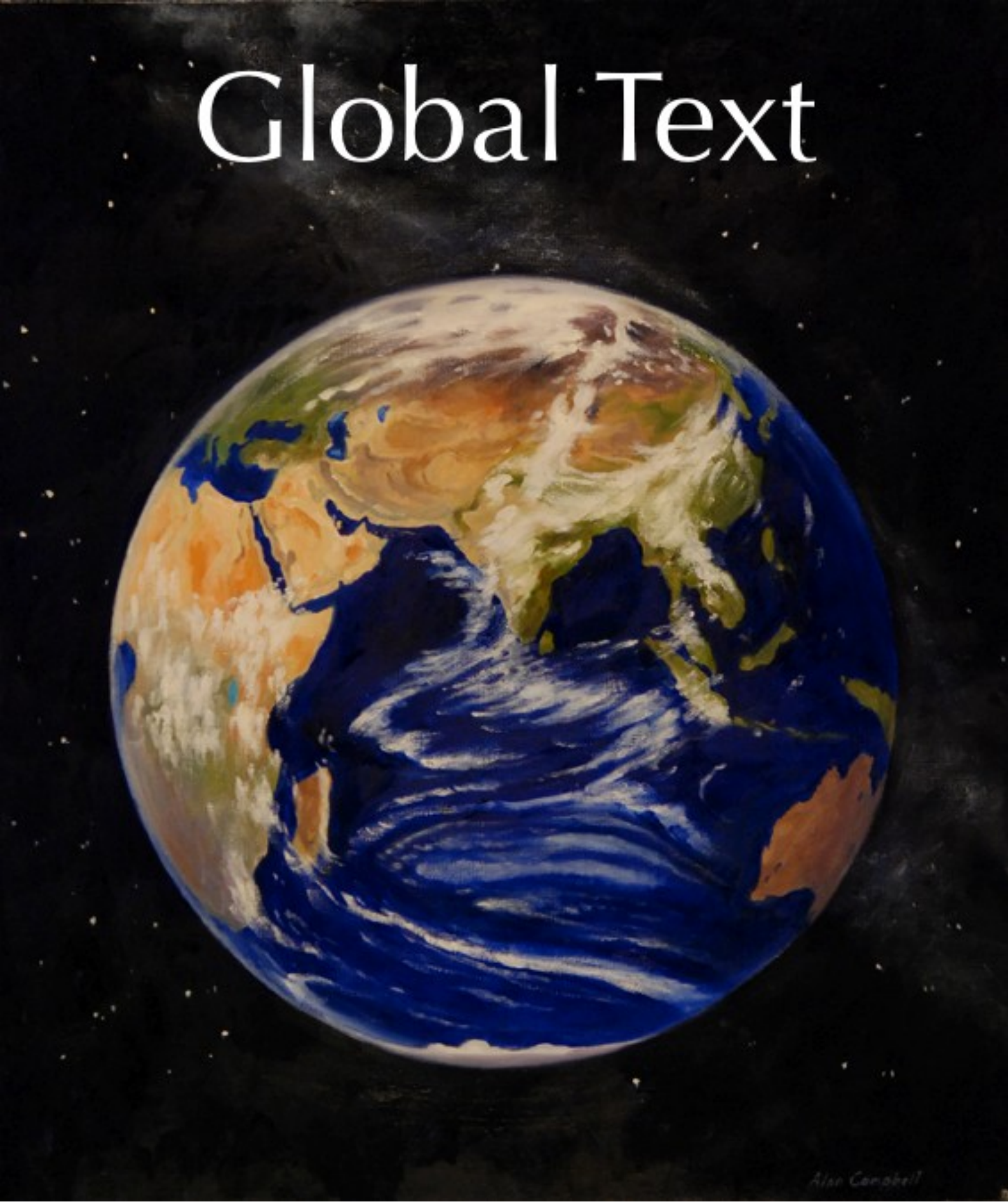
Global Text



Information Systems

Richard T. Watson (editor)

University of Georgia

Table of Contents

Preface

The Global Text Project (http://globaltext.org/) was initiated in early 2006 to develop a series of free, open content, electronic textbooks. A description of the Global Text Project is available on the project's.

The impetus for developing the information systems text as one of the first in the series is based on:

- The worldwide community of IS academics is a closely-knit community. Individuals know each other and have a long history of cooperating with each other on a global scale. Creation of an open content textbook will require the cooperation of the worldwide community of faculty and their students, as well as practitioners.
- The IS community, of all academic communities, should be the one that is an early adopter of technology that holds the promise of being able to create a state-of-the-art textbook.

The Information Systems textbook created by the community will be best-in-class, up-to-date, and, perhaps most importantly, made available at no cost to students anywhere in the world, but particularly to students in the developing world.

The overall approach of the text

Introductory information systems textbooks often present the topic in somewhat of a vacuum. That is, they focus on information systems without really succeeding in showing how IS is integrated in organizations, how knowledge workers are supported, and how important IS is for an organization's success. Many undergraduate students do not understand why they are required to take an IS course since they are not IS majors. Many also expect the introductory course to focus on personal productivity software. This textbook will teach students how to exploit IS in a technology-rich environment. It will emphasize why, no matter what their major, information and communications technologies (ICT) are, and increasingly will be, a critical element in their personal success and the success of their organizations. In other words, they need to be introduced to concepts, principles, methods, and procedures that will be valuable to them for years to come in thinking about existing organization systems, proposing new systems, and working with IS professionals in implementing new systems.

Students need to understand systems and the systems concept, and they need to understand the role of ICT in enabling systems. Students will learn the characteristics of good systems (e.g., intuitive, likable, error-resistant, fast, flexible, and the like). Knowing the characteristics of good systems will permit students to demand well-designed systems and to suggest how existing systems should be changed. Students need to understand the affordances, directions, and limits of hardware, software, and networks in both personal and organizational dimensions. They also need to appreciate that, as technical capabilities change and new ones arise, more opportunities to apply ICT for efficiency, effectiveness, and innovation are afforded. They need to understand the process for developing and implementing new or improved systems and the activities of IS professionals in this process.

The distinction between information systems and information technology

We distinguish clearly between information systems and information technology, a distinction that seems lacking too often as the terms are often used interchangeably. We define these terms as follows:

- An information technology transmits, processes, or stores information.
- An information system is an integrated and cooperating set of software directed information technologies supporting individual, group, organizational, or societal goals.

In other words, IS applies IT to accomplish the assimilation, processing, storage, and dissemination of information. Thus, PDAs, cellular phones, music players, and digital cameras as information systems. These devices use multiple information technologies to create personal information systems. Similarly other information technologies, such as database, networks, and programming languages, are used to created organizational systems.

1. Being a systems innovator

Editor: David A. Bray (Emory University, USA)

Contributors: Benn Konsynski and Joycelyn Streator (Emory University, USA)

Reviewer: John Beachboard (Idaho State University, USA)

Learning objectives

- define what broadly constitutes a "system" and an "innovation"
- describe examples of innovation
- describe how one might strive to be a systems innovator
- describe the benefits of innovation to society at-large

Introduction

Let us welcome you the modern age, so full of promise both in terms of human and technological progress! This is the first chapter of several in this textbook. Later chapters will discuss what information systems are, how information systems are integrated into the workplace, the role of knowledge workers alongside information systems, and how information systems link to the success of organizations. In this opening chapter, we address the role of innovation and being a systems innovator. Our goal is to motivate you to want to be a systems innovator, not just a maintainer of existing systems or a support of systems that have already been built.

We want you to innovate! Without systems innovators, it is quite possible that our modern age would not be so full of promise and potential. In fact, without systems innovators, humanity might never have reached modernity at all.

Several historians say we humans are "modern" when we do not automatically reject new or foreign elements in society. For human society, modernity begins when communities began to explore, tolerate, and accept the new and diverse. Thus, modernity includes a receptiveness of human societies to new ideas. Living in the modern age allows us to expect that modern enterprises and markets will tolerate and potentially reward to new ideas and new practice. In a modern age, those individuals who design insightful innovations (i.e. innovators) can be highly praised if their innovations are well timed, well designed, and well implemented.

As systems innovators, we welcome the modern age and strive to be open to new and beneficial ideas of change. Human societies value and evaluate new ideas by expected impact and effect. Modern markets and firms represent particular types of human organizations. Markets and firms can incorporate innovations by changing either their design or practices.

Being a systems innovator

Let us briefly consider the meaning of the essential words in the title: "systems" and "innovator" (defining "being" is something we will leave to the philosophers).

Systems are the object of particular designs. Broadly speaking, systems involve the organization of things, logical and physical. Systems include data, processes, policies, protocols, skill sets, hardware, software, responsibilities, and other components that define the capabilities of an organization. Systems include human and non-human aspects. The components, or parts, of a specific system can be either real or abstract. Components comprise an aggregate "whole" where each component of a system interacts with at least one other component of the system. Cumulatively, all the components of a system serve a common system objective. Systems may contain subsystems, which are systems unto themselves that include a smaller set of interactions among components for a more narrowly defined objective. Systems may also connect with other systems. The following diagram (Exhibit 1) illustrates an example system.

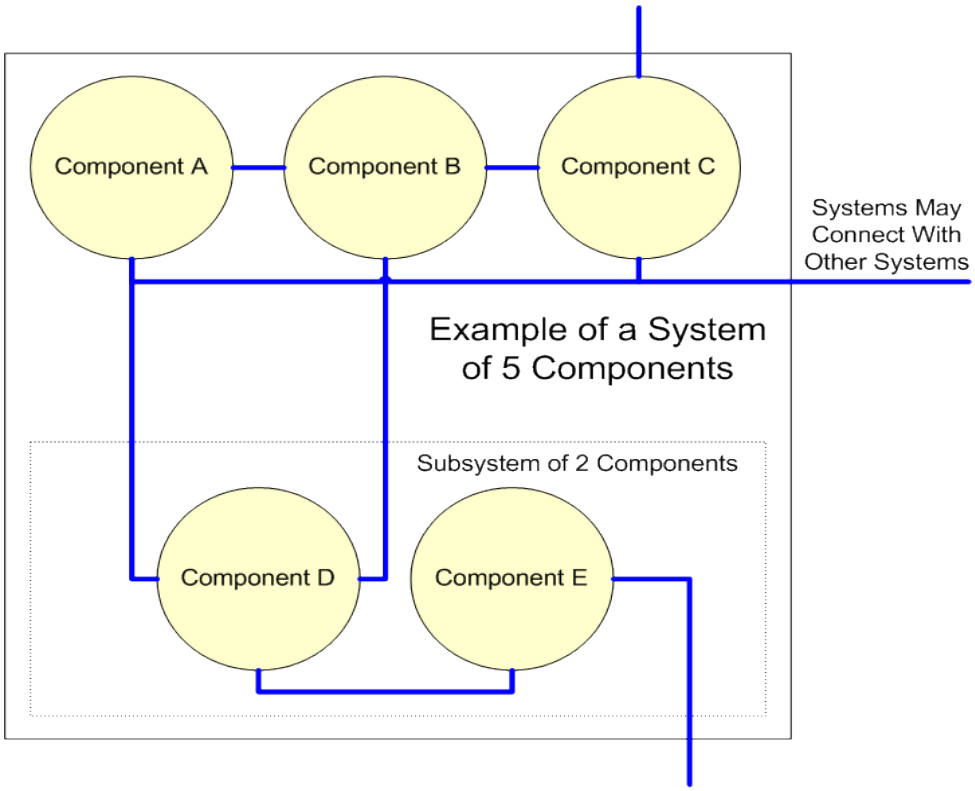


Exhibit 1: A sample system

Later on in this textbook, you will learn about information systems and how they support organizations. We will leave defining the specifics of information systems to those chapters—though will will say that information systems, as with all systems, consist of multiple components. All the components of a system serve a common system objective in support of the information system. Frequently this objective supports the purpose of the organization—the organization itself serving as a system of humans and tasks as well.

Later chapters will discuss the different components a of information systems and how these components should interact. Other chapters will discuss the consequences of components not interacting. For now, we want to motivate you to be interested in systems—and innovation.

Innovation is the process of "making improvements by introducing something new" to a system. To be noteworthy, an innovation must be substantially different, not an insignificant change or adjustment. It is worth noting that innovation is more a verb than a noun in our context. Innovation is similar to the word evolution, which derives from the Latin root for staying "in motion." Systems innovations often include an expectation of forward

motion and improvement. To be worthwhile, innovations must be worth the cost of replacement, substitution, or upgrades of the existing order.

The term innovation may refer to both radical and incremental changes to products, processes, or services. The often unspoken goal of innovation is to solve a problem. Innovation is an important topic in the study of economics, business, technology, sociology, and engineering. Since innovations are a major driver of the economy, the factors that lead to innovation are also critical to government policy-makers. In an organizational context, innovations link to performance and growth through improvements in efficiency, productivity, quality, competitive positioning, market share, etc. All organizations can innovate, including for example hospitals, universities, and local governments.

Rather than construct a narrow definition of innovation, it is useful to think of innovation as including, but not limited by, a few key dimensions. Successful innovations include these dimensions.

The first dimension is that of innovation form. Innovations manifest in many ways, but generally are either tangible or intangible. Tangible innovations result in new goods, services, or systems that you can physically touch. Examples include the introduction of new products or a style of architecture. Intangible innovations include the creation of new services, processes, modes of operating, or thinking. Intangible innovations might introduce greater efficiency into an existing process or create an entirely new way of doing something. For example, an innovation could reduce the time required to manufacture a car. This intangible innovation might translate into greater profits for a car manufacturer.

The second dimension is that of innovation degree. Innovation degree compares a particular innovation to that of the status quo. In 1980, a researcher named John Hage introduced the concept of “radical” versus “incremental” innovation—representing two different types of innovation. An incremental innovation introduces an idea, process, or technological device that provides a slight improvement or causes minor change in a normal routine. Sometimes the impact of incremental innovation may require only minor adjustments in the behavior, processes, or equipment associated with a system. A manufacturing facility upgrading to a new version of software that provides additional features to enhance existing operations is an example of an incremental innovation.

Conversely, radical innovations introduce an idea, process, or technological device that dramatically alters a current system. For example, if a manufacturing firm acquired a new technology that allowed the firm to completely redefine and streamline its production processes, then this new technology represents a radical innovation. Often radical innovations involve not only new technologies and processes, but also necessitate the creation of entirely new patterns of behaviors.

Systems innovators are individuals who design and implement innovations. To design refers to the process of developing a structural plan for an object. Systems innovators are individuals who transform the practice of organizations, markets, or society by making significant forward moving improvements.

What qualities are required of a good system innovator? Systems innovators seek to designs that improve on the old to take advantage of new technologies, new techniques and new practice and processes. We would suggest that systems innovators not only recognize that social and economic structures are all human-made, but also recognize that human structures are always open to changes, enhancements, and redesign.

It is important to recognize that systems operate within systems. Identifying the connections and layers of these systems will make you a successful systems innovator. Often identifying new connections or new layers that no one else has identified yet can provide new opportunities for innovation.

This book seeks to discuss with you the capabilities, approaches, and skills required of the systems innovator in the 21st century. How does one prepare for the assessment, evaluation, design, and implementation of the improvements to systems, particularly those that incorporate information technologies, particularly those systems that incorporate information technologies?

Systems innovators are designers

Sociologists note that humans are unique in their invention and adoption of tools. Among these human-made tools are the systems and procedures that govern, direct, and enable modern societies to function. These tools also include the systems that enable the actions of commerce and exchange. Systems enable patterns of work and reward and the conduct of participants in enterprise. For our modern age, systems have never been more relevant as the speed of society and the enhancement of information access and opportunity for social interaction increase. Almost all aspects of modern commerce, modern society, and modern life are connected the designs of humanity. Much of what defines the pace and practices of our modern age are systems and technology-enabled.

What factors are required for innovation? Good design and good management. We now discuss design and management.

First, designers matter. To be a designer implies the task of creating something, or of being creative in a particular area of expertise. Part of being a systems innovator includes being a designer. It is worth considering that the fields of “systems design” and “organization design” are similar as both incorporate creatable, changeable, and linkable elements.

Designers seek the requirements and expectations, identify the objectives and measurements of success, give structure to the elements, and form to the components of systems. Success or failure hinge on the ability of a designer to attain the proper requirements and expectations of a system. For example, a systems innovator plans to design a new cell phone network for 500,000 subscribers. Unfortunately, the innovator fails to include the requirement of future growth of the cell phone network to 2,000,000 individuals in five years. When the network is built, per the design of the innovator, new cell phone subscribers must be turned-away from accessing the network because of the omitted designer requirement. Since the designer failed to include the proper requirements, this omission diminishes the success of the system.

Second, managers matter. In addition to developing a structural plan for a system, designers must manage the process of systems development, to include overseeing systems implementation, adoption, and continuing operation. Design also sometimes involves the augmentation and extension of an existing system. Part of being a systems innovator includes the enhancement of an existing or legacy system with a new idea, method, or technological device. Extending the life of a useful system, or upgrading capabilities to better align with the enterprise objective, may be the best service of the systems innovator. Often, it is easier to enhance an existing system, than it is to decode, decipher, or replace such a system.

Social systems are tools designed by humanity. These systems reflect the bias and the values of the designers, or those that task the designers with requirements and expectations. Thus, designers, who create rules, influence

systems greatly. Essential elements of the process and product of system development include the unique style and preferences of a designer.

Designers leave their mark, their trail, and their values reflected in the tools they produce. Style and preferences also guide systems implementation. It is also important to note that systems are networks of interacting elements. Thus, the aggregate "whole" of a large system may be more capable, stronger, or beneficial than the sum of its individual components—or it might be less so. Systems amplify the strengths and the weakness of their design. Ideally, well-designed systems amplify the benefits of their individual components.

Innovations are new answers to problems

What are possible areas demanding innovation? What are different strategies for innovation? These are challenging questions with no easy answer. The concept of innovation has been widely studied, yet it remains a difficult topic to define. Merriam-Webster's online dictionary describes innovation as "the introduction of something new" or "a new idea, method, or device".

While this definition provides a good starting point for our discussion of innovation, there are still a number of dimensions to consider for a more thorough understanding of the concept. Careful observation of our surroundings reveals a multitude of innovations. Everything from electricity to running water, or from personal computers to cell phones, represents some form on innovation from past systems.

Innovations are not limited to tangible products. Innovations also occur when processes are dramatically improved. For example, through advances in cell phones, very little human effort is required to communicate a message across great distances quickly. More than 100 years ago, the similar transactions would have required significant manual work and time for a message to be sent by postal mail.

Many things can trigger innovation. An individual or team of individuals may seek to address an existing problem, respond to a new situation, or explore new ability.

While innovations typically add value, innovations may also have a negative or destructive effect as new developments clear away or change old organizational forms and practices. Organizations that do not innovate effectively may die or be destroyed by those organizations that do. Systems innovators are critical to our modern age. Innovators must insure that their envisioned innovations are appropriate to the environment of today and tomorrow.

Innovations are also reactions to change

While innovation can occur as individuals and groups wrestle with new problems, innovation can also be reactionary and occur as a response to unplanned changes. The ancient philosopher Heraclatus once said: "There is nothing permanent except change."

The statement is certainly true today in our high-tech world. Advances in computing power, communication technologies, and networking of computers around the world has quickened the pace at which dramatic change can occur across large and diverse groups of electronically connected people. Innovation often arises as a way of coping with, attempting to control, or benefit from changes.

Changes in the use of information technology often provide the impetus for innovation. There might be instances where local conditions encourage a particular innovation. For example, if past historical conditions prevented installation of wired telephone networks because they were too expensive, but now cell phone networks

are both more affordable and available; the innovation of cell phone networks might open up new capabilities for areas that previously did not have such technology. As cell phone networks networks become more prevalent, the ways individuals communicate, compute, and exchange information will change and local companies may seek to introduce cell phones with new features that adapt to these changing communications patterns.

Exciting times for systems innovators

We live in exciting times for systems innovators. Advances in electronic communications, airline transportation, and international shipping, increasingly connect the lives of multiple individuals throughout the world. Such connective advances are part of a greater trend known as globalization. For the modern age, globalization includes the opening of commercial markets, increased free trade among nations, and increased education for a larger number of people. With globalization, what you do may influence events on the other side of the world.

With globalization, environments for organizations, both businesses and world governments, are becoming more complex. The reasons for this increased environmental complexity include “the four V's”, specifically:

- increased Volume (from local to global context in terms of transactions)
- increased Velocity (faster transactions between people)
- increased Volatility (organizations change and reorganize faster)
- increased concerns regarding Veracity (the truth is harder to distinguish)

For systems innovators, it is important to recognize this perspective of increased complexity. This perspective is important both because it presents opportunities to innovate—by addressing the complexities and challenges mentioned above—as well as the risks associated with not innovating. Failure to innovate in an increasing complex and interconnected world may mean that your organization, be it a business or government, might become irrelevant and outdated quickly.

Increased complexity also makes the job of a systems innovator a bit trickier: an innovative solution needs to account for increasing complex environment. What may seem to be a straightforward solution may have unintended effects on a system or other systems connected to a system.

This leads to a second important perspective: systems operate within systems. Specifically, our world is a system of multilayered, interconnected systems. Homes connect to gas, water, and electrical systems that link to other homes. Traveling exposes us to systems of highways, public transportation, trains, planes, and ocean-going ships.

A business is an organization comprised of multiple workers, interdependent in the tasks they perform. Within the organization, there may be a system of monitoring the funds received into and paid out by the business—and accounting system. This accounting system would include humans (managers and accountants), accounting data, processes for managing accounting data, rules for recording accounting data, as well as technology components. Within this accounting system may be another system: an information system running a computer program dedicated to tracking electronically the accounts of the organization. A systems innovator looking to improve the organization may focus on the system of the overarching organization itself, the accounting system, or the information system dedicated to tracking electronically the accounts of the organization.

General insights into human (and information) systems

For systems innovators, it is important to note that all human systems are artificial. By "artificial", we mean that human systems would not exist naturally in the world without humans. No natural rules govern the systems humans create—whether the systems are governments, businesses, educational institutions, or information systems. This is not to say that the systems humans create do not have rules; rather, they often do! For systems innovators, you can influence and change the rules. Part of innovating is identifying when the rules of a system, be it an organization or information system, could be modified to provide a better benefit.

So what are rules? Rules are defined ways of interacting with elements in a system, often proscribing an action. One rule might be "do not steal". This rule means that individuals should not take an element that does not belong to them. Another rule might be "if an electronic message is received from Company ABC, route it to our Accounts Payable". With rules, it is important to note that they link elements with actions. Rules can form the policy of a system. By system policy, we mean rules that link actions to elements in a system.

Information systems include data and processes. Data can be logical values (true vs false), numbers, words, or strung-together sentences. Actions, known as processes, are required to actively exchange, transform, and move data. For a computer to "compute", processes actively manipulate data. Components of an information system detail the rules for what processes can do to data, under what circumstances. A systems innovator seeking to improve an information system might look to modify the data an information system contain or collect. Equally, a systems innovator might improve an information system by modifying what processes manipulate data—or an innovator might modify the policies of a system to reuse existing processes in new ways on data.

Recognizing that all human systems are artificial leads to another equally important perspective for our modern age: organizations are becoming like markets. By markets, we mean places where no one person is commanding everyone else. With marketplaces, you are free to wander to different vendors, try their wares, and are under no obligation to purchase their goods or services. No one is commanding you to buy from Company ABC vs. XYZ—you get to decide.

For organizations, this means that traditional "management" of individuals by command and control is increasingly becoming difficult in our complex, global world. Reasons for this reduced ability to command are partially dependent on globalization. Businesses may be partnered with other businesses where they do not have the ability to directly tell these other companies what to do. The same may be true for world governments. There also may be instances where organizations are competing with one another, perhaps to sell similar goods or services to you as a consumer—or perhaps to discover a new idea or innovation.

Cumulatively, these factors mean that organizations will be less able to command individuals or other organizations what they would like them to do, and instead have to rely on other mechanisms. These other mechanisms include using diplomacy to influence individuals or organizations, being smarter or stronger than other competing organizations, or giving rewards to elicit desired behaviors from individuals.

General implications for a systems innovator

So what does this mean for you as a future systems innovator? It means you should be mindful of the increasingly complex environment of our global world. You should seek out the connections and layers among systems. If you spot new connections or uncover a new layer, you may also identify radical innovations. Sometimes

the most important part of being an innovator is having the wisdom to know when to form partnerships and with whom to make friends.

It also means you should seek to identify what are the rules of a human system. You should be open to asking "why" a certain rule is in place and allow yourself to consider what would happen if that rule was changed. What would improve the current rules? Are there rules that no longer help as they previously did? You should also recognize that the ability for organizations to command others is decreasing. As an innovator, this trend is helpful, since it increases the chances for you to "market" and spread innovative ideas. This also means you need to consider how to influence and encourage others to adopt your innovations.

Specifically, as a systems innovator, you will need to "market" your innovation. Simply because you have thought of an innovation does not mean it will succeed. If you are not skillful at influencing others to consider and adopt your innovation, your innovation may not succeed. Further, you may need to be smarter or stronger than other innovators—and you may need to consider what rewards would encourage individuals to adopt your innovation. Sometimes an innovation itself can encourage individuals to adopt it, but this often is not immediate. By their nature, humans are not prone to change if they are relatively happy. Even if your innovation provides new benefits, you may need to consider wants to encourage individuals to shift from their old "ways" to your innovation.

Finally, it means you should recognize that innovation is necessary to deal with change. Change is constant in our world, so innovation also needs to be constant. Yes, innovation can be risky as sometimes an idea might not be incomplete, not right for the current environment, or not aligned with the needs of an organization. However, there is a greater, more certain risk that any system will become outdated without innovations. As a systems innovator, you should search, dream, and reach for the future.

How can I innovate?

It would be great if it were possible to describe systems innovation as a simple formula. However, this is not the case. Just as modern societies are open to differing views and ideas, there are many ways routes to innovation.

Sometimes, an existing issue or obstacle with a system prevents the achievement of a certain goal. Individuals may brainstorm solutions to this problem and a novel idea will emerge that provides a good fit for removing or minimizing the obstacle. In other cases, an individual who is unfamiliar with the obstacle may bring an entirely new perspective that leads to an innovative solution. For systems innovators, continuous exposure to new ideas on different topics can bring fresh perspectives to familiar issues, thereby triggering new ideas and insights.

Innovation also necessitates a careful balancing-act between risks versus rewards. Many new ideas promise a tremendous payoff and recognition. However, with increasing rewards often comes increasing risk. For example, introducing an entirely new information system to a company's operations department may hold the promise of making inventory management more efficient, producing faster product availability, and increased sales. At the same time, the initial implementation of a new information system probably will cause disruption within an organization, perhaps in the form of requiring new processes or employee training. When undertaking an ambitious effort, it is essential that a systems innovator be aware of the potential downsides and risk factors that will undermine success if not adequately addressed. Complex systems often have unexpected consequences, some of which are likely to be undesirable. Failed innovations are not only time consuming but can be costly and a source of embarrassment for a would-be innovator.

While it may seem wise to take the safe route and focus on smaller, seemingly less risky projects, this may mean addressing small problems or introducing ideas that have a minimal impact on a system's performance. For example, rather than addressing inventory management problems directly, simply upgrading the computers that run the inventory management without actually changing the software that manages the processes, might have a minimal impact on the core problems. In addition, systems projects can often grow in scope as the project progresses. What started as a small effort might uncover additional requirements or system dependencies, prompting a project that started out as a low risk to grow into a longer, larger, more risky endeavor. Systems innovators must balance the reward a potential innovation might provide with the risk that implementation or adoption of such an innovation may go awry.

In addition, systems innovators should appreciate the importance of appropriate timing. Sometimes innovations can be "ahead of its time" or "too late." When designing innovations, it is important to consider environmental factors. An innovation must fit the needs of an organization, market, or society. An innovation introduced out of phase can undermine a system and other innovation efforts. Remember our earlier example of a systems innovator planning to design a new cell phone network for 500,000 subscribers. The systems innovator failed to take into account the requirement of future growth of the cell phone network to 2,000,000 individuals in five years. A skilled systems innovator would have planned for both the present and future of his or her designed system.

For our modern age, systems innovators can design and create innovation in ways previously unavailable. Innovators must insure that their envisioned innovations are appropriate to the environment of today and tomorrow. Through technology, there are new ways for individuals to combine ideas for entirely new outcomes. This "re-mix" age allows recombination of systems elements to produce results greater than the sum of the parts.

What do innovations achieve?

Ultimately, any systems innovator is important in what their innovations achieve for organizations and individuals. Thus, it is appropriate to conclude discussion of "Being a Systems Innovator" with reflections on what ultimately are the fruits of innovation, and what makes being a systems innovator such an important and essential role for the fast-moving world of the 21st century. For a successful systems innovator, keeping a long-term view on the outcomes achieved from any future innovation is vital.

First, innovations marry insights and existing knowledge to produce new knowledge. Without new knowledge, your organization, be it a business or government, might become irrelevant and outdated quickly. By creating new knowledge, innovations are the only sustainable advantage. The present "ways" of systems, with time inevitably become old "ways" and outdated. For our modern age, that some individual or organization will eventually identify an innovative "way" better than the old "ways" is almost certain. Changes happen, and without innovation, organizations might become irrelevant quickly. New knowledge also allows your organization to gain positive benefits from previously unforeseen approaches or opportunities. These new approaches can help your organization grow or profit. Our world's future is made by innovations and new knowledge gained from these achievements.

Imagine individuals at the dawn of the 1900s. If you could go back in time and tell them about the modern world, what would be the "new" knowledge you would share with them? What innovations would be the most important to you? Would you discuss modern jets that travel the global daily? Or would you explain how we have sent rockets into outer space and astronauts to the moon? Or would you tell them about the Internet and personal

computers? Or would you talk about our use of antibiotics and modern medicines to treat diseases? What other innovations do you think are most noteworthy?

Now think about those individuals in the year 1900. Would they even believe some of the innovations you told them? How would they react if you tried to tell them about the ability to share electronic messages with people around the world in less than a second? How would you even begin to describe the ability to search for information, music, or videos on the Internet—recognizing that they did not even have television yet in the year 1900?

All of these innovations (and many, many more) occurred in less than 100 years, and our world is moving forward ever faster, and with ever more complexity, in our innovations and discoveries. With these innovations comes new knowledge, knowledge we now take for granted in our daily lives. This new knowledge improves our ability to work more productively, live longer and fuller lives, communicate across large distances, and perform tasks in hours that previously took weeks or months to complete.

Innovation also achieves shared knowledge. For innovations to succeed, they often must share (either within your organization or with the world) insights that one or two people previously may have observed or discovered. If you are a systems innovator and you realize a better way for your company to interact with its customers, you will need to share your idea with others to encourage its adoption. Equally, if you discover an improved way for individuals to manage their email messages, you may incorporate this innovation into a software product that you then make available to others (to buy or for free). The knowledge produced by innovators needs to be "shared" for their innovations to be truly realized and recognized.

Innovations achieve new products and profits

Second, innovations translate new knowledge into new products and profits, particularly for business (but also for organizations where performing efficiently is important). Even for governments, innovations can allow governments to save money or do more with the same amount of funds. The radio, the television, the personal computer, the cell phone—all inventions we take for granted today, were innovations that had to be dreamed of, experimented with, tested, and refined before they could be products and produce profits for businesses. Innovations take time and courage to see an idea through to reality. For example, websites like Amazon.com or eBay.com were once innovative start-up companies with untested ideas. Their different innovative visions were believed by some, uncertain by several, and publicly dismissed as not possible by several (at the time).

Systems innovations can produce increased profits for an organization either by producing new products or by producing new ways of doing old activities. Should you accept the challenge of being a systems innovator, you need to be in love with not just the new and exciting, but also with understanding the current context and history upon your area of focus. Past and present events provide a context to find innovations.

It is the mid 1990s and you are a systems innovator. As a systems innovator, you know that historically most people have to go to a bookstore to buy a book. They have either to call or visit the bookstore to see if it has a particular book, and physical bookstores can only carry a limited number of books. For rare or unique books, chances are your local bookstore will not have the product. Equipped with this knowledge of past and present events, you might think about launching a company where people can visit a central website, search through millions of books, and order the book online and have it delivered to their home. Such an innovation became Amazon.com, and produced millions of dollars for its founding innovators.

Again, it is the mid 1990s, you are a systems innovator, and you see a trend where hard drives increasingly are getting physically smaller with more storage space. You also notice a new audio compression technology that allows entire songs to be compressed into small files (called MP3s). Equipped with this knowledge of past and present events, you might think about building a device that would allow individuals to store MP3s on a portable hard drive with a nice, friendly interface for people to search and find the songs they want to play on this portable device. Such an innovation was Apple's iPod—which included not only a hardware device, but also an information system (a website, called iTunes.com) for people to find, purchase, and download the songs they would like to play on their iPods. This innovation also achieved both a new product and large profits for Apple and its Chief Executive Officer, Steve Jobs.

Innovations increase effectiveness

Third, related to the earlier two points, innovations increase the effectiveness of individuals and organizations. By effectiveness, we mean how well actions of an individual or organization lead to a desired outcome. If an individual has to do a lot of work to produce only a small amount of a desired outcome, the effectiveness of that individual's actions is low. Conversely, if an individual has to do minimal work to produce a large amount of a desired outcome, the effectiveness of that individual's actions is high.

Innovations can make existing ways of doing activities more effective and thus either more profitable or enriching for the participants. Sometimes the art of being a systems innovator is not necessarily about discovering something completely new, but instead is about "refining" some processes exist and making these processes better and more effective. The Internet is full of examples where existing ideas were translated into the digital world and made more effective. Email allows individuals to send electronic messages to each other and receive them in much faster time than it would take to deliver a hand-written message. Computers allow individuals to compose and edit documents electronically using a word processing program in ways that are much more effective than retyping the document numerous times and changing revisions manually.

Individual improvements in effectiveness can also translate into organizational effectiveness. If a team of people discovers an innovative way of rearranging how they work together, this innovation may translate into faster results or better outcomes for the team. For information systems, innovators are often striving to make not only the system work better and more effectively—but also the organizations of people who interact with the technology also work better and more effectively.

No human system is completely effective and all of our systems have the potential to be improved. As a systems innovator, your mission is to seek ways of increasing individual and organizational effectiveness. You want to discover innovations that require the minimal amount of work to produce the largest amount of a desired outcome. Challenge the unknown, not feasible, or impossible.

Summary

As we have discussed, systems are the object of particular designs. The components, or parts, of a specific system can be either real or abstract. Components comprise an aggregate "whole" where each component of a system interacts with at least one other component of the system. To innovate is to make "improvements by introducing something new". A noteworthy innovation must be substantially different, not an insignificant change or adjustment. Innovations can be tangible or intangible, radical or incremental.

Systems innovators are individuals who design and implement innovations. To design refers to the process of developing a structural plan for an object. Designers seek the requirements and expectations, identify the objectives and measurements of success, give structure to the elements, and form to the components of systems. Success or failure hinges on the ability of systems innovators, as designers, to attain the proper requirements and expectations of a system.

As a systems innovator, you should be mindful of the increasingly complex environment of our global world. You should seek out the connections and layers among systems. If you spot new connections or uncover a new layer, you may also identify radical innovations. Sometimes the most important part of being an innovator is having the wisdom to know when to form partnerships and with whom to make friends. You should also remember that all human systems are artificial. By "artificial", we mean that human systems would not exist naturally in the world without humans. Part of innovating is identifying when the rules of a system, be it an organization or information system, could be modified. You should seek to identify what are the rules of a human system. As a systems innovator, you should be open to asking "why" a certain rule is in place and allow yourself to consider what would happen if that rule was changed.

As a final important point, systems innovators achieve "magic". By "magic", we mean that innovations designed by systems innovators allow abilities or feats that were previously not possible or realistically feasible. If innovations allow the impossible to be technologically possible, innovations allow "magic". New, innovative technologies often allow such innovation, thereby helping humanity to reshape the natural world.

Humans have a long history of using new technologies to overcome the physical limitations of human beings. For example, the use of a plow and the irrigation of crops allowed humans to productively farm land and grow crops. By growing crops, humanity began to build settlements (which themselves began to use new technologies like levels, bricks, hammers, and nails). These technologies helped human civilization to grow. With civilization, humanity began to focus on things beyond immediate, short-term survival—to include education. Education is only possible because we have technologies that allow other human individuals to grow enough food for individuals beyond themselves. We can go to school because others will work while we are studying and provide the necessary resources for our society to function, including running water, electricity, healthcare systems, construction of buildings, transportation systems, and more.

In a sense, all the systems that we discussed at the start of this chapter are a result of innovations and human technologies that have allowed us to reshape our world. Civilization is possible by employing innovative technologies and systems that allow humanity to think beyond short-term survival and pursue education, research, global commerce, foreign relations, and even fun recreational activities like books, movies, and television.

Innovations are "magic"—they reshape the natural world. Humans use tools to accomplish tasks that were either not feasible or impossible. Innovative tools also increase the effectiveness of systems and individuals. Historically, human use of tools has allowed us to extend our physical abilities. Now, with information systems, there is the unique opportunity for human beings to extend not only their physical abilities, but also our cognitive abilities. Not only can we work better or faster, but also we might be able to think better or faster as an individual or organization. All of this will be possible through future innovations.

Innovative information systems in the last 40 years have already dramatically changed our world, to include faster, global transactions between people and the ability to collaborate and electronically share commerce,

government, or entertainment-related activities with millions of people. Innovative information systems of the future will achieve what we would label “magic” today. As a systems innovator, the fruits of your successful innovations will not only produce new knowledge, new products, profits, and increased organizational effectiveness —your innovations will also achieve that which previously was impossible or infeasible.

Our closing advice: **search for beneficial, new ideas. Through your efforts, bold innovations will produce the world of tomorrow.**

Exercises

1. You are able to go back in time to visit members of your local neighborhood in the 1900's. What would be the “new” knowledge you would share with them? What innovations would be the most important to you and why?
2. You are able to go forward in time to your local neighborhood in the year 2075. What do you imagine, as a time-traveler from the present, would be some of the future innovations that you would observe? How would they change human societies? What innovations do would be the most important to you and why?
3. You have been hired as a systems innovator to design a new cell phone network for 500,000 subscribers. You are wise enough to include the requirement of future growth of the cell phone network to individual additional subscribers. What other requirements might be worth considering when you design the system? What requirements might influence the success (or failure) of the designed system?
4. If you could work on designing any innovation, what would it be and why? Would you create something new or extend an existing system? What requirements and other concerns would you need to consider in designing your innovation? What benefits do you think would occur if you could achieve your innovation as you imagine it?

Chapter editor

David A. Bray is currently a PhD candidate at the Goizueta Business School, Emory University. His research focuses on "bottom-up" (i.e., grassroots) socio-technological approaches for fostering inter-individual knowledge exchanges. Before academia, he served as the IT Chief for the Bioterrorism Preparedness and Response Program at the Centers for Disease Control where he led the technology response to 9/11, anthrax, WNV, SARS, and other major outbreaks.

References

Argote, L., Ingram, P., Levine, J. M., & Moreland, R. L. (2000). Knowledge Transfer in Organizations: Learning from the Experience of Others. Organizational Behavior and Human Decision Processes, 82, 1-8.

Clippinger, J. H. (1999). The Biology of Business: Decoding the Natural Laws of Enterprise (1st ed). San Francisco: Jossey-Bass Publishers.

Cummings, J. N. (2004). Work Groups, Structural Diversity, and Knowledge Sharing in a Global Organization. Management Science, 50, 352-364.

Daft, R. L., & Weick, K. E. (1984). Toward a model of organizations as interpretive systems. Academy of Management Review, 9, 284-295.

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly, 13, 319-340.

Dawes, R. M., Orbell, J. M., Simmons, R. T., & Kragt, A. (1986). Organizing Groups for Collective Action. The American Political Science Review, 80, 1171-1185.

Galbraith, J. R. (1982). Designing the innovating organization. Organizational Dynamics.

Heckscher, C. C., & Donnellon, A. (Eds.). (1994). The Post-Bureaucratic Organization: New Perspectives on Organizational Change. Thousand Oaks, Calif: Sage Publications.

Kling, R. (1991). Cooperation, coordination and control in computer-supported work. Communications of the ACM, 34, 83-88.

Kling, R. (2003). Reconceptualizing Users as Social Actors IN Information Systems Research. MIS Quarterly, 27, 197-235.

Kuhn, T. S. (1962). The Structure of Scientific Revolutions. Chicago: University of Chicago Press.

March, J. G. (1991). Exploration and Exploitation in Organizational Learning. Organization Science, 2, 71-87.

Markus, M. (2001). Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Situations and Factors in Reuse Success. Journal of Management Information Systems, 18, 57-93.

Nonaka, I. (1994). A Dynamic Theory of Organizational Knowledge Creation. Organization Science, 5, 14-37.

Simon, H. A. (1969). The Sciences of the Artificial. [Cambridge: M.I.T. Press.

Wade-Benzoni, K. A., Tenbrunsel, A. E., & Bazerman, M. H. (1996). Egocentric Interpretations of Fairness in Asymmetric, Environmental Social Dilemmas: Explaining Harvesting Behavior and the Role of Communication. Organizational Behavior and Human Decision Processes, 67, 111-126.

2. 2. Achieving Efficiency and Effectiveness through Systems

Editor: Gabrielle Piccoli and Iris Liu (Cornell University, USA)

Reviewer:

Learning objectives

- understand the differences between information systems and information technology
- be able to identify the four components of information systems
- understand the relationships between the four components of information systems
- understand the reasons for having an information system
- be able to assess the value of information systems from the financial as well as managerial points of view

Introduction

An information system is designed to collect, process, store and distribute information. Although information systems need not be computerized, Information Technology (IT) plays an increasingly important role in organizations due to the fast pace of technological innovation. Today most information systems beyond the smallest are IT-based because modern IT enables efficient operations as well as effective management in organizations of all sizes. This chapter will first define the critical components of modern information systems, and then discuss how organizations achieve higher efficiency, better effectiveness, and improved coordination through their use.

What is an information system?

To understand what an information system is we need to first clearly differentiate it from information technology—with which it is often confused. Let's look at example. A manufacturing company with 1,200 employees used to pay the employees by checks. At the end of every month, the human resources staff would look at how much each employee should be paid and cut 1,200 checks; one for each employee. To collect their pay, the employees would have to go to the human resources office with their employee identification cards. Every employee would show his/her identification card to the human resources staff so the staff could ensure the checks were given to the right employees. Note that, while there is a system managing the payroll, the process of issuing paycheck requires no information technology and is completely manual.

One day an ill-advised employee visits the human resources office with a fake identification card and obtains the check which does not belong to him. As a result of the stolen check, the company suffers a loss—having to compensate the employee whose check was stolen.

Reacting to this event, the director of human resources considers installing a system that automates the end-of-month payment process. The information of employees' bank accounts will be stored in the system and their pay will be directly deposited into their bank accounts at the end of every month. The objectives of this new information system is to improve efficiency—saving the human resources staff's time in manually preparing 1,200 checks and verifying employees identification cards 1,200 times a month—and to improve effectiveness of the organization by reducing the possibility of lost checks.

After a few months, the human resources director finds out that the human resources staff are still preparing the checks manually for employees every month, and that the employees are still coming to collect their checks in person. In other words, the new technology is not being used. When the director investigates what went wrong with the system, the staff tell him that there is nothing wrong with the system. When an employee's account number is input, the system will notify the bank to deposit the salary into that account on every pay day.

However, upon further investigation, the director discovers several possible causes for the system's failure. First, the employees are reluctant to provide their bank account information for various reasons, such as not feeling comfortable with releasing their personal information. Also, when the wrong account number is entered and the money is deposited into the wrong account, nobody in the human resources department is in charge of contacting the bank to rectify the mistakes. This discredits the new system, and employees who encounter this problem no longer want their salaries directly deposited.

IT is not information system

As can be seen from the above example, information technology and information system are two related but separate concepts. In our example the IT component seems to be working quite well, yet the organization is not reaping the benefits of the time saved by human resources staff and employees. In other words, the system fails to achieve its objectives due to the failure of other components.

Let's look at another example. When the most famous banker in the Ching Dynasty, Mr. Hu Syue-Yan, established his first bank, Fu-Kang, in the mid-1800s, we can be absolutely sure that there were no computer systems in the bank! At that time, the services a retail bank provided were very similar to those offered today: a customer could deposit money in the bank and earn interests, borrow money, or remit money orders. All these activities had to be recorded to reflect a customer's current balance in the bank. That is, an information system needed to be in place in order to keep track of how much the customer deposited, how much the customer withdrew, how much the customer borrowed, and how much the customer transferred into other accounts.

How did Mr. Hu Syue-Yan's employees do so? Relevant information was collected, processed, stored and distributed using pen and paper. Thus, although a computerized technology was unavailable at the time, the bank's information system still achieved its goals—enabling the business to serve its customers. Again, here is evidence that information technology is not information system. Even though IT is often at the core of modern information systems, information technology and information system are two different concepts. But what is the difference? What are the components of an information system?

The four components of an information system

An information system is defined as a socio-technical system comprised of two sub-systems: a technical sub-system and a social sub-system. The technical sub-system encompasses the technology and process components, while the social sub-system encompasses the people and structure components. The critical insight from the

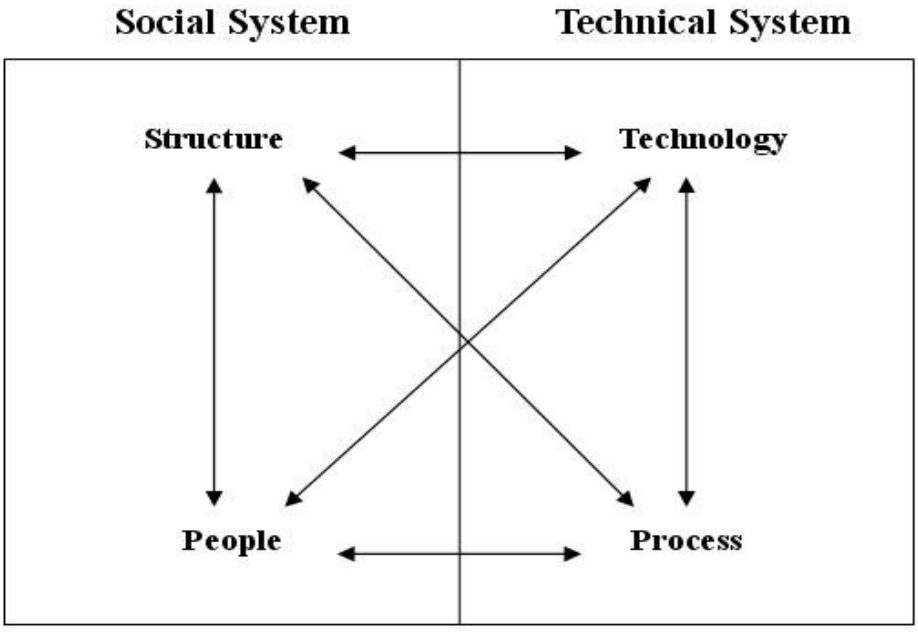


Exhibit 2: The socio-technical system

examples introduced earlier is that for an information system to perform and achieve its objectives, all four components have to be present and working together. We now define and describe the four components of a modern information system (see Exhibit 2).

Information technology

As discussed earlier, an information system needs not to use computers. However, modern organizations increasingly rely on information technology as the core of their information systems. We define information technology to include hardware, software and telecommunication equipment that is used to capture, process, store and distribute information.

Hardware is the physical equipment—such as a personal computer, a laptop, a portable computing device, and even a modern cell phone—used to process information. Software is the set of coded instructions (programs) that direct the hardware to perform the required tasks. A typical example is Google Docs—a word processing program designed to instruct a computer to create text documents. Telecommunication systems are the networking equipment enabling users and devices to communicate. An example of a telecommunication system is a telephone network, which allows two callers to interact by voice over a distance.

These three elements—hardware, software and telecommunication systems—comprise the IT component of an information system. For example, the technology components of the automated payroll system mentioned in the first example include:

- hardware—computers and printers
- software—the accounting software application designed to keep track of the salaries and the staff scheduling system designed to keep track of hours worked and how much each employees should be paid

- telecommunication systems—local and inter-organizational channels of communication and routing equipment designed to connect the company to the bank for automatic money transfers.

Process

A process is the set of steps employed to carry out a specific business or organizational activity. In other words, a process maps the set of actions that an individual, a group or an organization must enact in order to complete an activity. Consider the job of a grocery store manager and the process he engages in when restocking an inventory of goods for sale. The store manager must:

- check the inventory of goods for sale and identify the needed items
- call individual suppliers for quotations and possible delivery dates
- compare prices and delivery dates quoted among several suppliers for the same goods
- select one or more suppliers for each of the needed items based on the terms of the agreement (e.g. availability, quality, delivery)
- call these suppliers and place the orders
- receive the goods upon delivery, checking the accuracy and quality of the shipped items; pay the suppliers

Note that there are multiple viable processes that an organization can design to complete the same activity. In the case of the grocery store, the timing and form of payment can differ dramatically, from cash on delivery to direct transfer of the payment to the supplier's bank account within three months of the purchase. The critical insight here is that the design of the process must fit with the other components of the information system and be adjusted when changes occur. For example, imagine the grocery store manager purchasing a new software program that enables her to get quotations from all of the suppliers in the nearby regions and place orders online. Clearly the preceding process would need to change dramatically, and the store manager would need to be trained in the use of the new software program—in other words, changes would also affect the people component.

People

The people component of an information system encompasses all those individuals who are directly involved with the system. These people include the managers who define the goals of the system, and the users. In the opening example concerning the automated payroll system, the people component of the system includes the human resources director who wants to enhance an efficient and effective payroll process, the human resources staff who maintain the correct employee account information, and the employees whose salaries will be deposited directly into their account. An analysis of the opening example clearly shows that problems with the people component were partly to blame.

The critical insight here is that the individuals involved in the information system come to it with a set of skills, attitudes, interests, biases and personal traits that need to be taken into account when the organization designs the information system. Very often, an information system fails because the users do not have enough skills, or have a negative attitude toward the system. Therefore, there should be enough training and time for users to get used to the new system.

For example, when implementing the automated payroll system, training on how to enter employees' account information, how to correct wrong entries, and how to deposit the salaries into each account should be provided to

the human resources staff. The benefits of the system should be communicated to both the human resources staff and the employees in order to build up positive attitudes towards the new system.

Structure

The structure (or organizational structure) component of information systems refers to the relationship among the individuals in the people component. Thus, it encompasses hierarchical and reporting structures, and reward systems. The structure component plays a critical role in an information system, simply because systems often fail when they are resisted by their intended users. This can happen because individuals feel threatened by the new work system, or because of inherent human resistance to change. When designing a new information system the organization needs to be cognizant of the current and future reward system in order to create incentives to secure its success.

Relationships between the four components At this point it should be clear how information systems, while enabled by IT, are not synonymous with IT. Each of the four components discussed above can undermine the success of an information system—the best software application will yield little result if users reject it and fail to adopt it. More subtly, the four components of information systems must work together for the systems to perform. Thus, when the organization decides to bring in a new technology to support its operation, the design team must adjust the existing processes or develop new ones. The people involved must be trained to make sure that they can carry out the processes. If the skills of these individuals are such that they can't perform the required tasks or be trained to do so, a different set of individuals need to be brought in to work with the system. Finally, the design team must evaluate whether the organizational structure needs to be modified as well. New positions may need to be created for additional responsibilities, and old jobs may need to be eliminated. The transition from the old way of doing things to the new system needs to be managed, ensuring that appropriate incentives and a reward structure is put in place. Following is an example that illustrates the interdependence of the four components of information systems.

Mrs Field's Cookies (Ostofsky and Cash, 1988), one of the world's largest snack-food stand franchisors, which currently owns stores in the United States, Canada, Hong Kong, Japan, the United Kingdom and Australia, was started by a young mother with no business experience. Debbi Fields started baking when she was a teenager. Her cookies were so popular that she decided to open her first store in Palo Alto, California in 1977. When the business started expending, Randy Fields, Debbi's husband, believed that it was more important to keep the size of the staff small in order to enable the decisions making process to be faster and more accurate. He saw information systems as a way to avoid expanding staff while growing the business.

The system introduced at Mrs Field's that was used by the store manager on a daily basis was the day planner system. Every morning, the store manager entered information, such as day of the week and weather condition, into the system. Then, the system computed the projected sales and recommended when the cookies should be baked. The store sales were then periodically entered into the system during the day to adjust the projections and recommendations. Every day, the sales results were sent to the corporate database for review so the headquarters could respond quickly if any store was not performing well.

The objectives of building information systems: Having defined what information systems are, we now look at the reasons why modern organizations introduce them.

Efficiency

Efficiency is often referred to as "doing things right" In this chapter, we define efficiency in general terms as the ratio of output to input. In other words, a firm is more efficient when it produces more with the same amount of resources, produces the same amount of output with a lesser investment of resource, or—even better—produces more output with less input. The firm achieves efficiency improvements by reducing resource waste while maximizing productivity.

More output with the same input

To illustrate how organizations can be more efficient by introducing an information system, we provide an example of a hospital using an information system to manage patient information. Without a system to manage patients' personal and historical information, a doctor would need to ask a patient the same questions about allergies, family history and the like each time they visit the hospital—even if the patient has been to the same hospital and visited the same doctor a number of times before. As a result, the doctor's time is wasted in asking redundant questions each time the patient visits.

The most immediate solution to this information management problem is to create and maintain a folder for the patient containing their medical history, which is then used by any doctor treating the same patient in the future. The doctor retrieves the patient's historical information from the folder and saves time asking the same questions again.

With this simple information system, a doctor can serve more patients (more output) within the same amount of time (same input). An even higher degree of efficiency can be achieved by using a computerized information system. That is, doctors enter the patients' clinical results into a computerized database instead of writing on a piece of paper and filing the paper in a folder. When a patient returns to the hospital in the future, a doctor can obtain the patient's information at the click of a mouse. As a result, doctors can serve even more patients, since they do not need to search through all the patients' folders in order to find the specific information needed.

Same output with less input

One of the main objectives of any organization is to attempt to control costs and reduce the investment necessary to produce its output—in other words, most organizations are constantly trying to become more efficient by way of cost reductions. Information systems can help in this regard when they help lower costs, for example through a reduction in excess inventory, or by eliminating mistakes in operations.

Consider a grocery store as an example. If the store is able to better communicate with its suppliers, thus placing more recurrent orders for smaller quantities, it can minimize the costs of holding inventory (less input), yet be able to maintain the same level of service to its customers (same output). The store manager can also install a system that maintains inventory information. The data entered into the system are the items that are sold in the store and the quantity of these items. Every time an item is sold or ordered, the manager adjusts the quantity of the item in the inventory system. Without this system, the manager has to periodically go around the shop and the storage room to check if any items need to be restocked. After this system is installed, the manager can just look at the record to identify which items are almost sold out and need to be restocked. This also reduces the input (manager's time) to achieve the same output (restock all items).

Effectiveness

Effectiveness is often referred to as “doing the right thing”. In this chapter, we define effectiveness as the ability of an organization to achieve its stated goals and objectives. Typically, a more effective firm is one that makes better decisions and is able to carry them out successfully.

Responding better to the needs of different customers: An organization can create or refine its products and services based on data collected from customers as well as information accumulated from its operations. In other words, information systems help organizations to understand their customers better, and provide products and services customers desire. Doing so even helps organizations to provide personalized service if the organization collects customer data at the individual level.

We can once again use the grocery store as an example. The grocery store can accumulate information about what customers have been purchasing in the past and analyze this information to find out what items tend to sell well and at what time. The manager can also ask customers what kind of products and services they would like to purchase in the future, thereby attempting to anticipate their needs. With this information in hand the grocery store management can order products that will attract customers and stop ordering unpopular products.

Information systems can help organizations to improve product or service quality, or maintain the consistency of quality. To be able to improve product or service quality or to ensure consistency, organizations need information from the past as a source of error correction and as a reference point of improvement or consistency.

With the information system, which keeps track of the inventory in a grocery store, the manager can identify which items are popular and which are not. As a result, the manager can reduce the quantity ordered or stop ordering these slow-selling products.

In the same manner, a manufacturing company can collect information from quality control tests so as to analyze the most recurrent problems during the manufacturing process. The company can then find a solution to reduce these recurring problems. For example, a furniture manufacturer may find that the majority of the chairs produced do not pass quality control tests. The manager then reviews the results of these quality control tests and finds out that most of them fail because they are unstable. The manager can then look at the machine which produces the chairs and change the specification to rectify the problem. As a result, the quality of the chairs improves.

The company can also collect customer feedback on its products and services, and make improvements based on that feedback. For example, a telephone company collects customer feedback on their phone calls and then adds services such as call waiting according to customer suggestions. As a result, the telephone company can deliver products and services that fit their customers’ needs.

Responding better to the needs of different employees: An opportunity to improve effectiveness that is often overlooked involves better catering to the needs of the firm’s employees. This can be achieved by providing useful information to employees or faster access to information that helps them to perform their job. An information system can respond to the needs of employees by collecting data from various sources, processing the data in order to make it useful, and finally distributing it according to the needs of employees.

Another often overlooked opportunity to use information systems to fulfill the needs of employees is through empowerment. Empowerment represents the notion that the organization’s employees can be trusted to take on

more responsibility and make more independent decisions when they are given the information necessary to do so. Consider, for example, the employees of a large grocery store who typically receive and stock goods to be inventoried. If they have access to the appropriate information, such as original order forms and the invoices, they could be given responsibility to check, accept and even pay for the goods.

Better communication and coordination: Coordination is rooted in the ability to share information so that different individuals, different departments within an organization, or different organizations are brought together to pursue a common goal. Information systems support communication and coordination by better managing the distribution of information.

Communication consists in the exchange of information between two points, with the goal of having the recipients understand the sender's message. Communication is essential to every organization, as communication among employees ensures that they work together to carry out internal activities; communication between an organization and its suppliers ensures the suppliers provide correct materials for the organization to generate products and services to sell; and communication between an organization and its customers ensures that customers understand the products and services they are buying, receive confirmation when transactions occur, and are able to resolve problems that may occur encounter purchasing—the after-sale service.

Information systems can enhance communication by providing for more, and at times superior, channels. For example, the invention of electronic mail (email) has reduced the use of memos and written correspondence within an organization. As a consequence, the speed at which communication takes place improves. Multimedia communication elements, including images, sound and video files that employ a combination of presentation formats (text, graphics, animation, audio, and video) have also improved the richness of communication. These multimedia elements can be attached to an email and the email can be sent to suppliers or clients to better present or describe the parts wanted or the products and services provided.

For example, a salesperson from a hotel can attach a video clip with an advertising email to better illustrate the quality of its guest room. Such attachments can not be done by handwritten correspondence. Thereby, the quality of communication is improved. The invention of email has also reduced the use of the telephone. Now employees can read messages at their convenience without being interrupted by telephone calls while working.

Information systems not only improve point-to-point communication, but also within networks, which involves more than two parties. A computer network is a group of hardware (nodes in the network) with links to each other so that information can travel among them. A network helps organizations to collect information from and distribute information to different parties (such as suppliers, customers, and partners) in order to receive a more complete set of information of business activities, which then enhances coordination within the organization. For example, the operation department in an manufacturing company can collect information from the sales and marketing department to find out how many products need to be produced, information from the purchasing department to find out the costs of the parts, information from the executives to find out special about product changes, and information from quality control to find out how to improve product design and minimize defects.

Regulatory compliance

At times an organization will introduce information systems that may not improve the organization's efficiency, effectiveness, or enable it to communicate and coordinate better. This happens when regulations and laws require the organization to perform certain tasks—for example, recurrent maintenance on the machinery they use—or

produce some information—for example, tax reports. Regulatory compliance typically requires the organization to be able to create, manage, store or produce information—for example, maintenance logs or financial reports to compute taxes. In these situations the firm will introduce an information system.

Measuring the impact of an information system: The measurement of efficiency and effectiveness gives managers guidelines to assess the value of information systems. Without these measures, managers may be misled and make wrong decisions when investing in new technology and designing information systems. On the one hand, if the value of the system is underestimated, managers may cut back the allocated resources, which will result in foregoing the benefits of the new system. If the value of the system is overestimated, managers are wasting resources which could be used in other projects with higher returns. In this section, we introduce several established methods to measure efficiency and effectiveness improvements (or lack thereof) deriving from the use of information system.

Financial measures

A number of financial metrics have been proposed and used over the years to evaluate the costs and benefits associated with information systems implementation. In this chapter, we will discuss two of them: Return on Investment and IS Budgeting.

Return on Investment

Return on investment (ROI) is the ratio of monetary benefits gained from an investment to the amount of money invested.

$$ROI = \frac{estimated\ benefit - initial\ investment}{initial\ investment}$$

ROI looks at how the introduction of the information system enables the usage of resources to contribute to the organization. The organization can benefit from the introduction of the new system in various ways. First, a new system can reduce the costs of current operation by increasing efficiency. For example, a business can implement a new system which stores transactional information automatically, therefore saves the labor costs of data entry. Therefore, the estimated benefit in the above equation will be the differences in labor costs. Second, a company may modify the current system to take advantage of newly developed technology. For example, a bank which offers online banking can reduce the cost of mailing monthly statements to clients. Therefore, the estimated benefit will be the differences between the cost of mailing the statements before and after the installation of the system. Finally, a new information system may also support growing business transactions. For example, a retail store may switch to an Internet ordering system from a call center to be able to serve more customers. Doing so enables the business to respond to more customers at the same time by letting customers browse products and services online and enter ordering information by themselves. The estimated benefit is the extra revenue generated from online ordering.

In all three examples, the initial investment is the cost of bringing in the technology, setting up the new business processes, training the employees, and organizing the reporting and reward structures. In other word, it is the cost of designing, building and implementing the appropriate information system (as defined above). With this information, we can compute the ROI. An information system with a positive ROI indicates that this system can enhance efficiency and/or improve effectiveness of the organization.

The advantage of using ROI is that we can explicitly quantify the costs and benefits associated with the introduction of an information system. Therefore, we can use such metric to compare different systems and see which systems can help your organization to be more efficient and/or more effective.

The disadvantage of using ROI is that it may be difficult to justify the causal link between the investment in information systems and the gained benefits. For example, the extra revenue generated from online ordering may not be due solely to the introduction of the new system. It may be because your product is in the growing phase and rapidly increasing in popularity, how can you be sure that you would not have generated the increased revenues even without the new online ordering system? As a result, the benefits of the system may be overestimated. On the other hand, some customers may browse your products online but still order through the call center; therefore, you under-estimate the benefits of the system. As you can see, it is difficult to distinguish which part of the revenue is strictly due to the introduction of the new system and this will lead to an inaccurate ROI.

IS budgeting

The IS budget provides a reference point of efficiency at the firm level instead of the system level. An organization with relative less IS budget when comparing with similar organizations is considered to be more efficient since it achieves the same level of services (output) with less resource (input). The advantage of using IS budget as a reference is that the information needed can be obtained relatively easily from financial documentation.

IS budgets, as measure of efficiency, have some significant limitations however. For one, assuming that two organizations are very similar is an over-simplification of reality. Moreover, a firm's IS budget will depend on the systems it currently has, as well as the one it is currently developing.

Managerial performance measures

Effectiveness measures relate to how well a firm is able to meet its business objectives once it is enabled by the new information system, and therefore measures whether the system has improved the organization's effectiveness.

Information usage

Once an information system is implemented, the behaviors of acquiring and using information may be directly influenced. For example, a restaurant manager would not be able to make good staffing decisions without information about forecasted business. An information system which collects data of past business patterns and forecasts future business can provide the restaurant manager with sufficient information to make competent staffing decisions. Therefore, we can measure effectiveness by assessing information usage. Information usage can be evaluated by

- the extent to which the system is used
- the correlation between the system inputs and the business objectives

The system usage can be measured by the amount of queries needed to make managerial decisions. The correlation between the system inputs and the business objectives should be assessed to ensure the inputs serve their purpose. For example, an information system would not be effective if it was designed to forecast future business, but only allowed the input of supplier information.

Customer and employee satisfaction

Information systems should also be able to help organizations better respond to the needs of different customers and employees. Therefore, we can also assess the impact of information systems by measuring the extent to which

the system improves customer satisfaction, and the extent to which the system fits the needs of employees and owners. These measures can be obtained by self-reported surveys.

Summary

An information system, designed to collect, process, store and distribute information, is comprised of four critical components: technology, process, structure, and people. Technology and process represent the technical sub-system of an information system, while structure and people represent the social sub-system.

The technology component includes hardware, software and telecommunication equipment. A process is a set of actions that are designed to carry out a specific business or organizational activity. The people component include all of the individuals who are directly involved with the information system. Finally, the structure component refers to the relationship among the individuals in the people component.

These four components of information systems are interdependent. That is, changes in one component may affect the other components. The major reasons of organizations introducing a new information system are to enhance efficiency (doing things right), and/or to improve effectiveness (doing the right thing). Efficiency can be enhanced by reducing inputs while producing same or more outputs, or producing more outputs while using the same level of inputs. Effectiveness can be improved by better responding to the different needs of stakeholders. The impact an information system brought to an organization can be assessed from the financial point of view as well as from the managerial performance point of view.

Case

Royal Hotel's Espresso! Rapid Response Solution

The Royal Hotel in New York City, NY was a luxury all-suite hotel primarily serving an executive clientèle visiting Manhattan on business. These guests were busy and demanding as they used their suite not only as a place to sleep but also as a temporary office. The general manager stressed the importance of the high quality of service due to the high percentage of repeat guests. "Our guests are extremely discerning, it is completely unacceptable to have a light bulb out in the bathroom when the guest checks in, particularly if she is a returning guest", the general manager said.

To ensure the extremely high quality of service, the general manager decided to purchase and install M-Tech's Espresso! Rapid Response Solution. With this new technology, the housekeepers could report deficiencies directly to the computer, instead of verbally communicated to the maintenance department after they ended of their shift. The housekeepers just needed to dial a special code from the phone in the guest room and an automated attendant would walk them through the reporting process step by step in the language of their choice. Espresso! Then automatically generated, prioritized and dispatches a work order to a printer, fax, or alphanumeric pager. Therefore, the new system should be able to reduce the response time as the housekeepers did not have to wait until the end of the shift to tell the maintenance department, and sometimes they even forgot to tell the maintenance department. Also, Espresso! had a reporting function so that the management team could obtain information about most frequently occurring or recurring issues, top reporting and completing performers and so on. With this kind of information, the maintenance department could identify recurrent problems and stop them before they even occurred.

Upon installation, a week of on site training was also offered. The installation and the training session seemed to run smoothly. Employees appeared eager to learn about the new system. However, soon after roll-out the general

manager discovered that the employees had reverted to the old manual system and had rapidly lost interest in Espresso!

Case questions

- What are the elements comprising the four components (technology, process, people and structure) of new reporting system?
- Why do you think the new reporting system failed? In other words, which of the four components of the information systems failed to support the goal of the system?

Case

Lands' End's Custom Tailored Apparel Program

In October 2001, Lands' End, a direct merchant of traditionally styled clothing who offers products through catalogs and the Internet, announced its new IT-driven strategic initiatives, a custom tailored apparel program. By November 2002, 40 per cent of Lands' End's web shoppers were buying custom-tailored chinos and jeans, while 20 per cent of these shoppers were new customers.

The concept of this initiative is mass-customization, a process that uses the same production resources to manufacture a variety of similar, yet individually unique products. Experts have found that consumers were willing to pay more for custom apparel and footwear. Other than increasing sales, the custom tailored apparel program brought Lands' End other benefits, including enhancing customer loyalty and lowering the operating costs spent in creating, printing and mailing catalogs. However, withholding catalogs from Internet buyers does not generate online sales. Therefore, sending catalogs at the optimum frequency and pages to keep them apprised of new products is necessary.

Lands' End's proprietary products, strong distribution infrastructure and established brand made the company ready for this electronic commerce initiative. Also, Lands' End did not set up a separate Internet division; hence, avoided internal competition. To manufacture these individually unique garments, Lends' End partnered with Archetype Solutions, Inc (ASI). After customers entered sizing information on Lands' End website, the orders were sent to ASI and software produced electronic patterns and order files for each order, which were then sent via email to production facilities in Latin America or Asia. Manufacturers produced, inspected and packed the garments. The garments were shipped to a third-party shipping center in the US and then shipped to consumers. During the production process, the garments were scanned and the status was updated at each stage of the process. The status report for all orders was sent nightly to Lands' End. ASI contracted with retailers (i.e. Lands' End) and manufacturers. Retailers pay ASI a license fee, which include an annual fixed component based on number of categories and a per unit fee. Therefore, both retailers and ASI had the incentive to sell a lot of units. The manufacturers were also required to license manufacturing and tracking software from ASI. Therefore, the manufacturers need to be able to be adept and flexible, and able to learn new technologies fairly rapidly.

Case questions

- Why did Lands' End introduce this new information system? What are the benefits this new system brought to Lands' End?
- How can the executives of Lands' End assess the financial and managerial performance impact of this new IT-dependent strategic initiative?

References

Cash, J. I., & Ostrofsky, K. (1989). Mrs Field's Cookies. Harvard Business School Case Study.

Ives, B., & Piccoli, G. (2003). Custom made apparel and individualized service at Lands' End. Communications of the Association for Information Systems, 11, 79-93.

3. Achieving efficiency and effectiveness through systems design

Editor: Per Flaatten (Retired Accenture Partner)

Introduction

In the previous chapter, you learned that an efficient and effective information system (IS) is composed of people, processes, structure and technology. However, the process by which you can create an IS was not covered. This chapter describes the efficient and effective development of the technology part of an IS; other chapters to follow will describe the activities required for the people, process and structure aspects of the IS.

The approach we follow is to first define in general terms the sequence of activities required to go from the decision to create a new IS to its implementation and subsequent maintenance. We then describe the most important issues or difficulties that you may encounter during the process, based on the experience developers have encountered on projects in the past. The rest of the chapter—the bulk of it, in fact—is devoted to describing for each activity possible approaches to resolving the issues and avoiding the difficulties. These approaches are not the only ones that are possible; those that are mentioned here have been selected because they have been successful in the past, are documented in the literature (so you can learn more about them by consulting various reference works), and enjoy widespread acceptance in real-life IS departments.

Unless otherwise indicated, we assume that the IS being developed is a web-based system that processes business transactions of some kind, and that the project is of medium size—say 5 to 25 people on the team. This means that we do not consider the development of websites that are purely informational, nor for personal productivity, nor that result in a software product for sale to individuals or organizations.

Development process: from idea to detailed instructions

What the development process essentially does is to transform the expression of an idea for an IS—a problem to be solved, an opportunity to be taken advantage of—into a set of detailed, unambiguous instructions to a computer to implement that idea. The biggest problem is that computers are excessively stupid and will only do what they have been told to do. For example, suppose you create a billing program for an electric utility and specify that bills must be paid within a certain time or the customer's electricity will be cut off. Suppose further that a customer receives a bill stating that he owes USD 0.00 (he might have a previous credit). Contrary to a manual system where all the work is done by humans, a computerized system may well treat this bill as any other bill and insist on payment; it may even send a signal to the customer relations department to cut off power for non-payment. To avoid this, explicit instructions must be included in the billing program to avoid dunning for amounts of less than a certain limit.

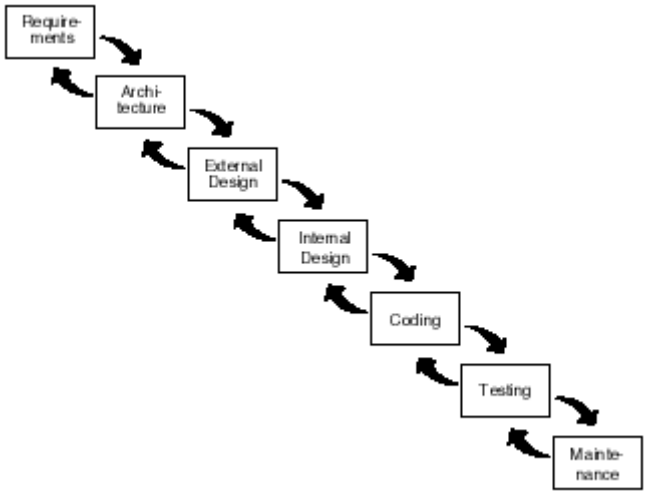


Exhibit 3.:Waterfall model

Systems developers must therefore pay attention to an excruciating amount of detail—not only when business goes on as normal, but anticipating all the exceptions that may arise. The exceptions may in fact amount to several times the work required for normal cases. The system then becomes, through sheer accumulation of details, more and more complex. This complexity is in itself a source of difficulty—it becomes hard to "see the forest for all the trees," to keep your eye on the big picture of the business benefits to be achieved, while at the same time making sure that every technical and business detail is right and finding out what went wrong whenever something does go wrong—as it invariably will.

From the earliest days of computer technology, the method for developing information systems has addressed the need to proceed from the general to the ever more detailed. The first well-known effort at formalizing the process came in 1970, in an enormously influential paper by W. W. Royce describing the waterfall model of the systems development life cycle.[1] Every author on systems development bases his or her work on some variation of this model, and we, too, have our favorite, depicted in Exhibit 5.

The work products or "deliverables" to be created during systems development start with the business case, the formal description of the rationale for developing a system in the first place, the results to be achieved and a cost-benefit analysis detailing planned development costs (often treated as an investment) as well as operational costs and savings. The business case is often considered part of project management rather than the development process per se; we include it here because it is the developers' best tool for not losing sight of the essential—the end result they are trying to achieve. As development work proceeds, developers make choices: one of the main factors in deciding which alternative to pick is the impact of the choice on the business case. For example, a choice that increases benefits may also increase costs: is the trade-off worth it? As a result, the business case must be maintained all along the life of the project as decisions are made.

The next deliverable in the waterfall approach is the information system's requirements. Requirements come in two flavors: functional requirements—what the system should do: processing, data and media content—and quality requirements—how well the system should do it: performance, usability, reliability, availability, modifiability and security.

1 Royce, Winston W. "Managing the Development of Large Software Systems," *Proceedings of IEEE WESCON*. August 1970.

The business objectives documented in the business case and the requirements, especially the quality requirements, dictate the architecture or overall shape of the information system being developed. The architecture describes the major components of the system and how they interact. It also documents design decisions that apply to the entire system, so as to standardize the solutions to similar design problems occurring at different places. Architectures or elements of architecture can be common to many applications, thus saving the project team time and money and reducing the amount of risk inherent in innovating. For example, many sales applications on the web use the concept of a "shopping cart", a temporary storage area accumulating product codes and quantities ordered until the customer decides that she has all she wants, at which point the application gathers all the products, computes the price, and arranges for payment and delivery. Shopping cart routines are available commercially from several sources.

- The design of the information system is at the heart of the development process. The design is in two parts:
- A description of how the system will appear to its users. This is called the external design or functional design.

A description of how the system will be operate internally, largely hidden from users. This is called the internal design (or technical design), and it lays the foundation for programmers to create the code which the hardware will execute.

The functional design specifies the interaction between the users and the system—what actions the user can take and how the system will react; it describes the inputs and outputs (screens, reports, messages exchanged with other systems); it establishes the different databases and their structure; and it shows at least a storyboard of the media content (text, graphics, photos, audio/video clips, etc.).

The technical design inventories the programs and modules to be developed, and how processing flows from one to the other. It also takes the architecture one step further in the implementation of some of the quality attributes, such as data integrity, fallback operation in case the system is unavailable, and recovery and restart from serious incidents. Finally, here is where any routines to create the initial data and media content required on day one of the new system's operation.

Code is the technical name for the programming statements and database specifications that are written in a language that can be understood by the technology. Creating code is a highly self-contained activity; the details depend on the environment and the subject will not be treated further in this chapter.

Throughout these steps, the initial idea for the system has been transformed into a set of computer instructions. Each step is performed by human beings (even if assisted by technology in the form of development tools) and is therefore subject to error. It is therefore necessary to conduct tests to make sure that the system will work as intended. These tests are organized in reverse order from the development activities: first, the code is tested, module by module or program by program, in what is called unit tests. Next come string tests, where several modules or programs that normally would be executed together are tested. Then follow integration tests, covering all of the software and system tests, covering both the software and the people using the system. When the system test has been completely successful, the information system is ready to for use. (Some organizations add an additional test, called acceptance test, to signify a formal hand-over of the system and the responsibility for it from developers to management. This is especially used when the software is developed by a third party.)

To put the importance of testing into perspective, note that it statistically consumes just about 50 per cent of the resources of a typical IS department.

With the system test complete and the initial data and media content loaded, the information system is put into production, and the development team is done. Right? Wrong! In fact, the most expensive part of the development process, called maintenance, is yet to come.

No sooner has a new system started to operate than it requires changes. First, users quickly discover bugs—errors in how the system operates—and needed improvements. Second, the environment changes: competitors take new initiatives; new laws and regulations are passed; new and improved technology becomes available. Third, the very fact that a new system solves an old problem introduces new problems, that you didn't know about beforehand. To illustrate this, take the standard sign at railway level crossings in France: Un train peut en cacher un autre ("One train may hide another one".). This caution alludes to the fact that you don't know what lies beyond your current problem until you have solved it—but then the next problem may take you completely unawares.

In theory, every change you make must go through a mini-life cycle of its own: business case, requirements, architecture, design, code and test. In reality, only the most critical fixes are done individually and immediately. Other changes are stacked up and implemented in periodic releases, typically a release of minor fixes every month and a major release every three to six months. Most often, the maintenance phase is performed by a subset (often around 25 per cent) of the initial development team. But since the maintenance phase is likely to last for years—certainly more than ten and not infrequently 20 years, the total cost of maintenance over the life of a system can eclipse the cost of initial development, as shown in Exhibit 6.

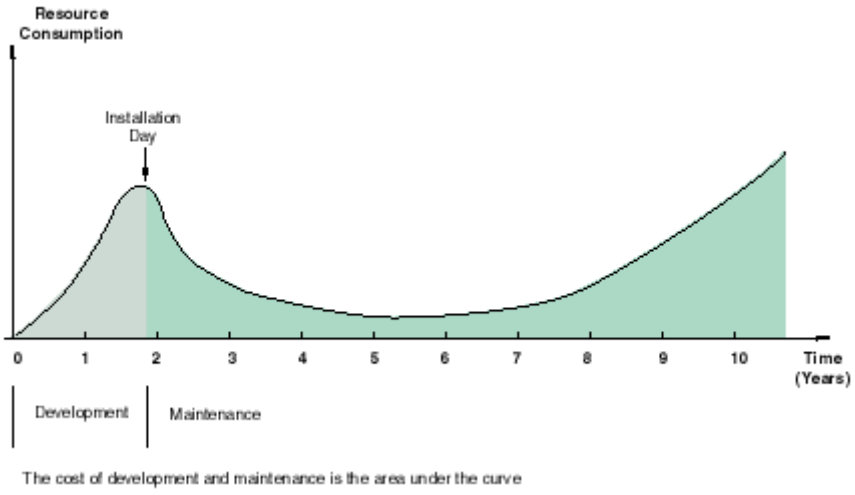


Exhibit 4.: Total costs of a system

(The increasing maintenance cost towards the end of the system life is due to the fact that the more a system has been modified, the harder it is to understand and therefore to make additional modifications to.

Issues

In this section, we will describe the difficulties that designers have historically had (and to some extent continue to have) in performing their tasks. These difficulties will help explain the widely accepted approaches, and some of the more innovative ones, that are the subject of the bulk of the chapter.

Cost

The first and most apparent issue with systems development is one of cost. From the earliest days, systems development has been seen as a high-cost investment with uncertain returns. It has always been difficult to isolate the impact of a new business information system on the bottom line—too many other factors change at the same time.

There are two components to total cost: unit cost and volume. Unit cost can be addressed by productivity increases. Volume can only be reduced by doing less unnecessary work.

System developer productivity was the earliest point of emphasis, as evidenced by counting lines of code as a measurement of output. (Lines of code is still a useful measure, but not the most critical one.) Both better computer languages and better development tools were developed, to a point where productivity is no longer the central issue of systems development. It is generally assumed that a development team is well trained and has an adequate set of tools.

Reducing the amount of unnecessary work is a more recent trend. Unnecessary work arises from two main sources: "gold plating" and rework.

Gold plating refers to the tendency of users to demand extras—features that they would like to have but that do not add value to the system. What is worse, developers have tended to accept these demands, mostly because each one seems small and easy to implement. The truth is that every time you add a feature, you add to the complexity of the system and beyond a certain point the cost grows exponentially.

Rework becomes necessary when you make an error and have to correct it. If you catch the error and correct it right away, no great damage is done. But if the error is left in and you don't discover it until later, other work will have been done that depends on the erroneous decision: this work then has to be scrapped and redone. Barry Boehm has estimated that a requirements or architecture error caught in system testing can cost 1000 times more to fix than if it had been caught right away[2]. Another way of estimating the cost of rework is to view that testing takes up an average of 50 per cent of the total initial development cost on most projects, and most of that time is spent, not in finding errors, but correcting them. Add the extra cost of errors caught during production, and the cost of rework is certainly over one-third and may approach one-half of total development and maintenance costs.

And this is for systems that actually get off the ground. A notorious study in the 1970s concluded that 29 per cent of systems projects failed before implementation and had to be scrapped (although the sample was small—less than 200 projects). These failures wind up with a total rework cost of 100 per cent![3] More recently, Bob Glass has authored an instructive series of books on large systems project failures[4].

Speed

In more recent years, concerns with the speed of the development process have overshadowed the search for increased productivity. If you follow the waterfall process literally, a medium-to-large system would take anywhere from 18 months to three years to develop. During this time, you are spending money without any true guarantee of success (see the statistics on number of failed projects above), with none of the benefits of the new system accruing.

2 Boehm, Barry W. *Software Engineering Economics*. Prentice-Hall, 1981.

3 GAO report FGMSD-80-4, November 1979

4 Glass, Robert L. *Software Runaways* and *Computing Calamities*. Prentice-Hall, 1998 and 1999.

It is a little bit like building a railroad from Chicago to Detroit, laying one rail only, and then laying the second rail. If instead you lay both rails at once, you can start running reduced service from Chicago to Gary, then to South Bend, and so one, starting to make some money a lot earlier.

Another factor that increases the need for speed is that the requirements of the business changes more quickly than in the past, as the result of external pressure—mainly from competitors but also from regulatory agencies, which mandate new business processes and practices. Eighteen months after your idea for a new system, that idea may already be obsolete. And if you try to keep up with changes during the development process, you are creating a moving target, which is much more difficult to reach.

Complexity

One of the main characteristics of information systems is that they are large, made up as they are of hundreds or thousands of individual components. In an invoicing subsystem, you might have a module to look up prices, a module to extend price by quantity, a module to add up the total of the invoice, a module to look up weight, a module to add up weights and compute freight costs, a description of the layout of the invoice, a module for breaking down a multi-page invoice, a module for printing... Each module is quite simple, but still needs to be tracked, so that when you assemble the final system, nothing is forgotten and all the parts work together. Compounding this is the fact that each module is so simple that when somebody requests a change or a refinement, you are tempted to respond, “Sure, that’s easy to do”.

And even though the components may be simple, they interact with each other, sometimes in unanticipated ways. Let us illustrate with an example—not taken from the world of IS, but relevant nonetheless. A large company installed a modern internal telephone system with many features. One of the features was “call back when available.” If you got a busy signal, you could press a key and hang up; as soon as the person you were calling finished his call, the system would redial his number and connect you. Another feature was “automatic extended backup”. This feature would switch all the calls that you could not or would not take to your secretary, including the case where your line was busy. If your secretary did not respond, the call would be sent to the floor receptionist, and so on, all the way to the switchboard, which was always manned. (This was in the era before voicemail.) The problem was of course that the backup feature canceled out the call-back feature—since you could never actually get a busy tone.

The effect of interaction between components in a business information system are often in the area of and quality requirements described earlier, such as performance, usability, reliability, availability, modifiability and security. None of these requirements are implemented in any one component. Rather, they are what are called emergent properties in complexity theory. For example, an important aspect of usability is consistency. If one part of the system prompts you for a billing address and then a shipping address, other parts of the system which need both should prompt for them in the same sequence. If you use a red asterisk to mark mandatory fields to be filled in on one screen, then you shouldn’t use a green asterisk or a red # sign on another screen. Neither choice is wrong—it is making different choices for the same function that reduces usability.

Finally, a critical aspect of complexity is the difficulty of reliably predicting system behavior. This means that you cannot be content with designing and coding the software and then start using it directly. You first have to test it to see whether it actually does behave as predicted (and specified). This test must be extremely thorough, because errors may be caused by a combination of conditions that occur only once in a while.

Unpredictability also applies to making changes to the system. This means that once you have made a change (as will inevitably happen), not only must you test that the change works, but you must also test that all those things that you didn't want to change continue to work as before. This is called regression testing; how to do it at reasonable cost will be discussed later.

Technology and innovation

One of the main drivers of information systems development is to take advantage of technological innovation to change the way business is done. As an example, take how the emergence of the Internet changed business practices in the late 1990s allowed new businesses to flourish (Amazon.com, Google, and eBay spring immediately to mind) and, to a lesser extent, existing businesses to benefit.

However, for every success, there were many failures. Every innovative venture carries risk, and while many dot-com failures were due to a lack of solid business planning, others failed because they could not master the new technology or tried to use it inappropriately. (Bob Glass's books refered to previously is filled with horror stories illustrating these dangers.) The problem is that if the technology is new, there are no successful examples to follow—and by the time these examples show the way, it may be too late, since others, the successful adventurers, may have occupied the space you would like to carve out for yourself. The difficulty, then, is to know how close to the leading edge you want to be: not too close, or you might be bloodied; and not too far behind, or you'll be left in the dust.

A related problem is that of change saturation. A mantra dear to business authors is "reinventing the organization". This may be good advice, but an organization cannot keep reinventing itself every day. Your most important stakeholders—customers, employees, even shareholders—may get disoriented and no longer know what to expect, and the organization itself may lose its sense of purpose.

Alignment on objectives

Any system development project is undertaken for a reason, usually to solve some operational difficulty (high costs, long processing delays, frequent errors) or to take advantage of a new opportunity (new technology, novel use of existing technology). However, many stakeholders have an interest in the outcome. Workers may resist innovation, regulators may fear social consequences, management may be divided between believers and skeptics, development team members may be competing for promotions or raises etc. If the objectives are not clearly understood and supported, the new system is not likely to succeed—not the least because the various stakeholders have different perceptions of what constitutes success.

Adoption

Once a new system has been created, the next challenge is to make people—employees, customers—use it. In the past, back-office systems such as billing, accounting and payroll were easy to implement. The users were clerks who could be put through a few hours or days of training and told to use the system; they had no choice. Today's system users may be less pliable and may refuse to go along or protest in such a way that you have to change the system, or even abandon it. As an example, Internet customers may "vote with their feet,, i.e. go to another website that provides the same service or goods at a better price or more easily.

Another example of how things can go wrong was recently provided by a large hospital organization that had created at great expense a system for physicians and surgeons. It was based on portable devices that the physician would carry around and on expensive stationary equipment at the patients' bedsides and in nursing stations. Three

months after the launch, it became apparent that practically all the physicians refused to use the system, and it had to be uninstalled, at the cost of tens of millions of dollars.

The issue of user adoption will be covered in more detail in Chapter 5, "System Implementation".

Useful life

The final issue we will consider is how to plan for a system's useful life. This is important for two reasons. First, as with any investment, this information is used to determine whether a new system is worthwhile or not. If you have a system that is expected to cost USD 5 million and bring in USD 1 million per annum, you know that the system must have a useful life of at least five years.

Second, planning the useful life of a system gives you at least a chance to decide how and when to withdraw or replace the system. Most development projects do not address the issue of decommissioning at all. As a result, systems live for much longer than anyone would have imagined. This is how so many systems were in danger of crashing on the first day of the year 2000—none of the developers had imagined that their systems would last so long. A perhaps more extreme example is that of the United States Department of Defense, which is reputed to have had more than 2,200 overlapping financial systems at one time[5]. Efforts to reduce this number have not been very successful, proving that it is much harder to kill a system than to create one.

Overall development strategy

Before we go into the various techniques and tools that are specific to each of the steps in the systems development life cycle, let us first look at the overall approach. The issues that you must address when you choose one approach over another have already been outlined above: development projects take too long to pay back; by the time they are implemented, the needs have changed, and the time elapsed before errors and shortcomings are experienced in operation make corrections costly. On the other hand, the system may require a critical mass of functionality before it can be truly useful, and a rapid implementation of a part of the system may do no good. Finally, If the system you are building is truly innovative, especially in its use of technology, the risks of failure are high and special precautions must be taken.

Iterative development

Iterative development is the standard approach today. It is characterized by the following:

- a series of short (3-6 month) development cycles, allowing for quick feedback from experience gained with the working system
- Each cycle delivers some significant, useful functionality.
- The early cycles focus on "low-hanging fruit"—functionality which is cheap to develop and has a high payback. Early successes give credibility to the project and enables you to do more difficult things later.
- The experience with the functionality in one cycle allows you to adjust the system in the next cycle.
- The project management style is called "time-boxing": each iteration has a deadline for implementation which cannot be exceeded. If some planned functionality takes too long to develop, you postpone it to the next cycle instead of delaying the cycle itself.

5 *San Francisco Chronicle*. May 18, 2003

- Maintenance follows initial development in a smooth transition (see the section on Maintenance later in this chapter). In fact it is difficult to tell when development ends and maintenance starts.

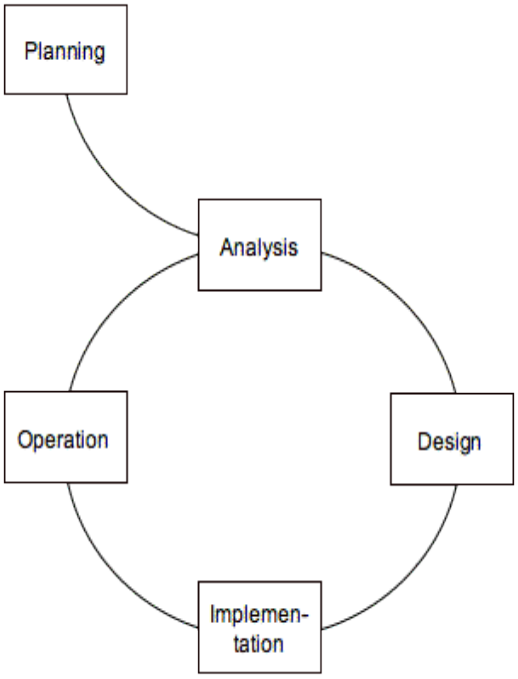


Exhibit 5.: Iterative development

The first version of iterative development was called Rapid Application Development, created by James Martin in the 1980s in response to the emergence of CASE (Computer-Aided Software Engineering) tools[6], and the approach has remained tool-intensive. Iterative Development also goes under the name Incremental Development, emphasizing that functionality is added gradually.

Alternative approaches: "Big Bang"

Occasionally, the circumstances of a given project may dictate that you use a "big bang" approach, where all the functionality of the planned system has to be delivered at the same time. The London Stock Exchange underwent two major transformations, one in 1986, when computerized and phone technology displaced face-to-face trading; and another one thirteen years later, when the phone technology was phased out, at least for the top 100 stocks. These two "big bangs" were successful, but both carried a built-in transition period where old and new systems coexisted.

In other cases, you may be building a brand new facility and want to use new technology, such as the computerized baggage handling at Denver International Airport in 1995. This system resulted in an unmitigated catastrophe and had to be scrapped at least temporarily.

The last temptation to adopt a "big bang" approach may be when you want to create an integrated enterprise-wide system, managing all the transactions and management data of an entire enterprise. Hardly anyone attempts this anymore after the consistently unsuccessful attempts in the early days of database management systems (around 1970) and Information Engineering (1985-1900).

In summary, rather than cede to the temptation of delivering everything at once, do your utmost to find a gradual approach, where you can learn as you go and change direction as dictated by reality.

6 Martin, James. *Rapid Application Development*. Macmillan, 1991.

Alternative approaches: Prototyping

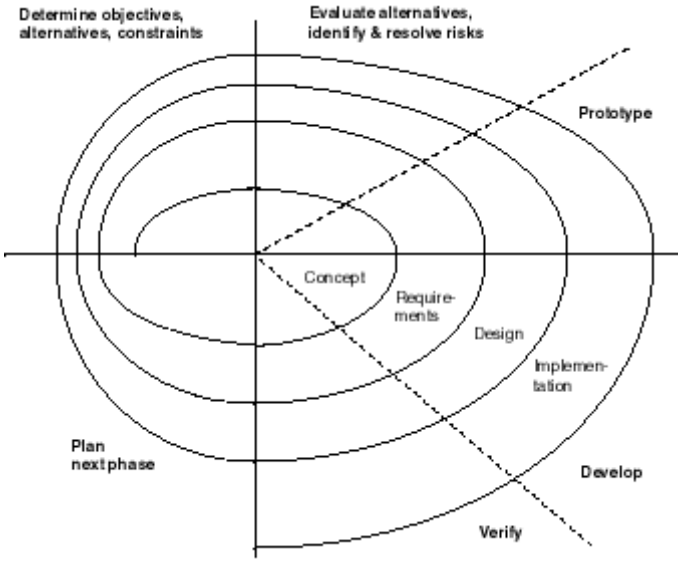


Exhibit 6.: The spiral approach

Prototyping is a variation on iterative development, used mostly for very innovative projects, those where the risk of failure is the greatest. Mostly, it is used with new technologies that requires new architectures. The basic principles of this approach were documented in an article by Barry Boehm, and illustrated by the following schematic which gave the apt name "Spiral Methodology" to the approach[7].

The principle behind prototyping is similar to iterative development, with the following two exceptions:

- Instead of starting with low-hanging fruit, start with the highest risk area. This enables you both to avoid spending large amounts on unfeasible systems and to learn the most from your errors, by making them early.
- Be prepared to throw away version 1 and even possibly version 2 of your system. The purpose of the prototype isn't to provide business functionality so much as it is to learn what not to do.

Requirements

Once you have decided on the approach to the development project, the next step is to find out in detail what the system should do—and, assuming you have opted for iterative development, what should take the highest priority. Then you need to formulate these requirements so that they can be used to drive the project, in particular when the requirements change in midstream.

Requirements elicitation

When you are eliciting system requirements, focus on what functions the system should perform, what data is required to support those functions, and how important the quality requirements—performance, reliability, usability and flexibility—are likely to be.

If the system you are planning is primarily intended to solve operational problems (high cost, high error rates, unresponsiveness, customer complaints...), you should spend some time getting familiar with the current

7 Boehm, Barry W. "A Spiral Model of Software Development and Enhancement." IEEE Computer, 21, No 5 (May 1988): 61.

operation, focusing on those people who actually experience the problem (as opposed to those to whom the problem or its symptoms are reported). This approach was systematic in the early days of information processing and was called "Analyzing the Current System." (One of the additional advantages of analyzing the current system was that it allowed the project team to learn about the business while doing something productive. Project teams today usually have to meet the expectation that they already know the basics of the business.)

There is less emphasis on the current system today, especially if the motivation for the system you want to build is to take advantage of technology to exploit a new opportunity. In that case, you want to focus on finding people who have a vision of what could be done, preferably in a group setting, where creative ideas are likely to flow.

Interviewing is by far the most common technique used in reviewing a present system. It relies heavily on such active listening skills as open-ended questions, appropriate words and phrases, acceptance cues, and restatement at both the direct and emotional level. Used properly, silence can also be effective.[8]

The most widespread group technique is probably Joint Application Design. JAD was developed in the early 1980s to overcome some of the difficulties caused by the more classic approaches to requirements analysis.[9]

The technique itself consists of gathering highly competent users (about ten) for a two-four day workshop to discuss, conceptualize, and analyze. IS personnel are present, mainly to listen, to record what is being said, and to document the users' contributions. The workshop is led by an independent person to whom neither the users nor the IS personnel report. The role of the leader is twofold: to elicit decisions by consensus and compromise rather than by fiat or majority vote and to keep the meeting on track so that the system to be built is the one that is discussed.

A particularly effective way of documenting what is being said is to create a prototype of the application on the fly. When you hear a user express an idea, it is particularly effective to be able to say, "Is this what you mean"? as you illustrate a screen layout, mouse clicks or navigation paths on a computer (projected on a large screen for everybody to see). The group can then react, critique, and build on what is shown.

There are a couple of traps you need to avoid with group techniques. The first is called "groupthink" and consists in the group losing its critical sense and going along with absurd or unworkable ides—partly because it is difficult for a group member to appear to be the only one who is against some idea. The other danger is related: a group may become overenthusiastic and push for "gold-plating," features and functions that look nice but serve no useful function—or not useful enough to be worth the cost. Don Gause and Gerald Weinberg have written an excellent and thought-provoking book on the subject.[10]

Finally, if the system you are planning is going to be used by outsiders (consumers, customers, suppliers, the general public) you need to devise a way to make their voices heard, for example by surveys or focus groups.

---

8 See for example: Gildersleeve, Thomas R. *Successful Data Processing Systems Analysis*. Ed 2. Prentice-Hall, 1985.

9 *Joint Application Design*. GUIDE Publication GPP-147. GUIDE International, 1986

10 Gause, Donald C., and Gerald M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House, 1989.

Requirements prioritization

It is impossible to satisfy all the requirements that a group of users and other stakeholders may come up with, especially if you are doing iterative development. The decision to include one set of requirements and not another is difficult. First, different stakeholders are interested in different results. Second, two requirements may be mutually contradictory; for example, higher performance may require more powerful hardware at greater cost. Third, some requirements may need to be satisfied before others can be implemented. The process needs to be seen as both fair and responsive to stakeholders' wishes.

In some cases, a straightforward economic analysis such as return on investment may be the right approach. This usually works best for requirements in the aggregate and less well for each detailed requirements. The difficulty is tying each individual requirement to a specific benefit and a specific cost. It is also a tedious job when the number of requirements is large. Thus, a prioritization scheme based on pure economics is best suited for the big-bang approach to systems development.

A similar approach, but based on risk (highest risk first) is appropriate for the prototyping approach to very innovative projects.

An alternative approach is Quality Function Deployment. This approach (introduced by Toyota in the 1970s to improve their process for designing automobiles)[11] sets up a series of matrices describing how well each requirement contributes to fulfill each business objective, then how each design feature (developed at a later stage) implements each requirement, and so on, thus illustrating qualitatively where the most benefit will be had. The unique part of this approach is that a group of stakeholder is asked to rank features pairwise—"I prefer B to A, C to B, C to E...." The end result is a series of ranked requirements where all the participants feel that their voice has been heard, thus helping to build consensus.

Functional requirements formulation

The requirements need to be documented unambiguously, to minimize later differences in interpretation. There is no standard format: text, schematics, and even prototypes can be used.

The most useful guideline is to document the requirements in plain language as testable statements. For functional requirements, this is pretty straightforward, for example: "After the customer has tried unsuccessfully to enter a Personal Identification Number (PIN) three times, the system will terminate the transaction, send an alert to the Security Department and retain the customer's card." This statement can now be cross-referenced to the design (in what module is this function implemented?) and to the test data (where are the instructions to the tester to try three wrong PINs and how is the result of this test documented?).

Plain language is useful for rigor and completeness; however, for any but the smallest systems, plain text becomes cumbersome and confusing. It is hard to communicate to users what the system will do and how it will accomplish its overall objective, and asking them to take a position on whether everything is covered is unfair. In particular, you are likely to hear, later on, comments of the type, "I assumed that this or that function was covered..." or, "It goes without saying that..." If these comments come as responses to processing errors once the system has been implemented, it is too late.

To address this problem, information systems professionals have developed a range of graphical representation tools, the three most widely used of which are decomposition diagrams (also known as Warnier charts after their

11 Hauser, John R., Don Clausing. "The House of Quality." Harvard Business Review, May 1988.

advocate, a French analyst named Jean-Dominique Warnier), data flow diagrams (DFDs), and entity-relationship diagrams (ERDs).

Warnier charts, an example of which is shown in Exhibit 7, show how a function can be decomposed into subfunctions. In turn, each subfunction can be further subdivided, all the way down to an elementary level. (The same hierarchical decomposition technique can be used in many other contexts, for example in outlining an essay.) The technique has the advantage of being easy to grasp and easy to use. However, it is not suitable for all purposes. It essentially can depict only a static structure, which must, in addition, be composed of mutually exclusive and completely exhaustive elements—MECE in analyst jargon. In other words, the decomposition must not leave anything out and no element can be used in more than one place.

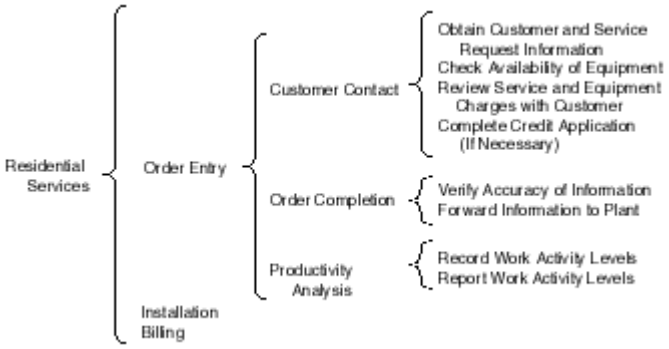


Exhibit 7.: A Warnier chart

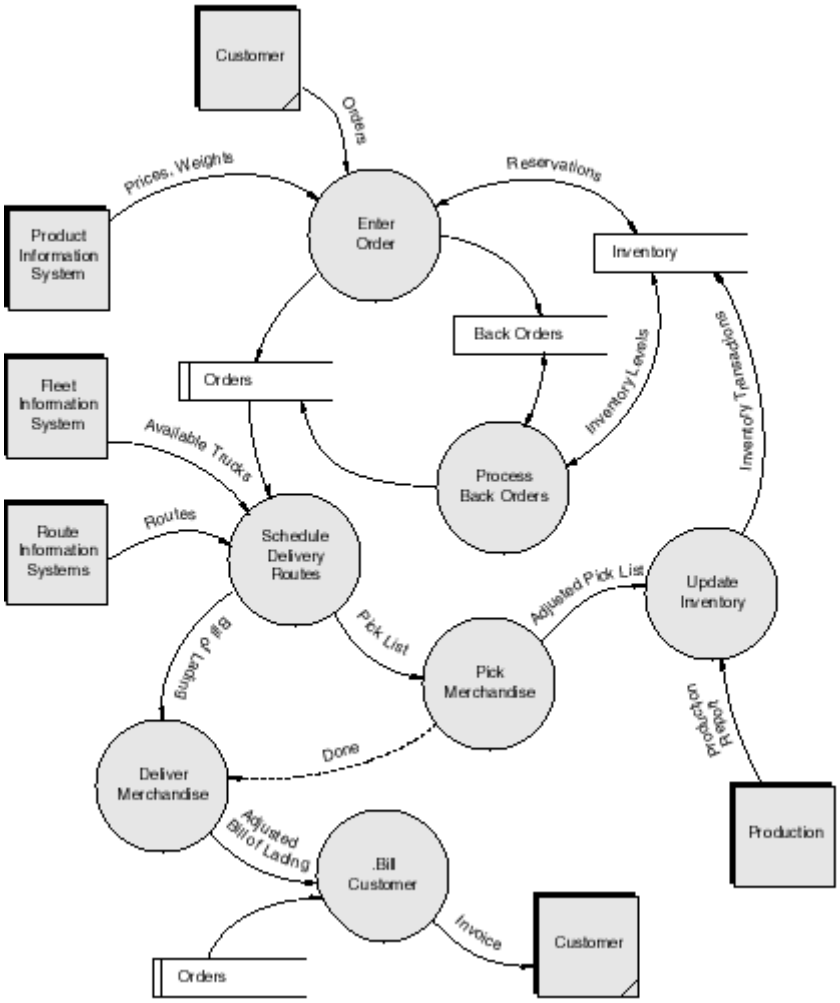


Exhibit 8.: A data flow diagram (DFD)

DFDs depict a flow of activities whereas the Warnier chart technique depicts static structures. An example is shown in Exhibit 8. Data flow diagrams are more powerful than Warnier diagrams but are a little more difficult to grasp and to draw correctly. They are best adapted to repetitive administrative work: processing a pile of customer orders that have come in through the mail, picking the orders from inventory, delivering them, mailing invoices and collecting payments.[12] (This was how most information systems were organized before the days of computers and in the early period of information technology; it is still used by utilities such as phone and electric companies for billing and payments; it goes under the name of batch processing.) As a result, DFDs are no longer as widely used as before. A variation that remains in widespread use is a diagram that shows the flow of activity down the page which is divided in columns, one for each organizational entity or system that intervenes in the processing of a single function. (This type of chart has been called “swimlane diagrams.”)

ERDs have a different purpose from Warnier charts and DFDs. They depict data structures: entities that correspond to things about which the system maintains data, such as customers, products, and orders; and relationships between them, such as the fact that each order belongs to one customer, but each customer may place several orders, and that each order consists of several products. In addition to identifying the entities and relationships, the model identifies the main attributes, or pieces of data, to be stored for each entity. An example is

12 Gane, C., and T. Sarson. *Structured Systems Aanalysis: Tools and Techniques.* IST, Inc., 1977

shown in . The ERD is an invaluable tool not only to point out dependencies and relationships among pieces of data, but also for the actual design of relational databases.[13]

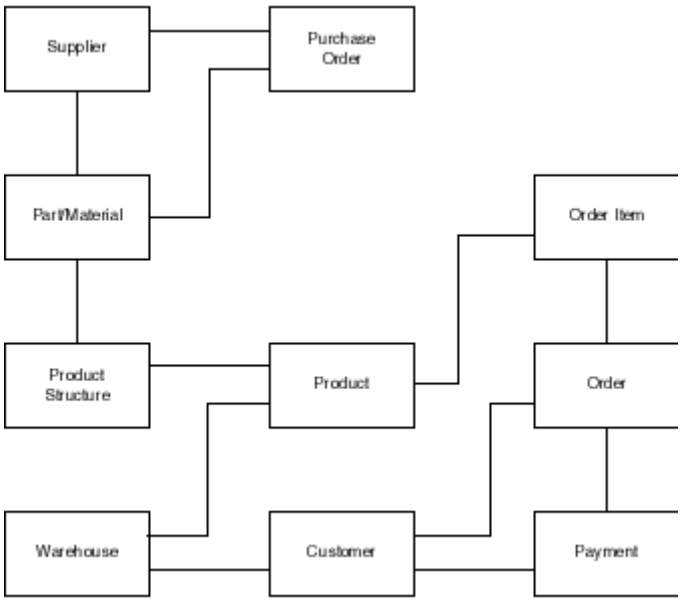


Exhibit 9.: ERD diagram

Other techniques are available for more specialized purposes, and you can also, if you have a specific need, develop your own. Bear in mind that you may then have to invest in training other people to use your homegrown technique.

Graphical techniques are indispensable tools for communicating and explaining requirements. But that is all they are. If a given technique falls short of this objective, do not use it. Above all, do not get caught up in the formalism and the stringent rules that some authors prescribe if these rules interfere with the main purpose. (But do remember that you are communicating not only with users and management, but also with your colleagues, who may have very specific expectations about the techniques you have selected.)

Quality requirements formulation

The formulation of quality requirements (performance, reliability, usability, flexibility...) should also be testable. This implies most often that you set a measurable, numeric goal, for example: "The system must be available to end users at least 98 per cent of the time in any given week". This cannot necessarily be tested directly before the system goes live. However, your design must include a way to keep track of whether the performance objective is being met.

The Software Engineering Institute has developed an entire design methodology driven by quality requirements that explains this process in detail (reference required).

Architecture

An architecture of any kind of complex man-made system, such as a cathedral or an office park, describes the overall structure of the system: its major components and how those components interrelate.

13 Chen, Peter. *The Entity-Relationship Approach to Logical Data Base Design*. QED Information Sciences, Inc., 1977.

The architecture of a business information system can be defined as the infrastructure and interfaces that enable system components (such as hardware, software and network communications) to work together to accomplish the system's purpose. The architecture is defined for the system as a whole, rather than for each of its components. In fact, the infrastructure and interface standards are usually shared among multiple systems. For example, all the web-based systems in an organization, whether Internet or intranet, might share the same architecture.

Generally, the system architecture is imposed on a project team by the organization's Information Systems (IS) department. Depending on the needs of the business system, the project team may request changes to the architecture, which are then typically implemented by a separate architecture team led by the organization's systems architect described in Chapter 3.

Sometimes, a new architecture is needed. This typically happens when an organization decides to use technology in a novel way to take advantage of a business opportunity. In such cases, a new architecture may be developed at the same time, or slightly ahead of, the business information system. A brand-new architecture can carry tremendous risks and should only be undertaken after careful consideration by the systems architect.

Importance of architecture

Could you build an information system without an architecture?

Yes, it is possible to build small, isolated information systems without a formal architecture, just as it is possible to build a log cabin without one. But as soon as you want to build a building which is larger or has multiple components—an energy efficient home, an apartment complex, an office high-rise—you need an architecture to show where the electric wiring, the plumbing, heating and air conditioning, the stairs and the elevators should go and how they should work together.

The main role of a system architecture is to help manage the complexity and size of modern business information systems. The architecture embodies important design decisions that have already been made. This is a constraint on the team, which is not free to make decisions that run counter to the architecture, but it also means that there is some kind of support within the organization—including knowledgeable people, development guidelines, reusable code, and implementation experience with the architecture—making the team's job easier.

The fact that many design decisions have already been taken has the following advantages:

- Risk reduction: first, the use of proven components and interfaces reduces the number of unforeseen problems arising during system development and operation; second, technological innovation is easier to control, because the architecture pinpoints the main risk areas, isolating them from what remains routine.
- Integration: well-architected systems are easier to integrate with each other and with legacy systems, because the architecture defines and describes in some detail the points at which two systems are allowed to interact.
- Productivity: an architecture is an investment—the stronger the architecture, the higher the productivity of the development team after an initial learning curve. Developers can in a certain sense "reuse" the technical decisions made by the system architect and less time is spent analyzing and testing alternative solutions.
- Reliability: the use of proven components and approaches that have already been successful in the specific environment in which you are working reduces the total number of errors and makes it easier to find those errors that do occur.

- Maintainability: standardized design and construction makes it easier for maintenance teams to correct, enhance and extend them to a changing business and technical environment.
- Predictability: an established architecture has well-known technical characteristics, making it relatively easy to analyze and predict performance, reliability, usability and other quality measures.

Selecting, extending or creating an architecture

On projects where both the business functionality and the technology are well understood, the project team will be best served by adopting an existing architecture, with minor modifications where needed. If the system is small enough, the architecture may not need to be specified in much detail. For example, a system may be simply defined as being web-based: this implies a host of technical decisions, as will be illustrated in later chapters.

Of course, if some element is absent from the architecture needed for a project, the team may be free to add it. This increases the risk and the cost, since no support will be available from the system architect and her team.

Where a new development project is driven by business innovation, it is best to adopt an existing, robust architecture. This reduces the development risks by not adding technical risks to the business risks.

On projects driven by technology—when someone has had a novel idea for applying technology in a way that has not been done before—you may well need a new architecture altogether. This architecture must at least be designed before the requirements process goes too far. In most cases, it will need to be developed and tested on a pilot project, to verify that the technology is workable and will help solve the business problem.

Finally, note that most of the systems "development" activity of existing IS departments is in fact maintenance. Maintenance requests very rarely have an impact on the system architecture.

In summary, in most cases, the system architecture will be given at the outset of a development project, or at least in its very early stages—certainly before requirements gathering is complete and before design has gone very far.

Design

To design a system is to decide how the requirements formulated as described in the previous section and approved by all the stakeholders are going to be implemented. The design process has two main focal points: the future users of the system (external, functional, or user design) and the IS professionals who will code and test it (internal or technical design).

External design

The first question of a design is, how will the users interact with the system?

Physical interaction

Physically, the interaction takes place via some device. In the generic case assumed in this chapter, this will be a personal computer hooked up to a network (often the Internet, but sometimes to the organization's internal network). The devices that are available are then a keyboard with a mouse, a screen with a given resolution, and a printer, most often based on inkjet technology. When the user population is the general public, you must be prepared to accommodate a variety of technical solutions: some users may have a lower screen resolution, some may have slow printers or no printers at all. The operating system and the browser—the software that enables the PC to communicate with the Internet—may also vary from user to user. You must therefore be careful to use the "lowest common denominator", the minimum capability likely to be available to all users.

Alternative means of interaction are available. Historically, users communicated with computers via forms that were filled in and transcribed to paper tape or cards in the form of patterns of holes. Early on, machine-readable data formats were developed such as optical character recognition (OCR) and magnetic ink character recognition (MICR). These had severe technical constraints but are still widely used in specialized applications such as check processing and invoice payments (where the account number and invoice amount are printed in OCR font on a slip that the customer detaches from the invoice and sends in by mail with the payment). The retail industry developed an industry-wide standard Universal Product Code (UPC) in the 1970s, embodied in bar codes on grocery packages, in books and on most other products. Similar in principle (but technologically quite different) are the magnetic stripes on the back of credit cards (often extended to membership cards that enable checking in and out as well as dispensing membership services). These forms of interaction require some specialized equipment, ranging from a hand-held scanning wand to an automated teller machine (ATM).

More recent developments include using touch-tone telephones for data entry. Automated response systems take phone callers through a series of menus to direct their calls either to a person with whom they can interact or to voice messages (either generic information useful to any customer or customer-specific messages, such as a credit card balance, generated on the fly). It is also possible for the caller to be asked to enter a series of digits—a customer number, a password or Personal Identification Number—or even alphanumeric text. Internet applications are emerging that are specifically designed to interact with mobile phones that use video and audio as well.

Another recent development is the use of Radio Frequency Identification (RFID), miniature transponders located in products, library books, electronic toll collection devices in cars and trucks, and cattle, to mention but a few uses.

For some very large systems, the development of a new input/output device may even be justified, as with the hand-held, wireless package tracking devices used by Federal Express.

In all cases, you must address two issues. First, how can the users get access to the interaction device? Are they likely to have a PC or a cell phone? Are they likely to adapt to using them as interfaces? (Cell phone texting may not be optimal for a system catering to retirees, nor may PCs with landlines connected to the Internet be the most appropriate solution in remote, non-electrified regions.) Alternatively, if the users are the organization's own workforce, what is the cost of equipping them? Second, whenever a user interacts directly with an information system, there is always room for doubts about the identity of that user, so you must decide what level of user authentication is necessary and how you can achieve it. There exists technology that is solely devoted to this authentication, such as fingerprint and retina scanner; other, frequently used solutions rely on password or PINs.

Interaction flow

The interaction flow follows directly from the functional requirements, and in particular the process descriptions or schematics (DFDs or "swimlane" diagrams) described in section 5 above. To revert to our default example of a web-based application using a PC with a mouse, a keyboard, a VDU and a printer, you need to develop detailed layouts of screens and printouts (you can use presentation software such as Powerpoint to mock up these) and descriptions of what the system does in response to each possible user action—entering text, clicking on radio buttons, check boxes and navigation arrows, or pressing the Tab or Enter keys, for example. For each item that the user can act on, also called a control, you describe all the possible user actions and for each of those, the system's

processing and possible responses. (This is called by some a Control-Action-Response, or CAR, diagram, an example of which is shown in Exhibit 4.)

CAR Diagram

Control	Control Type	Action	Response
About Us	button	click	hyperlink to About Us
Contact Us	button	click	hyperlink to Contact Us
Locations	button	click	hyperlink to Locations
Careers	button	click	hyperlink to Careers
Site Map	button	click	hyperlink to Site Map
Personal Finance	folder tab	click	display Personal Finance pane
Wealth Management	folder tab	click	display Wealth Management pane
Small Business	folder tab	click	display Small Business pane
Open Account	drop-down list	select	launch Open Account script for selected account type
User ID	text box	type	display text
Password	protected text box	type	display asterisks
Login Button	button	click	launch Check Password script with User ID and Password as parameters
Forgotten Password?	button	click	launch Password Reminder script with User ID as parameter
Select a Service	drop-down list	select	
Login Button 2	button	click	(see Login Button)
Learn More	button	click	hyperlink to selected service
Enroll	button	click	launch Enroll script for selected service
		Enter key	(see Login Button)

Exhibit 10.: CAR diagram

For each process, you have a choice between two basic interaction styles, system-driven and user-driven. System-driven systems take usually either a question-and-answer approach or a menu-and-forms approach. Take as an example an income tax filing application. One approach is to have the system ask a series of questions in a sequence that is determined by the designer, prompting successively for the taxpayer's name, ID number, marital status, income from salary, taxes withheld, and so on. This is a very appropriate method for an individual taxpayer, who does this only once a year and whose knowledge of the intricacies of what you have to report and on which form is limited.

For a tax preparer with multiple clients, a menu-and-forms approach may be better. The user can choose which form to fill in from a menu; in response, the system displays a screen image of the form and the user can fill it in, using the Tab key to move back and forth between fields and character, arrow, backspace and delete keys to fill in each field on the form.

A third form of system-driven interaction, often neglected because the interaction itself is minimal, is batch processing. The best example comes from periodic applications such as utility billing or bank statement printing. In these examples, the processing is initiated by a timer (every night at 1:30, or every Monday, etc...). An entire file or database is processed without human intervention and the result is distributed by electronic (email) or physical ("snail" mail) when the processing is complete. A useful trick for weekly or monthly applications that print out large volumes of data is to organize the process by slicing the input file into segments—as many segments as there are days in the period. For instance, for a monthly application, you could print out all the customers whose name begin with A one night, B the second night, and so on. This is called cycle billing and is in general use.

Other types of applications may be best served by a user-driven approach. One example is Internet navigation, where the user can look up information (for instance in an on-line product catalog) by navigating freely from web page to web page, following links or invoking embedded search functions. As he or she finds products that are of

interest, they are put in a shopping cart; at the end, when the user wants to confirm the order, the application takes on a system-driven character again, to enter delivery, invoicing and payment data.

Another example of a user-driven application is the spreadsheet. The user is given a variety of tools and functions: putting them together step by step enables him or her to complete the task at hand, whether it is to compute loan amortization calendars or organize the music collection of a church choir. Spreadsheet software is in fact so flexible that it is questionable whether to call it an application at all; it is more like a tool. In fact, many organizations create applications from spreadsheet software by distributing pre-formatted templates: budgeting and expense tracking are often treated this way. In this case, you could argue that the spreadsheet software is like a programming language and the template is the true application.

A final consideration is that of user identification and authentication. Most networked applications deal with resources that must not be tampered with—whether it is selling products or giving access to confidential information. In these cases, any user should be identified—who is it?—and authenticated—is the user really who he or she says? The usual procedure is to assign the user an identifier the first time he or she uses the application. This identifier is then stored by the system and permits subsequent activities. In addition, associated with the identifier is a password that only the legitimate user is to know. When logging in, the user supplies both the identifier and password. The combination of the two, if correct, authenticates the user who is then authorized to use the application.

The choice of style and flow thus depends on the application, but also on the user population. An inappropriate interaction style can cause a system to fail, especially if the users of the system are outside your control (e.g. the general population) and have a choice whether or not to use the system. This is a little less critical if the system is aimed at the employees of your organization, but a poorly designed interface can only have detrimental consequences: loss of productivity and lack of cooperation to cite only two.

The following are some guidelines for good interface design. The list is not exhaustive, but it contains some hints that you may not otherwise have though of; user interface design is a vast subject and you should reference the ample literature before starting.[14]

- Do not consider yourself a surrogate user, thinking, “This is how I would do it if I were to use the system”. You are not. You have a different educational and professional background. Get help from actual users to evaluate your design.
- Be consistent. Use the same controls and actions to accomplish similar functions. If you use feedback cues such as icons, color, fonts, sounds, etc..., make sure they are associated systematically with the same actions throughout. For instance, if you want to use a garbage can icon to discard unneeded data, then make sure you always use the same garbage can icon, in the same place on the screen, and with the same “Oops” function.
- Provide “Oops” functions at all points where it seems useful. Users are often afraid to use system functions, especially those they do not use very often, because they are afraid that the consequence of an error may be to break the system. Consistently having “Oops” or Undo functions available makes the system much more usable.

14 See for example: Shneiderman, Ben, and Catherine Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Addison-Wesley, 2004.

- Pay attention to users who have some physical impairment, such as color blindness, vision or hearing impairment, only one arm…Do not rely exclusively on color or sound to convey information.
- Make it very clear when a transaction, a data base update, or other irrevocable action has been performed, at what is called a commitment point.
- Do not overcrowd screens and reports, but don't commit the opposite error either: you do not want to devote a whole screen to a single item of data.
- Avoid distracting graphics and decorative elements, especially for high-usage systems. "Pizzazz" may be fun the first time, but users quickly get tired of it.
- Pay special attention to error processing. Make sure the user understands what is wrong and which steps to take to correct it. Anticipate that the user will try to perform unexpected actions and design the interface to react accordingly.
- Let the user choose the identifier and the password with as few constraints as you can. This is especially valid for external customers and occasional users. You can be more demanding with frequent users, especially within your own organization—assigning an ID rather than letting the user choose, forcing a mix of alphabetic and numeric characters for the password, making the user change the password periodically. (To force this discipline on occasional users—who may not use the system for months at a time—is counterproductive, because he or she is likely to compromise password security by writing it down.)

Data

In the typical case, data design consists of deciding how to organize the permanent data maintained by the application into a relational data base, made up of a series of tables—usually one for each entity identified in the entity-relationship model during the requirements analysis. Each line (record, tuple) of a table describes one particular occurrence of the entity type. For instance, in a customer table, there will be one record per customer; and each of these records will contain attributes or data items describing that customer: name, address, telephone number…

Each attribute is described with respect to its functional definition, its value domain, its physical representation on various media, and any restrictions on the operations that can be performed. This documentation decreases the number of programming errors and data base integrity problems during implementation and operation of the system.

For each entity, a key is chosen to identify it uniquely. A good key has no meaning outside of identifying the entity uniquely. In addition, a good key never changes its value for any given entity over the entire life of that entity.

For each relationship, an implementation is chosen. In most cases, relationships are implemented via foreign keys: an attribute of the entity is the key of another entity, serving as a pointer. For example, an order entity contains the customer number of the customer who placed the order. Then, when the order is processed, the application can access the customer data (for instance, to ascertain the delivery address).

One of the most important criteria for a good data design is that the data items (attributes) are assigned to the right table (entity). The objective is to avoid storing data redundantly. For example, in the order process outlined above, you want to store the customer's delivery address in the customer table rather than in each of the customer's orders in the order table. This isn't so much to save space as to insure consistency. If the same piece of data is

stored in several places, there is a risk that the values at some point will come to differ. Then, you are faced with the dilemma of choosing one, but which one?

The process of assigning attributes to the correct tables is called normalization. There is a set of precisely defined steps and criteria that you can follow to make this process almost foolproof. (Need reference here). The acid test is to run down all the functions of the system and ask yourself whether any of them would require updating the same piece of data twice.

Even though the occasion of all this work is the implementation of a single system, one of the main purposes of data base design is to ensure coherence of definitions and consistency of data values across all the systems of an organization. Thus, much of this work is ideally performed by a data administration section, which covers the activities of the entire IS department, not only the current project. If there is no data administration section, then one of the analysts on the project plays the same role, covering only the project.

Media content

The last aspect of external design that we'll cover is media content design. This refers to the design of applications whose main purpose is to convey information to the application users, whether it be products for sale, press releases, academic articles or similar content.

Media content can be in several different forms: text, still images, animations, video or audio, or any combination of the three. Designing the content and the presentation of the information is the affair of communications specialists. Your responsibility as a system designer is primarily to provide for the storage and retrieval of each component as needed. You must also decide how the media content will be maintained (say, by changing the photographs when a product is modified) and how to ensure that all the navigation links available to the user work correctly. It is easy to radically modify the content of a page; it may not be so easy to locate all the other pages that point to the one you've modified, to see whether those links still make sense. It is even worse when you suppress a page without suppressing the links to it, leaving the user high and dry in the middle of navigating the web.

Internal design

The internal, or technical, design of an information system is the description of how the system will be implemented. Most of the work done here is intimately dependent on the architecture adopted, but we can identify a few general principles that apply across the board. This will be the simplest and the most useful if we look at the end result of the design rather than the process to get there,

The internal design identifies and describes all of the components of the system: hardware, software, communications protocols. For each component, it describes which functions are fulfilled by, or allocated to, that component. The decomposition/allocation should meet the following criteria:

- Completeness. Each functional requirement (see section 5 above) is allocated to a component.
- High cohesion: Each component performs functions that belong logically together. For example, extending and totaling the lines of an invoice and the computation of sales tax arguably belong together; checking a user's password does not.
- Low coupling: Each component interfaces as simply as possible with other modules. For instance, an on-line ordering application might have a module to process credit card payments. The invoicing program

should just communicate the invoice amount to the credit card payment module, which, after processing the credit card data entered by the user should simply indicate to the calling module whether the payment was accepted or not.

One particular strategy of allocation is worth bringing out: object orientation. Whereas traditional systems allocate functions to modules based on the processing flow, object orientation uses the data structure as the criterion. Each object (corresponding to an entity in the entity-relationship diagram or a table in a relational database) is packaged with its code The functions that such an object can perform are typically to modify the value of an attribute (say, the customer address) or to send the value of an attribute back to the requesting program. The application still needs to implement a flow of processing (for example, log in, followed by browsing of the catalog, depositing items in a virtual shopping cart, checking out and paying). The object-oriented paradigm can be made systematic by considering each such process as an object in itself, even though it is not an entity in the entity-relationship diagram.

Object orientation has the advantage of making it easy to meet all three of the criteria enumerated above. The structure of the system is simpler and more reliable than traditional systems design. The disadvantages are that performance may be a problem and that the approach is less familiar, especially for IS professionals of a certain age.

Code

Completing the design process results in a series of detailed specifications. The translation of these specifications into a form that can be understood by the technological components of the system—PCs servers, mainframes, communications gear... This is called coding. (It used to be called programming, but it entails in fact more than writing programs—for example, database specifications).

How this is done is specific to each environment and will therefore not be further described here. Each programming language has its own peculiarities, and each IS department has its own habits; the best organized departments have in fact established coding standards to be followed by all team members. This has the advantage of making work more interchangeable: if everybody uses the same style, work on one component can be taken over by anyone. Maintenance is also greatly facilitated by standards.

Test

Why we test

We test a software program, an application, an entire business information system to detect errors—things that don't work the way they are supposed to. If the system doesn't work properly, we find out why and fix it (or abandon it, if we judge it too expensive to fix).

We test other things, too. New airplanes are tested before they are declared airworthy. Before buying a new car, you probably want to test-drive it. In the United States, the Food and Drug Administration mandates extensive tests, lasting several years, of any new medication before it can be prescribed or sold over the counter. Sometimes, the tests are insufficient, as with the case of the X-ray equipment that would, under certain circumstances, irradiate patients with 1,000 times the normal dose; as a result, patients died. In other cases, it isn't possible to test a product under realistic conditions: the Lunar Excursion Module could not be tested on the moon before the real moonshot.

Software—beyond a few hundred instructions—becomes so complex that its behavior cannot be accurately predicted just by inspecting it. Errors creep in through inattention, misunderstanding, or other human weaknesses. Just as no one would want to fly an untested plane, no one would want to use untested software.

Testing is a part—a critical one—of a more general process called software defect removal.[15] (Other techniques include inspection, prototyping, and automated syntax checking.) Even used in conjunction, these defect removal techniques cannot guarantee defect-free software. The most thorough of tests cannot address all the possible values of input data against all the possible states of the database. A defect may remain latent for several years until some highly unlikely combination of circumstances reveals it. Worse, the test itself may be defective: after all, the test, too, is designed and executed by fallible human beings.

In traditional systems development methodology, testing is seen as a discrete activity, which occurs once development is substantially complete. This view causes problems, because the cost of correcting errors becomes exponentially larger the more time passes during which the error remains undetected. An error that is discovered by its author as it is committed can be corrected immediately, at no cost. A requirements error discovered during operation may invalidate the entire system—and carry business interruption costs in addition. Experts estimate that correction costs can be a hundred times higher when the error is discovered after the system has been delivered. The reason for these high costs are that the longer the time between committing an error and discovering it, the more work will have been done and therefore have to be corrected or scrapped and recreated.

The V model of verification, validation, testing

Before explaining how testing is done, let us place it in a wider context. At about the time that the IS community discovered the high cost of error correction, the United States Department of Defense devised a set of processes to control the work of contractors on large, complex weapons systems procurement projects. Because of the increasing software content of modern weapons systems, these processes were naturally adapted to the world of software development. They also put testing in proper perspective as one means among many to ensure software quality.

Because of the shape of the graphical representation of these processes, the name V model was coined. An example of a V-model that applies to business information systems development is depicted in Exhibit 9.

15 Dunn, Robert. *Software Defect Removal*. McGraw-Hill, 1984.

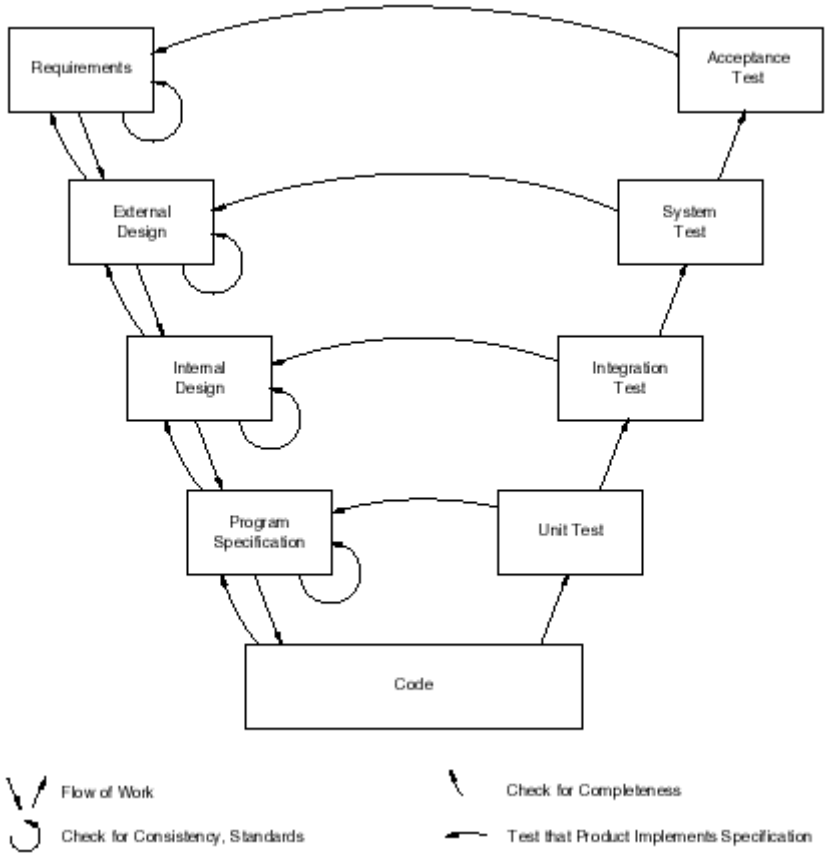


Exhibit 11.: A V model

The left-hand side of a V depicts activities of decomposition: going from a general objective to more and more detailed descriptions of simpler and simpler artifacts (such as program code in a business system). The right-hand side depicts integration: the assemblage into larger and larger components of the individual pieces of code created at the bottom angle. As each assembly or subassembly is completed, it is tested, not only to ensure that it was put together correctly, but that it was also designed correctly (at the corresponding level on the left-hand side of the V).

The final characteristic to be noted is that each phase corresponds to a hand-off from one team to another (or from a contractor to a subcontractor, in the original Department of Defense version of the V model). In the case of systems development, what is handed off is a set of deliverables: a requirements or design document, a piece of code, or a test model. The hand-off is only allowed to take place after an inspection (verification and validation)of the deliverables proves that they meet a set of predefined exit criteria. If the inspection is not satisfactory, the corresponding deliverables are scrapped and reworked.

Let us define verification and validation. **Verification** checks that a deliverable is correctly derived from the inputs of the corresponding activity and is internally consistent. In addition, it checks that both the output and the process conform to standards. Verification is most commonly accomplished through an inspection. Inspections involve a number of reviewers, each with specific responsibilities for verifying aspects of the deliverable, such as functional completeness, adherence to standards, and correct use of the technology infrastructure.

Validation checks that the deliverables satisfy the requirements specified at the beginning, and that the business case continues to be met; in other words, validation ensures that the work product is within scope, contributes to the intended benefits, and does not have undesired side effects. Validation is most commonly

accomplished through inspections, simulation, and prototyping. An effective technique of validation is the use of traceability matrices, where each component of the system is cross-referenced to the requirements that it fulfills. The traceability matrix allows you to pinpoint any requirement that has not been fulfilled and also those requirement that need several components to be implemented (which is likely to cause problems in maintenance.)

Detailed descriptions of various do's and don'ts of inspections are available from most systems analysis and design textbooks, as well as from the original work by Ed Yourdon. (reference here)

Testing, the third component of the V model, applies to the right-hand side of the V. The most elementary test of software is the unit test, performed by a programmer on a single unit of code such as a module or a program (a method or a class if you are using object-oriented programming). Later on, multiple units of software are combined and tested together in what is called an integration test. When the entire application has been integrated, you conduct a system test to make sure that the application works in the business setting of people and business processes using the software itself.

The purpose of a unit test is to check that the program has been coded in accordance with its design, i.e., that it meets its specifications (quality definition 1—see Chapter 1). It is not the purpose of the unit test to check that the design is correct, only that the code implements it. When the unit test has been completed, the program is ready for verification and validation against the programming exit criteria (one of which is the successful completion of a unit test). Once it passes the exit criteria, the program is handed off to the next phase of integration.

The purpose of integration testing, the next level, is to check that a set of programs work together as intended by the application design. Again, we are testing an artifact (an application, a subsystem, a string of programs) against its specification; the purpose is not to test that the system meets its requirements. Nor is the purpose to repeat the unit tests: the programs that enter into the integration test have been successfully unit-tested and therefore are deemed to be a correct implementation of the program design. As for unit tests, the software is handed off to the next level of testing once the integration test is complete and the exit criteria for integration testing are met.

The purpose of system testing, the next level, is to check that the entire business information system fulfills the requirements resulting from the analysis phase. When system testing is complete and the exit criteria for system testing are met, the system is handed off to IS operations and to the system users.

How to test

Prepare the test

Let us start with unit testing. To unit-test a program, you: first create test cases. A test case is the description of a unique combination of inputs and data base states. The secret to good, efficient testing is to plan the test cases so that each is unique and together they are exhaustive. In practice, you develop unit test cases from the program specification to make sure that all the processing rules and conditions are reflected in at least one test case. You create a separate test case for each state of the data base that could have an impact on how the program works. Pay special attention to boundary conditions, those that are abnormal or at the edge of normality. For instance, when processing a multi-line order, pay special attention to the first and the last lines of the order. Include orders that have one line, or none at all, or a very large number. Consider also abnormal cases, such as that of trying to create an order for a customer who is not recorded on the data base. Create different orders for different types of customers. Consider what happens for the first order of the day, when the order file is empty, or after a heavy day, when the order file becomes very large. Create order lines with very small and very large amounts. Are negative

amounts accepted? It is as important for a program to raise the proper exceptions for erroneous inputs as it is to check that valid data is processed correctly.

Once you believe the test cases are complete, cross-reference them against the program specification to ensure that each specification is addressed.

Once the test cases have been created, prepare the test scaffolding, the environment in which the test will be run. If the program is a user interface program, most of the test data needs to be laid out in a script for the user to follow. If not, input or transaction files must be loaded. This can be done with a test data load utility or it may require a specially written program. Next, the test data base (including web site content, if applicable) must be created. The best way to do this is to have the project team maintain a master database of shared test data from which each programmer can extract relevant tables, rows and files , modifying them where necessary for the purpose of the test. (You cannot use the database directly, since your program is likely to update it, thus creating unexpected conditions for everybody else.) If the project team doesn't have a common test data base, create your own, as you did for input and transaction data.

Next, consider what stubs and drivers you need to test your program. In many cases, a program needs to communicate with other programs, some of which may not have been created yet. A stub is a small program that can be called by the program being tested and can simulate the services rendered by the called program. A driver works the other way around: if your program is designed to be called by a program that doesn't exist yet, and you must write a simple program that activates yours.

Another useful tool is one that does screen capture. This tool records keystrokes and screen display changes; it enables you to repeat exactly the same test and to compare its output before and after you make modifications.

Before running the tests, you should prepare the expected results for each test case in each test cycle. Most developers do not do this. They believe that it is sufficient to run the test and review the output to see if it is correct. This is insufficient, for two reasons. First, people have a propensity to see positive results. Subconsciously, the brain will often rationalize an error and tend to think that the result is actually correct when it isn't. Second, preparing expected results is an excellent way to review the completeness and relevance of the test data.

Ideally, you would load the expected results on to a file in the same format as that produced by the test. Then, the comparison between expected and actual results can be done electronically—a much more reliable approach than trusting human faculties. A keystroke/screen capture facility comes in handy here, especially for programs that are part of the user interface. But even if you don't have such a tool, comparing a predetermined expected result against an actual one is a lot more reliable than just viewing the actual result to decide whether it is correct.

Preparing expected results is time-consuming and not much fun. Ultimately, however, by avoiding false positive results and by making it easier to repeat the same tests after correction, you save time and increase quality. However, since the cost is incurred earlier than the benefits materialize, project management has a tendency, when put under time pressure, to shortcut this indispensable investment.

Execute the test and record the results

With all the preparation that has gone on in advance of the unit test, executing it and recording results is as simple: just do it.

Well, not quite. If you do not have a scripting tool and you are testing at the user interface, you must observe an iron discipline of recording what you do and how the system responds, by taking notes and printing screens—or even photographing screens using digital photography or Polaroids. This is easier said than done.

Find and correct errors

When you have executed a test, review the output, either by examining the results of a program to compare the expected v. the actual results, or by scanning the results yourself. A discrepancy between the two may have one of the three following causes:

- The discrepancy is only apparent: the results are just in a slightly different format (such as a zero being unsigned in one and signed in the other). Correct the expected results so that the discrepancy does not repeat, wasting time every time the test is rerun.
- The discrepancy may stem form an error in the test data. This occurs quite frequently. The expected result itself may be wrong; there may be a clerical error in the test data; or the test data may not correspond to the intent of the test case. In the first two instances, the error is easily corrected. In the third case, it may be a little more complex, because the problem is a sign that the program specification may have been misunderstood.
- The discrepancy is due to an error in the program. In that case, correct the program and rerun the test. You must also make sure that the correction has not disturbed a part of the program that has already been tested, if necessary by rerunning all previously run test cycles. This process is called regression testing. In practice, you do not run a complete regression test every time you find a small error in your program—you can postpone it until the end. However, before you can hand off the program to the next phase, you need to rerun all the test cycles once after all your corrections have been completed.

Executing the other tests

The V-model introduced describes the flow of development and integration, specifying that for each development phase, there is a corresponding test. The V-model is only a schematic representation, however; in real life, more tests are required than those shown, and some things are difficult to test for—in particular the quality requirements, performance, usability, reliability and maintainability, and may require extensive work.

How much testing is enough?

A central question is, when are you done testing? We have already said that it is impossible to guarantee that a system is free of defects, no matter how long you test it. So when do you stop?

This is essentially an economic question. It can be formulated in two ways:

- The benefits of delivering the system, including its residual errors, outweighs the risks, or
- The cost of discovering additional defects is greater than the savings gained from finding and removing them.

Either case boils down to a judgment call. This is most apparent in the first formulation, which refers explicitly to risk. In the second view, we need to add the word “probable” in front of both the cost and the savings, to emphasize that it is hard to predict how much effort you will need to find the next bug and indeed how grave that defect is going to be.

The decision is illustrated by the defect removal curve in Exhibit 10.

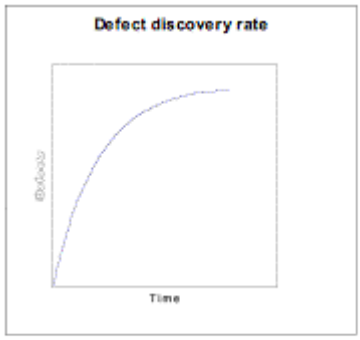


Exhibit 12.: Defect removal curve

Prototyping

According to the V-model, each test is assumed to be executed on the actual code after it has been developed. In many cases, preliminary tests can be devised that do not require a fully developed system. Rather, they work on a "synthetic" application, one which has the same technical characteristics as the application under development, but which doesn't really do anything of functional value. In the V-model, such tests are identified as prototypes; setting them up and running them is called prototyping. The benefit of prototyping over testing on a full system is that you can do it earlier; any correction you need to make to the specification you are prototyping will cost much less to make than if you wait until actual testing is possible.

If you create a prototype to test out some specification, it doesn't mean that you can omit the corresponding test later on. In fact, the test is still needed, because the implementation of the specification may be incorrect. But you should be able to run the test, and especially post corrections, in much less time than if you don't prototype.

A corollary of the view of testing that we have just described is that any deliverable created by an activity on the left side of the V-model is liable to be tested. It should therefore be couched in testable—concrete, operational—terms. A requirement that states, "The system must be extremely reliable," is not useful. Rather, use descriptions such as, "The system must be available to users during 99 per cent of its scheduled uptime" and "The system will produce no more than one Type 0 System Change Request per week.." The deliverable is not complete unless it also is accompanied by a test model, containing at least a description of the testing approach and ideally a set of test cases.

Another corollary is that test data have to be designed. You cannot use live data—data extracted at random from an existing application, at least not exclusively. (An exception may be made for volume testing.)

Maintain

Problems with the waterfall life cycle

When the waterfall life cycle was first published, development project planners used it as a generic systems development plan, translating each activity as a separate bar on a GANTT chart which couldn't be started until the previous activity was completed. This approach caused a number of difficulties and has largely been abandoned. The waterfall life cycle is now used almost exclusively as a training framework, because it distinguishes between the

different kinds of activities that take place during systems design—although it does not represent an actual chronological sequence of events.

The main reasons for the inadequacy of the waterfall concept are related to the need to manage changes to the system. We have already mentioned that as soon as a system has been launched, requests for changes start accumulating. In fact, these requests start almost as soon as the first systems requirements have been agreed to, whether it is because the business and technological environments keep changing or because verification, validation and testing (including prototyping) uncover defects that need to be corrected.

Specifically, the problems that arise are the following:

5. Projects take too long. By placing each activity on the critical path rather than doing some of them simultaneously, the elapsed time of a project cannot be shortened beyond a minimum of a year or more in the typical case. During the long development phase, costs are incurred; benefits do not start accruing for several years. This interferes with the realization of the economic benefits supposed to be brought by the system.
6. Requirements have to be finalized—"frozen" is the term most used—too early in the project. During a year or more of development, the world does not stand still, and what may have been useful at the start of a project may be irrelevant a year or two later. This became especially true when information technology started to be used as a competitive weapon in the early 1980s.
7. Users frequently do not get a full understanding of the system until it is almost done. Accepting a requirement or a design on paper is not at all the same as seeing and using a real system, and users who see a system in operation for the first time may quickly discover major problems. As a result, either the system has to be delayed by major corrections or it will be released on time, but with known defects.
8. Once the developers have handed the system over to operations, they quickly find, contrary to their (somewhat unrealistic) hopes and expectations, that they can't just forget about it. As people start using the system, they find undesirable system behaviors such as bugs, poor performance, etc. They also develop ideas for additional functions to add to the system, whether these ideas for modifications come from their own experience in using the new technology or are due to changes in the environment—imposed by the competition, regulators, and society at large. In other words, maintenance hits.
9. In fact, maintenance is already needed during development. As programmers test, they find bugs that have to be fixed. Some bugs are just errors they have made themselves, but sometimes errors are due to poor design or to a misunderstanding of the real requirements. Errors may in fact be found at any point during the development process. To correct these errors, you may have to go back one or several phases in the life cycle to scrap some work previously done and rework it. (Royce knew this: he provided the back arrows on his diagram to address this need. But project planners did not truly understand the problem. As a result, the maintenance activity—or bug fixing—that goes on in a waterfall project is usually conflated with the testing activity.)
10. On innovative projects, another difficulty arises. The team members may not know how to perform their designated activities in the best way, because they have never done it before. When this happens, designers

experiment by quickly implementing and testing a possible solution, thus anticipating the later stages of the waterfall life cycle.

Development vs. maintenance

All of the points enumerated above have been extensively analyzed and solutions have been proposed. Most of the solutions result in a shorter initial period of systems development—say, three to six months—after which periodic new releases of the system provides additional functionality every three months or so. While many authors had started promoting Rapid Application Development (RAD) in the early 1990s, this did not become generally accepted until the race to create web-based applications towards the end of the decade. However, by the time the e-bubble burst in March, 2001, RAD and incremental releases had become the norm for all but the most ambitious systems development projects. Even though the pace of creation of web applications has leveled off, we have not seen a return to large, big-bang projects.

A consequence of this trend is that maintenance has now become the normal mode of application creation. In the past, you might have a large team working for three years on the initial development of an application, followed by a team reduced to one-half or one-third the initial team, working for ten to twenty years to do maintenance. This would imply a ratio of initial development to maintenance of 1 to 2 or 1 to 3. With the new model, the initial development would take place with a smallish team over three months and maintenance would still last for ten or twenty years with a team of the same size or slightly smaller, giving us a ration of development to maintenance of 1 to 50 or 1 to 100 in extreme cases.

This tremendous shift has not attracted the interest of researchers and authors. IS departments have realized it, however, and the best-run organizations articulate their work around maintenance as the default activity, initial development being almost an afterthought.

In the new model, you do not have a unidirectional flow from requirements (supposed complete) through design to construction. Rather, you have system change requests flowing from any part of the process (wherever an issue has arisen) to any other part of the process (wherever the request has an impact).

Most IS departments use the following classification for their system change requests:

Type 0: Emergency fix. The system has a critical error and is not operational; no bypass is available.

Type 1: Error correction. The system does not work according to specifications. A temporary bypass is available.

Type 2: Enhancement. The system does not work satisfactorily: there are reliability, usability or performance issues

Type 3: Extension. The system needs new or modified functionality to respond to external changes or to increase its business value.

Type 4: No system change required. The request is one that can be satisfied by more information or by routine action (such as a password reset). This category is included here only for the sake of completeness.

In the standard case, Type 0 requests are addressed immediately. Type 1 and simple Type 2 requests are bundled together in what is often called a “dot” release, typically monthly (or in some cases weekly). More complex

Type 2 and all Type 3 requests are bundled in major releases, typically every three months (or in some cases every month).

The resulting life cycle might look somewhat like .

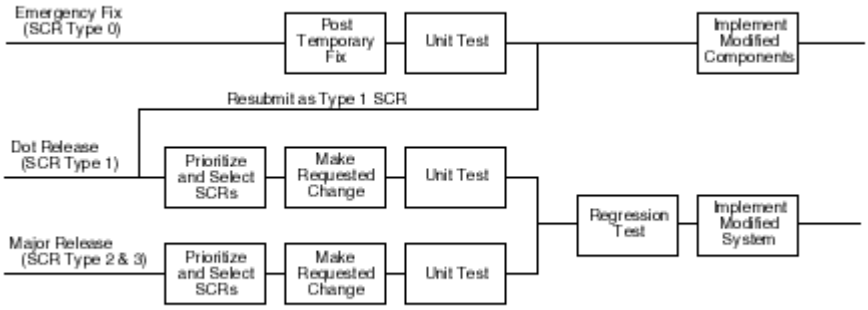


Exhibit 13.:A systems life cycle

Testing during maintenance

One of the major problems with maintenance is that modifying a system, whether by correcting a component or extending its functionality, there is a high risk of disturbing, unintentionally but unavoidably, functions that do not need to be (and in fact should not be) changed. This accentuates the need for regression testing: testing that not only does the modification work, but everything that was not modified continues to work as before. Any but the most trivial change may require you to rerun an entire system test.

During maintenance, you will create new programs or even subsystems (to address Type 3 SCRs—functional extensions) and you will modify existing programs (typically to address other types of SCRs, but sometimes also Type 3s). Unit testing new programs during maintenance is no different from doing it during development. Testing program modifications, however, can be a little less rigorous. In general, you will test only that the modification works, but perform no regression testing (at the unit level) to check that the unmodified parts still work as specified. This is possible because the risk of errors is relatively low and because the correction-induced errors you may have caused will most likely be caught by the system-test level regression test, which is not optional, whether your maintenance release contains major new functionality or not.

For emergency fixes (Type 0), the same rule holds. Regression testing of emergency fixes just is not cost-effective. However, in the next scheduled release, the type 0 fix must be backed out and resubmitted as a type 1 fix. This way, it will be re-implemented, unit tested at the program level and regression tested at the release level, just like other SCRs.

Correspondingly, the regression test at the system test level becomes more important. This has several consequences:

- The regression test is time-consuming and costly. However, it does not cost much more to test many modifications than just a few. This is a powerful argument for organizing maintenance in releases, as described in the previous paragraph.

- The system test from the first release becomes the regression test for the following releases. This is, in fact, part of the “product” as delivered to operations; no system is complete without it.
- When planning a new release, one of the first deliverables is the revised regression test model. For releases that have a lot of new functionality, you may also want to schedule other tests, depending on how you judge the risk of an unsatisfactory result. These tests are likely to be fairly infrequent and also fairly release-specific, and the maintenance of a regression test for these purposes is usually not economical.

4. Business process modeling and process management

Editor: Franz Lehner (University of Passau)

Learning objectives

- understand why process management is important
- be able to explain what a process is and the different types of processes
- state the difference between process orientation and other organizational principles
- understand the necessity of process modeling
- be able to draw simple ePK-diagrams
- be able to discuss the use software for process modeling
- be able to name and briefly summarize some methods used for the analysis of processes

Understanding the key steps of process management

The development of process management

To improve and maintain organizational efficiency, there must be a constant willingness for innovation and reorganization. Information and communication technology has become an indispensable aid and medium for efficiency gains. Organizational science and information systems are close partners in this pursuit of efficiency. A focus on process thinking is a feature of the modern organization. As a result, techniques such as Business Process Reengineering (BPR), Business Engineering (BE), Business Modeling (BM) have emerged to support the methods and goals of process management. In this chapter, the fundamental principles and perspectives are conveyed and the state of technology and its practical application are introduced.

Since the beginning of the 1990s, enterprises have put greater emphasis on analysis for optimizing business processes. A clear trend is observed is the shift in attention from a functional orientated organization to an alignment around business processes. In functional orientated organizations, the traditional functional departments such as procurement, production, logistics, finance, IS, marketing and so forth are dominant. Now, there is a new way of thinking: “The endeavour for optimum and profit generating satisfaction of the clients wishes should (…) come from a process orientated organization structure, in which the position and department formation would be conceived considering specific requirements in the course of the process for performance in the organization” (Striening 1988, 28). The goals are more precisely explained in the following section on the Concepts of Process Management, namely the optimization of the combined work of all functional areas independent of the

organizational structure and therefore there it is more common to find an overlapping of functions, areas, and departments.

Business processes are value added activities that produce a strategically valuable output for an organization. Business Process Management is the optimization, automation, as well as specific regulation and improvement of a business process. The tasks of Business Process Management, can be divided into Process Identification, Process Modeling, Process Analysis and Process Management. Process modeling documents the identification of processes in a standardized form, which is usually supported by software. These process models form the basis for process analysis and the adaptation or redesign of a process. Process management should ensure systematic planning, steering, and supervision of the execution of a process. At the same time, the process results will be controlled by the previously established measurement system that used to monitor performance and will be the basis for future process adjustments.

The general goal of process management is to increase client (customer) satisfaction as well as to improve the efficiency of business processes. Consequently, the firm should gain a productivity increase. Both a client and value added orientation is necessary. A client can be the final consumer of a product or a person within an organization.

Process management can be considered from three organizational perspectives: strategic, operational, and administrative:

- Strategic management is concerned with creating the general framework in which the business process are executed.
- Operational management creates the workflow for formalizing and automating business processes and applying information systems, as appropriate, to support process execution.
- Administrative management, typically the role of the IS unit, is responsible for developing the organizational and information systems infrastructure to support business processes.

What is a process and what are the different types of processes?

There are many definitions of a process Lehner et al. (2007), and we will discuss a few of these (see Exhibit 14). The selected examples illustrate that each perspective on a process each emphasizes a particular trait or characteristics of a process. As a result of these different views, there is no single definition.

Exhibit 14.: Definitions of a process

Definition	Source
Repeated tasks that arise in the performance of an assignment in different sectors of an enterprise. They are the recurring results of for an individual task with: • measurable input • measurable added value • measurable output	Fischer (1993, 312)
A succession of tasks, activities, and performances for the creation of products or services, which are directly connected with one another and in their sum determines the business management, technical production, technical administration, and financial success of the enterprise	Striening (1988, 57)

A manual, partly automated, or fully automatic business activity, which is performed following definite rules and leads to a particular goal. A business process creates, in this way, a valuable result for the client.	Oberweis (1997)
The content, timing, and natural sequencing of an object necessary to complete a business management function	Vossen and Becker (1996, 19)

Most of the definitions have in common that there is measurable information and measurable results, a definite beginning (this means that it has a starting occurrence or initiation) and an end, as well as a demand for value or a contribution for the creation of value. The main features of a process are depicted in Exhibit 15. Process management supervises all aspects of a process from the initial event until its completion. It goes beyond departmental or functional barriers, and sometimes beyond organizational boundaries, to cover the entire process. Process management is an integrated approach to managing processes.

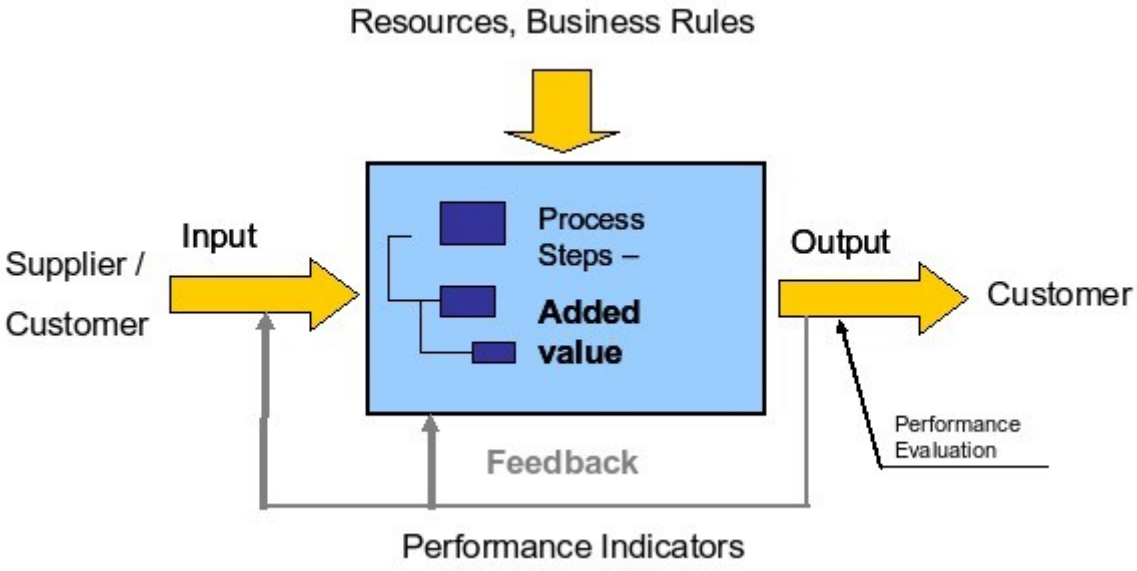


Exhibit 15: The structure of a process (Tonchia and Tramontano)

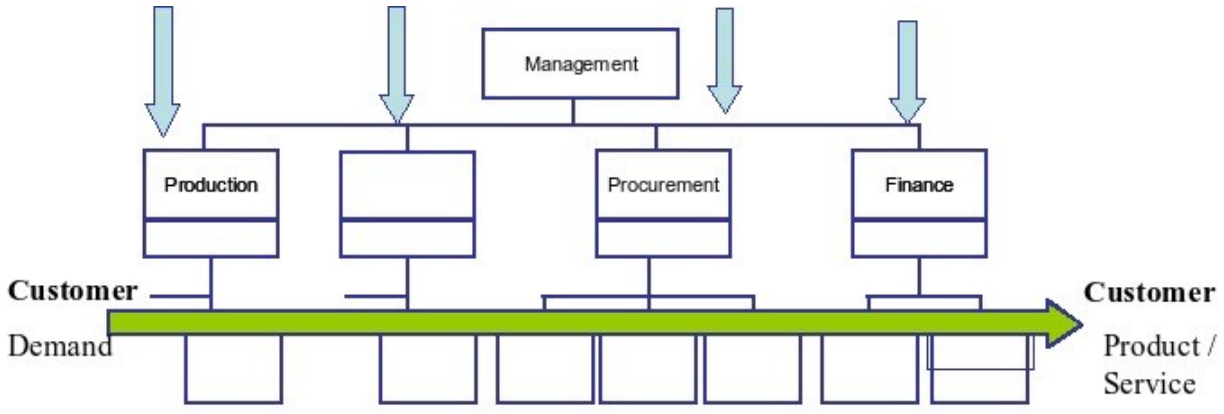


Exhibit 16.: Business process example

The classification of processes is another key aspect of process modeling. One can broadly distinguish between material processes (e.g. procuring, producing, storing and financing) and formal processes (e.g. planning, controlling and decision-making). Exhibit 20 lists a few examples of processes often found in enterprises.

A further distinction is whether a process is a main, service or support, and management and leadership process. Main processes are those that directly create value for an external client. They can be product related (e.g. production, R&D) or client related processes (e.g. order completion, distribution, acquisition). Main processes are sometimes also called core processes, because they are central to the strategic goals of the business. They are the means by which the business creates value. Service processes deliver value to internal clients and support other processes. (e.g. personal recruiting, maintenance, quality, and security). Management and leadership processes act upon main processes. Planning, accounting, and budgeting fit into this category.

Processes can be further subdivided into strong or weak structured processes. Strong structured processes are frequently repeated, structured data, and well documented. They are the day-to-day transactions of an organization. They are well suited to conversion to electronic workflow and document management systems. Weak structured processes are, on the contrary, characterized by low predictability and infrequent repetition and unstructured data. One uses groupware and communication systems to support weak structured processes.

Process orientation as prerequisite for process management

All over the world, one can observe an economic and societal restructuring. A dynamic business environment and the pressure for firms to increase their competency is forcing enterprise to develop new abilities. Typically, firms adapt to a new environment by following a learning-process to gain efficiency and flexibility. To increase their competitiveness, many enterprises reorganize their systems and structures. They apply one of the many “salvation plans”, such as “Business Process Reengineering”, “Business Process Design”, “Business Process Optimizing”, Work Flow Management”, “Business Modeling” or “Business Process Oriented Organizational Set-up”, just to name of few of them. The choice of methods is good, however careful examination of each of these methods reveals that they all are based on the same central foundation. The focus is on identifying business processes and modeling them. They have a common process orientation.

Process orientation is booming in information systems, and has also become an important basic component in many organizations. Indeed, process thinking has a long tradition in business administration. The innovation in process orientation is in the expanded context and in the use of computer software. Business processes can be analyzed in their entirety throughout the organization, and this analysis is shaped and steered with the support of process modeling software. Process models document the business processes and support their systematic analysis. As a result of this analysis, some processes can be implemented electronically using specialized software, variously know as Work Processing Systems, Workflow Management Systems, and Business Engineering Tools.

Process orientation, as it is understood today, is strongly connected with strategic thinking and organizational development (OD). Porter's value chain, probably the most well known of the general process models, is a highly useful tool for strategic thinking.

Process orientation gained considerable additional attention with the rise of the Business Process Reengineering (BPR). The work on BPR of a few American authors (e.g. Hammer and Champy 1993, Davenport and Short 1990, Davenport 1993) started a worldwide trend, moved the discussion of processes in to the center of much

organizational thinking about efficiency. Organization sought to use BPR to achieve reduce the complexity of internal operations and more efficiently achieve organizational goals.

To achieve the promise of BPR of simplifying organizational processes, it is of crucial to have an exact and thorough modeling of the targeted business processes. Process modeling is a methodology that explicitly or implicitly helps an enterprise to:

- understand the nature of its various processes (business processes, service processes etc.)
- recognize the resources necessary for the execution of each process
- rearrange the system of processes and resources (i.e. process orientation)
- to permanently improve processes

In general, an improvement in process can be reached through simplifying and standardizing the elements of a process and their relationship to each other. Through an automation of a chain of activity, for example, through new technology and information systems, the efficiency of the processes can often be increased. Also it is often possible to achieve efficiency gains by restructuring single parts of an enterprise. For example, by changing the order of the activities of an internal process or the sequencing of the procedures of a process.

The traditional organization is orientated around functions, hierarchies, competencies, departments, capacities and so forth. This leads in times of market change and competition to a critical disadvantage due to inflexibility, slow adaptation, and a loss of customer focus. Process orientation is a solution to these problems because it pays attention to products, value chains, and process connections. The goal is to ensure the processes are performed efficiently and effectively irrespective of organizational boundaries.

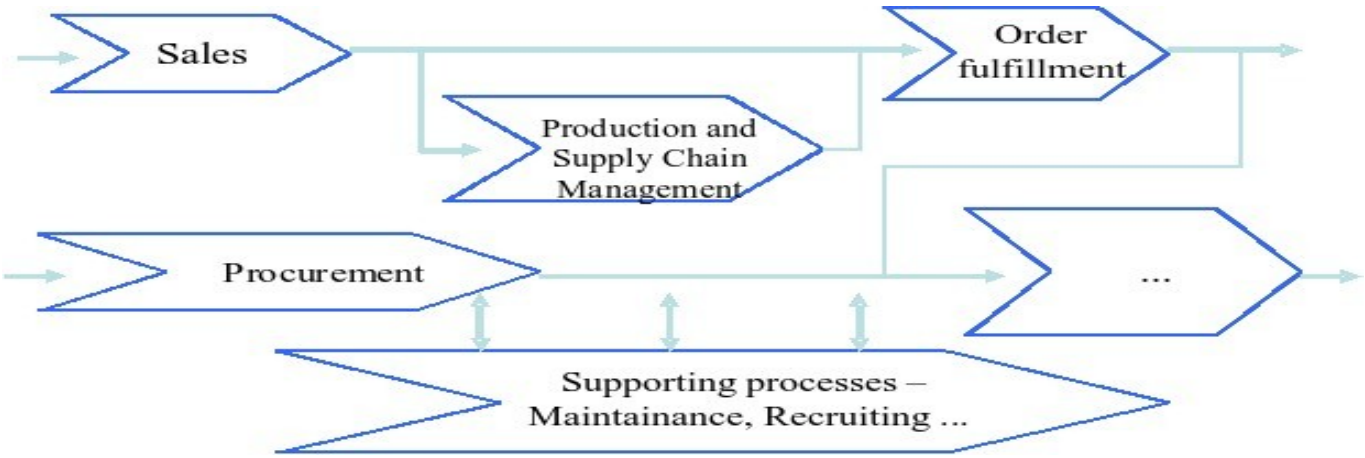


Exhibit 17: Process oriented enterprise

Process orientation is a philosophy of business, it is more than a methodology, and it implies no accepted actions of a single or connected system. However, methods and concepts have been developed, founded on process orientations, that have been widely adopted. As well as BPR, other names for this philosophy of business include Lean Management, Total Quality Management (TQM), and Process Cost Calculation. In the classical hierarchical organization, process management performs operates in parallel and complements the functional structure. A fundamental alteration of the organization's structure is not necessary. Finally, it is important to point out that process orientation should co-exist with other management approaches and management science applications.

What is business process modeling and what is it good for?

A purely verbal description of the sequencing of tasks is not suitable for describing a process because of the level of detail and interaction that must be captured. Graphical methods are clearer for showing the ordering of activities, identifying those that occur in parallel and detailing a task's resource requirements. Graphs are easier to read and convey the overall nature of a process.

Process modeling is a method for enabling an enterprise to document its processes and to recognize the resources required by each process. and to depict or to document them, that is to model them, as activities, events and resources. There is no generally accepted standard for process modeling and it is strongly influenced by the capabilities of the selected tool. In additionprocess modeling, most of the time, takes place as part of a larger project (e.g. business process reengineering or introducing a workflow system). The higher goals or the larger project usually strongly influences the approach to process modeling.

In practice, organizational problems often trigger process modeling (e.g. a decrease in turnover, a loss of market share, a decrease of the quality of work). It often takes place as a reaction to a critical concern. Because process modeling is time consuming, Ideally, it should be take undertaken without undue pressure for rapid change. Seeking to use process modeling as a quick fix to a critical problem is unlikely to lead to the best outcome because it encourages short cuts in modeling and analysis that are detrimental to a high quality solution.

There are two major approaches of process modeling. The first kind assumes the existing processes must be understood before taking action (e.g. Bevilaqua and Thornhill 1992). The second approach starts with the results that this process should accomplish. It argues that analyzing existing processes will not produce radical change. Hammer and Champy's BPR represents the second school. In reality, a synthesis of these two extremes would likely be most useful.

We now examine Nagl's (1993) four steps for the general order of procedures in process modeling:

- understand existing processes, their resource requirements, strengths and weakness, and identify any risk factors
- define planned processes and describe the current functional processes
- determine of the planned use of resources (future state) considering the critical success factors
- identify the stages of implementation (actual or current state), including describing the system of resources, use of process as well as the measures taken in different functional areas

The resulting process model should, among other things, do the following:

- identify and define processes
- support process analysis and make needs for improvement visible
- document the change in the order of a process as well as recognizing the effects of such changes

As already stated, business process modeling is concerned with the portrayal and description of all relevant aspects of a business process in an abstract model with the help of a formal or semi-formal language of description. At the same time, it actively supports the development of the new processes and reorganization of the business. Its

goal is to improve organizational efficiency and competitiveness. Process modeling supports these goals in diverse ways, including:

- Documentation: Business process modeling provides a simplified, but at the same time exact, description of the enterprise. All the elements and sections are described as well as their relationships and connections with one another. It provides the means to analyze emerging problems and to analyze their effects within the process or other related processes.
- Organizational analysis and reorganization: The sequencing of each process is analyzed in the course of the process modeling making possible to identify needless or awkward parts of the organization. As a result, parts of the process can be changed and organizational roles modified. As well, redundant tasks can be eliminated.
- Planning the use of resources: Because a process model provides an overall view of the enterprise and the exact knowledge of the way that the process functions, it is possible to determine the exact resource requirements of individual organizational units. Available resources can then be better planned and allocated. In addition, possible bottlenecks can be highlighted, so that the enterprise can react appropriately to relieve them.
- Process management, monitoring, and directing the course of a process: A process model delivers a business' leadership with an exact picture of the different trends and courses in the business. Input and output, distribution of resources, and the relationship between and in the individual elements of the enterprise are represented. Thus, more precise control of the course of business is possible. The organization should be more able to react in a timely manner to unexpected events.
- System and software development: A business process model gives management and analysts the opportunity to simulate proposed new processes before implementation. As a result, problems, possible improvements, and unanticipated outcomes are more likely to be recognized before implementation. It is cheaper and less disruptive to simulate change before making a final decision on the form of a new process.
- Creating a foundation for the establishment of a workflow management system: After successful business modeling, installing a workflow management is a possibility. A workflow system completely defines, manages and executes workflow processes through the execution of software whose order of execution is driven by a software representation of the workflow's process logic. Such systems are frequently used for document management.

The optimization of an organization is only possible if processes are modeled accurately, only then, does exact knowledge of possible improvements or problems become available. Many different elements can play a part in a process, and the more aspects that are recorded, the more exact will be the process model. Each organization needs to determine the appropriate detail for its process model to meet organizational goals.

According to the requirements of the respective organization, the following information needs may be relevant:

- Activities and structure of the process: This point is essential for modeling, it determines the purpose and structure of the process. It describes what happens and how it is to happen.

- Resources: This aspect has to do with the available internal as well as the inflowing external resources of a process. It defines what material is necessary for the correct sequencing of the process and the source of these resources. Data that are essential for the progress of the process are defined.
- Costs: An organization has a limited budget, and thus it is essential to have an exact list of the different tasks. Process Modeling can record the cost clearly for each individual process, recognizing where a redistribution or a change of budget is necessary or possible.
- Time: As with the cost, time also plays an important role in the schedule of a business. Through including time in process modeling, processes that last too long or create bottlenecks can be recognized and additional resources can be deployed to solve these timing issues.
- Exceptional incidents: The normal process is often the practice, though unusual events (e.g. lack of resources, short notice of deadlines changes) disturb process completion. Such disturbances need to be considered when modeling to ensure the process is fulfilled.
- Priority and role in the organizational structure: Each process has a definite status in the enterprise. Decision-making can be streamlined by the identification of organizational priorities. For example, if there is a shortage of resources, the decision as to where scarce materials should be delivered first is automatics because it is defined with the businesses processes.
- Communication structures: This item describes the internal communications structure of a process as well as its corresponding relationship to other units of the enterprise. For example, it describes what messages are exchanged between processes.
- Quality requirements: Quality requirements are included in process modeling to ensure customers' needs are met at the right standard. As well as defined quality requirements for output, quality standards for the process and specific dependencies are also recorded.
- Security requirements: With some processes, security factors have to be considered. For example, it might be important to prevent unauthorized access to internal data or minimize the risk of an accident in a manual process.

The preceding list is not necessarily complete. It shows that processes are very complex and that the information needs for process modeling are dependent on the purpose of the model.

Process modeling can serve multiple needs. The most important among them being:

- communication with co-workers and partners
- establishing a basis for understanding a process and analyzing it
- planning of work processes and exceptional situations
- implementation of a Workflow systems
- training people to use organizational processes
- input for software development
- a foundation of expertise for information management

For the systematic description of processes, several different notation formats are available. They vary from informal descriptive techniques (e.g. verbal or text based descriptions, flowcharts), to semi-formal techniques (e.g. ePK-diagrams, BPMN-diagrams, and UML-diagrams), to formal methodologies (e.g. Petri nets). Formal methods are based upon a theoretical foundation, usually based on diagram theory. It is helpful to use a common standard to make communication with others easier.

BPMN (Business Process Modeling Notation) <www.bpmn.org/> is an open standard for modeling, implementing, and performing business processes. It is similar in its notation to the activity diagram of UML (Unified Modeling Language), which is widely used in the software industry. An activity diagram can be complemented with an application or sequence diagram.

ePK (event controlled chain of process) diagrams are also widely used, and appear in ARIS platform and SAP software. Business processes are portrayed as chains of activities that can be connected through events.

Process description languages are typically strongly structured, often tabular or graphically oriented. They often use diagrams or structured language to capture detail. As a result, they are more suited for describing processes than everyday language.

Modeling with ePK

A major advantage of ePK-diagrams is that they are easy to read and are intuitively understood. An important disadvantage is in the limited ability for automatic analysis. A sample ePK diagram is shown in Exhibit 18. As a core element of ARIS, ePK diagrams will be treated when we cover tools for process modeling.

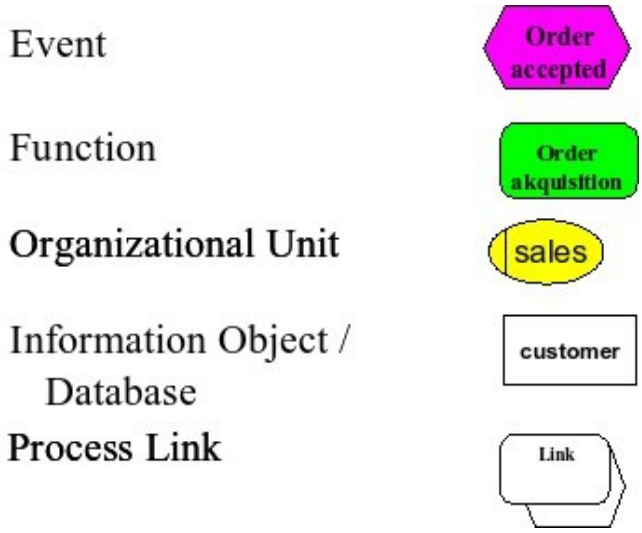


Exhibit 18: Example elements of an ePK diagram

We can describe the business processes within an organization in terms of events and function. Examples of events include an incoming order and sending out of invoice. The total of all possible events in an organization and its environment is defined as the organization's event scope. Functions include or describe a business management chain of activity (e.g. examination of creditworthiness). A function is a time consuming occurrence, which is triggered through a starting event and is completed by an ending event. Sometimes a function is also called a chain of activity, occupation, or operation. Functions can also be broken down or aggregated. An informational object describes the data that are processed by a function (e.g. a customer's record). Connectors describe the different forms of process branching (see Exhibit 19). In Exhibit 17, we see an example of an ePK diagram.

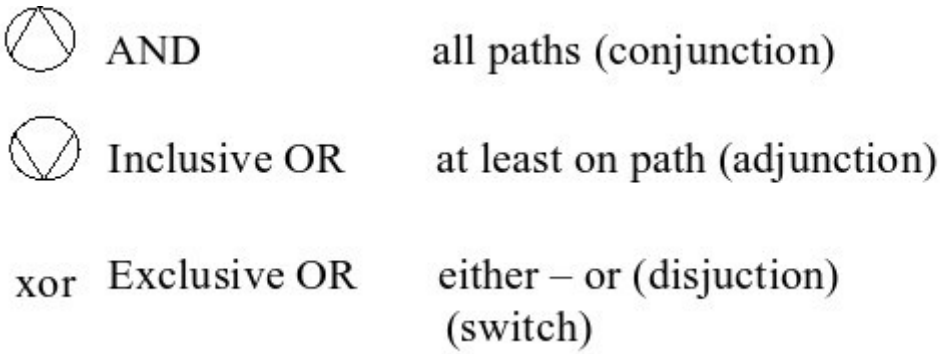


Exhibit 19: Connectors

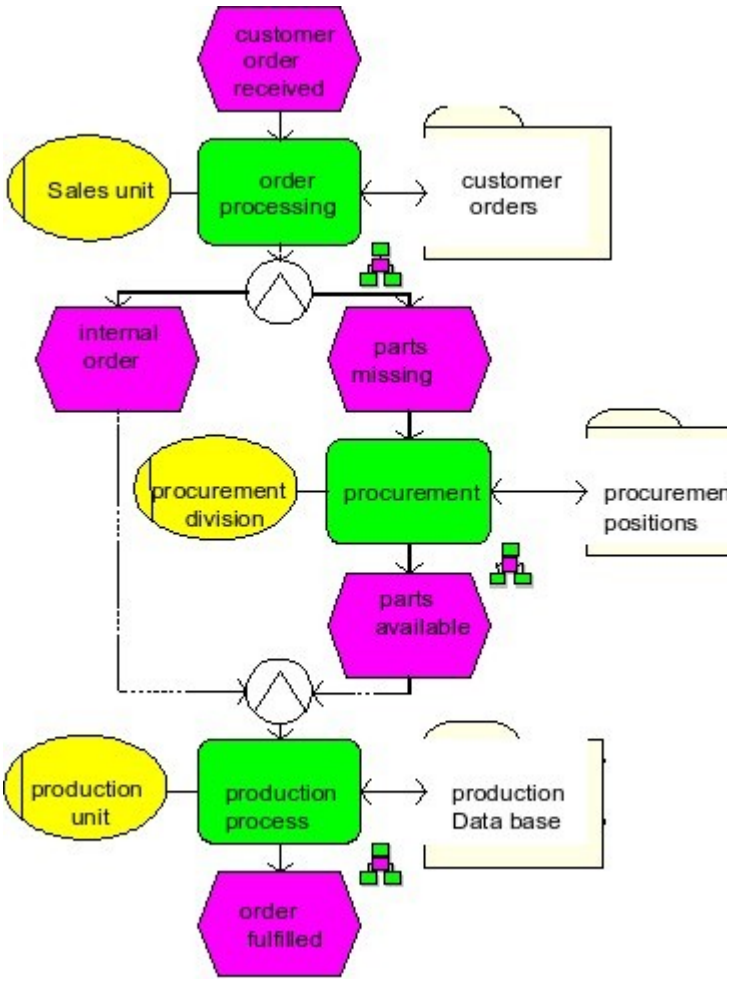


Exhibit 20.: an ePK diagram

Functions and events create an event controlled chain of process, with arrows showing the flow of the process. An incoming event initiates a function and a function when completed, in turn, initiates another event. Connectors can be used to combine parts of a chain. Complex chains can be broken down into layers, with high level layers showing the overall flow and lower layers recording the fine details. There are some common conventions for process modelling, including:

- Events (e.g. parts missing) initiate functions (procurement).

- Parallel of functions should be modeled with logical operators.
- A process model should flow from the top to the bottom.
- Clear and understandable terms should be used for functions and events.
- An organizational unit responsible for the execution of each function should be identified.

Software for process modeling and process support

The planning, realization, supervision, and control of processes can supported by software. The spectrum of software support includes simple modeling tools to Workflow Systems, Document Management Systems, EAI-tools (Enterprise Application Integration), Business Rules Engine (which can automatically perform certain defined processes). The supply and demand for process technology is still growing strongly. We discuss a few tools that are mainly used for process modeling and process analysis.

Process modeling software is often similar to CASE-tools (Computer Aided Software Engineering), which were originally created for software design and software development. CASE tools provide some support for process modeling.

Workflow management describes another group of tools, which are well-suited for chain of activity modeling. The tools are frequently used to support group work and are sometimes called groupware. The tools were designed to support office automation, office communication, and office organizational systems, and their development started in the 1980s.

Tools support the systematic definition, storage and analysis of data collected during process analysis and process modelling. In large projects, they are especially useful in coping with handling volumes of data. They promote process understanding and process thinking. However, they don't automatically indicate how to improve a process, that is something for humans to undertake.

A process modeling tool needs to support different views of a process (e.g. understanding a process across functions or departments).Tools should also be able to show processes at different levels of detail and integration. The goal is to have an Enterprise or Information Model. An enterprise model supports definition of:

- functions
- data objects, documents, and data flows
- processes
- organizational structures (organizational units and organizational set-up, employment structures).
- resources and other process features (e.g. cost, resource consumption, length of a run, frequency, volume etc.),
- process responsibility (e.g. process owner)
- process results
- process triggers (events)

An enterprise model creates the possibility of a range of views of the organization to suit different purposes (e.g. strategic planning or process improvement). Most process modeling tools offer the breadth of functionality to

support development of an enterprise model. The centre of a process modeling tool is a description language, which must fulfill certain requirements for it to be useful. The most important features include:

- Power of expression: A language must be capable of representing all relevant aspects of a process. It must capture all properties and relationships.
- Formalization and degree of precision: A description language must be flexible to adapt use to a particular project's goals. If the language is not flexible enough, it loses power of expression and might be unsuitable for some modeling tasks.
- Visualization: A modeling language should support multiple organizational views, so that all aspects of the various processes are represented. A graphical representation is very helpful, as one loses the general idea very easily with a plain text description of a system. The ability to change the level of detail in a process model is also useful. One should be able to fade out irrelevant facts for a closer inspection of the fine detail of a process or summarize different processes to gain a high level perspective of a process.
- Development support: A process modeling tool, ideally, should also support software development as software is often written to automate processes.
- Analysis and validation: In most cases process modeling enables analysis of internal processes and to represent different feasible process structures and sequences. The modeling language must support precise representation of existing and proposed processes and validation for the testing of redesigned processes.
- Performance simulation: An analysis of organizational processes often identifies possible improvements in the structure and working of processes. To ensure that the potential changes result in process improvement, it is necessary to test them before implementation. Process simulation is a way of testing a change before making a costly commitment to an organizational process change. Implementation without testing or simulation can have costly consequences if the new process is flawed.

The usefulness of a modeling tool is also determined by other factors beyond the capability to meet the preceding requirements. A tool must also be easy to use and support the interaction of the team working on the project.

Process models must be constantly updated to reflect organizational change if they are to remain of value. Only the employees that carry out a process, will know if the process has changed or its requirements modified. Ideally, model maintenance is undertaken by employees and not of the modeling specialists. Consequently, process modeling tools should be readily used by employees. Of course, they might need some appropriate training, but this should not be extensive because they are likely to be infrequent users of the tool.

Process analysis and the benchmarking of processes

Generally, a process improvement can be reached by simplifying and standardizing the process and its relationships. Automation of an activity chain (e.g. new technology and an information system) can lead to rapid performance increases. Also, restructuring parts of a process (e.g. a change in the internal sequencing of a process) can increase the reduce costs and the time for process execution. Some potential changes include:

- Automation (e.g. cessation of manual activity)
- Information level (e.g. better reporting on the stages of process execution)

- Process sequence (e.g. changes in the order of a process' steps and elimination of unnecessary steps)
- Control (e.g. improved monitoring of a process at key steps)
- Decision support (e.g. improving the information supplied to decision makers typically results in higher decision quality)
- Regional co-operation (e.g. better coordination among different locations)
- Coordination (e.g. better coordination between individual tasks and also between two or more processes)
- Organizational learning (e.g. collecting and transmitting strategically important information to key managers)

Time, cost, and quality play an important role in process improvement. Thus, the individual process objects and activities must be examined to record their duration, quality, and content. The capacity, the consumption, the results, and responsibility of individual processes must also be determined. A variety of data might be collected, including production time, punctuality in meeting deadlines, work capacity, processing time, waiting time, and transport time, postprocessing time, and error rates. Careful and detailed process modeling lays the foundation analysis and redesign. There are a number of process assessment approaches including:

- net value analysis
- cost calculation
- benchmarking
- profit value analysis
- controlling
- portfolio analysis
- strengths, weaknesses, opportunities and threats (SWOT) analysis
- process simulation
- total quality management
- six sigma

Benchmarking requires comparison of current process time, cost, and quality with earlier outputs of the organization or with a similar organizations in the same sector and of comparable size. It should methodically evaluate processes and products with their equivalent in the comparison organization. Benchmarking should identify target time, cost, and quality metrics for process redesign based on the best practices of the benchmarked organizations. The central question is: “Why and how are others performing better?”

Often benchmarking projects are carried out with the assistance of a consulting service, which collects and evaluates the necessary data of a group of similar organizations. Participants gain access to their performance data and a comparative analysis of their performance relative to the group average and to the anonymous best or worst results of the group.

Profit value analysis is used to compare alternative processes with respect to established goals.The analysis is organized into six steps:

11. Identify the criteria for assessment (e.g. short production time, high quality, low cost).
12. Determine the weight of each goal (e.g. 25 per cent, 25 per cent, 50 per cent).
13. Assess the alternatives and give each a score for each criterion.
14. Calculate the profit value by applying the weights to each score and summing these.
15. Complete a sensitivity analysis of the profit value to determine how sensitive the findings are to key assumptions.
16. Rank the alternatives.

Profit value analysis takes into account the importance of different goals. However, the evaluation of each alternative is often subjective because objective measures for each criterion are often lacking. In addition, there is a certain amount of subjectivity in selecting weights.

The purpose of **cost calculation** is to investigate process costs beyond an organization's boundaries so that the full cost of a product to customers can be determined. Process cost calculation started in the 1980s. Because of a changing business environment, it was necessary to depart from the traditional method. Overhead costs were increasing because of several factors, including manufacturing flexibility, shortened product life cycles, a wider range of products, and variations within products. The goal is to make the cost structure transparent as this often leads to considerable process rationalization and savings. Cost calculation can highlight overheads that are too high and common processes that increase the cost of many other processes.

Process cost calculation is not an alternative to traditional cost accounting calculations. They are complementary. Process cost calculation tries to compensate for some of the shortcomings of the traditional cost calculation system in regards to the basic causes of organizational overheads. Increasing cost transparency should lead to more efficient resource consumption, report accurately the use of capacity, and improve the product cost calculations. The goal is to improve the quality of decision making related to new product introductions and pricing policies.

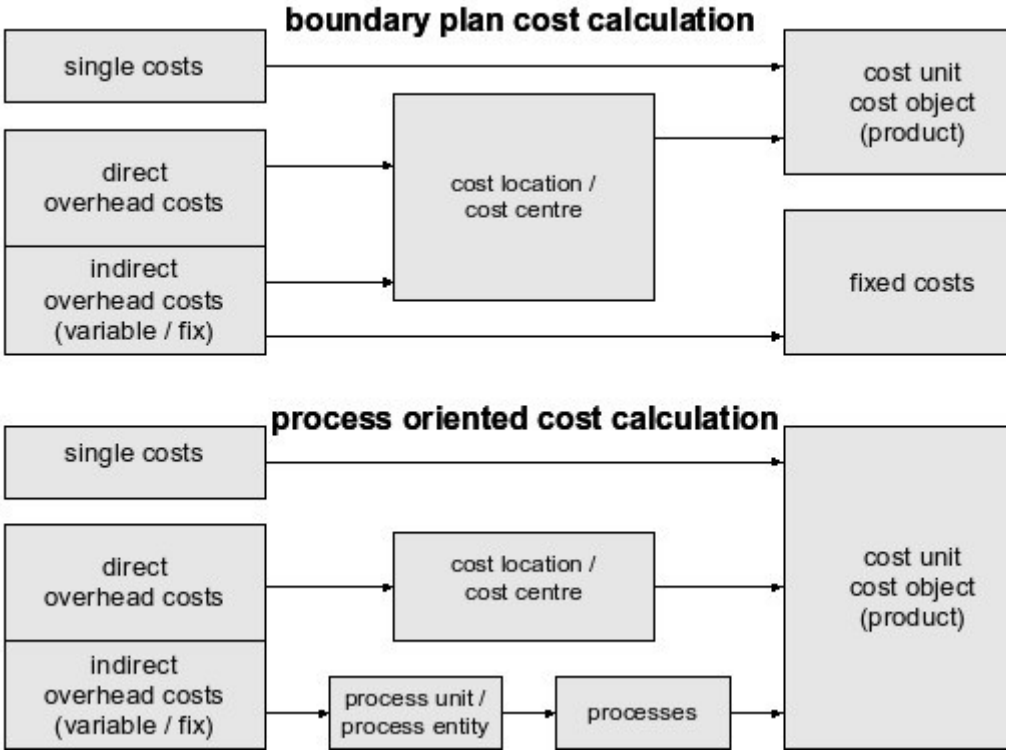


Exhibit 21.: Boundary plan cost calculation compared to process oriented cost calculation

Process cost calculation should document the cost drivers for each process. These cost drivers can be related to quality, time and efficiency goals that have an impact on organizational costs. Reducing cost drivers can decrease costs and improvement competitiveness. It might not be economical to undertake a complete cost analysis of all tasks, so process cost analysis should concentrate on those areas that are the main financial stress of the business. It should also focus on those areas for which no analysis of the basic cause of overheads has been completed.

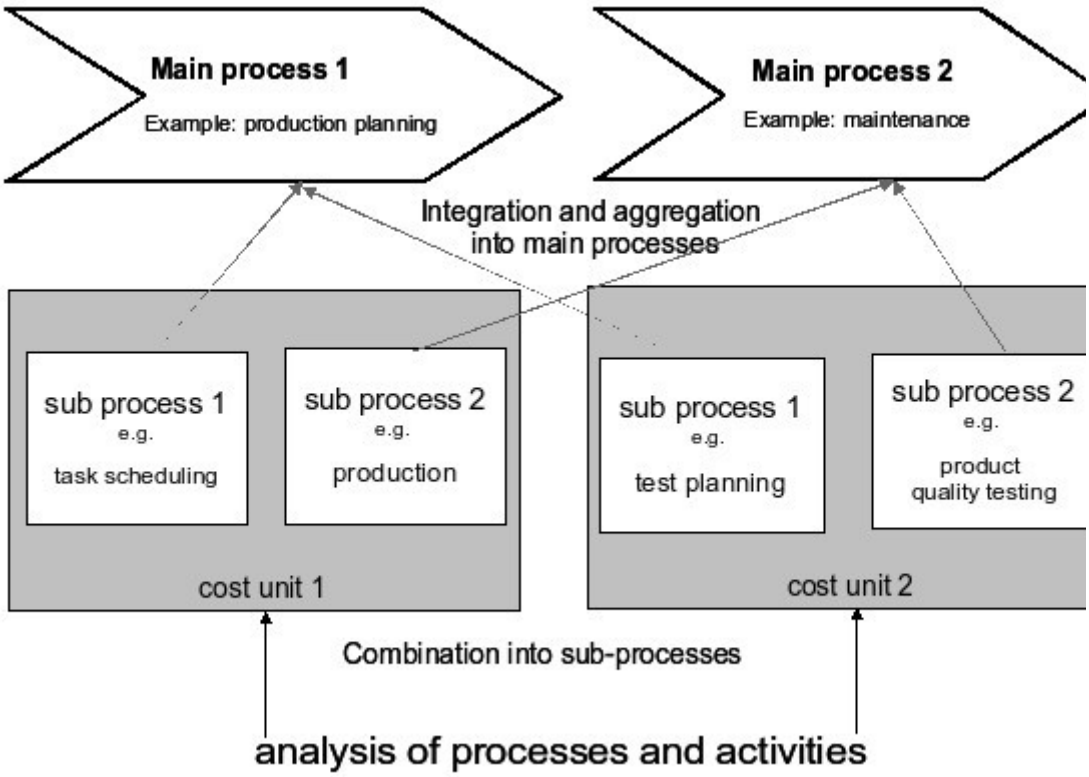


Exhibit 22: Process cost calculation (based on Horváth)

Process cost calculation makes no judgment about the value of a process and only considers the cost side. Therefore, it should be combined with qualitative procedures. For example, a particular process might have a considerable impact on an organization's reputation and this impact would need to be considered when reviewing the overall value of the process related to its costs.

A roadmap to process management

Process management should be an enduring element of an enterprise. Organizations need to continually evaluate about both their larger goals and the fine details of the steps that they take to achieve those goals. Process management provides support across this broad spectrum, as illustrated by Exhibit 10. Process management has become a basic requirement for nearly all organizations because it provides a methodical and systematic set of procedures for continuous improvement. It charts the roadmap to the future for most organizations.

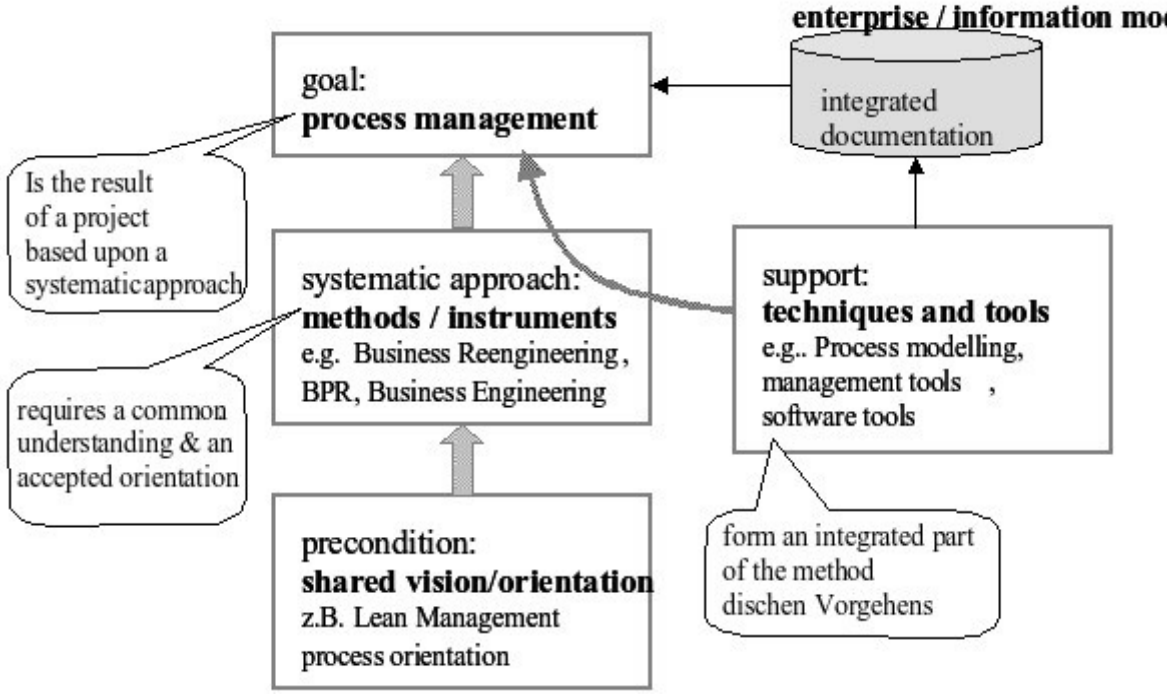


Exhibit 23: Steps in process management

In order for process management to be successful, there must be a person responsible for this task within the organization. Typically, the appointee is called a process manager. In addition, there can be process owners for key organizational processes. Process management occurs in parallel with the traditional functional structure of an organization. Thus, there is no need for a fundamental structural change, but there is a need to introduce new roles and recognize that management of a process often must span functional boundaries because processes span these boundaries.

Process management must support an organization's strategic success factors. Successful process management can create a competitive advantage and provide administrative efficiencies. It can also increase an enterprises' innovativeness when applied to areas such as research and development.

In general, the success of process management is dependent on the skill of the process analysts, who must be supported by appropriate tools for modelling and analysis. Successful automation of processes is reliant upon a number of systems, such as a workflow system, a process management system, or the quality of the information system in which the processes are embedded.

A five stage model can be applied for introducing business process management (Kleinsorge, 1994). The stages are completed sequentially.

1. Preliminary study: Those responsible for making the change are identified. A process manager is nominated for each process. The extent of the project should also be established at this stage as it might not be sensible to model all processes in an entire enterprise in one project.
2. Process modeling and documentation: All the elements (clients/customer, customer request, output, supplier, request to the suppliers, input, work capacity, etc.) of each process are recorded in a process model. This work is directed by the appropriate process manager. As well, the span and limits of these processes are documented.

3. Measuring achievement: Only a measurable process is controllable, so appropriate metrics must be determined for each process. These metrics should be highly objective and easily measured. Once the metrics have been defined, they should be measured for each process.
4. Active steering: Active steering (or controlling of a process) means that a process is managed to ensure high customer satisfaction, deviation from the predetermined goals is minimized, and problems are identified quickly and corrective action taken immediately. A sound measurement system will help in recognizing problems promptly and provide data for their rapid resolution. Active steering will increase customer satisfaction and efficiency and lower costs.
5. Continuous process improvement: Technological change and a dynamic business environment create the need for continuous process improvement. Organizations should strive to continually raise the quality of processes by reducing errors and required resources for execution. Errors in products and processes should not be tolerated and processes should be monitored constantly to identify opportunities for further improvement and lowering of costs.

Process management started with the Japanese concept of Lean Management, and now we have a wealth of approaches to process management. Organizations are increasingly combing process management and information systems to increase their efficiency and effectiveness. They realize that they need to redesign their processes first before automating them if they are to gain the greatest benefit from information systems.

Firms adopting process orientation can transform in many ways, including positive changes in:

- consumer orientation
- competence orientation
- concentration on creating added value
- effectiveness and efficiency
- delegation of responsibility and empowerment
- support for organizational learning

Business process management has become a core competency for many organizations and is thus closely related to strategic management. In addition, there is a strong linkage between information systems and process management. On the one hand, information is needed for the performance of processes, and on the other hand, new information is created by processes. Once again, we see evidence of the critical role of information systems in organizational success. Indeed, in today's information intensive economy, process management and information systems are critical partners in organizational success.

Exercises

Chapter editor

Franz Lehner.

References

BEVILAQUA, J.; THORNHILL, D.: Process Modelling. In: American Programmer 5 (1992), p. 2–9

COENENBERG, A.G.; FISCHER, T.M.: Prozeßkostenrechnung – Strategische Neuorientierung in der Kostenrechnung. In:DBW 51 (1991), Nr. 1, p. 21–38

DAVENPORT, T.H.: Process Innovation – Reengineering Work through Information Technology. Boston: Harvard Business School Press, 1993

DAVENPORT, T.H.; SHORT, J.E.: The New Industrial Engineering – Information Technology and Business Process Reengineering. In: Sloan Management Review 31 (1990), No. 4, p. 1–27

FISCHER, T.M.: Sicherung unternehmerischer Wettbewerbsvorteile durch Prozeß- und Schnittstellen-Management. In: Zeitschrift für Führung und Organisation 5 (1993), p. 312–318

GAITANIDES, M.: Prozessorganisation – Entwicklung, Anätze und Programme prozessorientierter Organisationsgestaltung. München: Verlag Vahlen, 1983

HAMMER, M. ; CHAMPY, J.: Business Reengineering – Die Radikalkur für das Unternehmen. Frankfurt/Main, New York: Campus Verlag, 1994

Hammer, M., Champy, J.: Reengineering the Corporation - A Manifesto for Business Revolution. New York 1993

HAMMER, M.: ReengineeringWork – Don't automate, obliterate. In: Harvard Business Review 4 (1990), p. 104–111

HORVÁTH, P. ; REICHMANN, T.: Vahlens Großes Controlling Lexikon. 2nd Ed.. München: Verlag Vahlen, 2003

HORVÁTH, P.: Controlling. 9. Auflage. M¨ unchen: Verlag Vahlen, 2003

Kleinsorge, P.: Geschäftsprozesse. In: Masing, W. (Hrsg.): Handbuch Qualitäts-Management, 3. Auflage, Carl Hanser Verlag, München, Wien 1994, p. 49-64

Lehner, F., Scholz, M., Wildner, St.: Wirtschaftsinformatik. Hanser Verlag, München 2007

NAGL, G.C.: Erfolgspotential Unternehmensprozeß – Modellierung von Unternehmensprozessen mit Computer Aided System Engineering. In: Zeitschrift f ¨ur Organisation 3 (1993), p. 172–176

OBERWEIS, A.: Geschäftsprozeßmodellierung. EMISA-Fachgruppentreffen 1997 – Workflow-Management-Systeme im Spannungsfeld einer Organisation, Oktober 1997

REMUS, U.: Prozessorientiertes Wissensmanagement. Konzepte und Modellierung, Universität Regensburg, PhD Thesis, Regensburg, 2002

STRIENING, H.-D.: Prozeßmanagement. Frankfurt/Main: Verlag Peter Lang, 1988

Tonchia, S., Tramontano, T.: Process Management for the Extended Enterprise – Organizational and ICT Networks, Springer, Heidelberg 2004

Vossen, G., Becker, J.: Geschäftsprozeßmodellierung und Workflow-Management. Modelle, Methoden, Werkzeuge. Bonn et al. 1996

5. Information systems methodologies

Editor: Salam Abdallah

Learning objectives

- define in general terms what is a methodology
- describe the role of methodologies in information systems
- describe the main features of methodologies
- describe some of the challenges when adopting a methodology
- break down methodologies into its fundamental components
- assess methodologies using an evaluation framework

Introduction

Information systems (IS) are affecting most management functions and have become essential to firms' competitive advantage and survival in the “new global digital economy”. Organizations are under pressure and they are responding by continually updating and aligning their IS strategies with their business strategic goals resulting in new information systems. Organizations need to also adapt to changes and align themselves with newly introduced information systems.

This alignment usually triggers a number of "transformational projects" across the organization that affect a large number of stakeholders; this may include new IS strategic planning, restructuring the organization and its business processes, security planning and building, and managing variety of applications (see Exhibit 30). Transformations and developments are usually carried out with guidance of methodologies to ensure business value and to provide the means to bring people, processes, and technology together. It is known that the field of IS contains a jungle of methodologies. It is estimated that there are hundreds of methodologies for different purposes and scopes and this reflects the fact that no single methodology is the best. Which will be the most appropriate methodology for a given project depends on the nature of the problem and the organization fitness.

Methodologies share general common philosophies and approaches and can play a central role in the working life of IS mangers. It is important for managers to have a conceptualized understanding of the philosophies behind methodologies and to be aware of the challenges behind adopting methodologies. Managers should be able to evaluate and select an appropriate methodology to ensure the project will be completed on time, on budget and according to specifications. This chapter provides a foundation for understanding and selecting methodologies.

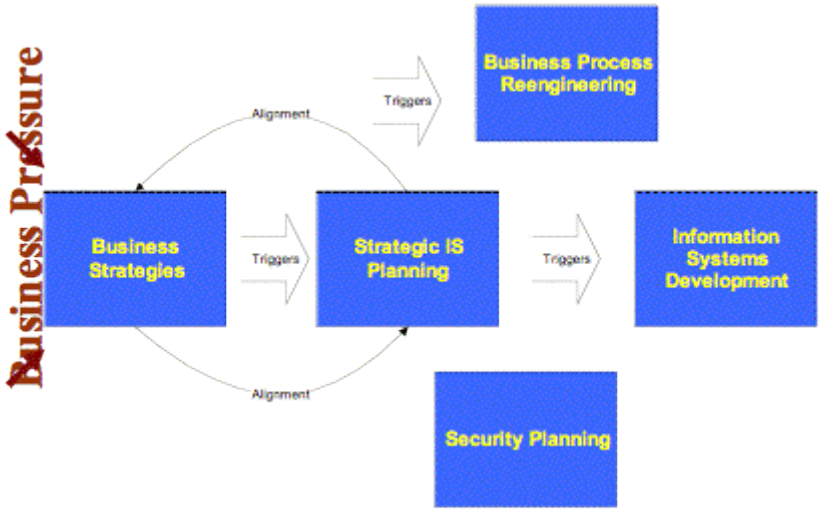


Exhibit 24. Business pressures triggering various types of projects

Why IS projects fail

Information systems related projects frequently fail; it has been reported that between 50 per cent-80 per cent of project fail. For example, projects in Business Process Re-engineering (BPR) have been failing at the rate of 70 per cent (Grant, 2002).

Different projects regardless of their nature have experienced poor quality outcomes, missed deadlines, over budget and cancellations. In a survey (Whittaker, 1999) the nature of the failure was the result of:

- the project budget was overrun by 30 per cent or more; and/or
- the project schedule was overrun by 30 per cent or more; and/or
- the project was canceled or deferred due to its inability to demonstrate or deliver the planned benefits

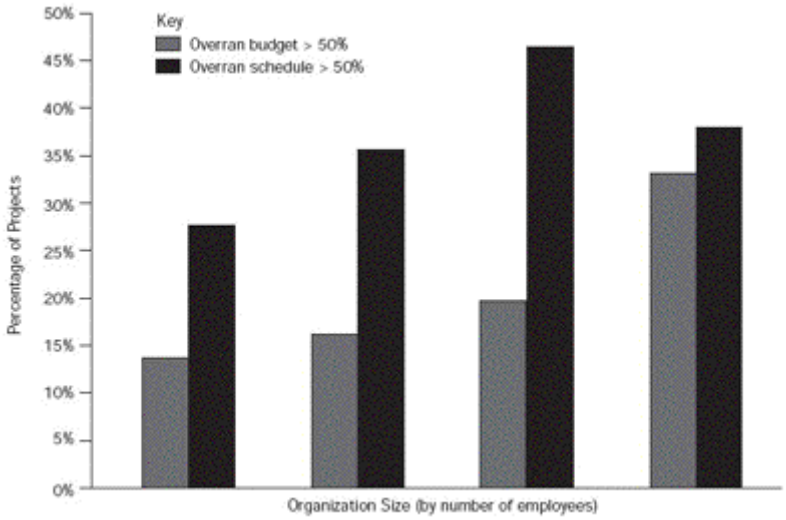


Exhibit 25. Serious budget and schedule overrun by organization size (Whittaker, 1999)
(B) Projects overrunning budget by 50 per cent or more (Whittaker, 1999)

One of the common stated reasons for failures is: organizations are not adopting a sound methodology or lack planning. Other common reasons of failure have been also stated, such as the business case of the project was weak in several areas, missing several components and a lack of management involvement and support (Whittaker, 1999).

Methodologies are necessary; they can provide organizations with useful ways to efficiently and effectively facilitate transformation to fully benefit from IS and to reduce the risks of project failures. Verner et al. (1999) surveyed twenty experienced software project managements, who agreed that the right choice of the methodology has an effect on the success of the project.

Methodologies defined

The terms "methodology" and "method" have been used interchangeably in the IS field. Usually the term "methodology" means a holistic approach to the problem-solving process and the word "method" is a subset of a methodology. Holistic means the methodologies include project management issues, education and training and support.

A methodology can be defined as "an organized collection of concepts, beliefs, values, and normative principles supported by material resources" (Lyytinen, 1987). Methodologies in general are viewed as problem-solving processes, to solve mainly ill-structured problems where the solution and the outcome are not easily predictable i.e. having high uncertainty. Methodologies provide users with the means to reflect on reality, determine the problems and solve them. They aim at reducing complexities, providing means of involving stakeholders and collecting requirements and capturing a holistic picture, allowing for control, and standardizing practices to reduce risks of project failure.

Ackoff (1981, p.354) argues that "to solve a problem is to select a course of action that is believed to yield the best possible outcome". Methodologies contain a set of practices to suit special circumstances and usually follow a life cycle aiming at an effective solution process and ensuring quality outcomes. Methodologies are usually based on different philosophies as a way to approach the problem and reaching a solution.

Solving problems may follow these three generic life cycle stages, see Exhibit 24.

- ***Plan***: This stage attempts to capture a holistic picture of the problem by involving all the relevant variables and stakeholders and laying down practical strategies and plan to solve the problem.
- ***Develop***: Develop the solution according to the plans.
- ***Manage***: Implement the developed solution and monitor and assess the results.

These three generic stages are the basis of most methodologies. Methodologies would consist of detailed descriptive steps to guide the user on how to proceed with these stages of solving problems and what work products or deliverables will be required as an output form each stage. Methodology also refers to a "well-tried method, often developed by experts and publicized through training courses and books" and these are called "formalized" methodologies, as opposed to "homegrown". Homegrown are methodologies developed inside an organization based on personal experience of managers or developers.

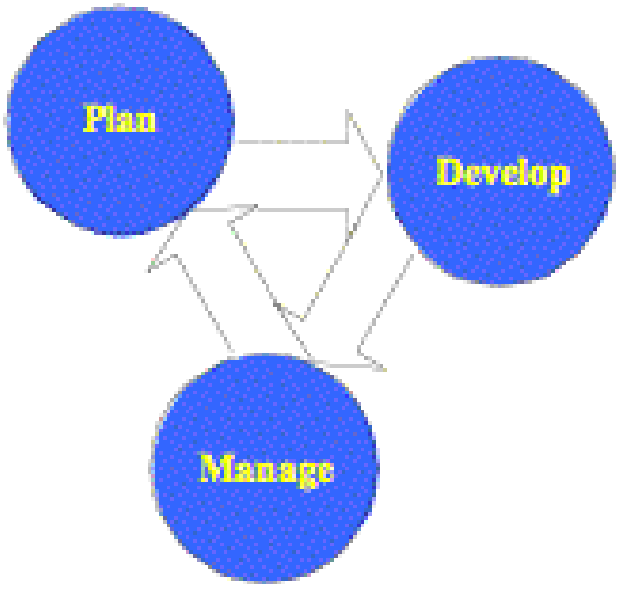


Exhibit 27.: Generic stages for solving problems

Methodologies are mostly developed and used in systems development. We find that development of other types of methodologies relies on knowledge and practices gained from systems development methodologies. For example business process re-engineering or information security relies on the established knowledge gained from the information systems development environment. Baskerville (1993) and Dhillon and Backhouse (2001) have all argued that information security methodologies are similar to information systems development methodologies since they consist of phases and procedural steps. Because of this, methodologies usually follow the same three generic stages but differ in their details and the expected work products.

Some of the common types of methodology used in the IS discipline are:

- Strategic Information Systems Planning (SISPM): This type of methodology seeks to integrate and to align an organization's strategic objectives with its existing information systems plan or business needs. SISP methodologies aim at assisting organizations in identifying opportunities by capitalizing on technologies for competitive advantage.
- Business Process Re-engineering (BPRM): Used to maximize corporate profitability by redesigning and transforming the organization's business processes.
- System Development Methodologies (SDMs): Used in designing different types of information systems such as transaction processing systems, enterprise information systems, decision support systems, etc.
- Information Security Methodologies (ISM): Assist organizations to establish a security plan to address vulnerability associated with unauthorized misuse of information.

To sum up, methodologies provide users with **ways of thinking, doing and learning.** Methodologies provide users with ways to approach the problem and controlling the process of solution development. They provide an appreciation of standards and they enforce a more disciplined and consistent approach to planning, developing and managing. Their deliverables may be checked for quality. A study has shown that the use of a structured software and maintenance methodology can contribute to the quality and business value (Nelson and Ghods, 2002). Methodologies are also seen as a way of learning by acquiring knowledge from previous projects and preserving it in a set format to guide organizations on their next project.

Waterfall model methodology

The oldest and most known methodology used to coordinate the efforts of information systems development is known as the “Waterfall”. The model is based on a generic life cycle stages to guide developers from an initial feasibility study through maintenance of the completed systems. Each stage of the model becomes the input for the following stage, see Exhibit 28. These stages are described as follows:

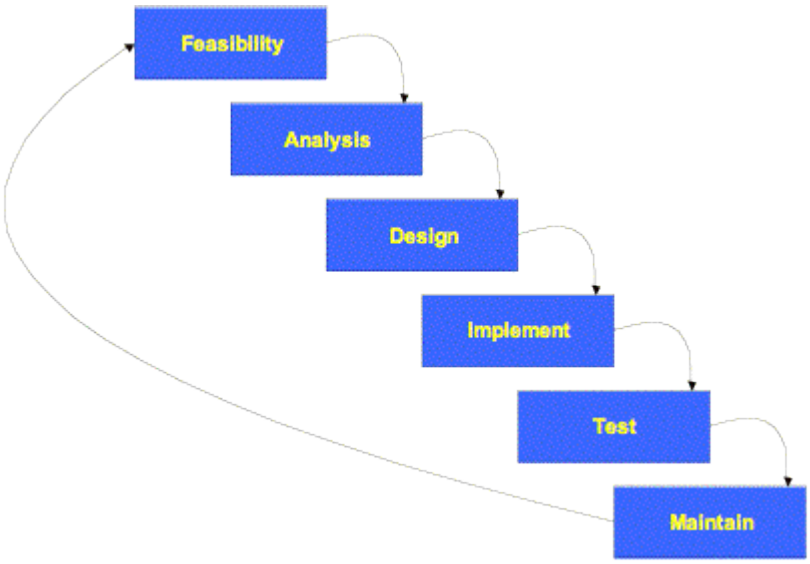


Exhibit 28.:Waterfall model methodology

Feasibility study: studies the existing systems to provide a clear understanding of the problem and to determine project goals

Systems analysis: analyzes end-user information and determines any deficiencies of the existing systems. Data collected in this stage will be used as part of the new system

Design: reatures and systems functions are described in details, including screen layouts, business rules, process diagrams, programming instructions and other various documentations

Implementation: designs are translated into code using programming languages

Testing: system is put into a testing environment to check for errors and bugs

Maintenance: rth ce system is put into production in a real business environment;hanges, corrections and additions start to crop and the system begins to evolve.

Evolution of Methodologies

Methodologies are considered useful and influence practice, therefore we find practitioners and academics continue to develop methodologies to improve projects success rates. Knowledge and lessons learned from previous practices are embedded into these methodologies; therefore, methodologies are evolutionary and their development is an endless effort. Methodologies have been in use since the 1960s and they have evolved in their approaches.

Tribal Era (1960-1970): During this period developers did not use a formalized methodology, the emphasis in this era was on understanding technologies and determining ways to solve problems. Little attention was given to the exact needs of users resulting in poor solutions that did not meet objectives. The effort was individualistic

resulting in poor control and management of projects. Excessive failures called for a systematic development approach.

Authoritarian Era (1970s to early 1980s): In this era, developers believed that adhering strictly to the details of methodologies ensured successful project and would meet management and users requirements. This era also produced undesired outcomes, because the emphasize was on adhering sacredly to methodologies and neglecting the fact that businesses are always in transit and have changing requirements reacting to business pressures. Methodologies in this era were seen as lacking comprehensive and flexibility.

Democratic Era (1980s to current): Unsatisfied with the restrictive approach of methodologies, new methodologies emerged with more maturity, that are more holistic based on philosophies, clear phases, procedures, rules, techniques, tools, documentation, management and training, including success factors such as the need for user involvement. Methodologies in this era produced better business modeling techniques and can be described as being incremental, participative (sociotechnical), systemic (holistic), and business oriented i.e. alignment of IS with business strategic goals.

In this era the concept of method engineering has also emerged. Method engineering is based on the philosophy that users should be able to mix and match between the different methodologies extracting specific methods, techniques and tools depending on the current problem to be solved, rather than adhering to a single methodology.

Also, recently agile or light methodologies have emerged, and they are based on the belief that projects should be small with minimal features. These methodologies are characterized as being adaptive to the environment, having small teams, and feedbacks are very essential, teams are self-organizing, informal in approach to development, flexible, participative and social collaborative.

Methodologies have been going through a phase of transformation—moving from a mechanistic to socio-cultural paradigm or from the hard to the soft approach. See Exhibit 29 for a comparison between the hard and soft methodologies.

Exhibit 29.: Comparisons between hard and software methodologies

Hard methodology	**Soft methodology**
Used to solve well-defined problems—simple problem and solution is known	Used to solve ill-structured problems—world problems are complex, ambiguous require novel solution
Focus on technical perspectives, in terms of solving the problem and controlling the project. They are rational and scientific based approach.	Social constructivism, humanistic, people views are important and joint solution constructed
Focus on the final outcome—reach a solution with shortest route	The focus is on the process to encourage knowledge building form multiple views and creative thinking

Methodologies basic structure

Methodologies usually have the same basic structure and share common terminologies. The basic structure of most methodologies will have the following three components:

- **Principles** are the guiding rules, standards and beliefs of the methodology. The collection of these principles makes up the philosophy and the aims of the methodology.

- **Processes** express the principle in the form of stages, steps, and tasks. The processes expound the principles. They contain advice and recommendations on how to proceed and complete deliverables using different techniques.
- **Services** may include different formats, they may include detailed documents to elucidate the process and principle expressed by practical examples, case studies, illustrations, guiding templates, documentations, etc. They contribute to the understanding and learning of the user. Services may also include various tools to assist the user in completing processes or provide advice and assistance.

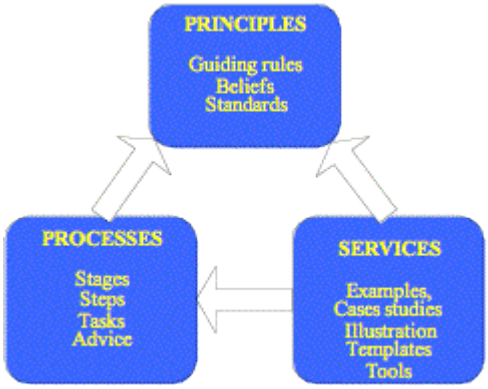


Exhibit 30.: Methodologies are basically collections of three elements

Selecting a methodology

Maybe one day you will be an IS manager and who will be involved in transformational projects, such as systems development, security planning, strategic planning, or business process engineering, and will be confronted with a difficult and critical task of selecting a methodology appropriate for the project. You should be able to carefully compare and select an effective methodology. Prior to selecting a methodology, you need to consider the following statements:

- When selecting a methodology choose one among proven methodologies. Choose methodologies that have been used successfully in other organizations.
- The benefits of using the methodology should be clear in your mind, in order to justify your reasons for adopting a methodology.
- What method and criteria will you use to choose an appropriate methodology? Appropriate in terms of usefulness and easy of use?

A useful generic framework that can be used to assist in the selection process is the NIMSAD, which stands for Normative Information Model-based Systems Analysis and Design (Jayaratna, 1994). NIMSAD is a tool that can assist users to evaluate and get better conceptual understanding of a problem to be solved and its environment. The framework also assists users to understand the complexity of the technical and social aspects of development. The framework is based on three important perspectives that the evaluator should consider in order to raise important questions and to answer them. These three perspectives are related to the ”problem situation”, ”methodology users” and the “methodology” itself. The main purpose of the framework is to evaluate these three perspectives.

- **The problem situation (the methodology context):** To get an understanding of the problem context, the problem solver should first develop an understanding of the organization. The organization may be understood by examining its main components, such as its people, processes, technology, information, material flow and structure. To get a clear picture of the problem, problem solvers need to gain deep understanding of these components and their interactions with each other. Jayaratna argues: "The richer the knowledge the intended problem solvers can obtain about the organization, the better the position they may be in for understanding the 'real' problems of the organization. It may also help them to make better judgments about the relevance of information to those in the 'problem situation' and ready to raise questions." Some of the questions that the methodology user may ask:

  Who are the clients? How strong is their commitment? Does methodology help in identifying clients and their concerns? What's the situation like? (simple or ill-structured) For which situation is methodology suitable? What does the situation demand, (identify problems, design solutions for already identified problems, implement an already designed solution)?

- **The intended problem solver (the methodology user**): An individual or group of people are responsible for the problem formulation and course of action to arrive at a solution. The problem solver could be a consultant, systems analyst etc. and could be from inside or outside the organization. The focus is on the role, rather than the person. The choice of the solution is usually reached in agreement between methodology users and stakeholder.

  The solution to the problem is greatly shaped by the mental constructs of the methodology users shaped by their personal characteristics, such as:

  - perceptual process
  - values/ethics
  - motives and prejudices
  - experiences
  - reasoning ability
  - knowledge and skills
  - structuring processes
  - roles
  - models and frameworks

  Some of the questions users may ask: *What level of abstract and technical thinking does methodology demand from the user? Do philosophical views advocated by methodology match user's view? What knowledge sets and skills does methodology require from the user? Are mental constructs of the user considered?*

- 3**The Problem-Solving Process**: This perspective examines the details of the process of solving the problem. The process has three primary phases, problem formulation, solution design and design implementation. Some of the questions that may be asked at each phase:

Problem formulation Does methodology offer assistance for boundary construction? What is role of client? Does methodology discuss particular methods of investigation? What techniques does methodology offer for expressing situation characteristics? What environmental (context) information is captured? What tools and techniques are available? What problems or problem types are of concern to the methodology? How does it help in deriving problem statements? Does it offer help in formulating notional systems?

Design phase: Does it help in formulating design solutions? What aspects cannot be captured by methodology? How experienced is user to be expected in the solution domain? Who decides on which solution to take?

Implementation Phase: What steps does methodology offer for developing the project? What does it offer in terms of tools and techniques? How does it help in handling major changes?

NIMSAD can provide a useful way to understand critically the problem being solved, users involved and the process of solution and the interactions between them. Once the methodology has been adopted it should be continually evaluated at the various stages of its use i.e. before, during and after. We should be prepared to adjust the methodology to fit better the problem situation or to abandon the methodology if no business value is being added.

Quality selection criteria of methodologies

Quality Attributes are the benchmarks that assess the methodology in terms of its content and operation. The quality attributes provide the means for understanding the fitness and suitability of a methodology. One needs to prepare a collection of attributes that are representative of the methodology requirements. For example, the attribute of being flexible (Can I change it?), comprehensive (Does it have everything that I need?), and practical (Is it useful and useable?).

Methodologies misuse

Ideally, methodologies should be adopted to become part of the working culture of practitioners as a way of completing projects successfully. Examination of practice on use of methodologies shows that methodologies have been adopted, or rejected, or in some cases has been misused. Have a look at the different ways how users have been misusing systems development methodologies.

1. Users are using methodologies as a cover up and not following the principles and the guidelines of the methodology, for several reasons:
   - to impress clients and win contracts or as a political protection for civil servants should projects go wrong
   - for image building to demonstrate that things are under control
   - to please their managers
   - to feel secure that they are doing that right thing, which is seen as ”social defense”
2. Follow a methodology that they learned previously e.g. from their previous managers, and continue using the methodology blindly without questioning its merits.

3. Users are misinterpreting the methodology's principles to suit their personal needs and desires and this causes deviation from the intended principles, which may eventually lead the user to abandoning the methodology. Studies show that 80-90 per cent of methodologies are being modified.

4. Some of the practitioners (60 per cent) do not fully understand methodology's principles and they end up with partial use.

Summary

Methodologies are extensively used by IS professionals to compete various tasks that involve planning, building and managing plans or systems. Methodologies can assist in reducing the risk of project failure, since they allow users to capture the real picture of the problem, develop strategies to solve the problem and to take actions; in other words, they can provide users with a way of thinking and doing and ensuring quality. Therefore, methodologies can be influential and careful evaluation and selection must precede the adoption of methodologies. Methodologies should be adopted to add business value and not to be used as cover up. Key Terms

Ad-hoc methodologies	Methodolgies structure
Agile Methodolgies	Methods
BPRM	Problem situation
Formalized Methodolgies	Problem-solving processing
Generic cyclic stages	SDM
Hard methodologies	SISPM
Homegrown methodogies	Soft methodology
ISM	The intended problem solver
Method engineering	Transformational projects
Methodolgies	Waterfall

Exercises

1. Search and reflect (individually or in groups) on the various methodology definitions and formulate your own definition of what a methodology is. Discuss and compare your definition in class.
2. Methodologies use many common terminologies that were not covered in this chapter. Define the following common terms: Paradigm, Techniques, Task, Tools, Iterations, Prototyping, Design Specifications, Notations, and Modeling.
3. Find a formalized methodology that is well documented on the web, briefly examine its details, and prepare a brief report on the methodology in terms of its general basic structure i.e. principles, processes, and services. Also identify the main stages of the methodology.

4. Find a formalized methodology that is well documented on the web, briefly examine its details, and prepare a one page report on the methodology using the dimensions provided by NIMSAD. Students are advised to select different methodologies to include, but are not restricted to: ISDM, SISP, ISM, BPRM.

Discussions questions

- The more detailed the methodology the better it is. Do you agree?
- Do you think one day will come where all practitioners will follow a methodology strictly and according to its principles?
- How can we ensure that methodologies are not being misused in an organization?
- Group project: Break class into groups and debate the issue of adopting or not adopting methodologies for a specific type of transformational project. The debate should explore issues from a technical and business perspective.

Chapter editor

Salam Abdallah

References

Ackoff, R. L. 1981, 'On the Use of Models in Corporate Planning', Strategic Management Journal, vol. 2, no. 4, pp. 353-359.

Ang, J., Shaw, N. & Pavri, F. 1995, 'Identifying strategic management information systems planning parameters using case studies', International Journal of Information Management, vol. 15, no. 6, pp. 463-474.

Avison, D. E. & Fitzgerald, G. 2003, Information Systems Development: Methodologies, Techniques and Tools, McGraw-Hill, Maidenhead.

Badenhorst, K. P. & Eloff, H. P. 1990, 'Computer security methodology: Risk analysis and project definition', Computer & Security, no. 9, pp. 339-346.

Baskerville, R. 1993, 'Information systems security design methods: Implications for information systems development', ACM Computing Surveys, vol. 25, no. 4, pp. 375-414.

Brinkkemper, S. 1996, 'Method engineering: Engineering of information systems development methods and tools', Information and Software Technology, vol. 38, pp. 275-280.

de Koning, W. F. 1995, 'A methodology for the design of security plans', Computer & Security, no. 14, pp. 633-643.

Dhillon, G. & Bakchouse, J. 2001, 'Current directions in IS security research: Towards socio-organizational perspectives', Information Systems Journal, no. 11, pp. 127-153.

Earl, M. 1994, 'How new is business process redesign', European Management Journal, vol. 12, no. 1, pp. 20-30.

Fitzgerald, B. 1998, 'An empirical investigation into the adoption of systems development methodologies', Information & Management, vol. 34, pp. 317-328.

Grant, D. 2002, 'A wider view of business process reengineering', Communications of the ACM, vol. 45, no. 2, pp. 85-90.

Jayaratna, N. 1994, Understanding and Evaluating Methodologies, McGraw Hill Book Company Europe.

Kettinger, W., Teng, T. C. J. & Guha, S. 1997, 'Business process change: A study of methodologies, techniques, and tools', MIS Quarterly, vol. 21, no. 1, pp. 55-80.

Lederer, A. & Sethi, V. 1996, 'Key prescriptions for strategic information systems planning', Journal of Management Information Systems, vol. 13, no. 1, pp. 35-62.

Lyytinen, K. 1987, 'Taxonomic perspective of information systems development: Theoretical constructs and recommendations', in Boland & Hirschheim, (eds.), Critical issues in Information Systems Research, pp. 3-41.

Mingers, J. & Brocklesby, J. 1997, 'Multimethodology: Towards a framework for mixing methodologies', Omega International Journal Management Science, vol. 25, no. 5, pp. 489-509.

Mumford, E. 1998, 'Problems, knowledge, solutions: Solving complex problems', in Proceedings of the international conference on Information systems, Helsinki, Finland, pp. 447 - 457.

Olle, T. W., Hagelstein, J., Macdonald, G. I., Rolland, C., Sol, G. H. & Van Assche, J. M. F. 1991, Information Systems Methodologies: A Framework for Understanding, 2nd edn, Addison-Wesley Publishing Company, Wokingham, England ; Reading, Mass.

Roberts, T. J., Gibson, M., Fields, K. & Rainer, K. J. 1998, 'Factors that Impact Implementing a System Development Methodology', IEEE Transactions on Software Engineering, vol. 24, no. 8, pp. 640-649.

Valiris, G. & Glukas, M. 1999, 'Critical review of existing BPR methodologies: The Need for a holistic approach', Business Process Management Journal, vol. 5, no. 1.

Wastell, D. G. 1996, 'The fetish of technique: Methodology as a social defence', Information Systems Journal, vol. 6, pp. 25-40.

Wastell, G. D., White, P. & Kawalek, P. 1994, 'A methodology for business process redesign: Experiences and issues', Journal of Strategic Information Systems, vol. 3, no. 1, pp. 23-40.

Whittaker, B. 1999, 'What went wrong? Unsuccessful information technology projects', Information & Management & Computer Security, vol. 7, no. 1, pp. 23-29.

6. Implementing systems

Editor: Kevin C. Desouza (University of Washington, USA)

Contributors: Leslie L. Mittendorf, Lokesh Ramani, Prem Kumar, Yared Ayele, Matt Sorg, Stephanie Morton, Ting-Yen Wang, Kelly Ann Smith, Benji Schwartz-Gilbert, Jaret Faulkner, and Kendal Sager (students at the Information School, University of Washington, USA)

Reviewer: Ron Vyhmeister (Adventist International Institute of Advanced Studies, Philippines)

Learning objectives

- understand why social problems must be considered along with the technical problems when implementing systems; understand the dangers of ignoring social issues when implementing systems
- discern different ways of adjusting to social issues when implementing systems
- be able to understand how ethics plays a role when implementing systems, especially the difficultly in defining this issue
- understand the significance and social/technical problems inherent in storing and disseminating information
- attain a general appreciation for the problems that emerge when conducting change management
- realize the roles people and company cultures play when implementing systems
- appreciate a modern socio-technical problem (Knowledge Harvesting), the challenges it presents, and some of the solutions that have been recommended

Introduction

The most important advantage that any organization can have over another is in the organization's knowledge workers. Workers that can create inventive processes and solutions to problems can improve the organization's everyday functions as well as its planning for future goals and objectives. The exploitation of this advantage can mean a reduction in the expenditure of both time and capital on the organization's projects, thus improving margins in both respects. However, this goal requires that the knowledge worker, whose main job is to assemble ideas rather than sellable products, be given the resources and time to solve new problems, rather than having to reinvent solutions to old, and often previously solved, problems. Within the Information Technology sector is a growing area of development focused on creating programs to reduce the amount of time knowledge workers must spend creating, recreating, learning, and relearning the processes and solutions to past problems. This area is known as Knowledge Harvesting, or more technically, Information Capital Management.

Knowledge Harvesting can be most simply defined as a set of processes or systems designed to capture and preserve the knowledge of an organization. In this case a process is considered a codified set of rules or guidelines designed to elicit certain key pieces of information, from the entity it is used on, and organize this information into a uniform and consistent package. This allows a knowledge worker familiar with the structure of the packages to

extract the useful information they may need without having to search for it. Secondly, a system is considered the physical technologies and techniques used to facilitate or support the Knowledge Harvesting processes. Some examples of technology systems include servers or computer terminals while system techniques could be exit interviews or electronic answer and question databases. Though these two pieces of Knowledge Harvesting exist and can be used by themselves, they are most effective when combined together. In this way they make up the fundamental building blocks for the harvesting of knowledge from the employees of an organization.

In order to analyze this subject to the fullest possible extent we have divided it into five major areas of discussion: the demand for Knowledge Harvesting within organizations, the affect Knowledge Harvesting has on knowledge workers, the effects from and impact on organizational culture, current Knowledge Harvesting practices with specific focus on a major case study, and finally, recommendations we have developed for the improvement of the Knowledge Harvesting functions in each of these areas. This structure allows us to progress through an organization from the bottom up and thus gives clear insight into the importance that Knowledge Harvesting is playing now and will play in the future.

Demand for Knowledge Harvesting

With the development of new technologies, business is moving at an increasingly faster rate. In order to stay at the leading edge of the market, organizations have to keep up or face being left behind by the competition. Knowledge Harvesting is one way that organizations can keep their employees well-educated and train new employees quickly. It is a great setback when organizations lose a veteran employee, and with the baby boom generation nearing retirement, “organizations will face a major exodus of institutional knowledge as their most experienced employees leave the work force” ("Intelligence at risk: capturing employee knowledge before retirement” 14). Without any way of capturing the knowledge from these employees, years of valuable experience will be lost. In fact, in a recent survey of over five hundred fulltime workers, more than twenty-six per cent of respondents noted that their organization would let them retire without any form of exit interview or transfer of knowledge. Fifty per cent of the five hundred surveyed said that their organization had absolutely no formal way to capture knowledge prior to losing an employee (Knowledge Harvesting - KM Toolkit: inventory of tools and techniques). The fact that these employees are allowed to leave without passing on any form of their experience means that new employees will have to continuously rediscover and solve past mistakes, creating a major loss of efficiency.

Through the implementation of Knowledge Harvesting systems, many hindrances created by the loss of experienced employees can be avoided. One of the most notable benefits of Knowledge Harvesting is that vital knowledge is not lost when experts leave the company (Knowledge Harvesting - KM Toolkit: inventory of tools and techniques). If this experience is properly harvested and stored, the knowledge of a few key individuals can then be made available to others who need it without creating a crippling dependence on the expert. This Knowledge Harvesting technique can also aid in training new employees. Since past experience is taken into account, the learning curve will be shortened due to the fact that the individual will no longer be reliant on their own problem solving skills; they can use past experience to solve problems more quickly.

Formalizing the Knowledge Harvesting Process

Many aspects of an organization can contribute to successful Knowledge Harvesting. In an idealistic setting, the organization has a fully developed knowledge management process and employees frequently contribute and extract knowledge from the organization. However, the actual process of harvesting knowledge is an intricate mix of

methods, behaviors, and motivations. Although these factors complicate knowledge harvesting, understanding them provides insight into the process of knowledge harvesting. As discussed earlier, an understanding of knowledge harvesting can allow organizations to use it to provide value to the business and eliminate redundancy.

Understanding where knowledge harvesting should occur is one crucial portion of formalizing the knowledge harvesting process in an organization. Knowledge is exchanged in two broad settings, formal and informal (Coakes). Formal settings include situations like exit interviews and official meetings for project updates. Informal settings are much more varied, ranging from chatting at water coolers to visiting a co-worker's office. Arguably, most of the knowledge sharing in an organization takes place at informal locations, because these interactions are more frequent and comfortable due to the informal nature. However, harvesting knowledge from informal interactions is challenging because the interaction is unstructured. Also notable is the role that geographic proximity plays in Knowledge Harvesting. While co-workers on a co-located team rapidly acquire knowledge from team members, the same process occurring across teams or office buildings is much slower (Brown).

Employee Contribution and Resistance

With the understanding of where knowledge harvesting occurs, the rewards or incentives for an employee to contribute knowledge can be explored. As a concrete definition, a reward is a condition that can motivate an employee to contribute knowledge with the expectation of receiving something in return (Coakes). Coakes explains research divided rewards into two categories, intrinsic and extrinsic. The extrinsic category of rewards includes increased salary or additional vacation time. As an example of an extrinsic reward, employees who contribute knowledge to a company system could be rewarded during performance reviews. The intrinsic category of rewards includes job satisfaction, decision making power, and improved career prospects. One particularly interesting example of an intrinsic reward can be seen in the case of the Xerox technician who received a standing ovation at a company conference because the technician had contributed many useful articles to Xerox's knowledge management system (Brown). Knowing these rewards helps information systems builders understand what motivates employees to contribute to knowledge harvesting activities. Thus, understanding and utilizing rewards systems represents a substantial part of successful knowledge harvesting.

However, in spite of rewards, many employees consider reasons to resist information harvesting. Most popular among these reasons is self-preservation. A number of studies have found that in organizations implementing knowledge databases, organization members have been willing to use the system to search for information, but have been reluctant to submit their own information to the system (Cress 371). The sharing of information presents a social dilemma for members of an organization; information is often perceived as power, and sharing of information without a guarantee that others will do the same could mean the relinquishing of power and degradation of one's position within the group (372). Another employee concern is the possibility that others will take credit for knowledge that an employee contributes. This concern is particularly strong in situations where a reward system can be abused to benefit those who steal ideas. Additionally, employees may feel that harvested knowledge will be used for negative purposes. As Harrison explains, on any team of computer programmers, there will always be a top performer and a bottom performer (6). With knowledge harvested about the performance of each team member, the intention of management becomes a concern for the employees. With positive intention, management could provide training and career development materials for the worst performers in the team. However, with negative intention, management could choose to fire the weakest performers. Finally, some industries and organizations have competitive cultures that encourage individualism and discourage cooperation.

For example, stock brokers within a large financial firm might act as individualists in order to retain their clientele and to keep secret their investing strategies. With these reasons to resist knowledge harvesting, the systems and processes of knowledge harvesting face additional challenges to successful implementation.

Issues of Organizational Culture

Organizational culture can be defined as the "character or personality of the organization" (Ribiere 32). It encompasses the norms and values of the organization, as well as the accepted behaviors and actions of organization members. The culture of an organization has a significant impact on the sharing of knowledge among the organization's members, a key influencing factor in a successful Knowledge Harvesting process. It is therefore essential for the organization to develop a culture in which knowledge sharing is encouraged.

While developing objective measures of organizational culture is a matter of debate, Davenport and Prusak have identified four main aspects of culture whose presence increases information sharing. The first, altruism, is a general feeling of goodwill in which members of an organization share their knowledge for the greater good of the organization, without expecting anything in return (Ribiere 51). Reciprocity, the second characteristic, is the belief that if one member of an organization contributes useful knowledge, others will also contribute knowledge that will help that member (52). The third factor, repute, is the belief by members of an organization that contributing knowledge will improve their reputation within the organization, which can lead to tangible benefits such as job security and bonuses. Trust is the final characteristic, considered the most important of the four. Without trust, Davenport and Prusak find that all knowledge management efforts will fail (54). Trust can be fostered internal job characteristics such as benefits as work environment, as well as external characteristics such as job satisfaction (57).

In his research into the factors affecting success of knowledge management initiatives, Ribiere defined success as growth of project resources, growth of knowledge shared within a team, valuable input from more than just one or two team members, and some evidence of financial return on investment (78). Among the organizations that had successful knowledge management initiatives, Ribiere identified high levels of trust and solidarity among organization members as characteristics that promote success (128). Cress et. al, found that members of an organization often overestimated the cost of contributing to a knowledge database, but when they understood the value of their contributions in relation to the actual costs, they were more likely to contribute their knowledge (Cress 375). They did discover, however, that in some extreme situations, members would not contribute information simply to hurt other members of the organization, even when it decreased their personal benefits (374). This shows the importance of altruism, reciprocity, repute, and trust as cultural factors within the organization; if these characteristics are present in the culture, it is unlikely that such extreme behavior would occur.

Current Practices in Knowledge Harvesting

Clearly, there is a demand for knowledge harvesting. However, in order to successfully harvest knowledge, an organization must go much further than just making technological changes. On the contrary, successful knowledge harvesting is largely based upon organizational adjustments; technological changes simply support these organizational activities (Laudon and Laudon, 446). The good news is that most successful knowledge harvesting plans use very similar organizational activities, allowing us to extract a general organizational plan to successfully harvest knowledge (Eisenhart, 52; Chua, 252). Knowledge Harvesting Inc.'s process for knowledge harvesting does a good job of capturing the processes used by most successful knowledge harvesting implementations. According to Knowledge Harvesting Inc., harvesting requires the following organizational activities: focus—deciding upon the

knowledge that needs to be sought out; find and elicit—identifying the experts and interviewing them; and organize—categorizing, expanding, and pruning the results in an appropriate manner (Eisenhart, 52).

To illustrate an implementation of this harvesting process, we will examine how the Army's Center for Army Lessons Learned (CALL) harvests knowledge. Even though CALL does not consciously use Knowledge Harvesting Inc.'s process for harvesting, we will notice how CALL still distinctly adheres to the same organizational procedures outlined by Knowledge Harvesting Inc.

Case Study

The first step in the process of knowledge harvesting, focus, involves determining what knowledge to harvest. Many firms determine what knowledge to harvest by having the harvesters meet with management. Note that the harvesters can either be employees from inside the company or external consultants. Through the meeting, the harvesters can determine the target audience and choose to harvest knowledge based on the target audience's needs (Eisenhart, 50). CALL takes this process a step further, by not only deciding upon the content to harvest based on the target audience, but also upon the significance of the data to be harvested—"potential for generating…future strategic value" (Chua, 254). This means that valuations of knowledge will have to be determined; CALL works with senior Army officers to identify specific learning objectives that the harvested knowledge needs to fulfill. This latter method resolves, from the very start, the potential for information overload due to harvesting too much knowledge and overwhelming target audiences with superfluous data (Chua, 257). However, in addition to avoiding information overload, excessive focus can also cause CALL to "miss" information; high-potential information is lost due to its lack of fitting the established criteria (fitting a team's needs and fulfilling a learning objective).

Depending on the chosen method, once the knowledge to be harvested has been determined, the next step is to "find and elicit" the knowledge. Find and elicit refers to discovering the experts in a knowledge area and then interviewing or extracting the desired knowledge from them. Common strategies for the find and elicit step employ a first-person or second-person approach. Cerabyte Inc. produces software, called Infinos System, which supports a first-person style. The system asks experts a series of leading questions while he or she works through a process in order to obtain best practices and procedures knowledge. "The questions are designed to help employees clearly articulate best practices by encouraging them to think, clarify, and record their actions" (Duffy, 60). Georgia-Pacific, a forest products company, uses a second-person format of extracting knowledge. The company first directly interviews both identified experts and the target audience to determine gaps in knowledge. The harvester then interviews the experts again to fill in these gaps (Eisenhart, 50). CALL's method to find and elicit knowledge utilizes a unique third-person approach of gathering knowledge, which places less emphasis on the 'expert' and more emphasis on outside observers. To gather information from identified experts—soldiers in the field—CALL first assembles a data collection team of about eight to fifty "subject matter experts…from various units across the Army" (Chua, 254). The team members are selected specifically for their removal from the event being studied, in order to promote objectivity. The team members are also selected such that many different fields of study are represented, which "enables deep knowledge to be collected for each event" and enhances "the reliability and validity" of gathered knowledge (Chua, 255). Additionally, the use of outsiders from different specialties allows for "fresh ideas" to be added to analysis (Chua, 258). The CALL data collectors are then dispatched alongside troops to observe, capture, and document in the midst of the event (Chua, 257). The collectors document with a variety of media in order to capture the reality of the situation as fully as possible, usually employing video and photographs, as well as diagrams and written descriptions. This detail not only allows end-users to gain from the knowledge by

'reliving' the experience (Chua, 255), but also to retain the context of the event (Chua, 257). Occasionally, the collectors will also interview a wide range of personnel to gather feedback and interpretations in order to get more perspectives on the event. Note that a peripheral benefit of this method is that it is not affected by an organization's culture of sharing. CALL's method is not centered on the individual expert and his or her desire to share knowledge; rather, CALL focuses on third-party observers to make unbiased observations from multiple viewpoints. Despite its benefits, CALL's avoidance of the expert is also the third-person approach's biggest shortcoming. Many companies have found that their knowledge harvesting projects are reliant upon the fact that their harvested knowledge has come directly from experts. Employees are more receptive of harvested knowledge that come first- or second-hand from experts. They claim that information is more "memorable" and "educational" if the sources are fellow employee-experts (Keelan).

Organize, the third step in the Knowledge Harvesting process, refers to categorizing, fixing, adding, or pruning the gathered information. In this stage, the harvester asks: "If I was in this situation, would I get to the same place [the experts] did [using this gathered knowledge]?" If the answer is no, then the harvester needs to better organize or edit the gathered knowledge (Eisenhart, 52). CALL utilizes a very extensive human-centered approach to the organize step. First, the data collection team communicates within itself to organize individual insights. These insights are then sent to CALL headquarters to be analyzed by another group, the analysts, who act a conduit, "seek[ing] input and insights from other Army experts". The analysts are ultimately responsible for further organizing and editing in an attempt to construct new knowledge from the disparate pieces of knowledge sent in from the data collectors. After indexing this new knowledge, it is transmitted electronically for review by other professionals in the Army. Finally, after review by Army professionals—upon which the knowledge is codified and divided into 'lessons'—the data collectors return to CALL headquarters and review all of the compiled lessons. Through a process called the 'murder board', the data collectors decide which lessons are thrown out and which are important enough to be packaged for distribution (Chua, 255-256). Despite this exhaustive and complete approach to organizing, the amount of work that must be done—in addition to the lack of use of technology—to reach this point of completeness dramatically increases the delay in getting harvested information to the target audience. This could pose a major problem if the harvested knowledge is time-sensitive in the environment (who wants obsolete information?) or within the target audience (what team wants knowledge that it needed last week?).

Implementation Differences

Amidst the presence of general organizational processes for knowledge harvesting, however, there is a lot less similarity in how firms have implemented each process. Some have used technological approaches, some have used human-based approaches, some have used interviewing, and some have used observations to support knowledge harvesting processes. Despite such variety in implementation practices, this has helped firms to distinguish themselves and their knowledge harvesting agendas from competitors.

We have already discussed how many firms implement focus by determining target audiences and their knowledge needs. We have also seen how CALL also focuses on the 'strategic value' of harvested knowledge. Yet another method to determine what knowledge to harvest is to focus only upon knowledge has been successful in the past. Currently, firms have implemented this solution with a technology-based approach; Intellectual Capital Management (ICM) software is employed to focus solely on extracting and recording employee expertise during successful business process creation or change. This method tries to capitalize on "practices that have proven efficient and effective" (Duffy, 59).

Find and elicit has even more varied implementation than focus. We have already observed some of these different implementations in the first-, second-, and CALL's third-person approaches. However, whereas these approaches defined humans as the experts from whom to gather knowledge, some implementations choose non-human experts to 'interview.' Hewlett-Packard's implements find and elicit by using software to examine the company's own technical notes, frequently asked questions, help files, call log extracts, and user submissions (Delic, 75). Roche Labs, a healthcare company, gathers desired knowledge by 'interviewing' global news sources, specialty publishers, health care Web sites, government sources, and the firm's proprietary internal information systems (Laudon and Laudon, 424).

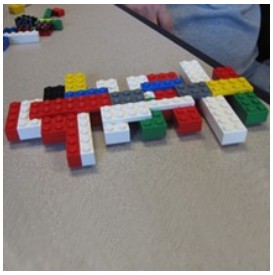
Lastly, the organize process has also seen a number of different implementation. We have seen how CALL uses a human-centered approach. Another example of this is KPMG, an international auditing and accounting firm. To organize gathered knowledge, KPMG employs an extensive staff of "analysts and librarians" to categorize elicited information and to assess its quality (Laudon and Laudon, 424). Another method of organizing is to use a software-centered approach, which can be exampled by Cerabyte's Infinos System. After eliciting best practices knowledge, the system itself will try to ascertain bottlenecks, risks, and tradeoffs. The system will then interact with users and experts to solve these issues (Duffy, 60).

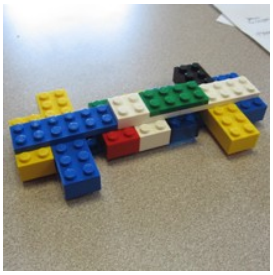
Bridge Building

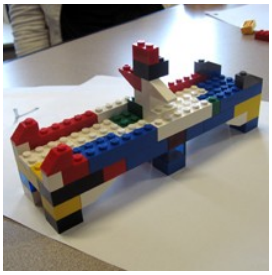
To illustrate the importance of Knowledge Harvesting in training new employees and gaining knowledge for organizations, we constructed an experiment to compare and contrast two different systems. One system had an outline of the knowledge that should be entered into system; the other contained no guidelines whatsoever and allowed entry of any free-form comments. A group of students was divided into two groups of equal size and given the same instructions: Build a model bridge from Lego blocks that spans a six-inch ravine and is at least two inches wide. The two groups developed different designs based on their interpretations of these instructions. Neither group had any idea that they would need to explain how to build their bridge via written instructions. After the groups were done building, Group A was given a detailed form asking for certain sketches of their model, what problems they had run into, and any other comments they would like to add. Group B was merely told to write instructions on how to build the bridge. The two sets of instructions represent the aforementioned Knowledge Harvesting systems.

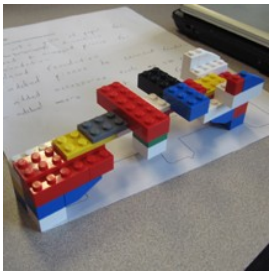
When the groups traded instructions and were asked to reproduce each other's models, both groups struggled to reconstruct the bridges, but for very different reasons. The instructions that Group A wrote prompted them to draw the different layers of their bridge, which they did, but not completely to scale. The form did not specify whether layer one was the top or bottom layer. When Group B received these instructions, they had a general idea of what the bridge should look like, but due to the lack of knowledge about the organization of the layers and the size of the pieces, were unable to reconstruct the bridge exactly. The instructions that were written by Group B were very vague and contained a great deal of information that Group A did not find pertinent to the building of the bridge.

Exhibit 31.: Bridge building exercise


Group A Original


Reconstruction of A by B


Group B Original


Reconstruction of B by A

This situation is not far from what businesses must cope with today. Effective Knowledge Harvesting helps create value for the organization and makes it easier to train new employees, because the usefulness of certain information is already determined. Group A, by answering prompts similar to that which would be given in an exit interview, was able to establish exactly what information should be passed the next group in order to construct the bridge. However, Group B could not establish what information was essential, and when the instructions were passed on, Group A had a hard time determining what information would help them remake the bridge and what was extraneous.

Group B also had a hard time constructing their bridge despite the fact that they had most of the essential information. The problem in this case lies more so in the presentation of the information. Group B had all of the information that they needed, but they could not determine where to place each of the layers nor could they determine the scale of the drawing in the directions. If the experiment could be repeated, the follow up form would be further revised so as to better represent the way in which the bridge could be constructed.

An organization implementing Knowledge Harvesting techniques, such as exit interviews, informal transition processes, or periodic meetings with employees, has an advantage when it comes to retaining knowledge that is important to the organization. As with the bridge building experiment, when an exit interview is conducted, the company gains possession of information that previously would have left the company along with the employee. Rather than valuable experience being lost when the employee leaves or is required to pass a project on to another team, the company maintains control of that knowledge. The training process, as illustrated in the bridge building example, can also be shortened with effective Knowledge Harvesting. The employee being trained no longer has to solve every problem already solved by previous employees. Through a brief description of the problem and solution, the new employees can now overcome obstacles without wasting time rediscovering the solution, and organizations can save time and money. The organization would also have to go through a revision process to make sure that the information is being presented to the employees in a way such that it is helpful.

Recommendations

Based on our analysis of the demand for and factors affecting Knowledge Harvesting, we created a list of general recommendations that organizations should heed if they expect to remain competitive in the knowledge market of the near future. By following these recommendations, the knowledge created by their employees can be successfully harvested and then, in turn, used to increase the productivity and uniqueness of the organization's projects and processes. These recommendations can be summarized as: a broad implementation of standardized Knowledge Harvesting techniques in organizations, the creation of a cooperative work environment, the creation of a definitive

rewards system, and the disassociation between the idea of sharing knowledge and that of being replaced within the organization.

The broad adoption of a standardized Knowledge Harvesting system is, as we have demonstrated, becoming an ever pressing demand on organizations today. The baby boomer generation, which makes up a large part of the current workforce of knowledge workers, is steadily moving towards retirement. For many organizations this will mean a loss of much of their experience and knowledge, unless they put into place a dedicated system to harvest these from their departing employees. However, the system must be standardized across the entire organization, as differences in technique or thoroughness can create problems with organizing and sharing the collected pieces at a later date.

Second, the creation of a working environment centered on sharing or cooperating with fellow employees is essential to the Knowledge Harvesting process. If employees feel that they are part of a larger team working towards a common goal, it will become easier for them to share their own knowledge in the hopes that it will further the team as a whole. In contrast, a work environment that fosters individualism and the hoarding of knowledge will make individual workers reluctant to share their own knowledge with others, as it may give others an advantage over the sharer. One possible way organizations might accommodate for this is by structuring employees into teams centered on certain projects, and then rewarding those teams or individuals who contribute the most in the form of shared knowledge or experience. This same idea could then be extended to the individual level by rewarding those employees who contribute knowledge to a knowledge base and then have their contributions used by their co-workers. These developments could have the effect of both providing an incentive to share and allowing the more experienced employees to pass on knowledge to those with less experience through direct interaction.

Directly related to the creation of a strong group work environment is the creation of a definitive rewards system that helps to facilitate knowledge sharing. This reward program would have to track both the amount of work any given employee accomplishes and how much knowledge they contribute to other employees of the organization. One idea for this could be a Wikipedia-like database where information is collectively edited and shared, but that would keep track of those who contributed the information for tracking purposes. Though in many cases the reward system developed would have to accommodate for the organizational culture in which the information is being shared, as each organization might handle group collaboration differently (e.g. one organization might hold group sessions where ideas are traded and discussed, while another might rely on an intranet-based discussion board).

The final, and possibly one of the most essential, recommendations for the implementation of Knowledge Harvesting processes is the disassociation of sharing knowledge with the fear of being replaced within the organization. In many cases, employees are hesitant to share their knowledge as they believe that the company will eventually decide that the employees are no longer useful, and can thus be replaced with less experienced and lower paid workers who can use the experienced employees' harvested knowledge. In order to dispel this myth, organizations will have to make an effort to show their employees that sharing knowledge is beneficial to them, their fellow employees, and the organization as a whole, while also demonstrating that it is the employees who do not share with others that are replaced. This sort of organizational culture will create a sense of the necessity of sharing while also giving employees a greater sense of pride and security in their jobs.

Although these recommendations are far from exhaustive in respect to actions which organizations can take to ensure the preservation of their current employees' knowledge, they are the essential pieces needed to create an

effective Knowledge Harvesting system. The production of usable artifacts of knowledge, and especially the most effective and extensive methods for doing so, can be expected to become a major issue within organizations, both large and small, in the coming years. Those organizations which take these initial steps towards creating a Knowledge Harvesting system will be the same organizations that can expect to be at the forefront of their industries and the public sphere of the future.

Editor

Kevin Desouza

References

Brown, John Seely and Paul Duguid. "Balancing Act: How to Capture Knowledge Without Killing It." Harvard Business Review May-June 2002: 73-80.

Chua, AYK, W Lam and S Majid. "Review - Knowledge reuse in action: the case of CALL." Journal of Information Science 32.3 (2006): 251-260.

Coakes, Elayne, ed. Knowledge Management: current issues and challenges. Hershey, PA: IRM Press, 2003.

Cress, Ulrike, Joachim Kimmerle and Friedrich W. Hesse. "Information exchange with shared databases as a social dilemma: the effect of metaknowledge, bonus systems, and costs." Communication Research 33.5 (2006): 370-391.

Delic, Kernal A and Lahaix, Dominique. "Knowledge harvesting, articulation, and delivery." Hewlett-Packard Journal 1998: 74-81.

Duffy, Jan. "Managing Intellectual Capital." Information Management Journal 35.2 (2001): 59-63.

Eisenhart, Mary. "Gathering Knowledge While It's Ripe." Knowledge Management (2001): 48-53.

Harrison, Warren. "Whose Information Is It Anyway?" IEEE Software 20.4 (2003): 5-7.

"Intelligence at risk: capturing employee knowledge before retirement." Industrial Engineer August 2005: 14.

Keelan, Tim. "Keane Inc. Utilizes StoryQuest, iPods and Podcasting to Support Sales Training and Organizational Change." Business Wire 3 February 2006.

"Knowledge Harvesting - KM Toolkit: inventory of tools and techniques." 31 May 2005. National Electronic Library for Health. Crown. 26 November 2006 <http://www.nelh.nhs.uk/knowledge_management/km2/harvesting_toolkit.asp>.

Laudon, Kenneth C. and Jane P. Laudon. Management Information Systems: Managing the Digital Firm. New Jersey: Pearson Education, 2006.

Ribiere, Vincent Michel. Assessing knowledge management initiative successes as a function of organizational culture. Dissertation. The George Washington University. District of Columbia, 2001.

7. How hardware and software contribute to efficiency and effectiveness

Editor: Larry Press (California State University-Dominguez Hills, USA)

Reviewer: Geoff Dick (University of New South,Australia)

Learning objectives

- describe and quantify the improvement in electronic technology used in information processing systems
- describe and quantify the improvement in storage technology used in information processing systems
- describe and quantify the improvement in communication technology used in information processing systems
- describe and differentiate between the major software development platforms: batch processing, time-sharing, personal computers, local area networks and the Internet.
- describe and give examples of the following modern, Internet-based software categories: open source, applications with user-supplied content, composite applications, software as a service, mobile, portable and location-aware applications, long tail applications, and collaboration support applications.
- describe impediments to innovation and the adoption of new applications: limited communication and data center capacity, inequity in access to information technology, organizations with vested interests in the status quo, and concerns over privacy and security.

Information technology is improving at an accelerating rate. This opens the way for innovative applications, which make organizations and individuals more efficient and effective. This chapter outlines hardware progress, which has led to new forms of software and software development. Software evolution has brought us to the current era of Internet-based software. After describing some of the characteristics of Internet-based software, we ask whether the progress we have enjoyed will continue and conclude with a discussion of some of the non-technical issues, which tend to impede that progress. The sections read as follows:

- Hardware progress
- Software progress
- Internet based software
- Will the progress continue?
- Bumps in the information technology road

Hardware progress

Technology is improving rapidly. It seems that a new cell phone or computer is outdated the day after you buy it. This is nothing new. Consider manned flight for example. The Wright Brothers first flight in 1903 lasted only 12 seconds and covered 37 meters.[16] Once we understand the science underlying an invention, engineers make rapid improvements in the technology. Within 66 years of that historical first flight, Apollo 11 landed on the moon.

Would you guess that that information technology progress is slowing down, holding steady, or accelerating? It turns out that it is accelerating—the improvements this year were greater than those of last year, and those of next year will be still greater. We often use the term exponential to describe such improvement. Informally, it means that something is improving very rapidly. More precisely, it means that the improvement is taking place at a constant rate, like compound interest. In this section, we consider three technologies underlying IT—electronics, storage, and communication. Each of these technologies is improving exponentially.

Sidebar: Exponential growth

Take, for example, a startup company with USD 1,000 sales during the first year. If sales double (100 per cent growth rate) every year, the sales curve over the first 12 years will be:

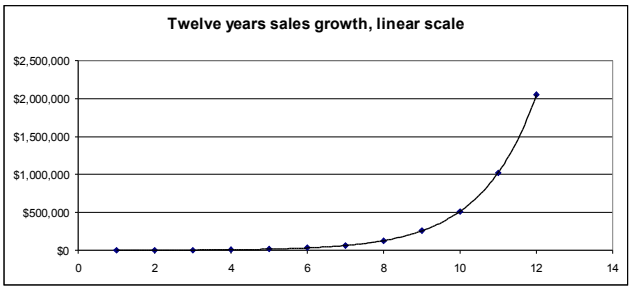


Exhibit 32: Exponential growth graphed with a linear scale

Sales are growing exponentially, but the curve is almost flat at the beginning and then shoots nearly straight up. Graphing the same data using a logarithmic scale on the Y axis gives us a better picture of the constant growth rate:

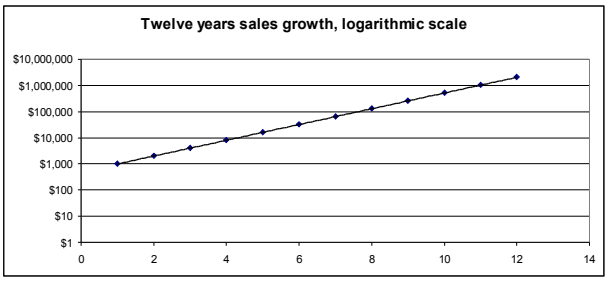


Exhibit 33: Exponential growth graphed with a logarithmic scale

Note that since the growth rate is constant (100 per cent per year in this example), the graph is a straight line. You can experiment with differing growth rates using the attached spreadsheet.

16 http://www.nasm.si.edu/galleries/gal100/wright1903.html.

Of course, nothing, not even technology, improves exponentially forever. At some point, exponential improvement hits limits, and slows down. Consider the following graph of world records in the 100 meter dash:[17]

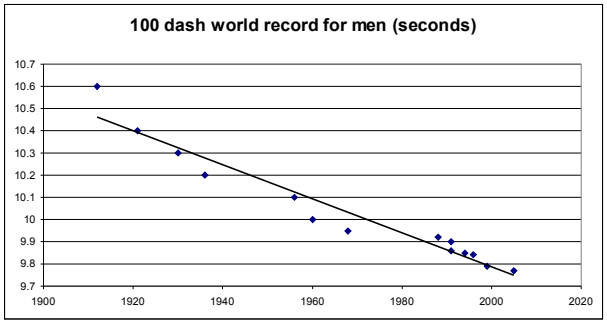


Exhibit 34: Linear improvement

There is steady improvement, but it is at a roughly linear rate. The record improves by a constant amount, not at a constant rate.

Progress in electronic technology

Transistors are a key component in all electronic devices—cell phones, computers, portable music players, wrist watches, etc. A team of physicists at the Bell Telephone research laboratory in the United States invented the first transistor, shown below.

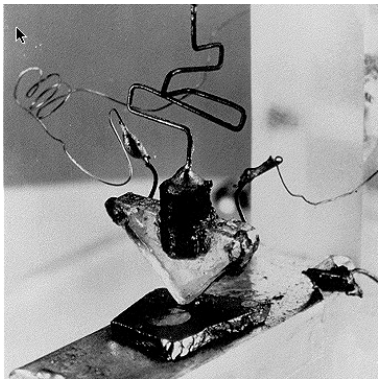


Exhibit 35: The first transistor

This prototype was about the size of a 25 cent coin, and, like the Wright brothers' first plane, it had no practical value, but was a proof of concept. Engineers soon improved upon the design. In 1954, Texas Instruments began manufacturing transistors. They were about the size of a pencil eraser, and several would be wired together to make a circuit for a device like a transistor radio. In the late 1950s, engineers began making integrated circuits (ICs or chips) which combined several transistors and the connections between them on a small piece of silicon. Today, a single IC can contain millions of transistors and the cost per transistor is nearly zero.

Consider the central processing unit (CPU) chip that executes the instructions in personal computers and other devices. Most personal computers use CPU chips manufactured by Intel Corporation, which offered the first commercial microprocessor (a complete CPU on a chip) in 1971. That microprocessor, the Intel 4004, contained 2,300 transistors. As shown here, Intel CPU transistor density has grown exponentially since that time:[18]

17 http://en.wikipedia.org/wiki/World_Record_progression_100_m_men.
18 http://www.intel.com/pressroom/kits/quickreffam.htm.

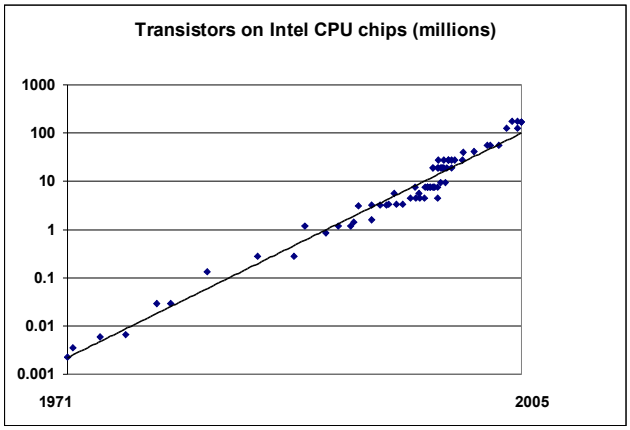


Exhibit 36: Improvement in electronic technology

With the packaging of multiple CPU cores on a single chip, transistor counts are now well over one billion.

Intel co-founder, Gordon Moore, predicted this exponential growth in 1965. He formulated Moore's Law, predicting that the number of transistors per chip that yields the minimum cost per transistor would increase exponentially. He showed that transistor counts had pretty much doubled every year up to the time of his article and predicted that improvement rate would remain nearly constant for at least 10 years.[19]

We have used Intel CPU chips to illustrate exponential improvement in electronic technology, but we should keep in mind that all information technology uses electronic components. Every computer input, output or storage device is controlled and interfaced electronically, and computer memory is made of ICs. Communication systems, home appliances, autos, and home entertainment systems all incorporate electronic devices.

Progress in storage technology

Storage technology is also improving exponentially. Before the invention of computers, automated information processing systems used punched cards for storage. The popular IBM card could store up to 80 characters, punched one per column. The position of the rectangular holes determined which character was stored in a column. We see the code for the 10 digits, 26 letters and 12 special characters below.[20] [21]

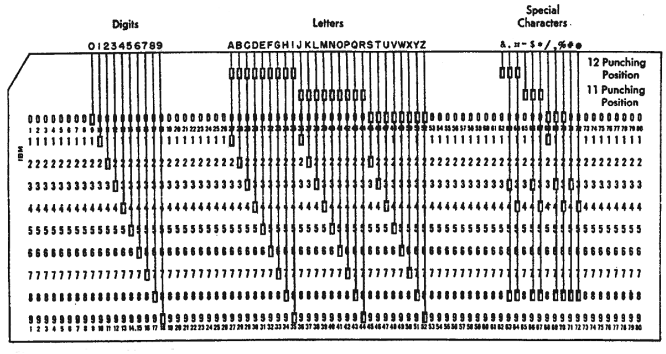


Exhibit 37: Punch card code for alphabetic and numeric symbols.

19 http://download.intel.com/research/silicon/moorespaper.pdf.
20 http://www.columbia.edu/acis/history/026-card.jpg
21 Http://www.cs.uiowa.edu/~jones/cards/history.html.

Punch card storage was not very dense by today's standards. The cards measured 3 ¼ by 7 (3/8) inches,[22] and a deck of 1,000 was about a foot long. Assuming that all 80 columns are fully utilized, that works out to about 48,000 characters per cubic foot, which sounds good until we compare it to PC thumb drives which currently hold up to 8 billion characters.

Every type of data—character, audio, video, etc.—is stored using codes of ones and zeros called bits (short for binary digits). [23] Every storage technology distinguishes a one from a zero differently. Punched cards and tape used the presence or absence of a hole at a particular spot. Magnetic storage differentiates between ones and zeros by magnetizing or not magnetizing small areas of the media. Optical media uses tiny bumps and smooth spots, and electronic storage opens or closes minute transistor "gates" to make ones and zeros.

We make progress both by inventing new technologies and by improving existing technologies. Take, for example, the magnetic disk. The first commercially available magnetic disk drive was on IBM's 305 RAMAC (Random Access Method of Accounting and Control) computer, shown below.

Exhibit 38: RAMAC, the first magnetic disk storage device

IBM shipped the RAMAC on September 13, 1956. The disk could store 5 million characters (7 bits each) using both sides of 50 two-foot-diameter disks. Monthly rental started at USD 2,875 (USD 3,200 if you wanted a printer) or you could buy a RAMAC for USD 167,850 or USD 189,950 with a printer. (In 1956, a cola or candy bar cost five cents and a nice house in Los Angeles USD 20,000).

Contrast that to a modern disk drive for consumer electronic devices like portable music and video players. The capacity of the disk drive shown here is about 2,700 times that of a RAMAC drive, and its data access speed and transfer rate are far faster, yet it measures only 40x30x5 millimeters, weighs 14 grams, and uses a small battery for power. The disk itself is approximately the size of a US quarter dollar.

22 This was the size of the US paper currency at the time Herman Hollerith invented the machines that used them. His company eventually became IBM.

23 For example, the American Standard Code for Information Interchange (ASCII) code of the letter *Q* is *01010001*.

11.

Exhibit 39: A modern disk storage device, manufactured by Seagate Technology

Progress in communication technology

People have communicated at a distance using fires, smoke, lanterns, flags, semaphores, etc. since ancient times, but the telegraph was the first electronic communication technology. Several inventors developed electronic telegraphs, but Samuel Morse's hardware and code (using dots and dashes) caught on and became a commercial success. Computer-based data communication experiments began just after World War II, and they led to systems like MIT's Project Whirlwind, which gathered and displayed telemetry data, and SAGE, an early warning system designed to detect Soviet bombers. The ARPANet, a general purpose network, followed SAGE. In the late 1980s, the US National Science Foundation created the NSFNet, an experimental network linking the ARPANet and several others—it was an internetwork. The NSFNetwork was the start of today's Internet.[24]

Improvement in the Internet illustrates communication technology progress. There are several important metrics for the quality of a communication link, but speed is basic.[25] Speed is typically measured in bits per second —the number of ones and zeros that can be sent from one network node to another in a second. Initially the link speed between NSFNet nodes was 64 kilobits per second, but it was soon increased to 1.5 megabits per second then to 45 megabits per second.[26]

24 Press, Larry, Seeding Networks: the Federal Role, Communications of the ACM, pp 11-18, Vol. 39., No. 10, October, 1996, http://bpastudio.csudh.edu/fac/lpress/articles/govt.htm. A one page history of the early Internet and the ideas behind it http://bpastudio.csudh.edu/fac/lpress/471/hout/netHistory/.

25 Packet loss and data transfer rates are also common measures of communication quality. Researchers monitoring Internet performance have observed that between the spring of 1997 and fall of 2006 packet loss rates between North America and other regions of the world have typically fallen between 25 and 45 per cent per year. Packet loss rates to developed nations are under 1 per cent, but Africa has the highest loss rates and is falling further behind the other regions. They found comparable improvement when monitoring data transfer rates. (xx reference to SLAC site).

26 The letters *k* and *m* are abbreviations for *kilo* and *mega* respectively. One kilobit is 1,024 bits and a megabit is 1,048,576 bits. For most practical purposes, we can think of them as 1 thousand and 1 million bits.

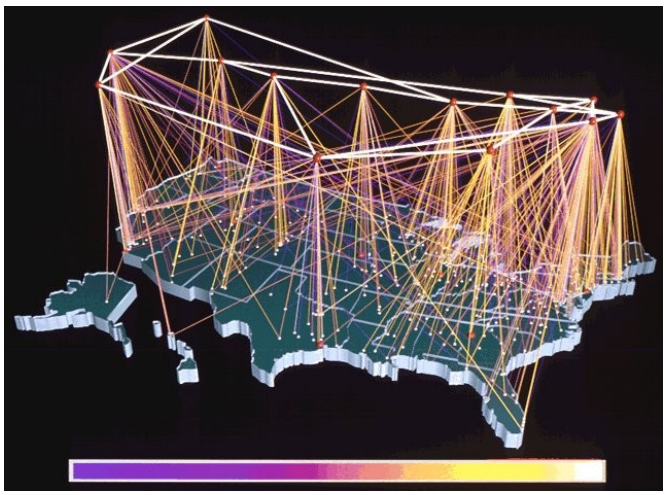


Exhibit 40: The NSFNet backbone connected connected 13 NSF supercomputer centers and regional networks

Sidebar: Commonly used prefixes

Memory and storage capacity are measured in bits—the number of ones and zeros that can be stored. Data transmission rates are measured in bits per a unit of time, typically bits per second. Since capacities and speeds are very high, we typically use shorthand prefixes. So, instead of saying a disk drive has a capacity of 100 billion bits, we say it has a capacity of 100 gigabits.

The following table shows some other prefixes:

Prefix	$1{,}024^n$	English term	Approximate number
kilo	1	Thousand	1,000
mega	2	Million	1,000,000
giga	3	Billion	1,000,000,000
tera	4	Trillion	1,000,000,000,000
peta	5	quadrillion	1,000,000,000,000,000
exa	6	Quintillion	1,000,000,000,000,000,0000
zetta	7	Sextillion	1,000,000,000,000,000,0000,000
yotta	8	Octillion	1,000,000,000,000,000,0000,000,000

Note that the numbers in the fourth column are approximate. For example, strictly speaking, a megabit is not one million (1,000,000) bits it is 1,024 x 1,024 (1,048,576) bits. Still, they are close enough for most purposes, so we often speak of, say, a gigabit as one billion bits.

Capacities and rates may also be stated in bytes rather than bits. There are 8 bits in a byte, so dividing by 8 will convert bits to bytes and multiplying by 8 will convert bytes to bits.

The NSFnet was the first nationwide Internet backbone, but today there are hundreds of national and international backbone networks. High speed links now commonly transmit 10 gigabits per second, and a single fiber can carry multiple data streams, each using a different light frequency (color).[27] Of course, progress continues.

27 A gigabit is 1,024 megabits.

For example, Siemens researchers have reliably transmitted data at 107 gigabits per second over a one hundred mile link and much faster speeds are achieved in the lab.[28]

There has been similar improvement in local area network (LAN) technology. Ethernet is the most common LAN technology. When introduced in 1980 Ethernet links required short, thick cables and ran at only 10 megabits per second. Today, we use flexible wires, Ethernet speed is 10 gigabits per second today, and standards groups are working on 40 and 100 gigabits per second.[29]

Individuals and organizations also use wireless communication. The WiFi[30] standard in conjunction with the availability of license-free radio frequency bands led to the rapid proliferation of wireless local area networks in homes and offices. When away from the home or office, we often connect to the Internet at WiFi hotspots, public locations with Internet-connected WiFi radios. We also connect distant locations with wide-area wireless links when installing cable is impractical. And we can use satellite links to reach remote locations, but they are expensive and introduce a delay of about .24 seconds because of the distance the signal must travel.[31]

Cellular telephone networks are also growing rapidly. There were 1,263 million cellular users in 2000, and that had increased to 2,168 five years later.[32] (The number of wired telephone lines fell from 979 million to 740 million during the same period). Cellular communication has improved with time, and, in developed nations, we are deploying third and even fourth generation technology, which is fast enough for many Internet applications.

28 http://www.siemens.com/index.jsp?sdc_p=fmls5u01426061ni1079175pcz3&sdc_bcpath=1327899.s_5,&sdc_sid=33487116105&.

29 Gittlin, Sandra, Ethernet: How high can it go?, Network World, 11/22/06, http://www.networkworld.com/research/2006/112706-ethernet-10g.html.

30 *WiFi* is a trade name belonging to a trade association, the WiFi Alliance. It is an association of manufacturers who make equipment based on technical standard 802.11 if an engineering society, the Institute of Electrical and Electronic Engineers. The WiFi alliance was formed to resolve standardization differences, publicize and market the technology, and test and certify equipment as being IEEE 802.11 standard compliant.

31 Radio signals propagate at the speed of light (300 kilometers per second) and the round trip distance to a communication satellite is around 70,000 kilometers.

32 International Telecommunication Union database, http://www.itu.int/ITU-D/ict/statistics/.

Software progress

The hardware progress we have enjoyed would be meaningless if it did not lead to new forms of software and applications. The first computers worked primarily on numeric data, but early computer scientists understood their potential for working on many types of application and data. By 1960, researchers were experimenting non-numeric data like text, images, audio and video; however, these lab prototypes were far too expensive for commercial use. Technological improvement steadily extended the range of affordable data types. The following table shows the decades in which the processing of selected types of data became economically feasible:

Exhibit 41.: Commercial viability of data types

Decade	Data Type
1950s	Numeric
1960s	Alphanumeric
1970s	Text
1980s	Images, speech
1990s	Music, low-quality video
2000s	High-quality video

But none of this would have been possible without software, and we have seen evolution in the software we use, and, underlying that, the platforms we use to develop and distribute it.[33] Let us consider the evolution of software development and distribution platforms from batch processing to time sharing, personal computers, local area networks, and wide area networks.

Internet resource:

Listen to TCP/IP co-inventor Vint Cerf's Stanford University presentation on the Internet and its future.

In this historic video, Cerf's collaborator Bob Kahn and other Internet pioneers describe the architecture and applications of their then brand new research network.

Batch processing

The first commercial computers, in the 1950s, were extremely slow and expensive by today's standards, so it was important to keep them busy doing productive work at all times. In those days, programmers punched their programs into decks of cards like the one shown above, and passed them to operators who either fed the cards directly into the computer or copied them onto magnetic tape for input to the computer. To keep the computer busy, the operators made a job queue—placing the decks for several programs in the card reader or onto a tape. A master program called the operating system monitored the progress of the application program that was running on the computer. As soon as one application program ended, the operating system loaded and executed the next one.

Batch processing kept the computers busy at all times, but wasted a lot of human time. If a programmer made a small error in a program and submitted the job, it was typically several hours before he or she got the resultant error message back. Computer operators also had to be paid. Finally, professional keypunch operators did data entry using machines with typewriter-like keyboards that punched holes in the cards. This tradeoff of human for computer time reflected the fact that computers were extremely expensive.

33 The software platform evolution outlined in this section was based on prototypes which had been developed in research labs many years earlier. We are giving the chronology of their commercial deployment, not their invention. A short survey of that invention is given in Press, L., Before the Altair — The History of Personal Computing, Communications of the ACM, September 1993, Vol. 36, no 9, pp 27-33.

Time sharing

By the early 1960s, technology had progressed to the point, where computers could work on several programs at a time, and time-shared operating systems emerged as a viable platform for programming and running applications. Several terminals (keyboard/printers) were connected to a single computer running a time-sharing operating system. Programmers entering instructions or data entry operators used the terminals. They received immediate feedback from the computer, increasing their productivity.

Let's say there were 10 programmers working at their own terminals. The operating system would spend a small "slice" of time—say a twentieth of a second—on one job, then move to the next one. If a programmer was thinking or doing something else when his or her time slice came up, the operating system skipped that person. Since the time slices were short, programmers had the illusion that the computer was working only on their own job and they got immediate feedback in testing their programs. The computer "wasted" time switching from one job to the next, but it paid off in saving programmer time.

Exhibit 42: An early timesharing terminal

Time-sharing terminals were also used for data entry, so we began to see applications in which users, for example, airline reservation clerks entered their own data. Professional keypunch operators began to disappear.

Personal computers

Time-sharing continued to improve resulting in a proliferation of ever smaller and cheaper "mini-computers". They might be the size of a refrigerator rather than filling a room, but users still shared them. As hardware improved, we eventually reached the point where it was economical to give computers to individuals. The MITS Altair, introduced in 1975, was the first low-cost personal computer powerful enough to improve productivity. By the late 1970s, programmers, professional users and data entry workers were using personal computers. They were much less powerful than today's PC, but they began to displace time-sharing systems.

Exhibit 43: MITS Altair

Programmers could write and test programs on their own machines—they could, for the first time, own their own tools and be independent of employers. They could also work at home, and hobbyist programming began to spread. Secretaries and typists began using personal computers for their work, using desktop computers with built-in programs for creating documents—the original "word processors". Knowledge workers—managers, engineers, and others—began using personal computers to increase their productivity. Software companies quickly offered programs for word processing, maintaining small databases and doing spreadsheet calculations on personal computers.

The first personal computers used the same types of terminals as time-shared computers. They consisted of a keyboard and a printer or screen, which could display characters, but not pictures or diagrams. As such, the user interface of early personal computer operating systems was similar to that of time-sharing systems. The computer displayed a prompt indicating that it was ready for a command, which the user typed. For example, the user could erase a file called myfile.txt by typing the command:

```
> delete myfile.txt
```

As personal computer hardware improved, it became feasible to move from character displays to displays which could turn the dots making up the characters on and off individually. This made it possible to display and work with images as well as characters. Graphic displays, when coupled with a pointing device like a mouse, ushered in applications and operating systems with graphical user interfaces (GUIs). One could now delete a file by dragging icon or small picture representing it to a trash can on the screen. The Apple Macintosh, introduced in 1984, was the first low-cost personal computer with a GUI operating system.

Exhibit 44: The original Apple Macintosh

Users liked GUIs because they did not have to memorize commands to use a program, and operating systems like the Macintosh OS and Windows have come to dominate personal computer desktops. However, character-oriented user interfaces are still popular among technicians and system administrators who operate and maintain our networks and back-end computers. They find typing commands faster than using a GUI once they have the commands memorized. It is also possible to write small programs containing several commands to automate common multi-command tasks like creating user accounts.

Personal computers with GUI operating systems like the Macintosh OS and Windows quickly made their way into homes and organizations. Their low cost encouraged the founding of many companies to develop software for the personal computer platform. Some of these, for example, Microsoft, Lotus, Adobe, and Electronic Arts, are major companies today.

Local area networks

At first, personal computers were stand-alone productivity tools. To share data with a colleague, one had to store it on a floppy disk or some other removable media and hand deliver it. Time sharing systems had enabled users to communicate with each other and conveniently share data, but early personal computers did not.

The solution to this problem was hardware and communication software for connecting the computers in an office or a building, forming a local area network or LAN. There were several competing LAN technologies at first, but they were proprietary. Because it was an open standard, Ethernet eventually became dominant.

Once connected to a LAN, users could share common databases and documents as well as hardware like printers. Each user had his or her own computer, and common computers were added to the network. These computers were called servers, since they were programmed to offer a service like printing or database management to the user's computers, the clients.

The LAN became the next important software development platform. Programmers developed client-server applications in which they separated the user interface program, which ran on the user's personal computer, from the application logic and databases, which ran on servers. Programmers also had to develop software for hardware services like sharing files, printers, and modems. As we see below, client-server applications are also prominent on the Internet. When you use the Web, you are running a client program like Internet Explorer or Firefox on your computer (the client) and retrieving information from Web servers.

Sidebar: Internet client options

The client-server model has moved from the LAN to the Internet. Several client options are found on the Internet. The most common is the Web browser. Early browser-based applications retrieved Web pages with hypertext markup language (HTML) tags added to control the formatting of the text and images.[34] Any computer, regardless of its operating system, could use these applications, and very little skill was required to add HTML tags.

However, the pages were largely static text and images. The next generation of Web clients could execute simple programs included in Web pages, making dynamic behavior like cascading menus and buttons that changed appearance when clicked possible. These programs are generally written in the Javascript programming language.

AJAX, asynchronous Javascript and XML, uses Javascript to download content without displaying it while the user looks at a page. Consider for example Google maps. While a user is looking at a portion of a map, adjacent portions are downloaded and cached. When the user scrolls the display, the adjacent portions are displayed immediately.

An applet is a program that is automatically downloaded and executed when the user links to an HTML page that contains it. Applets are useful when a fairly complex client program is needed for an application. They are most commonly written in the Java programming language.

A Web browser can be augmented using a plug-in like the Macromedia Flash player, which enables your Web browser to show animations and movies. Once you install this plug-in, your browser will use it to play Flash movies from any Web site.

34 For example, the tag starts boldface text and the tag ends it.

For complex tasks rich user interfaces and specialized server software and databases, a Web browser may not suffice. In those cases, the user must install a custom client program. That program, not a Web browser, would be used to interact with the server. These applications can be fast and complex, but the user must be able to install them. Examples of websites that require custom clients include Apple's iTunes client, Google Earth and Microsoft's Virtual Earth 3D.

Wide area networks—the Internet

Around the time organizations were rolling out their early LANs, the Internet was beginning to spread within the research and academic communities. The Internet was not the first wide area network (WAN). Earlier research networks had preceded it, and IBM and Digital Equipment Corporation both marketed widely used WAN technology to large organizations. There had also been several commercial WANs, which resembled time-sharing systems with remote users dialing in using personal computers. The Internet was different in two key ways. First, it used public domain communication software developed by researchers while the IBM and DEC networks were proprietary, requiring one to use their hardware and communication software. Second, the Internet was a network of networks, an internet (small i). An organization could connect its LAN to the Internet regardless of networking hardware and software they used internally.

The Internet also differed from the telephone, cable TV and cellular phone networks. Those networks were designed to provide a specific service—telephony or broadcast video, while the Internet was designed to provide low-cost communication between computers connected at the edge of the network. In the Internet model, the users who connect to the network invent and provide applications and services, not the network operator. The Internet was designed from the start to be an end-to-end, "dumb" network that could connect complex devices (computers) at its edges.[35]

This end-to-end architecture was critical to the rapid expansion of the Internet. Any user could invent and implement a new application. For example, when Tim Berners-Lee invented the World Wide Web protocol, he wrote Web client and server programs, and allowed others to copy them. Anyone could use his software to create websites and to retrieve information from other's websites. Contrast this to, say, the telephone network where only the telephone company can decide to offer a new service like caller ID. On the Internet, anyone could be an inventor or an entrepreneur—innovation and investment took place at the edge of the network.

Programmers realized they could use the Internet protocols within their organizations as well as on the Internet. They began developing applications for their intranets—accessible only to authorized employees—and for extranets, which only selected stakeholders like suppliers and customers could access.

Internet based software

The rapid spread of the Internet is well known, and its size and power have made it today's dominant software development platform. We have gone beyond client-server applications, and new forms of software are transforming our individual work, organizations, and society. The Internet has facilitated and given rise to open source software, user contributed content, composite applications, software as a service, mobile, portable and location-aware applications, long tail applications, and applications in support of collaboration.

35 For more on the end-to-end design of the Internet see: Saltzer, J. H., Reed, D.P. and Clark, D.D., End-to-end Arguments in System Design, Second International Conference on Distributed Computing Systems (April 1981) pages 509-512. Published with minor changes in ACM Transactions in Computer Systems 2, 4, November 1984, pages 277-288 and Isenberg, David S. The Dawn of the Stupid Network, Networker 2.1, February/March 1998, pp. 24-31.

Open source software

Programmers typically write programs in a programming language. This is called the source program. The source program is compiled (translated) into a machine-language object program that the user's computer can execute. Commercial software companies generally distribute only the object program along with a license to use it on a single computer. The source program is a trade secret.

With open source software, source and object code are publicly available at no cost, and programmers making modifications or improvements often feed them back to the community.

While sharing of open source software dates back to the batch processing era, the practice has thrived on the Internet platform.[36] The culture of the academic and research community which developed the Internet was conducive to sharing. The Internet also eliminated the cost of software distribution and enabled programmers to form a community around an open source project and communicate with each other.

While the Internet is now open to commercial activity (it was not at first), open source software is still widely used. Users may not realize it, but many websites, for example Google or Yahoo, run on open source software. The Linux operating system is perhaps the most significant open source program. Linux has undergone continuous improvement since its launch, and is now found on everything from large mainframe computers to cell phones.

In 1991, while he was a student in Finland, Linus Torvalds posted the first version of Linux on the Internet. The email announcing its availability ends with the line: "Any suggestions are welcome, but I won't promise I'll implement them :-)." In keeping with the last sentence of his announcement, Torvalds has retained editorial control of Linux. This is typical of open source projects. One person is often in control, with a relatively small group of active developers contributing significant upgrades and extensions. A greater number find and correct errors. Some people do this as part of their work, others as a hobby.

Perhaps the greatest contribution of the open source community is not its software, but the culture and organizational openness it engenders. Open source projects and management techniques have spurred the need for applications that support collaboration and have facilitated the creation of distributed organizations.

Internet resource:

Perspectives on Free and Open Source Software, edited by Joseph Feller, Brian Fitzgerald, Scott A. Hissam and Karim R. Lakhani, is an excellent anthology on open source software. There are 24 chapters from a well-known group of authors on a wide variety of topics. You can download the book from http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=11216&mode=toc or purchase a printed copy.

Many open source projects are coordinated through online repositories like Sourceforge. Sourceforge hosts over 140,000 open source projects. Project pages allow for downloads of source and object programs, documentation, bug reports, feature requests, and many facilities for enhancing communication among the programmers.

User supplied content

With the moves form batch processing to time sharing to personal computers, users did more and more of their own data entry. The Internet extends this capability significantly. Users typically complete transactions without involving vendors. We pay income tax, renew automobile licenses, purchase goods for our homes and businesses,

36 SHARE, a users group for IBM computers was formed to share software in 1955. Software was distributed in card decks or on 9-inch tape reels.

and contract for business and consumer services online. This is often more convenient and economical for the user, and nearly eliminates the transaction cost. Organizations are also able to aggregate and mine transaction data, looking for patterns and preferences. This is common in high-volume retail applications like the Netflix movie rental service or online book sales. The retailer recommends movies or books based upon prior purchases of the user and others.

Going beyond transaction processing, the Internet enables applications that derive value from user supplied data. Take, for example, the case of Amazon.com, the first online bookstore. When Barnes and Noble, a large bookstore chain, established an Internet site, many predicted they would quickly displace the upstart Amazon since they were selling essentially the same books. Amazon prevailed because they encouraged users to write reviews of the books they purchased. Both companies sold the same books, but the customer reviews differentiated Amazon.

Many of today's Internet applications center on user supplied content. websites allow users to publish their own music, videos, and photographs, and others to find and retrieve them. Users supply descriptions of everything for sale in online marketplaces like Ebay and Craigslist. Millions of users contribute to blogs (web logs) on the Internet, and many organizations use them to communicate internally and with outside stakeholders like customers, suppliers and shareholders. The book you are now reading is a wiki, created by inviting users to draft and subsequently revise and improve chapters. The online encyclopedia Wikipedia is perhaps the best known example of an application based solely on user content. Organizations often use wikis for applications like internal knowledge sharing, project planning and documentation. Audio and video podcasts, recorded material that is delivered automatically to subscribers, are also common on the Internet and within organizations.

We might ask what motivates people to contribute explicit content without pay. Why take the time to improve a Wikipedia article or submit a bug fix to an open source program? Why contribute to the book you are now reading?

Motivation includes:

- creating or writing something, perhaps a program or reference source, for one's own use
- having the pleasure of knowing others with shared interest may find it useful
- having fun creating it
- enhancing one's reputation

As Yochai Benlker says: "We act for material gain, but also for psychological well-being and gratification, and for social connectedness."[37] These motivations in conjunction with the fact that "the declining price of computation, communication, and storage have, as a practical matter, placed the material means of information and cultural production in the hands of a significant fraction of the world's population-on" lead to significant, non-market information production.

How many millions of hours did Internet users contribute yesterday? What is the economic value of those hours — how does the Gross Contributed Product compare to the Gross National Product? What will it be in ten years? Might this non-market economy one day rival the market economy in importance? (Stay-at-home mothers and grandmothers might ask the same question). If so, what are the implications for organizations or management or the entertainment industry?

37 Benkler, Yochai, The Wealth of Networks, www.benkler.org.

Composite applications

Early websites were pure client-server applications—a user retrieved information from a single server. Soon, applications began to emerge that, without the user knowing it, combined information from multiple servers. For example, when one purchased a book from Amazon, it was shipped via United Parcel Service (UPS), but one could track its shipment from the Amazon website. Behind the scenes, the Amazon server queried the UPS server to get the current shipping information then relayed it back to the user in an "Amazon" web page. This required agreement and coordination between programmers at Amazon and UPS. The UPS programmers had to give the Amazon programmers technical details on how to access information from the UPS server—they had to define the UPS application program interface or API.

But, Amazon was not the only UPS customer who would like to provide shipment tracking information, so UPS and others began publishing their APIs so others could access their services. Today, companies commonly publish APIs and encourage others to create composite applications, often called mashups, incorporating data or computation from their servers.[38]

Exhibit 32 illustrates a mashup. The application at Housingmaps.com is used to rent and sell apartments and houses. In this case, we see listings in the USD 1500-2000 range in the western portion of Los Angeles. The data were retrieved from Craigslist.com, an online application listing items for rent and sale, and displayed on a map from Google.com. It is noteworthy that the map on the left side is not a static picture. It offers the full function of Google's mapping service—scrolling, zooming, panning and alternating street maps (shown here) and satellite photos. The links on the right hand side retrieve detailed descriptions of the rental offerings from Craigslist.com.

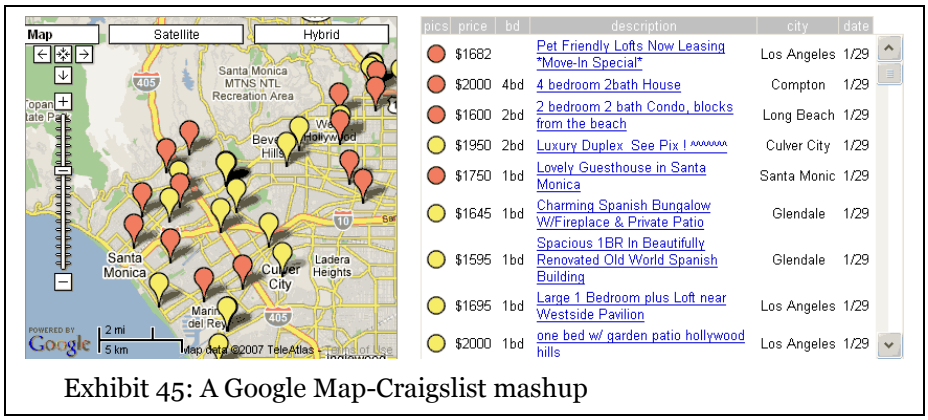


Exhibit 45: A Google Map-Craigslist mashup

While Google and Craigslist currently offer access to their applications on the open Internet at no cost, others charge for such services. For example, Amazon operates an extensive online store, and they offer access to the technology that powers that store to others. Organizations as varied as Target Stores, the National Basketball Association, and Bohemica.com's bookstore on Czech culture all use Amazon.com Web services.

Amazon, Google and Craigslist have extensive datacenters operating around the clock. They have made very large investments in programming and building expertise. The cost of expanding and maintaining Google's global map database is very high, and Craigslist has a vast store of user contributed data. But, since their APIs are simple, it is relatively easy for a programmer to incorporate these valuable assets into a new application. Applications with open APIs are becoming part of our information infrastructure, part of today's software development and delivery platform.

38 One directory, http://www.programmableweb.com, listed 2,300 such mashups in January 2007.

Sidebar: Using a Web service to add audio to an application

I maintain a blog as a supplement to a course I teach. The blog is at http://cis471.blogspot.com/, and, if you visit it, you will see a link at the bottom of each post that reads listen to this article. In the right-hand column, you will also notice a link reading Audio RSS Feed.

If you follow the first link, you will hear a slightly stilted sounding woman reading the text of the article. The program that does this text to speech conversion is quite complex and requires a fast computer, but that complexity is hidden. A server at Talkr.com does the conversion and the link to it required adding only one line of HTML code to the blog template:

```
<a href='http://www.talkr.com/app/fetch.app?feed_id=25222&perma_
link=<$BlogItemPermalinkURL$>'>Listen to this article </a>.
```

Adding the Audio RSS feed converts the blog to a podcast. A user who subscribes to the RSS feed automatically receives audio recordings of articles when they are posted. Again, adding this complex feature required only a single line of HTML code in the blog template:

```
<a href="http://www.talkr.com/app/cast_pods.app?feed_id=25222"> Audio
RSS Feed</a>.
```

Adding these audio features brought the blog into compliance with university regulations on accessibility by blind people. Doing so took only a few minutes because of the simple API used by the service at Talkr.com.

Software as a service

Since the earliest days of computing, users have had the choice of owning and operating their own computers or running their applications at a remote service bureau. Whether one used the service by submitting a deck of cards or by typing information at a time-sharing terminal, the service bureau owned and operated the equipment.[39]

As computer costs fell and millions of computing professionals were trained, in-house computing grew much more rapidly than the service bureau business. Today, nearly every organization operates its own IT department and either purchases software or develops it themselves; however, a trend back to software as a service may be emerging. As network speed and reliability increase, the case for hosted applications improves.

Advantages to running software as a service include:

- The application can be run from anywhere, including by mobile workers.
- There are savings in in-house IT infrastructure and personnel.
- The service provider has specialized expertise and skilled employees.
- The software vendor gets rapid feedback from users and can make changes.
- The software can be continuously upgraded and debugged.
- Upgrades do not have to be installed on client machines.
- Initial investment by the user is minimized.

On the other hand, there may be problems with software as a service:

39 Running jobs on remote computers was one of the key goals of the funding for ARPANet, the network that preceded the Internet. ARPA wanted the researchers they funded to be able to run jobs on each other's computers. (See this historic paper).

- Intruders may be able to access the application.
- The service might not be as flexible and well-tailored as software written in house.
- The service vendors may go out of business.
- The service vendor may alter the service, pricing, etc.
- Data may be insecure at the vendor's site, particularly if it is out of the country.
- Exporting your data may be difficult or impossible, locking you into the service.

It is clear that the decision as to whether to use a software service or to run your own software internally involves both business and technical considerations. Even if a service does everything you want, and it is cost effective, it will be a bad choice if the company servers are frequently down or the company goes out of business. Business contracts and service level agreements are as important as technical specifications.

Internet resources:

SalesForce.com was an early, successful vendor of software as a service. Company founder Marc Benioff expects hosted software to largely replace user-owned and operated software as networks spread and improve. Benioff feels hosted software levels the playing field between large and small organizations by enabling them all to use the same cheap, reliable programs. He also encourages outside programmers to use Salesforce.com tools to develop their own hosted applications. He describes his vision in this conference presentation.

The University of Arizona has turned to Google for electronic mail, and plans to use other Google services. ASU IT director Adrian Sannier explains why he made that choice in this blog entry. Sannier feels that consumer applications, not military or business, are now driving innovation.

Mobile, portable and location-aware applications

The Internet platform combined with wireless technology enables a new class of mobile, portable and location-based applications. In many parts of the world, it is now possible to connect to the Internet while outside the home or office. The connection may be fully mobile, for example, while walking or moving in a car, or portable, for example, using a laptop computer in an airport.

Today, we make these connections using the cellular telephone network or a WiFi access point.[40] The cellular networks are large, generally national enterprises, while WiFi networks are decentralized. They may be as small as a single access point in a café or hotel lobby or as large as a municipal network covering a city. WiMAX, a third wireless technology for Internet access, is partially standardized and may become important in the future.

Both WiFi and cellular networks are growing rapidly. Third generation cellular networks, which are fast enough for many Internet applications, are being deployed in many cities and developed nations. WiFi networks are being deployed by large and small entrepreneurs and, increasingly, by cities.[41] Unless heavily congested, Wifi access is generally faster than third generation cellular access. Some devices can connect to either cellular or WiFi networks, automatically choosing WiFi if it is available.[42]

40 WiFi technology was initially developed for LANs inside home and offices, but it has been, somewhat unexpectedly, used in wide area networks as well.
41 See http://www.muniwireless.com/ for coverage of municipal networks and their applications.
42 See, for example, http://www.theonlyphoneyouneed.com].

Laptop computers are generally used for portable access, but they are too large and consume too much power for mobile applications. There are competing designs or form factors for mobile access. Some devices are ultra-mobile PCs like the Microsoft Oragami shown below.[43] Others are fairly standard looking cell phones.

Exhibit 46: Microsoft's Oragami

Between these extremes, we see devices like the Apple iPhone, shown below. These smartphones combine telephony, Internet access, music and video play, and contact and address books in one device. The marketplace will determine the most popular form factor.

Exhibit 47: Apple iPhone

In addition to communication functions like email and instant messaging, mobile workers are able to access the Web and corporate databases. For example, a sales person might book an order or check the production or delivery status of an open order while at a customer's location or real estate appraiser or agent might send a photograph or video from a listed property to a central database. An October 2006 survey found that 64 per cent of mobile workers use their smartphones to access enterprise applications and business data.[44]

Mobile connectivity also enables location-aware applications. One may determine their exact location in several ways. If connected to the Internet, various databases and services are available. For example, Mapbuilder converts street addresses to geocodes (latitude and longitude) and displays the location on a Google map. The global positioning system (GPS) may also be used. A GPS receiver uses the signal transmission time to orbiting satellites

43 http://www.microsoft.com/windows/products/winfamily/umpc/default.mspx.
44 Malykhina, Elena, Leave the laptop at home, Information Week, October 30 2006, page 47-9, http://www.nxtbook.com/nxtbooks/cmp/infoweek103006/index.php?startpage=51.

to determine location. Where satellite signals are blocked, say by tall buildings, other techniques like estimating the distance to cell towers or TV broadcast antenna may be used.[45]

Emergency service (wireless 911) is a key location-aware application. It led the United States Federal Communication Commission to mandate location-awareness for cell phones sold in the US. While not yet fully implemented, the increased demand has driven cost down dramatically.[46] While emergency service is driving the market, we can anticipate applications like directing ads to people in specific locations, location-specific search, taxi and delivery fleet scheduling, and automatic geocoding of photos when they are taken.

The long tail

In a conventional store, stocking a new item for sale is costly. Inventory must be acquired, shelf space allocated and the merchandise displayed. The cost of adding an item for sale at an online store is essentially zero. Low stocking cost leads to the concept of the "long tail".

The long tail was documented in a 2003 study of Amazon.com. Eric Brynjolfsson and his colleagues estimated the proportion of Amazon book sales from obscure titles.[47] As we see in Exhibit 17, 47.9 per cent of Amazon's sales were of titles ranked greater than 40,000th in sales. This leads a large online store like Amazon to carry approximately 3 million books compared to 40-100,000 for a large brick and mortar store. The result is increased profit for the vendor, increased choice for the customer, and increased incentive for authors with niche products.

Exhibit 48.: The long tail of Amazon book sales

Sales rank	Per cent of sales
>40,000	47.9%
>100,000	39.2%
>250,000	29.3%

The long tail is not limited to books. It is even more important in the case of information goods, where it is not necessary to deliver a physical object like a book. Consider, for example, the online music market. Both Apple iTunes and eMusic, which specializes in music from independent labels, offer over 2 million songs in their online catalogs.[48] Google advertising is another illustration of the long tail. A large per cent of their ad revenue comes from advertising by millions of small, often local companies paying just a few cents per click (AdWords) or millions of websites, which display Google ads (AdSense). Vendors of software as a service also target the long tail—small organizations wishing to use packages previously affordable only by larger organizations.

Internet resources

Chris Anderson has extended and popularized the notion of the long tail, see:

- Anderson's initial long tail article
- Anderson's long tail talk

45 Receiving GPS signals in doors has been difficult in the past, but the latest GPS chips can detect signals with power of only 0.00000000000000000001 watts, enabling them to be used indoors. (http://www.infoworld.com/article/07/01/31/HNubloxgps_1.html).

46 The cost is under USD 1 in some circumstances. See http://www.eetimes.com/news/semi/showArticle.jhtml?articleID=196900828.

47 Brynjolfsson, Erik, Hu, Yu Jeffrey and Smith, Michael D., "Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers" (June 2003). MIT Sloan Working Paper No. 4305-03 Available at SSRN: http://ssrn.com/abstract=400940 or DOI: 10.2139/ssrn.400940.

48 http://www.emusic.com/about/pr/pr20061213.html.

- The Long Tail blog
- Anderson's *The Long Tail* book
- Interview on *The Long Tail* and the way he wrote his book
- Review of the book

Collaboration support applications

The visionaries, who imagined, funded and worked on the research that lead to today's platforms and applications, understood that networked computers would allow for collaborative communities of common interest. For example, in 1968, J. C. R. Licklider, who was instrumental in conceiving of and funding time-sharing, personal computers and network research wrote:

What will on-line interactive communities be like? In most fields they will consist of geographically separated members, sometimes grouped in small clusters and sometimes working individually. They will be communities not of common location, but of common interest.[49] (Emphasis in the original).

The early time sharing systems allowed people to send messages and share files, but there was little if any software for explicit support of collaboration. Perhaps the first major application of collaboration support software was in decision support rooms where users sat at computers connected to a local area network and shared a common, projected screen. These group decision support systems (GDSS) are used primarily for group decision making and consensus building. They are equipped with software for brainstorming, outlining, voting, etc.

A GDSS room supported collaboration among people who were working at the same time (synchronously) and the same location. With the advent of the Internet as a platform, software was developed to support people working synchronously at different places, for example, software for chat, video conferencing, instant messaging, voice telephony, sharing the same screen views and editing documents. Today, distant colleagues can open a chat window, wiki document, telephone session, and shared workspace and work together for hours.

Virtual worlds like Second Life are also used for synchronous collaboration.[50] The following shows a panel discussion in a virtual classroom[51] belonging to the Harvard University Berkman Center.[52] Each panelist and attendee designs an avatar, which they control and move around in the virtual space. We see the panelists' avatars seated at the front of the room and those of the attendees are seated around them. There are also shared spaces like the "flipchart" to the right of the panel. Organizations are experimenting with meetings, press conferences, concerts, and other synchronous events in Second Life.

49 J.C.R. Licklider and Robert W. Taylor, The Computer as a Communication Device, Science and Technology, April 1968, http://scpd.stanford.edu/sonnet/player.aspx?GUID=6F704E8F-352F-42E0-BC58-3F9014EA1F81.
50 http://www.secondlife.com.
51 http://www.flickr.com/photos/pathfinderlinden/174071377/.
52 http://www.vedrashko.com/advertising/2006/06/panel-on-marketing-to-virtual-avatars.html.

Exhibit 49: A virtual classroom in Second Life

We also have software that supports collaboration among people working at different times (asynchronously) and at different places. Examples here include email, voice mail, listservers, online questionnaires, threaded discussions or forums, blogs, and wikis. There are a also a number of network-based suites of standard productivity tools—software for word processing, spreadsheets, database and presentation. These have all of the advantages and disadvantages of other software as a service, but the documents they create are available online for sharing and co-authoring. In some cases, these suites may replace conventional desktop productivity tools; in others, they will complement them.

Collaboration support software has facilitated a trend toward increasing collaboration and distributed organizations, for example:

- business extranets allowing employees from multiple companies to work together (outsourcing)
- business intranets allowing employees from different locations within a firm to work together
- ad hoc organizations set up for a particular event or emergency like the Tour de France or a hurricane
- ad hoc organizations (perhaps within a company) set up to do a short-term particular project
- open source software communities
- cooperative industry organizations like the WiFi Alliance
- loosely-coupled interest group like those who have purchased a particular product or are thinking of doing so

It has also reduced the cost of employees telecommuting—working from home full or part time. Network-based software also facilitates the outsourcing of programming, graphic design, accounting, x-ray interpretation, and other information intensive work. Telecommuting and outsourcing require skilled management, but can result in significant savings for organizations and benefit to employees and society if done well.

Internet resources:

For a website devoted to GDSS, see http://www.dssresources.com/.

A brief discussion of the applications of group decision support is found at http://www.groupsystems.com/resources/custom/PDFs/Gartner-web_conferencing_amplifies_d_138101.pdf.

For a brief history of GDSS, see http://dssresources.com/history/dsshistory.html.

Will the progress continue?

The exponential improvement we have seen in hardware has enabled the invention of new software and applications. If this improvement and innovation continues, it will have profound effects on individuals, organizations and society. But, will the progress continue?

At some point, physical limits cause exponential hardware improvement to level off. Today, semiconductor companies are mass producing chips with a feature size of 65 nanometers (billionths of a meter), and that is expected to drop to 22 nanometers by 2011.[53] Halving feature size every two years after that would imply transistors the size of a single atom around 2033. Clearly, the current technology will reach physical limits before that time. Does that mean that exponential improvement in electronic technology will end? Not necessarily, since we may shift to a different technology.

The logic circuits of the earliest programmable computers were constructed using electromechanical relays. Those gave way to vacuum tubes, which gave way to discrete transistors then integrated circuits. As we approach the limits of current electronic technology, we may see a shift to something else, perhaps three-dimensional fabrication or growing of organic logic devices. We have seen similar progression in storage and communication technology. During the computer era, we have used punched cards and paper tape, magnetic tape, drums and disks, optical disks and electronic storage. Communication technologies have also shifted from the time of the direct-current telegraph to today's multi-wavelength optical links.

As we reach the end of the exponential growth period of a given technology, entrepreneurs and inventors have increased incentive to find replacements. But, what if we hit a double dead end? What if we reach the limits of IT technology and fail to find a replacement?

That is a possibility, but many argue that it is unlikely. Consider, for example, inventor and futurist Ray Kurzweil[54] who predicts continued exponential improvement in information and other technologies far into the future.[55] He notes that new information technologies like writing, mathematical notation, printing, computers, and computer networks accelerate learning and the accumulation of knowledge.

The global distribution of research also gives us reason to be optimistic. Much of the largesse we are reaping today is rooted in research. Modern personal computers, computer networks, and applications were conceived of and prototyped decades ago. For example, Ivan Sutherland, shown here, used a room sized "personal computer" to develop very early image processing software. There have been many hardware and software refinements in the ensuing years, but a modern personal computer running a graphic design program is an obvious descendent of Sutherland's Sketchpad program.

53 http://www.networkworld.com/news/2006/121306-amd-seeks-efficiency-in-making.html?page=1.
54 http://www.kurzweilai.net/.
55 Kurzweil, Ray, Law of Accelerating Returns, Lifeboat Foundation special report, 2001. http://lifeboat.com/ex/law.of.accelerating.returns.

Exhibit 50: Ivan Sutherland operating his Sketchpad graphic design program in 1963

Much of the research underlying today's hardware and software took place in the United States. Today, important research is taking place in many nations. As technology reaches developing nations with unique problems and ways of seeing the world, we will see even more innovation—they will think outside our boxes.

Internet resource:

Listen to Ray Kurzweil's presentation of his view of exponential improvement in technology and knowledge in a talk at the Technology Entertainment and Design Conference.

Bumps in the information technology road

While we cannot know the future with certainty, it seems safe to say that information technology will continue improving at an accelerating rate for some time. But, even if that is the case, there are important barriers to its adoption and application. Is spite of technical progress, we are facing insufficient communication and data center capacity to handle the next wave of video traffic. If we add the capacity, we will also need large amounts of electrical power. Equitable access to information technology will also require policy innovation. Organizations with vested interests in the status quo and intellectual property assets may try to slow the adoption of technology. Privacy and security concerns are also impediments.

Capacity to handle video traffic

Video traffic is growing rapidly. Movies, television shows, news and documentaries, and amateur video are appearing on the Internet, and organizations are using video conferencing and podcasts to communicate and disseminate information internally and externally. One can imagine that rather than sponsoring network television programs, organizations will begin producing their own programs and distributing them over the Internet. But, video files are large. A single episode of an hour long TV drama recorded in the small screen iPod format is over 200 megabytes.[56]

Video traffic will require innovation and large investments even in developed nations. Telephone and cable companies favor separating and charging for video traffic, in order to justify capital investment. Others have proposed technical solutions. One possibility is peer-to-peer networking, which automatically uses excess capacity on user's computers to store and distribute files, appears to have great potential[57]. However, wide spread peer-to-peer networking would also require capital investment in "last mile" connections between Internet service providers and users' homes and businesses.

56 Cringley, Robert, Peering into the future, The Pulpit, March 2 2006, http://www.pbs.org/cringely/pulpit/2006/pulpit_20060302_000886.html.

57 Norton, William B., Video Internet: The Next Wave of Massive Disruption to the U.S. Peering Ecosystem, September 29, 2006, http://www.pbs.org/cringely/pulpit/media/InternetVideo0.91.pdf.

Data center capacity and electric power

As technology changes, the costs and benefits of centralization of resources change. In the batch processing and time sharing eras, computer operations were centralized within organizations. The personal computer enabled significant decentralization. Organization continued operating large mainframe computers, but the personal computers on our desks and in our homes outpaced them. In the early days of the Internet, organizations and even users to operated their own servers. While many still do, economies of scale are leading to re-centralization, and servers are increasingly located outside the organization. Every major city has large buildings loaded with communication and computing equipment. The One Wilshire building,[58] shown below, looks like any downtown Los Angeles office building from the outside, but inside it is packed with racks of servers and communication equipment.

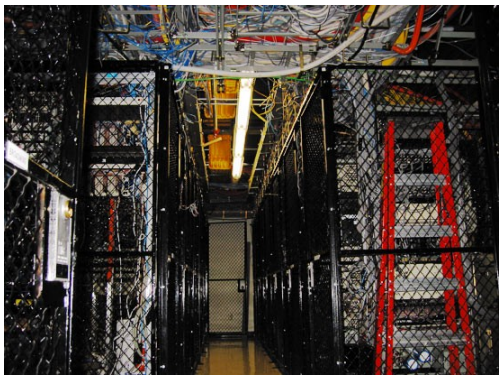


Exhibit 51: One Wilshire, a Los Angeles data center

Data center capacity is in great demand. Keeping up with video and other Internet activity will require rapid expansion and improved density. The economies of scale are such that datacenter ownership will become highly concentrated, which may lead to non-competitive markets, exposing users to non-competitive prices and service.

Data centers also require great amounts of electric power for equipment operation and air conditioning. We see large operators like Google and Microsoft building data centers near sources of low-cost power like The Dalles, Oregon.[59]

Inequity

The hardware and software advances we have spoken of are only widely available in developed nations, and even within those nations, there is substantial inequality. Information technology requires money and education, which are scarce in many parts of the world. The following map, showing values of the International Telecommunication Union Digital Opportunity Index for each nation, illustrates this "digital divide".[60]

58 For a photo tour of One Wilshire, see http://www.clui.org/clui_4_1/pro_pro/exhibits/onewilshire.html#.
59 Gilder, George, The Information Factories, Wired, October 2006, http://www.wired.com/wired/archive/14.10/cloudware.html.
60 World Information Society Report 2006, International Telecommunication Union, http://www.itu.int/osg/spu/publications/worldinformationsociety/2006/report.html.

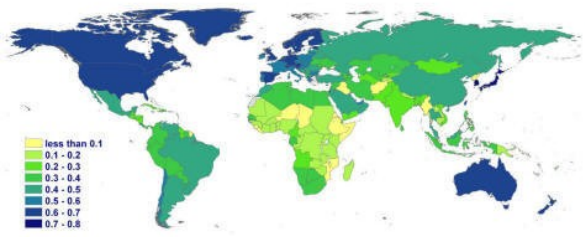


Exhibit 52: The digital divide

Since telecommunication infrastructure is economically and socially important, nearly every nation has a telecommunication policy. Some nations, like China, Singapore, Korea, Japan, and Chile enact policies and make investments designed to provide ample capacity, others take a more laissez-faire approach, counting on the efficacy of markets by encouraging privatization, independent regulation and competition. [61] There is growing evidence that this laissez-faire policy has reached its limits and is failing to generate the investment needed to keep up with forthcoming software and applications.[62] If that is the case, we need policy innovation as well as technical progress.

Vested interests can impede progress

Every nation enacts laws and policies regarding communication and information technology. The vested interests of telephone, cable TV, and media companies and government agencies influence this policy. Political considerations are often more important than technological considerations.

For example, repressive governments face a "dictator's dilemma" in that free and open communication is important for the economy, but can undermine a regime, and, to some extent, all governments curb open communication. Many nations take steps to filter information coming in and out and restrict access to computers and the Internet. This may have to do with maintaining political control or with enforcing cultural norms, for example, against pornography. The Open Net Initiative monitors government Internet filtering, as shown in the following map.

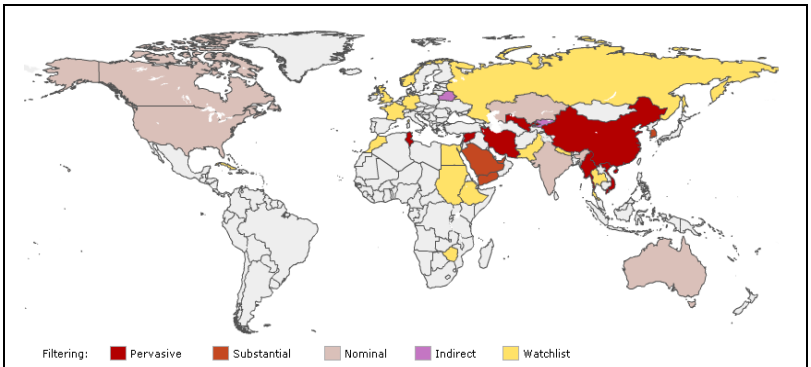


Exhibit 53: Government Internet filtering levels

Incumbent telephone companies and telecommunication ministries may fear a loss of revenue. For example, telephone companies and ministries have traditionally received significant income for completing international calls, but calls between two Internet-connected computers are free, and international calls routed from the Internet

61 Consider, for example, Chile's Digital Agenda, http://www.agendadigital.cl/aws00/servlet/aawscolver?2,inicio,,1,0.

62 See, for example, Press, Larry and Dumans, Marie-Elise, The Digital Divide Report: ICT Diffusion Index 2005, United Nations Conference on Trade and Development (UNCTAD), 82 pages, Geneva, July, 2006. (summary, full document).

to a standard telephone cost only a few cents a minute. Those who stand to lose from such improvements often lobby against new competition.

For example, in 1996, the United States passed a Telecommunication Act designed to increase competition. William Kennard, chairman of the United States Federal Communication from 1967-2001 was charged with implementing the act. Near the end of his term, he expressed his frustration that "all too often companies work to change the regulations, instead of working to change the market," and spoke of "regulatory capitalism" in which "companies invest in lawyers, lobbyists and politicians, instead of plant, people and customer service" (Kennard, 2000). He went on to remark that regulation is "too often used as a shield, to protect the status quo from new competition - often in the form of smaller, hungrier competitors — and too infrequently as a sword — to cut a pathway for new competitors to compete by creating new networks and services."[63]

Incumbent wireless communication companies—radio and television stations and cellular operators—lobby to keep their licenses to broadcast at specified frequencies in a geographic area. These spectrum licenses are valuable assets for their owners, but the rapid growth of WiFi would not have occurred if it did not use license-free radio frequencies. The current licensing regime is based on old technology. Modern radios are able to switch transmission modes to avoid interference, but to take full advantage of those advances, we will have to overcome vested interests and open more license-free spectrum.

Internet resource:

Bill Moyers' Public Broadcasting System documentary The Net at Risk illustrates the conservative impact of organizations with vested interest in the United States.

Intellectual property restrictions

Considerations of intellectual property may impede information technology application. Publishers fear illegal distribution of copyrighted material on the Internet. They use techniques like encryption to add digital rights management (DRM) to their material. So, for example, most songs purchased at Apple's iTunes store may only be played on Apple devices and movies on DVDs cannot be copied into a standard video files on disk drives. Illicit programmers attempt to circumvent DRM techniques, and indeed, copyrighted material is widely available on the Internet. The threat of piracy tends to keep audio, video and print material locked away.

Open media advocates say DRM harms honest people who wish to make legitimate backup copies of material or to play it on more than one device—for example watching a movie on either a computer or a TV screen. They argue that lobbyists for powerful media companies, not consideration of the public interest, often determine copyright law. The United States exemplifies this. In 1790, Congress established a copyright term of 14 years. The term has been frequently extended (11 times in the last 40 years), and is now the life of the creator plus 70 years. Media companies contend that without strong copyright, they have no incentive to produce content.

There has also been copyright innovation. Richard Stallman and other early open source programmers discovered that their public domain programs were incorporated in proprietary products. In response, Stallman invented the Gnu General Public License (GPL).[64] Under the GPL, an author retains copyright, and others are licensed to use the work as long as the derivative work also has a GPL license. Creative Commons (CC) licensing

63 Kennard, W. (2000), "Internet Telephony: America Is Waiting", Voice Over Net Conference, September 12, 2000, http://www.fcc.gov/Speeches/Kennard/2000/spwek019.html.
64 http://www.fsf.org/.

offers other alternatives.[65] For example, one CC variation allows others to copy, display or perform, but not modify, a work as long as they credit the author and attach the same license to any derivative work.

There has also been marketplace innovation. For example, we saw that http://www.emusic.com has a library of over two million songs, all of which are sold without DRM. Apple's Steve Jobs reports that only 3 per cent of the songs on a typical high capacity iPod music player use DRM, the rest are unprotected,[66] and suggests doing away with DRM for music. (Copies may still be watermarked, making each slightly different so they can be traced to their source). Many print publishers, for example MIT Press Journals and the University of Michigan Press Digital Culture imprint, charge for printed copies, but offer downloaded versions without DRM.

Internet resource

Podcasting pioneer Doug Kaye spoke about the value of free information in a keynote presentation at the Portable Media Expo. Kaye feels you maximize the value of content by making it free and having no impediments (like registration or charging) to accessing it. You can listen to this 3m 19s excerpt or the entire presentation.

Listen to Stanford Law Professor Larry Lessig's presentation on the history of US Copyright and Creative Commons. You can also read the transcript.

Security and privacy

As organizations and individuals interconnect and become dependent upon information systems, security and privacy threats arise. We are all too familiar with criminal schemes to exploit these interconnected systems. Thieves have stolen files with credit card and other information on countless individuals. Millions of computers are innocently running "zombie" software, which can be used to mass mail spam (unsolicited email) or flood a target server with fake information requests in denial of service attacks. Criminals impersonate banks or other organizations in phishing schemes designed to trick users into logging onto their websites and divulging sensitive information. Governments also wage "information warfare" in trying to access and compromise other government and military systems.

Efforts to protect the privacy of honest individuals may be in conflict with efforts of governments and law enforcement agencies trying to gather intelligence on terrorists or criminals. For example, Skype Internet telephony software encrypts conversations. This protects the privacy of honest citizens discussing sensitive family or business matters, but it also protects criminals. Government agencies now have the technical capability to track purchases, Internet searches, website visits, etc. of individuals or large groups of people.

These concerns impede the acceptance of information technology and its applications. They also raise the cost and reduce the efficiency of those that are implemented.

Summary

We have traced the exponential improvement in electronic, storage and communication technology during the last half century. That change has enabled qualitative shifts in the dominant software development and delivery platform, beginning with batch processing, and extended by time sharing, local area networks, personal computers and now computer networks.

65 http://creativecommons.org/.
66 Jobs, Steve, Thoughts on Music, February 6, 2007, http://www.apple.com/hotnews/thoughtsonmusic/.

Internet based software is now becoming dominant. Moving to the network platform facilitates open source software, applications based on user-supplied content, composite applications, software as a service, mobile, portable and location aware applications, long-tail applications, and applications in support of collaboration.

It seems likely that this progress will continue as existing technologies are refined and new ones invented. Still, we can see barriers, which will slow the adoption of new applications. Video applications will require enormous expansion of our current communication and data center capacity and vast amounts of electric power. Inequity—a digital divide—within and among nations will limit the access to the benefits of our progress. Conservative vested interests work to preserve the status quo and we also face barriers in intellectual property restrictions, security and privacy.

Exercises

1. Exhibit 33 illustrates exponential growth in electronic technology, as measured by transistors on a chip. Research improvements in storage technology measured in cost per bit, and plot a similar graph.
2. Does your nation have a technology policy? Briefly describe it, providing links to relevant documents on the Web.
3. Does the government, private enterprise or a combination of the two operate telephone service? Cellular telephony service? Internet service? Is there competition in these three areas or is there only a monopoly or oligopoly?
4. How does Internet service in your nation compare with others? For example, what is the cost of home DSL service compared to other nations? How does the speed compare? What per cent of average monthly household income is an Internet account in your nation? Is high speed Internet service available in every part of your nation?
5. How long would it take to download a 200 megabyte copy of a one-hour television show using a 1 megabit per second DSL link? How long would it take to transmit it across a 10 gigabit per second backbone link? Estimate the number of copies of a popular TV show that might be distributed to homes and the time that would take using DSL.
6. Telephone and cable TV companies differentiate between telephony, video broadcast and Internet, charging separately for each service. Water companies could also decide to enter the service business, differentiating and charging separately for drinking water, garden water, etc. That sounds kind of goofy doesn't it? Design a spoof water bill that makes fun of the idea. What would your water billl look like if water service were delivered by your telephone company?
7. Create a blog using blogger.com or another online service and post something of interest to this course each week.
8. Select a Wikipedia article on a topic with which you are familiar and interested, and make a substantive improvement to it.
9. Find an online forum on a topic with which you are familiar and start a new thread.
10. Post a review of a book or other item that you already own on Amazon.com.

11. Netflix, an online video rental service, offers 70,000 DVD titles. Estimate the total number of bits in 70.000 DVDs. Assuming the average movie is 4 gigabytes, how many high-capacity PC disk drives would be required to hold all of their titles? Estimate the number of years it will be before the entire Netflix library will fit on a single disk drive.
12. Contact the IT department at your school or an outside organization and find out if they are still running any batch processing applications.
13. A GPS receiver can detect a signal with only 0.00000000000000000001 watts of power. How does that compare with the power consumed by a light bulb in your home? How does it compare to the power of the signal from a WiFi radio?
14. It is interesting that, at this time, we are witnessing a fundamental shift in CPU chip architecture. As the transistor density continues to increase, designers are now placing multiple CPUs on single chips. This shift to parallel processing is promising, but raises yet unsolved programming and inter-processor communication problems.

Chapter editor

Larry Press, Professor of Information Systems at California State University-Dominguez Hills, has been studying the global diffusion of the Internet, with an emphasis on policy and technology in developing nations, for over a decade. Dr. Press and his colleagues at the Mosaic Group developed an early framework to characterize the state of the Internet in a nation, and they, and others, have used this framework in many national case studies and surveys. This work has been supported by organizations including Rand, The International Telecommunication Union, SAIC, UNDP, UNCTAD, and the US State Department as well as governments in developing nations. Dr. Press was also an organizer and instructor in the World Bank/Internet Society workshops, which trained over 2,500 networking leaders from nearly every developing nation.

Dr. Press has worked in both industry and academia, consulted to a variety of industrial and multilateral organizations, and published numerous articles and reports. He has published two books, edited two book series, and been an editor or contributing editor for several magazines, trade publications and academic journals and periodicals. He has also received awards for excellence in teaching. His MBA and PhD in information processing are from University of California, Los Angeles.

8. Utilizing data for efficiency and effectiveness

Editor: Ron L. Thompson (Wake Forest University, USA)

Reviewer: Geoffrey N. Dick (Australian School of Business, Australia)

Learning objectives:

- describe competing views of how decisions are made within organizations
- describe ways in which data may be used to improve decision making, including:
    - controlling business processes
    - automating decision making
    - supporting complex decisions
    - augmenting knowledge

Introduction[67]

In a previous chapter, you learned how data are typically collected, stored and updated within information systems. In this chapter, we take more of an applications perspective. That is, we examine how individuals and organizations use information systems (and more specifically, computer software applications) to make use of the data that have been collected and stored.

To put this topic into context, we start by examining organizational decision making. Since decision making is at the heart of organizational operations, it is important to understand the link between the use of information systems and efficient and effective decision making.

Organizational decision making

In any organization, decisions must be made every day. Let's consider a grocery store, as an example. Someone needs to decide how many employees need to be on duty, and what tasks they need to perform (who should be working the check-out lines and self-checkout lines, who should be re-stocking shelves, who should be preparing fresh baked goods, etc.). Decisions need to be made about inventory control issues (e.g. what items need to be re-ordered, and how many of each), pricing issues (e.g. what items to place on sale, how much to reduce their prices), and so on. In addition to the normal, daily decisions, there are many others that occur less frequently (e.g. whether or not to hire another employee, and if so, whom), and some that occur quite infrequently (e.g. whether or not to open another store in another location). While decision-making environments vary substantially, they typically have one thing in common. More specifically, ready access to good data, information and (in more complex environments) knowledge may lead to better (more efficient and effective) decision making.

67 Some of the material in this chapter has been adapted from Thompson and Cats-Baril (2003).

There are different views of how decision making does (and should) occur within organizations. We start with a quick overview of the rational perspective, and then briefly discuss alternative views.

Rational view

The rational view of decision-making describes an ideal situation that some people argue is difficult to achieve. In an ideal world, organizational members charged with making a decision will ask all the correct questions, gather all the pertinent information, discuss the situation with all interested parties, and weigh all relevant factors carefully before reaching their decision. This ideal world rarely exists, however. In reality, most decision makers are faced with time pressures, political pressures, inaccurate or insufficient information, and so on. As a result, many decisions that are made might appear irrational when viewed from the outside. Still, it is useful to employ the rational decision making model as a starting point.

Herbert Simon (a Nobel Prize winner) proposed a model of rational decision making near the middle of the 20th century (Simon, 1960). His model includes four stages: (1) intelligence (is there a problem or opportunity?), (2) design (generate alternative solutions), (3) choice (which alternative is best?), and (4) implementation (of the selected alternative). The basic model of how rational decision-making should proceed is:

- **Intelligence phase**—collect data (and information) from internal and external sources, to determine if a problem or opportunity exists. To the extent possible, ensure that the data are accurate, timely, complete, and unambiguous.
- **Design phase**—generate possible alternative solutions. Ensure that as wide a selection of alternatives as possible are considered.
- **Choice phase**—select the best alternative solution. Identify relevant criteria for evaluation, as well as appropriate weighting for each criterion, and use these to objectively weigh each alternative.
- **Implementation**—perform whatever steps are necessary to put the selected alternative into action.

As an example, think of getting dressed in the morning—you gather intelligence, such as the weather forecast (or by looking out the window). You consider issues such as what you are doing that day, and who you are meeting. You think about alternative clothing options, while considering constraints such as what clothes you have that are clean. You might try on various combinations or outfits, and then make a decision.

Rationality assumes that the decision maker processes all information objectively, without any biases. In addition, this view of rational decision making implies that decision makers have a clear and constant purpose, and are consistent in their decisions and actions.

While many people strive for rational decision making, and intellectually acknowledge that rational decision making is a preferred goal, the realities are often far from this ideal. For example, many decision makers do not take the time to collect all relevant data, nor do they use well-known idea-generation techniques to help them generate a wide selection of alternatives.

In the above example (getting dressed in the morning), you may not go through all of the steps; instead, you might just pull on the first clean clothes that you find, and occasionally will find yourself inappropriately dressed (clothing that is too warm, or too informal for a meeting that you forgot about, etc.).

Alternative views

Behavioral theorists argue that the rational view of decision making is too simplistic. Although decision makers may go through the four phases of the rational model, they do not normally do so in any straightforward manner. Many decision makers, especially those with managerial responsibilities, have work that is fragmented, brief and varied. They face constant interruptions (telephone, email, impromptu meetings), and can only spend a short amount of time considering most decision situations. They are often under considerable stress, which can further degrade their ability to follow a rational approach to making all of the decisions that are needed.

As a result, decision makers often take shortcuts to reduce the burden of making decisions. They typically rely on a network of contacts (both inside and outside of their organization) to help them collect data and to come up with alternative solutions to a particular problem or opportunity. The term "satisficing" has been used to describe a common approach, which is to settle for the first alternative that seems "good enough", rather than expending additional time and effort to collect all available data and to consider all possible options.

In addition, human beings have many biases that can influence their decision making. We may be biased by our upbringing and educational background, by the opinions of persons whom we look up to, by our experiences in similar situations in the past, and so on. We also have personality traits (such as dogmatism, or a lack of creativity, or low willingness to accept risk). These biases and personality traits often keep decision makers from considering a full range of alternatives.

Also, it has been noted that decisions are often reached based on "political" motivations, rather than as a result of rational thought and consideration. For example, a purchasing agent might decide to obtain goods from a supplier whose products and services are inferior (lower quality products, higher price, etc.) than those offered by a competitor, because he knows that his boss is a good friend of an important individual who works for the supplier. While rational decision making implies putting organizational goals above departmental or individual ones, we know that this does not always happen.

In addition to situational factors (too many decisions requiring attention, inadequate data available, not enough time to consider options fully, and so on), decision makers are human beings with limitations. We can only keep so much information available in our short-term memory (which makes comparing options more difficult), we are poor at seeing trends in data, and we are slow (and often inaccurate) in accessing information stored in our long-term memory.

The reason for considering these issues, is to acknowledge they exist and then design information systems that can help us overcome individual and situational limitations, to try and help us move closer to having the ability to use a rational decision making process. For example, designing systems that provide summarized data (with access to more detailed data on demand) makes it easier for decision makers to retrieve the information they need, which increases the probability that they will do so (rather than taking shortcuts). Similarly, we can design systems that help decision makers to see trends in data, to compare multiple options simultaneously, and to provide more transparency in decision making (which reduces the probability of decisions being made for political reasons).

As an example, several large cities exist in dangerous areas—near nuclear power stations or in areas that prone to hurricanes, tornadoes, or floods. City administrators have plans to evacuate in cases of emergency—but think of the complications. The time of the emergency (which will affect where people are and what transportation is available), the amount of warning, the public transportation capacity (roads, railroads, airports), the severity of the

emergency, the cost of the disruption—all these need to be considered and possibly traded-off against one another. Ideally, planners will have developed comprehensive plans in advance that will have considered these issues and developed plans to minimize loss of life and economic disruptions. One way to assist these planners is to provide them with robust information systems that help forecast the impact of different possible scenarios, and also to help them weigh trade-offs among competing criteria.

Decision environments (degree of structure)

Obviously, not all situations that require decisions are the same. While some decisions will result in actions that have a substantial impact on the organization and its future, others are much less important and play a relatively minor role.

One criterion that may be used to differentiate among decision situations is the degree of structure that is involved. Many situations are highly structured, with well-defined inputs and outputs. For example, it is relatively easy to determine how much to pay someone if we have the appropriate input data (e.g. how many hours worked and their hourly pay rate), and any relevant decision rules (e.g. if the hours worked for one week are greater than 40, then overtime pay needs to be calculated), and so on. In this type of situation, it is relatively easy to develop information systems which can be used to support (or even automate) the decision.

In contrast, some decision situations are very complex and unstructured, where no specific decision rules can be readily identified. As an example, assume that you have been assigned the following task: "Create the design for a new vehicle that has at least a four-foot long truck bed, is a convertible (with a retractable hard-top roof), gets at least 50 miles per gallon of gasoline, has a high safety rating, and is esthetically pleasing to a relatively wide audience." There is no "optimal" solution to this task; finalizing a design will involve many compromises and trade-offs, and will require considerable knowledge and expertise.

With this brief introduction, we move to a more detailed discussion of the role of information systems in decision making.

Decision making: systems view

A previous chapter introduced the idea of viewing an organization as a system that acquires inputs, processes them, and generates outputs. The organization interacts with its environment, in that it acquires inputs from the environment (e.g. purchasing parts from suppliers), and creates outputs that it hopes will be accepted by the environment (e.g. through sales of products to customers). The organization also receives feedback from the environment, in the form of customer compliments or complaints, etc. This way of perceiving an organization is typically referred to as the systems view, in that the organization is essentially viewed as a system operating within an environment.

As discussed in a previous chapter, it is also possible to break the organization into a series of smaller sub-systems, or business processes. For example, we might view the purchasing function as a system that accepts inputs (e.g. materials requests from the production process), processes them (e.g. reviewing pricing and delivery details for a variety of suppliers), and generates outputs (e.g. purchase orders forwarded to specific suppliers). The purchasing process also receives feedback (details concerning orders received by the inventory control function, etc.).

For many organizations, these business processes (organizational subsystems) are supported by information systems. Before describing how this occurs, we need to define a few terms.

Data, information and knowledge

In a previous chapter, definitions and examples were provided to help differentiate between the terms data, information, and knowledge. As was discussed, the term data is generally used in reference to representation of raw facts. This might include mathematical symbols or text that is used to identify, describe or represent something, such as a temperature or a person. We should also note that this definition of data is considered by some people to be rather narrow; the term is sometimes also used to include images, audio (sound), and video. For the purposes of our discussion in this chapter, we will focus on the more narrow definition of data.

Again referring to a previous chapter, information is data that is combined with meaning. A temperature reading of 100 will have a different meaning if it is combined with the term Fahrenheit or with the term Celsius. Additional meaning could be added if more context for the temperature reading is added, such as whether the reading was for a liquid (e.g. water) or a gas (e.g. air), or knowledge that the "normal" temperature for this time of year is 20 0.

As such, the term information is generally used to imply data combined with sufficient context to provide meaning for a human being.

Knowledge can be thought of as information that is combined with experience, context, interpretation and reflection. We will expand on this definition as we discuss specific examples later in this chapter.

Information System as an Input-Process-Output Model

One way of viewing possible relationships between data, information and knowledge is to consider an information system from the perspective of an IPO (input-process-output) model. On the input side we have data, as discussed previously. These data are then massaged or manipulated in some way (e.g. sorting, summarizing, filtering, formatting) to obtain information. Note that the transformation of data into information may be completed by a person (e.g. using a calculator) or by a computer program, although for our purposes we are typically more interested in situations where computer programs are employed.

A simple example could be the "what if" type of analysis that an electronic spreadsheet package offers. We can use our current understanding of a situation to develop a model of how sales will go up or down by a certain factor based on the amount we spend on advertising and other factors such as price.

The resulting information is used by a human to reach decisions (how many people to hire, how many products to produce, how much to spend on advertising). The outcomes of these decisions are observable results, such as sales volume during a certain time period, or the number and size of back-orders, etc. If these objective outcomes (results) are monitored and examined, then knowledge may be gained (e.g. how to avoid inventory shortages, or how to balance inventory carrying costs against costs associated with product shortages).

The role of feedback

In addition, the observable results can be used to provide feedback into the system. This feedback can be used to help improve knowledge. The improved knowledge can then be used in two ways. First, it can be used to determine what (if any) changes are needed in the way data are transformed into information. For example, it might be decided that summarizing sales data by product category and time period is not adequate, and that a breakdown by geographic region is also needed. If such a determination is made, the result could be a change to a computer program to provide sales reports in a new format, showing information that was previously hidden within the raw data.

The second way that knowledge can influence the information system is that it can be used by decision makers to help them interpret information, influencing future decisions and actions. An example could be a review of outstanding debts in the light of prevailing or expected changes in interest rates. In this way, the quality of the decisions reached should hopefully improve over time, leading to more effective actions.

Organizations do not operate within a vacuum; they interact continuously with their environment. As such, organizations need to constantly adjust to changes in their environment. Similarly, information systems should not be viewed as being static. As new knowledge is obtained, information systems are modified, updated and expanded to address challenges or to take advantage of opportunities.

Using data to improve decision making

Most important management activities can be viewed from a decision making perspective. Within organizations (especially larger ones), numerous decisions are being made on a continuous basis. For example, decisions about how to design an assembly line in a production facility, and how to structure work tasks for employees working on the line, will have direct and substantial impacts on the efficiency and effectiveness of the use of resources (employees, production materials). Not surprisingly, many organizations have expended considerable resources to acquire or develop information systems that are designed to help improve the efficiency and effectiveness of organizational decision making.

A wide variety of terms have been used to describe information systems that are designed to support the decision making of organizational members. These include decision support systems (DSS), group decision support systems (GDSS), executive information systems (EIS), knowledge management systems (KMS), and business intelligence (BI). Additionally, the term expert system (ES) is often used to describe systems that attempt to augment human knowledge by providing access to reasoning used by experts in reaching their decisions.

On occasion, the term managerial support system or management support system (MSS) is used as an umbrella term to encompass these diverse (yet related) types of information systems (Benbasat and Nault, 1990; Clark et al., 2007). While each type of system has some unique aspect (e.g. DSS are designed to support one individual, while GDSS are used by groups; EIS are geared toward the unique monitoring and control needs of individuals that are higher in the organization; and so on), they also share some common elements. At their core, all are designed to improve decision making within organizations.

Rather than examining each of these related types of system in detail, we will focus on the functions that organizational members need to perform and decisions they need to make, and then show how information systems may be used to support them. In doing so, we'll also see how different types of management support systems can come into play.

Controlling

One important function that needs to be performed within organizations is that of control, and managers are frequently charged with controlling certain organizational processes (or functions). Data, and information, are generally essential components for aiding control.

As an example, consider the task of managing an assembly line in a production facility. For the purposes of illustration, we'll use the example of a production facility that assembles office chairs. The facility obtains parts from suppliers, assembles them into chairs, and releases the chairs to customers. The customers are distributors who then sell the chairs to the ultimate buyers (mostly larger companies that buy office chairs in bulk).

Note that for the moment, it isn't important to distinguish who has the responsibility for ensuring that the assembly line uses resources efficiently in creating chairs that are of sufficiently high quality; it could be a shift manager, or it could be the employees working on the line. Either way, certain data need to be captured, and certain information created, in order to control the operations of the assembly line.

To be more specific, assume that a decision has been reached to keep track of each part (chair back, right chair arm, etc.) that is used as input; this is accomplished by ensuring that each part has a UPC (Universal Product Code) bar-code affixed to it when it is received from a supplier, and each part's bar-code is scanned by a bar-code reader before being used in a chair's assembly. When a part is scanned, the information contained on the bar-code is copied and stored in a production database. In addition, as each part is added to the chair moving through the assembly line, a record is kept (in the database) of the time at which the part was scanned. When the chair has been completely assembled, it is placed inside a plastic bag, and either a UPC bar-code or an RFID (radio-frequency identification) tag is attached (depending on the needs of the customer). The bar-code is then scanned (or the RFID tag is read by an RFID reader), which records the time at which the chair was completely assembled and ready for storage (or shipment). This record is also added to the production database.

One way of using data to control this process is to constantly monitor the length of time it takes from when the UPC bar-code for the first part is scanned to when the bar-code (or RFID tag) for the assembled chair is scanned. By recording this information over a period of time, it is possible to obtain a distribution of observations (e.g. the mean [average] length of time taken to assemble a chair is 15 minutes, and the standard deviation is 1.5 minutes). Using this information, it would be possible to write a computer program to monitor the times taken as each chair is produced, and notify someone if the time taken is excessively long or unusually short. Note that a person (or group of people) would determine the rule for identifying exceptions (based on past experience), and the software would be programmed to enforce the rule.

Using this approach, a shift manager could be alerted, for example, when a chair takes longer than normal to be assembled. The shift manager could then investigate possible reasons for the delay (e.g. a temporary delay occurred when there were several defective left chair arms in a pallet; the immediate supply of left chair arms was depleted faster than the right chair arms, and a fork-lift truck had to be sent to the parts storage area to retrieve another pallet of left chair arms). As a result of this delay, the shift manager might institute or revise a policy to reduce the possibility of a similar delay occurring in the future.

Note that the scenario described above is only one of many possibilities of how this business process might be designed and controlled, and hence how an information system could be designed to support it. For example, an alternative would be to have the employees on the assembly line responsible for controlling the assembly process. Instead of notifying someone of the time taken after a chair has been completely assembled, it would be possible to compare the time from the start of assembling a chair until each part is scanned, and therefore it would be possible to know much sooner if a problem is occurring. As a general rule, the business process should be designed first, and then the information system should be designed to best support the process.

Automating decisions

Whenever possible, organizations strive to automate certain types of decisions. Automating decisions can be much more efficient, since you are essentially allowing a computer program to make a decision that was previously

made by a human being. In addition, automating decisions can lead to more consistent and objective decisions being reached.

Earlier in this chapter, we discussed the issue of the degree of structure for decision situations. Basically, the more highly structured the decision situation, the easier it is to automate it. If it is possible to derive an algorithm that can be used to reach an effective decision, and the data that are needed as input to the algorithm can be obtained at a reasonable cost, then it typically makes sense to automate the decision.

For example, the chair assembly company discussed previously might find it possible to automate the decision as to when to request the transportation of a pallet of parts from the parts storage area to the assembly line. This could be done by scanning not only the parts that are used in the chair assembly, but also any defective parts (which could be tagged as defective and noted as such in the database through a touch-screen monitor, keyboard or mouse at a workstation on the assembly line). By monitoring all parts that are removed from the temporary storage on the assembly line, the information system could determine when a pre-determined re-order level has been reached, and issue a request to the next available fork-lift operator to deliver the needed parts to the assembly line.

Davenport and Harris (2005) offered a framework for categorizing applications that are being used for automating decisions. Most of the systems that they describe include some type of expert system, often combined with aspects of DSS, GDSS, and/or EIS. The categories they provided include:

Solution configuration—these systems are employed to help users (either internal staff or customers) work through complex sets of options and features to select a final product or service that most closely meets their needs. Examples might include configuring a large or medium-sized computer system or selecting among a wide variety of cellular telephone service plans. The underlying computer programs would involve a type of expert system, including a set of decision rules that have been obtained from experts from the decision context.

Yield optimization—describes systems which use variable-pricing models to help improve the financial performance of a company, or to try and modify the behavior of people in some way. One example would be an airline, where 10 different people on the same flight might pay 10 different amounts for their tickets, depending on when they purchased the ticket, how frequently they fly with that airline, how full the flight is when they book their ticket, and so on.

Routing or segmentation decisions—these systems perform a type of triage for handling incoming requests for information or services. Examples include systems that balance the loads on Internet Web servers by routing requests to different servers, or systems for insurance companies that handle routine insurance claim requests and only route exceptional (unusual) requests to human claims processors.

Corporate or regulatory compliance—these systems ensure that an organization is complying with all internal or external policies in a consistent manner. For example, mortgage companies that want to sell mortgages on the secondary market have to ensure that they comply with all of the rules of that market when they are preparing the original mortgage. Similarly, insurance companies have to comply with federal and state regulations when writing insurance policies.

Fraud detection—these systems provide a mechanism for monitoring transactions and noting possible fraudulent ones. The approach used might be very simple, such as checking for a credit card's security code (in addition to the credit card number, to prove the person physically has the card in their possession). Other

approaches can be quite sophisticated, such as checking for purchases that seem to be out-of-character for the credit card holder (based on past purchasing history). By automatically identifying potentially fraudulent transactions, and then having a human operator contact the card holder to verify the transaction, the credit card company can reduce fraud losses and increase their customers' satisfaction.

Dynamic forecasting—organizations all along a supply chain can decrease their costs of operations by reducing the amount of product (raw materials, work-in-process, finished goods) that they hold in inventory. Dynamic forecasting systems (that use historical sales data etc.) help manufacturing companies align their customers' forecasts with their own internal plans. This in turn helps them to reduce their inventory carrying costs and make more efficient use of their production resources (facilities and people).

Operational control—these systems monitor some aspect of the physical environment (such as wind speed or rainfall amount) or some type of physical infrastructure (such as an electrical power grid or a communications network). If an unusual event occurs (such as a sudden surge in electrical power at one point in the electrical grid), the system automatically performs some type of action (such as shutting down some nodes and re-directing power over others).

Many management decision situations are not highly structured, however, and hence cannot (or should not) be completely automated. Next we describe systems that are designed to support decision making, rather than automate it.

Supporting complex decisions

In an unstructured decision context, there may be numerous factors or variables that need to be considered. Often, an attempt to find the "best" decision with respect to one factor will lead to a poor solution with respect to another. Even when the situation is very complex, however, it is often possible to use information systems to help support the decision-making context.

An entire branch of management theory and practice, termed management science, has evolved to try and bring more structure to unstructured decision situations. Management science is based on the application of mathematical models, and draws fairly heavily on the use of statistical analysis techniques. Examples include the use of regression analysis (to assess possible empirical relationships), simulation (to identify potential solutions by varying certain assumptions), and optimization models (to generate a "best" solution when resource constraints exist).

When a mathematical model fits well with reality, it may be possible to create an information system to help automate the decision. When the fit is less than perfect, we need to augment the use of the model with the judgment of a human decision maker. The term decision support system is sometimes used to describe information systems that are designed to help address unstructured decision situations. Many decision support systems use management science techniques to provide decision makers with alternative options.

Decision support systems do not necessarily need to be large, complex information systems. For example, a sales manager for the chair assembly company might use spreadsheet software to develop a forecasting model that could be used to predict demand levels for a product (line of chairs). After building the model to include criteria believed to impact demand (price, success rates for promotional [marketing] campaigns, etc.) the sales manager could use it to help forecast demand and then decide what demand levels to forward to the production group.

Consider a more complex example. In the early 1990s, American Airlines was faced with the daunting task of scheduling about 11,000 pilots and 21,000 flight attendants on close to 700 airplanes on flights to over 200 cities. In addition, they had certain constraints, such as the maximum time pilots and flight attendants can be in the air during a specific time period. This problem could generate between 10 and 12 million possible solutions (U.S. News & World Report, 1993).

The scheduling challenge facing American Airlines (and every other major airline) is very complex. When you consider overtime costs, labor contracts, federal labor mandates, fuel costs, demand for routes, and so on, it is obvious that there is no perfect solution. If a solution is derived that is "best" for one dimension (e.g. reducing overtime pay), another dimension (such as holiday preferences) will likely be compromised. To address the situation, American Airlines spent over two years working on a scheduling system that used management science techniques. The result was an information system that saves the company between USD 40 and USD 50 million per year, by reducing wasted flight crew time.

The scheduling information system uses data such as flight crew availability (e.g. federal mandates concerning necessary "down time" between flights, etc.), flight crew capabilities (e.g. pilots licensed to operate certain types of airplanes), flight crew preferences (e.g. base airport, requested vacation days, etc.), airplane characteristics (seating capacity, range, etc.), and route characteristics (e.g. distance, historical demand, etc.) as input. It then uses the optimization rules and logic embedded in the software to generate possible schedules that satisfy as many of the constraints as possible. The proposed schedules could be viewed as information, which is then used by decision makers (those responsible for scheduling airplanes and flight crews) to produce final schedules.

Decisions concerning the scheduling of airplanes and flight crews to flight routes has observable outcomes (e.g. the number of times a flight is delayed because a flight crew was delayed, the number of flights crew members complete over a given time period, and so on). These outcomes can be measured and examined, leading to greater insights (knowledge) which can then be used to fine-tune the rules and logic in the scheduling information system.

Knowledge management

When experienced people retire or leave an organization, typically their knowledge leaves with them. In addition, many larger organizations (e.g. major information technology consulting firms) have many people who have similar responsibilities (e.g. IT consulting) that could benefit from each others' experiences, but because of the numbers involved (and geographical separation) personal communications among the employees is not practical. A type of information system that is designed to help address these situations is often referred to as a knowledge management system (KMS).

Knowledge management systems can take many different forms, but the basic objectives are to (a) try and facilitate communications among knowledge workers within an organization and (b) try to make the expertise of a few available to many. Consider an international consulting firm, for example. The company will employ thousands (or tens of thousands) of consultants across numerous countries. It is quite possible (in fact, quite likely) that one consulting team in, say, Spain is trying to solve a problem for a client that is very similar to a similar situation that a different consulting team in Singapore already solved. Rather than reinventing a solution, it would be much more efficient (and effective) if the team in Spain could use the knowledge gained by the team in Singapore.

One way of addressing this situation is to have case histories for all client engagements posted to a case repository, which employees from all over the world can access (using the Internet) and search (using a search

engine). If the case documentation is of good quality (accurate, timely, complete, etc.), then the consultants will be able to share and benefit from each others' experiences and the knowledge gained. Unfortunately, however, it is often difficult to get employees to contribute in a meaningful way to the knowledge base (since they are probably more concerned about moving forward on their next client engagement, rather than documenting their experiences with the last one). In order for such systems to have any chance of working successfully, management may need to considered changes to reward systems and even to the organizational culture.

A different approach to knowledge management focuses more on identifying (and storing) details about the expertise of employees, and then allowing other employees to locate and contact these internal experts. This approach also has weaknesses, however, since the "experts" may spend so much time responding to requests and educating other employees that they have little time for completing their own work. As a result, employees may hesitate to volunteer information about their expertise.

Business intelligence

The term business intelligence (BI) is generally used to describe a type of information system which is designed to help decision-makers spot trends and relationships within large volumes of data. Typically, business intelligence software is used in conjunction with large databases or data warehouses. While the specific capabilities of BI systems vary, most can be used for specialized reporting (e.g. summarizing data along multiple dimensions simultaneously), ad hoc querying, and trend analysis.

As an example, consider a large grocery store chain. Similar to most competitors, the store keeps track of all purchases, down to the item category level. By that, we mean that the store keeps track of all the items that are purchased together (e.g. a package of diapers, a bag of potatoes, etc.) on a single receipt. The detailed data is captured and stored in a large database (and possibly copied into a data warehouse). Once that data is available, data analysts use business intelligence software to try and identify products that seem to be purchased together (which can help in product placement decisions), evaluate the success of marketing promotions, and so on.

When you consider the extent of the data that is captured at a check out, (goods purchased, prices, combinations, what is not purchased, date and time, how payment was made, and that much of that data can be combined with other data such as information available about the purchaser on loyalty cards, advertising campaigns, weather, competitors activities etc.) you begin to see the extent of the possibilities.

As with knowledge management systems, the value of business intelligence systems can be hindered in several ways. The quality of the data that is captured and stored is one concern. In addition, the database (or data warehouse) might be missing important data (for example, the sales of ice cream are probably correlated with the temperature; without temperature information, it might be difficult to identify why sales of ice cream increase or decrease). A third challenge can be that while data analysts may know how to use the BI software, they may not know too much about the context for the organizations operations. In contrast, a manager may know the organization, but not know how to use the BI software. As a result, it is not uncommon to have a team (a manager paired with a data analyst) to try and get the most information (and/or knowledge) out of a business intelligence system.

Conclusion

Even if people wish to use a rational approach to decision making, significant obstacles face those who attempt it. Decision makers within organizations are frequently faced with stressful environments of tight deadlines, interruptions, multiple issues requiring attention, and inadequate information.

Information systems may be used in many ways to improve decision making. Systems which capture raw data and transform it into information can help control business processes. Decisions which are very structured may be automated. For more unstructured or complex decision environments, decision support systems can help bring more structure and provide an initial assessment of alternatives. Expert systems can help capture the expertise of one or more humans, and make that expertise more widely available. Knowledge management systems can be used to both retain organizational knowledge when people leave, and to facilitate knowledge sharing among organizational members. Business intelligence systems are used to identify trends and relationships within large databases or data warehouses.

In short, information systems can be used in many ways to improve both the efficiency and effectiveness of organizational operations and decision making.

Exercises

1. Identify someone you know that works for a relatively large organization (e.g. a bank, insurance company, manufacturer, electric utility, etc.). Ask that person to identify an information system (along the lines of the ones mentioned in this chapter) that is used by the organization to automate or support decision making. By interviewing your contact person (and possibly others within the organization, if appropriate), try to address the following issues:
    a. Describe the data that is used as input to the information system. How is the data obtained? How is it input into the system? How good is the data (e.g. using characteristics such as accuracy, timeliness, completeness)? How expensive is it to obtain the data?
    b. How were the decision rules or algorithms that are used by the system developed? By interviewing experts? By reviewing historical data?
    c. What outputs (e.g. recommendations, decisions, actions) are generated by the system? Does the system make decisions, provide recommendations, or both?
    d. Does the company keep track of the outcomes of decisions made (or recommended) by the system? If so, how?
    e. Does the system get updated or modified (e.g. by updating the decision rules)? If so, how?
    f. In what ways does the system improve the efficiency and/or effectiveness of the organization? Does the organization attempt to measure the impact of the system? If so, how?
2. Think of an organization that you are familiar with (e.g. a university, a bank, an online retailer). Identify:
    a. Two decisions that must be made on a regular basis which are highly structured, and hence probably could be automated. Are these decisions currently automated, or performed by humans? If they are not automated, how challenging would it be to do so? Would it make sense to create a system (or systems) to

automate them? Consider issues such as how many people are currently used to make the decisions, how much their salaries are, how long it takes them to make decisions, etc.

b. Two decisions that are unstructured, and hence probably should not be automated. Even if they are not automated, are there ways that an information system could provide decision-makers with information that could help them make the decisions? What information would help? Is it currently available to the decision maker?

References

Anderson-Lehman, R., Watson, H.J., Wixom, B.H., and J. A. Hoffer (2004). Continental Airlines Flies High with Real-Time Business Intelligence. MIS Quarterly Executive. 3(4), 163-176.

Benbasat, I. and B.R. Nault (1990). An Evaluation of Empirical Research in Managerial Support Systems. Decision Support Systems. 6(3), 203-226.

Clark, T.D., Jones, M.C. and C.P. Armstrong (2007). The Dynamic Structure of Management Support Systems: Theory Development, Research Focus, and Direction. MIS Quarterly. 31(3), 579-615.

Davenport, T.H. and J.G. Harris (2005). Automated Decision Making Comes of Age. Sloan Management Review, 46(4), Summer, 83-89.

Fahey, L. and L. Prusak (1998). The Eleven Deadliest Sins of Knowledge Management. California Management Review. 40(3), Spring, 265-276.

Kling, R. and P.V. Marks (2008). Motivating Knowledge Sharing Through a Knowledge Management System. Omega. 36(1), 131-146.

Lamont, J. (2005). Predictive Analytics: An Asset to Retail Banking Worldwide. KM World. November/December. (Available online at

Leidner, D.E. And J. J. Elam (1995.) The Impact of Executive Information Systems on Organizational Design, Intelligence, and Decision-Making. Organization Science. 6(6), 645-664.

Negesh, S. (2004). Business Intelligence. Communications of the Association for Information Systems. (13), 177-195.

Nevo, D. and Y. Chan (2007). A Delphi Study of Knowledge Management Systems: Scope and Requirements. Information & Management. 44(6), 583-597.

Simon, H.A. (1960). The New Science of Management Decisions. New York: Harper and Row.

Thompson, R.L. and W.L. Cats-Baril. (2003). Information Technology and Management, 2nd Edition. Burr Ridge, Illinois: McGraw-Hill/Irwin, Chapter 8.

U.S. News and World Report. (1993). December 6, pp. 48-49.

Chapter editor

Ronald L. (Ron) Thompson is an associate professor of Management in the Babcock Graduate School of Management at Wake Forest University in Winston-Salem, North Carolina, USA. His has published in a variety of journals, such as MIS Quarterly, Information Systems Research, Journal of Management Information Systems, Omega, and Information & Management. Ron has served as an associate editor for MIS Quarterly, and has

presented at national and international conferences. He holds a Ph.D. from the Ivey School at the University of Western Ontario in London, Ontario, Canada. He has also served as a faculty member at the University of Calgary (Calgary, Alberta, Canada) and the University of Vermont (Burlington, Vermont, USA). His current research interests include communication in Information Systems (IS) projects and the use of advanced data analysis techniques in IS research.

9. Managing data for efficiency

Editor: William McIver, Jr., National Research Council Canada

Dedication: Roger "Buzz" King

Learning objectives

- understand the concepts of data, information, and knowledge
- understand the role that data management plays in achieving efficiency in an organization
- understand the issues involved in representing reality through data modeling

Introduction and Data Modeling

Introduction

The management of data is ultimately about representing, capturing and processing information about some aspect of reality. Organizations, communities or individuals must determine—perhaps with the help of a data management professional—which aspects of reality contain the knowledge they need to perform their tasks. Such knowledge is represented at a fundamental by data. An organization might be interested in managing data about any number of domains, including, but not limited to:

- employee information
- customer information
- documents
- part inventories
- product orders
- service orders
- geographic or spatial information
- environmental conditions
- information systems logs

The key elements of effective data management for any organization are: data models, an accurate and flexible representation of the concepts to be managed within the organization; information systems, a technical implementation and arrangement of data, software and hardware that provides for efficient processing of the data specified in the data model; and social processes, an appropriate organization of humans which allows the information system to be used in a safe and effective manner. This chapter will explore each of these areas.

Purpose of this chapter

The purpose of this chapter is to provide a set of tools for understanding data management from conceptual, technical, and social perspectives. The intention is that this chapter will serve people involved in various management roles in an organization, including:

- **general clients**: those who depend on a data management system to carry out daily tasks, but who are not involved directly in the technical aspects of the system's design or operation
- **managers**: those who manage people and operations through the use of a data management system and may or may not be be directly involved in the system's technical aspects of its design or operation
- **technical clients**: those who are involved directly in the operation and maintenance of a data management system
- **systems analysts**: those who are directly involved in the planning, design, implementation, and maintenance of data management systems.

People occupying the roles listed above will most likely have different level of expertise in data management, but ideally everyone occupying those roles should share some common understanding of the issues involved in managing data within their organization. For example, a person whose role in an organization is management must be able to communicate her data management needs with systems analysts and technical users. A manager must also be able to reason about the implications of technical issues communicated to them by systems analysts on their management tasks.

Example questions for managers:

Will the elimination of a certain database remove essential information needed to manage the organization?

Are expenses from new hardware recommended by technical users justified in terms of new efficiencies that would be realized?

On the other side, technical users and systems designers must be able to translate the organizational and social issues communicated to them by users and managers into appropriate features or modifications to the information systems they design and operate.

Topics covered

This chapter is organized into two parts. Chapter 1 provides a conceptual perspective on data management issues. It focuses, in particular, on the use of data to represent reality, otherwise known as data modeling. Chapter 2, which is preparation, extends the discussion in Chapter 1 to examine the ways in which systems are constructed for managing data. This includes the use of database management systems and the role of networking in managing remote and distributed collections of data.

What is not covered in this chapter

The topics covered in this book span or are related to a number of different computer science and information science domains, most especially *database systems*. This chapter covers several topics that are traditionally taught within database systems courses or some systems design courses, namely *data modeling* and *database systems* architectures. There are many database systems concepts that are beyond the scope of this chapter, however. These include: theoretical approaches to data modeling, including normalization theory; a comprehensive treatment of

indexing in database management systems, and programming language aspects of data management. These topics may be included in future versions of this chapter. Feedback is welcome from instructors and students alike as to their needs.

How to use this chapter

Key terms introduced in the chapter are presented first in **bold** typeface. This chapter presents conceptual and theoretical discussions interleaved with one or more related examples. This is designed to show the reader how the concepts and theories being presented at the moment are applied in real data management contexts. Readers are encouraged to develop the examples further based on their own interests.

The examples presented in this chapter are described in the section *Data management domain examples* below. Examples are referenced with the abbreviation *"EX"*. followed by the two or three letter code representing the domain example and a number representing its order within that domain example. Thus, the second example involving the *Weather monitoring* domain would be referenced as *"EX. WM-2."* Some of the domain examples presented in this chapter are not be used until the forthcoming Chapter 2 of this chapter, but the reader is encouraged to draw parallels for them while considering the examples in Chapter 1.

The role of data management technologies in achieving organizational efficiencies

One of the most important benefits that information systems provide to organizations is the ability to manage data at rates and quantities that far exceed human abilities. As computerization of organizations has evolved so too has the ability of organizations to manage ever more complex and numerous tasks. Consider the following realities:

- **postal and courier services**: The Universal Postal Union (UPU) estimates that in 2006 the number of "domestic letter-post" items delivered in Europe and the Commonwealth of Independent States was 17,591,462,361 [UPU 2007]. The courier service FEDEX reported that between 2005 and 2008 it handled an average of 3,476,000 packages per day, including an average of 496,000 packages per day for international delivery. To facilitate these deliveries it coordinates a worldwide system of over 600 aircraft, 40,000 motorized vehicles, 1,400 stations, and 140,000 employees respectively [FEDEX 2007a ; FEDEX 2007b].
- **air travel**: The US Federal Aviation Administration (FAA) reported that in 2006 the number of passengers who boarded airplanes in the US was 716,818,000 [FAA 2006].
- **health care**: As part of its mandate the World Health Organization (WHO) collects and reports on statistical information on over 150 health indicators for 193 countries [WHO 2007].
- **telephone services**: The International Telecommunication Union (ITU) estimated that by August 2007 there would be 3 billion mobile telephone subscribers in the world. African countries accounted for an estimated 198 million mobile telephone subscribers in 2006 alone [ITU 2007].
- **banking services**: It is now common for a bank to manage several million individual accounts for customers and provide access to their services via hundreds of branch offices and automated teller machines.
- **library services**: The US Library of Congress began with 3,000 items. Its collection now comprises over 134 million items with 10,000 new additions daily [LoC 2007].

Organizations in most of these sectors operated prior to the development of computerized data management systems. Posted letters were once sorted, routed, and delivered to letter carriers by hand. Airlines once processed reservations and handled baggage without computers. Telephone calls were once set up by hand without access to an electronic number database and after that calls were established mechanically based on detecting the numbers that a caller dialed. Libraries and banks also managed their customer services without the benefit of computerization. Humans' abilities to manage these endeavors has limits, however. These operations could not have **scaled** to the sizes they are now without automation.

To be more specific, none of these situations would be possible without powerful data management technologies. The services delivered in each type of organization described above necessitates the storage, processing, and tracking of millions of data items. Further complicating this is the fact that most data items have critical relationships with other data items which must also be maintained reliably. Data about an airline passenger, for example, represents not only their identity, but the flights on which they are booked, payment information for their tickets, and information about the baggage they checked before boarding their flight. By implication an airline passenger's reservation also represents changes to the current inventories of available seats on each the aircraft they will be flying. Make one error in any one of these areas of data management and a passenger may find themselves in the wrong city without their luggage; an airline may accidentally book multiple passengers for the same seat or conversely fail to realize that they still have seats available to sell. Proper data management allows organizations to perform well in the face of complexity and volume.

As the reader realizes by now the phrase “data management” means far more than performing numeric calculations which are the stereotypic duties of computers. The basic responsibilities that data management technologies have are more far reaching. They include:

- **data models**: mechanisms that allow clients to specify what data are to be managed including the logical relationships amongst them and constraints which must hold
- **storage management**: providing mechanisms for storing data in a logically coherent and space efficient manner
- **access methods**: providing mechanisms for locating desired data amongst a very large collection of data and retrieving them efficiently
- **query processing and data manipulation**: providing mechanisms that allow clients—people or software—to create, examine, change, and delete data in a convenient manner
- **security**: providing mechanisms for making data secure from unwanted access
- **transaction processing**: providing mechanisms which allow multiple clients to simultaneously examine and change data without destroying its logical coherence
- **application program interface**: providing a mechanism by which other software systems can make use of the data management system

Correctly designed and operated data management technologies allow systems such as those described above to run reliably and efficiently. Data management technologies are used most often in conjunction with other technologies. Well-designed data management technologies combined with other technologies such as barcodes

and optical scanners allow courier services and airlines to track packages and luggage and ensure that they are delivered to their proper destinations. Effective data management technologies combined with switching devices—themselves computers—allow mobile telephone service providers to route high volumes of calls to the proper phones and to associate text messages and voicemail with the correct recipients. Libraries are now better able to keep track of their massive holdings and to maintain an accurate accounting of which borrowers have which items. Likewise banks are better able to keep track of their customers' accounts and the transactions that take place against them.

Representing reality through data management[68]

Data management is ultimately concerned with representing some aspect of reality that must be recorded, analyzed, or communicated. The complete representation of reality itself—our world, the universe—is an intractable problem as we will see shortly, but even representing some small aspect of it can be a daunting a task.

The information management professional must remember that many of these issues are often not obvious to people outside of the discipline.

The aspect of reality that is chosen to be represented depends upon two main determinants. The first determinant is the nature of the **domain** for which the data are to be managed. Common generic data management domains are:

- **objects:** These can be physical or conceptual entities such as automobiles, train or airplane reservations, or health records. Data management in this domain involves recording and tracking objects.
- **events**: These can be any type of occurrence for which a record is desired, such as a business transaction or a bank deposit. Data management in this domain involves recording in a highly reliable fashion any facts about events necessary for reconstructing them at a later time.
- **organizations**: These can be individual businesses, communities, governmental agencies, or departments within larger entities. Organizational data management in this domain involves recording and tracking: objects within the organization, including people; events and processes within the organization; and relationships between processes and entities.
- **physical phenomena:** These can be any observable occurrences such as: geological conditions, weather conditions, or astronomical processes. Data management in this domain involves recording any measurements necessary for deducing, at a later time, an understanding of a phenomenon.
- **multiple domains:** This can be the management of data across several domains given above or multiple instances of the same domain. A common example is the need to manage data about multiple organizations, each having their own data management scheme, in a coordinated fashion. Data management in this context may involve the development of a global data model that encompassed the data models of the individual domains or it may involve the creation of mechanisms to map between the data models in the constituent domains.

One requirement that is common to most data management domains is the need to support queries. A query in this context is a question that is posed to and answered by a data management system using the data it is managing.

68This section is inspired by the original edition of William Kent's cult classic book: Data and Reality [Kent 1978].

The second determinant of what aspects of reality are to be represented is the set of tasks to be performed with the data that are to be managed. Suppose we are designing a data management scheme for a grocery store. For example, suppose that one of the tasks the store needs to perform is tracking its inventory so that it can know when certain items must be replenished. The aspects of reality that must be represented then include not only product numbers and their quantities for the inventory, but information about which of those products have been purchased by customers.

The concepts discussed in this section apply to many forms of data management. Most can be applied using pencil and paper. That is, they do not necessarily require computers and database management software. Technology will be discussed later. Concepts surrounding the nature of data will be discussed in the remainder of this chapter.

Data management domain examples

We need to be able to represent any aspects of reality that humans find important to remember. What books are in their library and who has which books checked out? Which people have reserved a seats on our buses, trains, or airplanes? What motor vehicles are registered within our province or country? What was the weather like where we live on a given date? These are but a few common examples of realities which are represented by data.

We will identify several canonical examples here that will be used throughout the following sections of this chapter to explain data management concepts and techniques. Each example represents some aspect of some reality about which it is common to manage data. Each of these examples brings unique challenges to data management. Each domain example description is given here along with the abbreviation that will be used to refer to it in the various example in this chapter.

- **Weather monitoring (WM)**: This example involves the recording of temperature across a geographic region.
- **Motor vehicle registration (MVR)**: This example involves recording license information about all vehicles in a geographic region.
- **Library management (LM)**: This example involves the tracking of library books through the processes of borrowing and return.
- **Train reservations (TR)**: This involves recording seat reservations for train trips.
- **Manufacturing (MAN)**: This involves the tracking of parts that are used to fabricate a product.

  There are many other are many other examples that could be included here. The reader should be able to how these examples can be extended into other data management domains.

What are data?

A **datum** is either a mathematical quantity, a set of symbols, or some combination of the two that is used to represent a **fact**. Datum is the singular of data. Facts that are represented by data may be natural objects or phenomena, human-derived concepts, or some combination of the two. Several fundamental issues are involved in using a datum in data management:

Relationships: What relationships exist between a datum and other data?

- Form: In what form must a datum exist?

- Meaning: What does a datum actually mean?

First, to be useful, a datum is usually managed in relation to some other data. For example, it is usually not useful to record and manage someone's name only. We are more likely interested in the relationships between that person—as represented by their name—and other data, such as their address, their telephone number, or anything else that is necessary for a particular set of data management tasks.

Second, the form in which a datum is presented carries a lot of meaning. In fact, it is often the case that most *types* of data must be presented in certain formats in order that they are understood. Standard conventions are usually developed for the visual layout of a given data type. Consider the conventions that apply to the presentation of telephone numbers, addresses, dates and times. One difficulty in managing data is that many conventions vary by region and country. For example, 2.5 hours past noon is represented variously as "2:30 p.m.", "14:30", or "14 h 30".

Third, what a datum might mean is the topic of next section.

EX. WM-1:

Current measurements determine that the quantity *-10* in the Celsius scale represents the temperature outside the author's window as he writes this. To make sure that this quantity is understood to be a temperature value and not something else, we may wish to add a symbol indicating the unit of measure to be degrees. The '°' symbol has become customary for this. Thus, we might have *-10*°. Further, in a world with different temperature scales, it is important to specify on which scale our quantity is to be interpreted. We would customarily add *"C",* to indicate the Celsius scale. Thus, we have *"-10° C"*.

The datum *-10° C* does not make sense to us unless we are given a context such as the time and location of the reading. Thus, we will likely want to keep temperature in relation to a number of other data quantities or symbols. One representation that we might record is the following grouping of data `<2006-12-04 07:31, Fredericton, -10° C>`, where *date* and *time*, *city name*, and the *temperature* are all represented together. In data management, this type of grouping is called a **tuple**.

The characters *"° C"* when presented together with a numeric value help us understand that the datum *-10* is a temperature reading. Likewise, the other characters present in the tuple help us realize the domains to which each other datum belongs: *date/time*, *city*, and *geographic coordinates*.

Question

- Why do we need to represent the city in the form of its name when we have its precise geographic coordinates?

EX. MVR-1 :

Governments have a need to identify motor vehicles on their roads. This is normally done by assigning to each vehicle a unique identifier, a unique set of symbols. A motor vehicle identifier is affixed to vehicles on license plates—its presentation format—and stored within some data management systems—its storage format—along with other information, such as the owner, make, and model of the vehicle.

A common format is *%c%c%c %d%d%d,* where *%c* represents a single alphabetical character and *%d* represents a single digit. So an example license plate code would be:

```
ABC 123
```

A clear pattern of characters separated by spaces is present in the presentation format of our license plate schema.

Questions

- Should the presentation format of a datum be the same as its storage format in an information system?
- What value does the space have in the presentation format?

Information and meaning

The concept of **data** is usually distinguished from the concepts of **information** and knowledge. If a datum is a fact about the real world, the **meaning** that we derive from it is **information**.

EX. WM-2:

Some data can have objective meanings. One objective meaning of *-10° C* is that it is below the freezing point of water.

Data are often given subjective meanings. A subjective meaning of *-10° C* for the author is usually the following: "It's cold!" Given that it is subjective, this temperature reading could be interpreted differently by other people. It may be considered "warm" by people who live in colder climes.

Meaning may be conveyed in the data format conventions developed for a given data type.

EX. MVR-2:

Authorities such as the police and tax officials can derive meaning from our motor vehicle identifier depending on how the codes are assigned. Suppose the **fields** of characters and numbers are each given a role. They could each be designated to represent something specific beyond providing a unique identifier for a vehicle. The first field, *%c%c%c*, could be used to represent cities or areas of the province. The second field, *%d%d%d*, could be used to provide a unique number for a vehicle within an area. Suppose all vehicles within the town of Fredericton, New Brunswick are assigned values in the first field of their licenses starting with the characters *'FR'*. Examples would be:

```
FRA 102
FRA 037
FRZ 321
```

Thus, people who know these roles would be able to derive some knowledge about the vehicle's place of registration by looking at the first field.

Sometimes it is necessary to distinguish one meaning from another. In other words, we may wish to represent such distinctions in the data themselves. This data modeling problem helps give us a partial understanding of what knowledge is.

EX. WM-3:

If the author thinks that *-10° C* is cold, what about *-9° C or -11° C?* We can measure the numeric differences between these quantities. In a data management system, however, it may be necessary to define what "cold"

means in a given context. In this case, we might define the following **rule** in our data management system to establish the meaning of cold.

```
IF temperature <= 0 ° C THEN cold = true;
```

Knowledge

Knowledge constitutes an additional level of meaning that we derive from information through some process. Sometimes this process is observational. What happens when it gets cold? Sometimes it is computational where data are processed to arrive at the higher level of meaning that represents knowledge. Data when interpreted within a certain context yields information having meaning. Information when applied within a certain context yields knowledge. A more important component in the concept of knowledge is that it also represents how we apply the information contained within it.

EX. WM-4:

Meanings of data have impacts on how an organization runs. This is an extension of information to knowledge. People who manage aircraft, for example, will be concerned with the impact of cold weather on its fleet. Observational knowledge of the cold told engineers long ago that ice can build up on an airplane reducing aerodynamic lift and potentially impeding the performance of its control surfaces. In extreme cases, this can have disastrous consequences. Here we see that how we manage data can sometimes have life-critical implications.

Thus, we are compelled to transform this type of knowledge into a computational context as the following set of rules extended from our last example.

```
IF temperature <= 0 ° C THEN cold = true;
IF cold == true THEN notify ground crew to de-ice aircraft.
```

By this point in the text it is clear that representing reality—facts, information, and knowledge—can be tricky. This is the topic of the next section.

Data versus reality

"Die Welt is durch die Tatsachen bestimmt und dadurch, dass es alle Tatsachen sind." [The world is determined by the facts, and by these begin all the facts.] [69] (Ludwig Wittgenstein, *Logische-Philosophische Abhandlung)*

We stated earlier that representing reality is an intractable problem. There are too many data types and relationships that must be identified and modeled, and some method for continuously collecting all of the data about reality would have to be implemented. Further, reality changes and so any representation of it would also have to be changed. An even trickier philosophical issue is the recursive reality that the data management process itself would also be a part of reality and, thus, it too would have to be recorded and managed.

Thus, one can only hope to represent some relatively simple aspect of reality. This can still be a challenge as we have begun to see. More specifically, this challenge can be decomposed and classified into a number of different problems, including, but not limited to, the following:deciding what objects or concepts to represent

- identifying objects or concepts inside the reality

69(§ 1.1)

- deriving new data from existing data
- representing multiple entities of the same type
- representing entities that have complex structures
- representing relationships between entities
- restricting or controlling data values

We examine each of these problems in the remainder of this section.

What must be represented?

The fundamental problems of data modeling include: deciding what entities must be represented within a chosen aspect of reality, what characteristics of those entities must be represented, and how best to balance the solutions to both of the above problems.

For any given aspect of reality, there are many real world objects or abstract concepts that we could choose to represent. This is part of the activity known as **data modeling**. The decision of what to represent is constrained, in principle, by the limited time and resources that an organization has to put into data modeling. This includes a limited capacity to design a representation scheme, called a **schema**; and a limited capacity to collect data to **populate** the **data collection** defined by that schema.

Practicality—beyond principle—dictates that the design of data management systems should be as simple as possible. As complexity in a schema increases the computation necessary to process it and maintenance requirements necessary to correct and modify it increase significantly [Banker 1993]. Thus, it is necessary to decide on a constrained version of reality that we wish to represent.

In data modeling, we will call the basic unit of representation an **entity**. Sometimes they are called **objects**—even if they represent non-physical entities. Any entity in the world can be described in terms of one or more characteristics. In data modeling, we call these **attributes**.

EX. WM-5:

We last chose to represent temperature readings using the following schema:

```
temperature_readings(date-time, city-name, temperature)
```

The entities in this case are the individual temperature readings. Each entity is described in terms of the attributes: *date-time*, *city-name*, and *temperature*.

Is it realistic to say that each city has only one temperature at any one time? The concept known as a "city" establishes both real geographic and conceptual boundaries around which we wish to record the real, physical concept of temperature.

We know very well that there may very well be different temperature readings at any one time at different locations within a given city. So which one do we record?

Fredericton is a relatively small town of *131.23 km²* by Canadian standards, thus one temperature reading may suffice. The average resident of Fredericton could easily comprehend the idea of a single temperature reading in relation to the entire city since it covers a small area. That one reading would suffice in informing a resident about whether to wear warm clothes or not.

On the other hand, there are certain data management domains where greater precision in the data is necessary. The operations staff at the airport, for example, will not be satisfied with a reading taken from the city centre. They must have a precise reading at their location since temperature can greatly impact the operations of the aircraft and the working conditions for the ground crew.

This last issue in the Weather monitoring example—the one of precision—is what is often referred to in data modeling as **granularity**.

Granularity

Granularity in the context of data modeling refers to the level of specificity or detail at which something is represented. Almost all data modeling problems offer the opportunity to represent some aspect of reality at ever greater levels of detail. Representations that contain a lot of detail are said to be **fine grained** with respect to granularity. Conversely, **coarse grained** representations contain less detail. Coarse grained representations are also often referred to as **high level**. The data modeler must ask the question: "How fine grained must the model be?"

EX. WM-6:

There are several dimensions along which we can increase the level of detail of this particular representation. Weather monitoring includes the dimensions of *time* and *space*. We are keeping temperature readings by time and geographic location. We have chosen thus far to represent the geographic location at a level of granularity of whole cities.

As was discussed above, some users of weather data may call for a finer-grained representation. To satisfy the requirements of organizations that manage airports and similar agencies, for example, we would have to adjust our current weather monitoring data model to be finer-grained with respect to geographic area. That is, our monitoring should include more locations within a given geographic area and perhaps mandated locations, such as airports, irrespective of other considerations.

The other dimension of granularity impacting Weather monitoring that was discussed above is *time*. In this context, we can adjust the granularity by increasing or decreasing the frequency with which we take and record measurements. Again, different users may have different requirements of the same type data. Individuals may be satisfied with an hourly or even daily temperature reading if they need only a sense of how hot or cold it is. Other types of users of weather data, such as airports, will require more frequent readings.

Temperatures can shift significantly in most locations. This is another aspect of representing reality that can have a critical impact on people as was discussed in our earlier discussion about deicing of airplanes. Thus, to meet the needs of airports, we must be sure to collect temperature data with high enough frequency that they are able to see temperatures rise or fall past critical points when they need to see it.

The level of granularity at which we choose to represent something brings with it simultaneous advantages and disadvantages. There are trade-offs between cost and accuracy where granularity is concerned. With a fine-grained representation, we can answer more questions with our data; however, the cost of storing a collection of data will rise as its granularity increases. Any detail we add to a data model means that more data must be collected to

populate it. Obviously, any increase in the frequency that data are collected into a given data model will also result in higher storage costs. Data management professionals must calculate these costs when designing a system.

EX. WM-7:

The Global Climate Observing System (GCOS) Network consisted of 1,016 stations around the world as of January 2007. Each station collects data anywhere from every five minutes to twice daily.

Suppose every GCOS station collects data every five minutes according to the current *temperature_readings* schema. Suppose further that *date-time, city-name*, and *temperature* requires *64*, *40*, and *4* bytes of data storage respectively. We would then require *31,104* bytes/day for each GCOS station or *31,601, 664* bytes/day for all GCOS stations in the network. The annual storage requirements would, thus, be *11,534,607,360* bytes. (The reality is that the real GCOS stations collect far more data with each reading.)

Question

- Can you derive these answers?

Independent of the level of granularity, one key aspect of representing some part of reality is the need to record information about things—objects, concepts, locations, etc.—in such a way that they can be accurately identified later.

Identity

In creating a data model, it is almost always the case that multiple entities must be represented and, further, that multiple **types** of entities must be represented. If entities within a collection of data lack **identity** it will be difficult, if not impossible, for users to find data they need. It is also important to be able to distinguish between entities of different types.

Giving each entity an identity—or identifier—gives users of a data management system a way of specifying which entities are of interest to them when they ask questions of or perform tasks with a collection of data.

EX. WM-8

In the previous discussion of this the Weather monitoring example, we decided that it is necessary to increase the granularity with respect to geographical locations of the temperature readings that are collected. That is, for a given *date-time* value it should be possible to record the collection of multiple locations within a single geographic entity such as a city. For example, for the city of Fredericton, we wish to take readings not only at the city centre, but at the airport as well, and perhaps other key locations.

Since multiple readings in this context represent different locations within a geographic entity—in this case a city—we must decide how to distinguish the readings from one another. Our current schema is: *temperature_readings(date-time, city-name, temperature)*. Suppose that on 4 December, 2006 at 07:31 in the morning, the temperature at Fredericton city center is 5° C while at Fredericton airport it is 0° C. If we use the current schema to store readings at Fredericton's city centre and at the airport at the same time (i.e. Date-time), we might naturally decide to record the readings as follows:

```
<2006-12-04 07:31, Fredericton, 5° C>
<2006-12-04 07:31, Fredericton, 0° C>
```

This problem is that this approach does not allow us to distinguish between Fredericton city centre and Fredericton airport. We could change the schema or data values we use to distinguish between each reading. Arguably the value for attribute *city-name* in the example above should be *"Fredericton",* not *"Fredericton airport"* or *"Fredericton city centre",* but this would not help us to uniquely identify each location within Fredericton.

One approach might be to use geographic coordinates as *discriminators*. Thus, we could change our schema to: *temperature_readings(date-time, city-name, latitude, longitude, temperature)*. Representing Fredericton city centre (*66° 32' W*) and Fredericton airport (*66° 10' W*), we would then record the readings such as:

```
<2006-12-04 07:31, Fredericton, 45° 52' N, 66° 32' W, 5° C>
<2006-12-04 07:31, Fredericton, 45° 52' N, 66° 10' W, 0° C>
```

For the same *date-time* value, we now have two readings, one for the city centre and the other for the airport, which is to the west of the city centre.

Would this suffice? How would the average person know which reading is for the airport and which one is for the city centre? Not many people memorize geographic coordinates in the form of latitude and longitude. Thus, we must find another way to discriminate between locations.

We probably should retain the coordinates, however, because in aviation and other application domains they could be important. One approach is to change the role of the second **field** in our scheme. We could make its role more general in geographic terms by calling it *location-name*. Values in this field will now be understood to indicate the names of any type of geographic entity—at least geographic entities that are deemed important in this application domain.

Thus, we could make use of names to such as "Fredericton City Centre" and "Fredericton Airport" in storing temperature readings. This would give us the following:

```
<2006-12-04 07:31, Fredericton City Centre,
  45° 52' N, 66° 32' W, 5° C>
<2006-12-04 07:31, Fredericton Airport,
  45° 52' N, 66° 10' W, 0° C>
```

Would this suffice? Technically, it might work. The geographic coordinates are likely to be associated with only one geographic name. If "Fredericton City Centre" and "Fredericton City Hall" effectively share the same geographic coordinates (*45° 52' N, 66° 32' W*), but we want to allow users to be able refer to either one in looking for temperature readings and assuming we record data using both location names, they could still distinguish between the two readings by the location name. A more common problem, however, is for the same name to refer to multiple entities, particularly cities and people.

What if we measure data for the city of Albany and its airport? There are at least two cities named Albany in North America: Albany, New York (*42.6525, -73.75667*) and Albany, Georgia (*31.57833, -84.15583*). Thus, we might be faced with:

```
<2006-12-04 07:31, Albany City Centre,
  42.6525, -73.75667, 14° C>
```

```
<2006-12-04 07:31, Albany Airport,
  42.74806, -73.80278, 14° C>
<2006-12-04 07:31, Albany City Centre,
  31.57833, -84.15583, 37° C>
<2006-12-04 07:31, Albany Airport,
  31.53528, -84.19444, 37° C>
```

We could address this problem in several ways. Perhaps we could add more attributes to specify locations, such as country, state or province, or county. The problem is that different location name would require different combinations of such location names.

Assigning identities to entities is often necessary for data modeling, but it is often not sufficient. To be useful, an identity must also be **unique**.

Uniqueness

The assignment of unique identifiers to entities is often the ideal way of distinguishing one entity from another. There are several ways to create a unique identifier. One way to create unique identifiers is to use a combination of existing attributes whose data values, taken together, are certain to be unique within a given data management context.

EX. WM-9:

The approach that we have used to uniquely identify locations for temperature readings has been to combine location names with geographic coordinates. While technically sufficient, we saw that it was less than ideal in terms of **usability**. Users would be able to distinguish between identical location names (e.g. Albany City Centre) using the coordinates, which would have to be different; however, coordinates would not be the most intuitive type of data for must users.

EX. TR-1:

Suppose we are designing a data management system to handle train reservations. One data modeling task would be to represent the trains. What we mean by "trains" in this context will determine what combination of attributes might be a suitable unique identifier.

Suppose we take "train" to mean a route between an origin and a destination. In this case, we could achieve uniqueness by combining attribute values for *origin* and *destination*. This assumes that our train network does not have duplicate city names. A schema suitable for uniquely identifying train routes might be the following: *trains (origin, destination)*. Note: the use of bold within a schema indicates the attributes that make up a unique identifier or key. The following is an example data collection that follows this schema:

```
<Montréal, Halifax>
<Halifax, Montréal>
<Montréal, Ottawa>
<Ottawa, Montréal>
```

Note, that this schema has the advantage of yielding a unique identity for reciprocal routes. For example, <*Montréal, Halifax*> ≠ <*Halifax, Montréal*> and <*Montréal, Ottawa*> ≠ <*Ottawa, Montréal*>.

What if our train service offers multiple trips on the same route? For example, there are at least two <*Montréal, Ottawa*> trips each day, in both directions. In this case, *origin* and *destination* would not guarantee uniqueness. The condition that prevents uniqueness—multiple trips—points to one possible solution, which is to use those data values that distinguish multiple trips on the same route. In this case it is *time*. Thus, we might modify our schema to the following: *trains (origin, destination, time_of_day)*.

We may find situations where it is important to distinguish between the same train route on different days of the week or specific dates. If a passenger orders two round trips between Montréal and Ottawa on different days, we might record those reservations as follows in our database:

```
<"Jim Smith", Montréal, Ottawa, 08:00, 2007-09-01>
<"Jim Smith", Ottawa, Montréal, 20:00, 2007-09-01>
<"Jim Smith", Montréal, Ottawa, 08:00, 2007-09-07>
<"Jim Smith", Ottawa, Montréal, 20:00, 2007-09-07>
```

As we can infer from the data values that *origin, destination, time_of_day,* and *date* must be combined to uniquely identify a route; we might add the *passenger_name* to attempt overall uniqueness of each record. Why does that not work?

It is easy to see that the combining of attributes can easily become tedious. We quickly went from juggling two attributes to four attributes. If we add the *passenger_name* attribute, we are still not guaranteed uniqueness in certain cases. What are they?

Thus, we need another approach to creating unique identifiers.

The use of multiple attributes to guarantee uniqueness can become quite messy. Often the most reliable approach to creating unique identifiers is to use a single, special attribute to which values are assigned that are guaranteed to be unique. Such an attribute is usually called a **key** or **object identifier.** One approach is simply to assign a unique number as the identifier to each entity.

EX. WM-10:

Instead of using a combination of location names and geographic coordinates to distinguish between the locations where a temperature reading was recorded, we could devise a set of unique identifiers for each station. Thus, if we have location names that are the same, like Albany, New York and Albany, Georgia, we can create a unique code for each. This is actually the approach taken by weather monitoring networks around the world. For example, Environment Canada assigns a unique five digit code, called an "index number", to each of the monitoring stations in its network. As an example, the stations listing "Fredericton" in its name are recorded as follows [EnvironmentCanada 2007]:

Index Number	**Station Name**	**Lat**	**Lon**	**Other attributes …**
…	…	…	…	…
71668	FREDERICTON CDA CS, NB	45 55N	66 36W	…
…	…	…	…	…
71700	FREDERICTON A, NB	45 52N	66 32W	…

Index Number	Station Name	Lat	Lon	Other attributes ...
...	...	...	...	...

Note that there are two stations in Fredericton with different geographic coordinates, each having its own *Index Number* attribute value. The other parts of the name have some meaning to those in the Weather Office. Index Number *71700* represents the Fredericton Airport (cf. A website such as http://maps.google.com and enter the coordinates).

Manually working out unique identifiers for each weather monitoring station works since there are a finite and slowly growing number of them. In the Weather monitoring example, however, we can imagine other entities that will have to be modeled for which this method of assigning unique identities is not feasible. There is a rapidly growing collection of individual temperature readings from each site in the network. It would not be possible for humans to manually assign unique identifiers to each temperature reading record that is stored. We would need to have our data management system generate unique identifiers somehow.

EX. TR-2:

It is common in transportation networks to assign a unique number to each route. People in a given location often know the routes by these numbers. "If you want to go to Ottawa, take the *35*, *37*, or *39*," a Montréal resident might say. Recall, that the issue we had with uniquely identifying train routes had not only to do with unique <*origin, destination*> pairings, but times of day. Thus, VIA Rail assigns a unique number to each <*origin, destination, time_of_day*> combination. Thus, the Montréal to Ottawa route has unique numbers for each time of day. An example of the routes listed previously in *EX. TR-1* with their unique "Train numbers" is as follows:

Train Number	Route name	Origin	Destination	Departure Time
35	Montréal-Ottawa	Montréal	Ottawa	15:05
37	Montréal-Ottawa	Montréal	Ottawa	16:45
39	Montréal-Ottawa	Montréal	Ottawa	18:00
14	The Ocean	Montréal	Halifax	18:30
15	The Ocean	Halifax	Montréal	12:35
34	Ottawa-Montréal	Ottawa	Montréal	15:10
36	Ottawa-Montréal	Ottawa	Montréal	16:25
38	Ottawa-Montréal	Ottawa	Montréal	17:55

As we discussed above in Weather monitoring example *EX. WM-10*, the hand assignment of unique train route codes is feasible because there are a fixed number of routes and new routes are added infrequently. The train reservations, on the other hand, will not be amenable to this approach. The reasons are the same as given for the temperature readings. Need to have our data management system generate unique identifiers somehow.

Question

- Some trains do not travel each day of the week. How would you model that?

It has been seen in *EX. WM-10* and *EX. TR-2* that sometimes uniqueness can be worked out manually. Often times, it is necessary to automate the process of assigning unique identifiers. There are a number of ways this can be done. Software developers can create services that give each requester a unique identifier. Thus, each time a new entity is created, a unique identifier is requested from the service. A few basic ideas about how this can be done are given in the following examples. In some cases it is not necessary to write new software to generate unique identifiers since most database management systems provide services for doing this.

EX. WM-11:

As was discussed in example *EX. WM-10*, we must somehow generate unique identifiers for each temperature reading. One approach is to simply ask the database management system used to store our temperature readings to generate a new unique identifier each time we create a new temperature reading record. That is, each time we store a new record in our collection, the database management system software would be instructed to insert a unique piece of information in that new record. A common method for a database management system to do this is to simply keep a counter. Each time a new record is created, the counter is incremented by 1 and the new value is given. Using this method, we might have something like the following for our temperature readings:

Reading Number	Station Index Number	Date	Time	Temperature
1	71668	2007-08-14	08:00 AM ADT	-10
2	71627	2007-08-14	07:00 AM EDT	-15
3	71700	2007-08-14	08:00 AM ADT	-11
…	…	…	…	
107	71668	2007-08-14	08:05 AM ADT	-11
108	71700	2007-08-14	08:05 AM ADT	-12

In the example above, we see that the ***reading_number*** attribute is simply an integer whose values are obtained from the database management system. It provides each record with a unique identity. This approach works as long as there is a central authority for generating the unique identifiers. In this case, that authority is the database management system. In examples WM-10 and *TR-2* the central authorities where Environment Canada and VIA Rail respectively.

What happens, however, if each region or each station in our weather monitoring network has its own database management system. In this scenario, the records would be stored in regional collections, each management by its own system, and then consolidated periodically into a national collection so that the data can be used together. In this scenario, how do we guarantee that the database management system responsible for Fredericton's two weather stations generates identifiers that are unique from those of some other region, let's say Montréal? If each database management system simply increments an integer, then we

have no practical way of insuring that each system generates identifiers that are different from the other systems.

One solution would to combine the integer values assigned by each system with some unique location or region identifier. Suppose we assign the Atlantic region in which Fredericton exists the code *"ATL"* and the Eastern region in which Montréal exists the code *"EAST"*. We might then unique reading numbers such as the following:

Reading Code	Station Index Number	Date	Time	Temperature
ATL.1	71668	2007-08-14	08:00 AM ADT	-10
EAS.1	71627	2007-08-14	07:00 AM EDT	-15
ATL.2	71700	2007-08-14	08:00 AM ADT	-11
...	...	...	...	
ATL.107	71668	2007-08-14	08:05 AM ADT	-11
ATL.108	71700	2007-08-14	08:05 AM ADT	-12

A critical problem with simple schemes for selecting unique identifiers is that they are often not secure. This is of special concern if such values are to be used by humans or if they can somehow be intercepted by other information systems.

At least two types of security are of concern here: (1) security from imitation identifiers and (2) security from errors introduced into the identifier itself. Ideally, one should not be able to guess a valid value for a unique identifier. Someone might attempt this in order to co-opt records for malicious reasons.

The introduction of errors is related in a way to the risk in (1). That is, if it is easy to guess a valid value for an identifier, then it is probably also very easy to introduce an erroneous value into a system. Common data recording errors for humans include the transposition of data (e.g. 5,401 vs 4,501) and the addition or removal of data. Thus, an ideal scheme for generating unique identifiers or other types of codes is one which is said to be **error-correcting**. For example, if given an identifier a human should not be able to introduce an error into the code that produces another valid identifier. The corollary to this is that the introduction of such an error should be detectable. Another way to view this is that the code should be **self-validating**. That is, it should be possible to determine the validity of a code value by examining the value itself.

EX. TR-3:

We could employ the same approaches that were just discussed in *EX. WM-11* to identify the reservation records discussed in *EX. TR-1*. Other ideas might be explored, however. One problem with simply selecting unique identifiers from a sequence is that they carry no useful information other than a guarantee from the database management system from which we get them that they are unique. In the last iteration of the design of our reading-number attribute in *EX. WM-11* above, we can now discern regions from identifier.

Likewise, in creating a suitable identifier for train reservations, it would be useful if, beyond being unique, one could glean critical information from them upon examination. For example, an official from the train network might be able to offer additional help to a passenger when presented with such an identifier. We

must also be wary of: (1) the possibility of creating imitation reservation numbers or (2) the ease with which errors can be introduced into an identifier's data value.

In case (1), suppose people know that the train network issues reservation numbers that are integers. It would then be very easy for people to create fake reservation numbers. With a fake reservation number—and lacking any other security measures—someone might then be able to co-opt someone else's reservation for their own or to discover information about the holder of the reservation. Ideally, we should use a unique identifier scheme for which it is not be possible for someone to guess a valid value.

Case (2) can be seen as related to case (1). It should not be possible for someone to introduce an error into a reservation number that produces another valid reservation number. Suppose Jim Smith makes a reservation over the telephone and is told by the operator that his reservation number is *"1010"* while he mistakenly writes down *"1001."* Suppose that *"1001"* is someone else's reservation number. Jim Smith now has created an identifier that can potentially violate someone's privacy. Train officials cannot easily tell that it is erroneous and, thereby, help Jim Smith to get his correct reservation number. This is a simple example, of course. In reality, other data, such as identifying information, are often used to cross check such things as reservation numbers.

EX. LM-1:

The International Standard Book Number (ISBN) is one scheme that has been developed that addresses the issues of error detection and the introduction of meaning into the structure of data values, as discussed above in example *EX. TR-3*. Each identifier is composed of unique numbers for *group* (i.e. language group or country)*, publisher, title,* and *check digit* (cf. http://www.isbn.org). An example of the 10 digit version, called ISBN-10 is: 1-40207-067-5.

```
1 := English-speaking country group
40207 := Publisher number
067 := Title (within the publisher's collection)
5 := Check digit
```

If we have the lists of codes for the different language groups, publishers, and each publisher's titles, the value of a ISBN itself will lead us to all of that information. The *check digit* is a value calculated using the values of the group, publisher, and title fields. Thus, if the check digit does not match the value calculated using the other digits, then we know there is an error in that particular ISBN number. The check digit is calculated using the following formula, where d_i is the i^{th} digit from the left:

$$\text{check digit} = d_1+2d_2+3d_3+4d_4+5d_5+6d_6+7d_{17}+8d_8+9d_9 \bmod 11$$

Thus, the first 9 digits of the ISBN 1-40207-067 would be used as follows to calculate the check digit:

$$\begin{aligned}\text{check digit} &= 1+2*4+3*0+4*2+5*0+6*7+7*0+8*6+9*7 \bmod 11\\ &= 170 \bmod 11\\ &= 5\end{aligned}$$

Note: The ISBN-10 standard specifies that if the remainder is 10, then the character "X" is to be used as the check digit so as to keep the number to 10 digits.

A data management system can then easily validate any value presented as ISBN-10 by the following rule:

```
IF (d1+2d2+3d3+4d4+5d5+6d6+7d17+8d8+9d9 mod 11) == check digit THEN
   ISBN is correct
ELSE
   ISBN is invalid.
END IF
```

Following our discussion in example EX. WM-11, we can see that this particular approach to generating identifiers counts on a central authority to define the *group* and *publisher* codes, but that once a publisher has been assigned its own publisher number, it can assign its own titles without interfering with any other publisher. Of course, it must notify some central authority of the new titles it creates.

Assigning values

Another level of reality is represented by the values we choose to assign to the attributes of the entities in a schema**.** Typically, data management software, particularly database management systems, can support the storage of various basic **data types**. These include types that represent the well-known numerical and logical domains integer, real, and boolean. Obviously, we need to be able to represent words in some way. For this, such systems allow the storage of **character** or **string** data. Strings are sequences of characters. Most software systems used for data management also permit the representation of other more specialized types such as dates, times; or arbitrarily complex combinations of other data types. The data modeler must decide on the types of values to be stored that best suit their use in a data management system. For example, if it will be necessary to perform arithmetic calculations using an integer value, the data values should be integers and not string versions of integers (i.e. *10* vs *"10"*).

EX. WM-12:

The reader should note that we used two data types across past examples to express geographic coordinates. One was the *traditional degrees, minutes (and seconds) form*, where North/South relationships and West/East relationships relative to the equator and prime meridian are represented with characters *"N"*, *"S"*, *"W"*, and *"E"* respectively. Values in this form look like the following:

```
latitude = 45° 52’ N, longitude = 66° 32’ W
```

Another form that is more common these days is a *degree decimal* representation of coordinates. The above coordinates in degree decimal form are:

```
latitude = +45.866667, longitude = +66.533333
```

Coordinates in this form make it easier to perform mathematical calculations with the values, such as determining the distance between locations. North and South are represented by positive and negative latitude values, respectively. West and East are represented by positive and negative longitude values, respectively.

Suppose, however, that we need to cross-reference coordinates in our data collection with existing documents that employ the traditional form. In this, case it may be advantageous to store our data values in the same format.

One alternative is to derive values in the form we need them from one chosen data type.

Another data modeling problem, beyond the question of data types, is to determine how the values are to be interpreted. Another way of viewing this is as the type of units that a value represents. This is particularly the case when physical phenomena are being represented for which various scientific units apply. This data modeling issue relates to the earlier discussion of the meaning of data.

EX. WM-13:

If we examine a data value for an entity's attribute in a data management system, the value alone will not tell us what it means. Suppose we decide to represent temperatures in our *temperature_readings* schema as integers. If we then examine the value of the *temperature* attribute in some record, we will see only an integer, say *-10*. What would *-10* mean though? We as humans know from the name of the attribute that it represents the physical quantity known as temperature.

The problem is that we also know that *-10* could represent a temperature on one of several scales. The question is then: what type of units—beyond integers—does this value represent? Some geographic regions still use the Fahrenheit scale. Some use Celsius. Some scientific domains assume the use of the Kelvin scale. Thus, it is necessary in our schema to somehow indicate what type of **units** this attribute is to represent.

This is a complex issue in data modeling. Addressing it ultimately entails the use of what is called **meta-data**. Meta-data are data about data. This will be discussed later. For now, let's focus on the representation of the data values themselves.

We could decide to have two attributes for the Fahrenheit and Celsius temperature scales and record the equivalent values in each attribute. This may not be the best approach, however. We will need twice the storage space now to record temperature values. We will also have to spend twice as much time recording values.

All decent modeling or design choices have trade-offs. That is, there are advantages and disadvantages to each. Some modeling choices are just bad!

What would be useful here is to be able to store the temperature in one chosen scale and to be able to ask to derive it in another scale.

Derived data values

Data modeling options have varying degrees of disadvantages in terms of the amount of **space** and **time** they require.

In deciding what must be represented by data, it is also possible to decide what need not be represented explicitly in the data. What need not be represented explicitly depends on the nature of the data. For example, in cases where multiple unit systems exists, it may make sense to store values in only one of the possible systems and then provide conversions to other systems. Data values that are represented this way are known as **derived data values**. Using a derived data mechanism, we need not stored an actual value in a data collection. Values can be calculated on demand instead, provided that the necessary input data values exist in the collection. It is important to note that derived data need not be limited to numeric data values.

EX. WM-14:

Suppose we must support users with temperature readings in both metric and imperial units. Using a derived data mechanism, we might modify the latest version of our temperature_reading schema in example *EX. WM-11* to arrive at the following:

```
temperature_readings(
   reading_code,
   station_index_number,
   data,
   time,
   celsius_temperature,
   fahrenheit_temperature(celsius_temperature))
```

By this syntax, we mean that the attribute *fahrenheit_temperature* is actually a function and that it derives its value using the value of the *celsius_temperature* attribute. Using some type of programming language, we would specify to the data management system the following algebraic equation in defining the *fahrenheit_temperature* attribute:

```
fahrenheit_temperature(celsius_temperature) :=
   1.8  x celsius_temperature + 32
```

Question

- What are the trade-offs in choosing a derived data value approach in this case?

Another case where derived values are beneficial is in the calculation of **aggregate data values**. Aggregate data values are derived using a collection of data values. The most common examples are statistical measures such as the average, mean, or median values of a collection of numbers. Other aggregate measures such as the sum, maximum value, or minimum values of a set of numbers are common as well. Database management systems typically provide **built-in** services or **functions** that calculate such measures over a collection of data specified by the user.

EX. WM-15:

Suppose we must support users with the average, maximum, and minimum temperature readings at each station for the last 24 hours. One method would be the following:

Given the schema:

```
stations (station_index_number, location_name)temperature_readings(
   reading_code,
   station_index_number,
   date,
   time,
   temperature)
station_statistics (
   unique_id,
   date,
   time,
```

```
  station_index_number,
  avg_temperature,
  max_temperature,
  min_temperature)
Collect current temperature readings from all stations.
```

g. Store each reading collected during Step 1 as an entity in the *temperature_readings* data collection.

h. For each *station_index_number* in the stations data collection, calculate the *average*, *maximum*, and *minimum* temperature values recorded in the *temperature_readings* collection over the last 24 hours; and store each calculation in the station_statistics data collection.

i. Wait for a specified time interval then got o Step 1.

Step 3 in our method above will cost us space, but it will also cost us in computational time. The latter will come about with the need to read values from the *stations* and *temperature_readings* collections in order to calculate the values to be stored in the *station_statistics* collection.

Using a derived data value approach, we can save space and potentially time by calculating the aggregate statistics only when requested by a user.

Questions

- What approach would you choose if there are likely to be many requests for statistics?
- Can you think of a way to optimize the procedure above in terms of computational time?

The costs in space expenditures are not zero for derived values. In a computer system at least, some amount of space is required to store the methods that are required to derive a value. These are traditionally called **stored procedures** in database management systems. For example, our temperature conversion formulas would be stored in the form of a small computer program. Many database management systems allow such procedures to be stored with the data. In other cases, developers decide to implement the procedures within external computer programs.

As was discussed at the start of this section and but can be seen more clearly by now, deciding what should be represented in data that we collect should depend on finding the best ways in which it will be understood and used.

Representing composite entities

Certain types of entities are better understood—and, thus, modeled—as being comprised of not just a set of attributes, but of other whole entities. These are called **composite entities** or **composite objects**. The entities which comprise another entity are referred to here as **constituent entities**. Sometimes they are referred to as **components** or **child objects** of an encompassing **parent entity** or **parent object**. It is important to note that many types of entities cannot exist in their own right without the existence of their constituent entities.

EX. TR-4:

One trip on one train route in our example can be modeled as a number of composite entities. The equipment that is to be used is itself is a composite. There may be several engines and any number of cars. A train company would certainly use a data management system to keep track of each of these pieces of equipment individually. For a given trip, the company would also need to know which of these individual pieces of

equipment are used together. Which engine did we send to Ottawa today with train *39*? Thus, we might look at a composite model of the equipment used for a trip as follows:

```
train_equipment(trip_id=2007-08-17 18:00, ... )

↳engines
  ↳{engine(id=E105, model=General Electric P27,... ),
    engine(id=E2711, mode=Budd Diesel #2, ... )}

↳passenger_cars
  ↳{passenger_car(id=P143,model=Bombardier 101,
    capacity=100,...),
    passenger_car(id=P2008,model=Bombardier 101,
    capacity=100,...),
    ...  ...
    passenger_car(id=P2009,model=Bombardier 101,
    capacity=90,...) }

↳dining_cars
  ↳{dining_car(id=D013, model=Bombardier D101,...)}

↳caboose(id=C007, model=Bombardier C101,...)
```

Thus, the train_equipment entity can be seen as having many components: collections of engines, passenger_cars, a dining car, and a caboose. Each of the constituent entities exist by themselves in their own right, but in order to understand the concept of train equipment for a particular trip, they are **aggregated** together as a single composite entity.

Question

- Can you create models of passenger reservations looking at them from a composite context.

Relationships

Another aspect of reality that must often be captured are the relationships that exist between entities in the real world. Relationships can be viewed in a variety of ways. From one perspective, relationships can be used to define the logical structure of a set of entities. Another important perspective is that which defines what various entities mean to each other.

The preceding section discussed the concept of a composite entity, an entity that contains other entities. A composite entity is one that defines structure. It represents the structure of containment between parent entities and their constituent entities. These are often called **has-a** relationships.

EX. TR-5:

The train-equipment schema discussed in example *EX. TR-4* defines **has-a** relationships. (There are other types of relationships in that schema that we will discuss below.) Using the **has-a** relationship, we can make the following assertions about the *train-equipment* schema:

```
train_equipment has-a set of engines
train_equipment has-a set of passenger_cars
train_equipment has-a set of dining_cars
train_equipment has-a caboose
```

The concept of data collections that have been referred to extensively in this section imply the set-theoretic concept of **membership.** A data collection can be viewed as a set, though most often not in strict set-theoretic terms since duplicate elements may sometimes exist. Nonetheless, an entity of a given type is often defined in terms of set of zero or more entities of that type.

EX. TR-6:

The train-equipment schema discussed in example *EX. TR-4* is a composite entity that is partly composed of several types of sets of entities. Note the use of braces (i.e. "{" and "}") in that example. The *train-equipment* schema contains a set of *engines*, a set of *passenger_cars*, and a set of *dining_cars*. From the train_equipment entities perspective, it has several **has-a** relationships with respect to these sets as we discussed in example *EX. TR-5*. Relationships also can be viewed from the perspective of the elements of those sets. Using the **element-of** relationship, we can make the following assertion about the *train-equipment* schema:

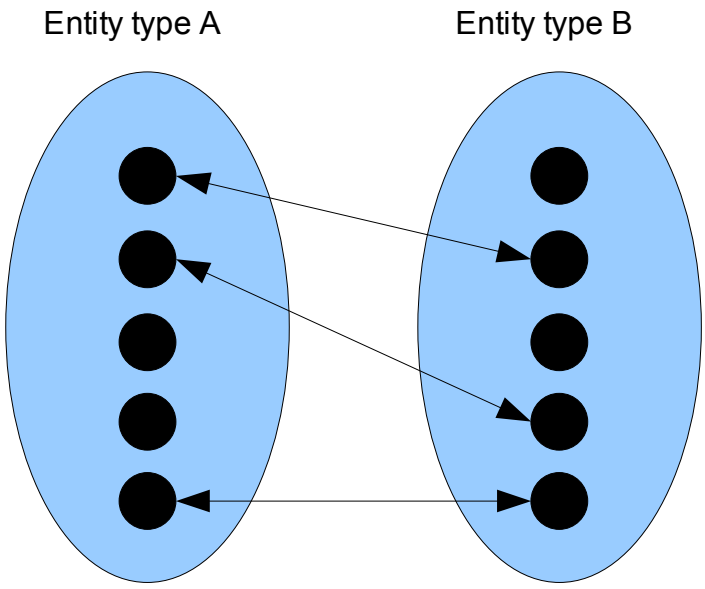
passenger_car having id=P143 is an **element-of** *passenger_cars*

Another perspective on relationships describes the **cardinality** of the associations between entities. There are three basic types of cardinalities: **one-to-one**, **one-to-many**, and **many-to-many**. These can be visualized in the following ways:

Cardinalities

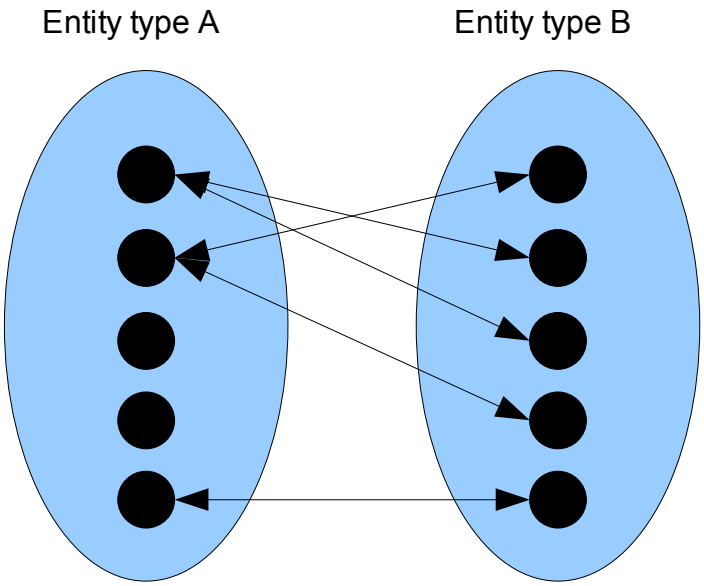
one-to-one

We depict a 1:1 relationship between type A and type B.



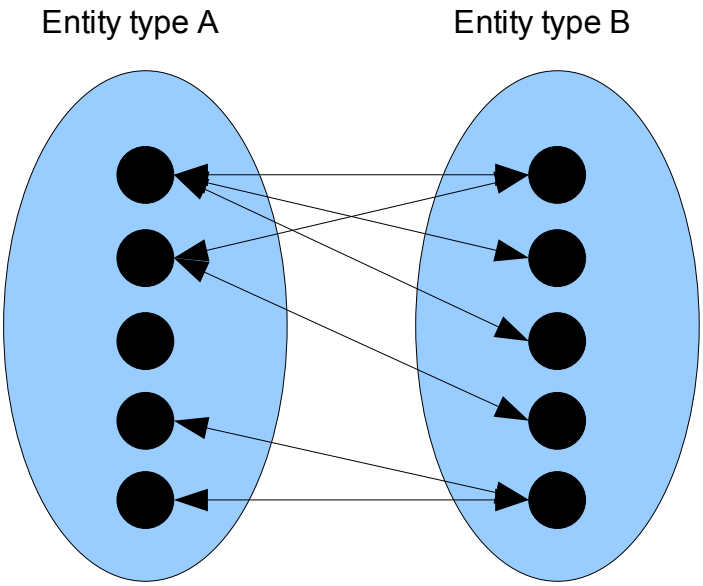
one-to-many

We depict a 1:m relationship between type A and type B.



many-to-many

We depict a **m:m** relationship between type A and type B.



Meta-data

Reiterating**, meta-data** are data about data. One key theme in this discussion on data modeling has been the problem of understanding the meaning of the values of data in a collection. The meaning of a datum can be understood in a set-theoretic sense. The number "10" is easily understood in terms of its membership in the domain Integer, for example. It is not possible looking at a value in isolation what its purpose is or what it represents. Human understanding might make use of the attribute names, but as has been shown, even those are limited in

explaining the contents of the attribute. For example, the attribute name “temperature” does not convey what scale its values represent. The purpose of meta-data is to fill these gaps in meaning.

Data can be explained at different levels. Thus, there are different levels of meta-data. At the most fundamental level are data about the physical storage properties of data. This is known as the **physical schema**. Meta-data at this level can include information about file formats in which data are stored, properties of the devices used for storage, and the bit-level and byte-level representations of data values.

In discussions above, data have been explained at a logical level. This is a level that is free of considerations about the physical details of data formats and their storage. This is known as the **logical schema**. Meta-data at this level describes the data model: entities, attributes of entities, data types of attributes, value constraints on attributes, and relationships between entities.

Higher levels of explanation can be given concerning data. As discussed above, meta-data are necessary for describing the meaning of data beyond what is usually described by a logical schema. Descriptions at these levels comprise what are sometimes called **semantic data models** (cf. [Hammer 1978]). Meta-data at this level may: provide human descriptions of attributes and define the value system within which data should be interpreted (e.g. Celsius vs Fahrenheit).

Long-standing research has been done to facilitate levels of understanding of data above the logical schema. That is, to facilitate understanding of data that approaches that of artificial intelligence.

Constraints

In a data model, the values that attributes can take are constrained at a basic level by the data type of the attribute. For example, if the attribute “number” is defined as an integer, it may only hold numeric integer values, such as the number 10; it would not be allowed to hold string values, even the string “ten”.

It is often the case that a data model must enforce ***constraints*** on data values even if they are technically legal from the standpoint of the domain of an attribute. For example, even if an attribute is defined as an integer, it may be necessary to restrict which integers it will be allowed to hold. Ideally, a data management system allows constraints to be defined within data models that enforce such restrictions.

A common situation where constraints on data values are necessary is in representing entities that have a fixed set of values. For example, an application that asks users to input a postal address can and should enforce the correctness of certain values that are a part of an address. Postal codes, provinces or states, and even city names in a given country exist within fixed sets. Thus, even if the *postal code* and *province* attributes are of a string data type, the particular string values they are allowed to hold can be restricted to the values contained in the official lists of postal codes and province names, respectively.

The principle of using constraints to enforce attribute data values can be applied to any type of data, even those of mixed types. Database management systems typically provide mechanisms that allow designers to specify constraints at the same time they specify a schema.

Another category of constraints within a data model involve the **cardinality** of **relationships** between entities. Thus, they are called **cardinality constraints**. These are used to restrict the types of cardinalities that exist between entities, as discussed in *the section on relationships*. Corresponding to the discussion in that section, the cardinality of relationships between two types of entities can be constrained at a basic level to: one-to-one, one-

to-many, or many-to-many. A one-to-one constraint between *entity type A* and *entity type B* in a schema says that a given entity of type A may be associated with only one other entity of type B. One-to-many and many-to-many are used similarly.

A cardinality constraint may go further, however. It is sometimes necessary to specify a minimum cardinality in a given relationship. For example, given a one-to-one relationship between *entity type A* and *entity type B*, the following specific types of cardinality constraints might also be added:

c. *1..1* := one entity type A *must* be associated with one entity of type B

d. *0..1* := one entity type A *may* be associated with zero or on entity of type B

Given a one-to-many relationship between entity type A and entity type B, the following specific types of cardinality constraints might also be added:

1..m := one entity type A *must* be associated with one or more entities of type B

a. *0..m* := one entity type A *may* be associated with zero or more entities of type B

b. *0..x* := one entity type A *may* be associated with zero or up to *x* entities of type B

c. *1..x* := one entity type A *must* be associated with at least one or up to *x* entities of type B

Given a many-to-many relationship between *entity type A* and *entity type B,* the following specific types of cardinality constraints might also be added:

0..m := each entity *may* be associated with 0 or more entities of the other type

a. n..m := each entity *must* be associated with 1 or more entities of the other type

Another perspective on cardinality constraints is as dependencies. A dependency defines a relationship that is *required* between entities.

EX. TR-7:

In our *train_equipment* schema, some cardinality constraints are required. For example, that *train_equipment* must include at least one *engine* in its set of *engines*. This may seem obvious to us, but in terms of data management, we must try to architect our systems to ensure that erroneous data are not entered. The use of cardinality constraints helps in this cause. In defining the *train_equipment* schema then, we can specify a minimum number of engines.

In reality in this particular application domain—the management of trains—we may need to derive the minimum cardinality of the *engines* set based on other factors.

Question

- Without necessarily being an expert on trains, can you make educated guesses about the variables that should be considered in determining the cardinality of the engines set?

Editor

Dr McIver is a Senior Research Officer in the National Research Council of Canada Institute for Information Technology (NRC-IIT), where he is a member of the People-Centered Technologies Group. His research is multidisciplinary, covering computer science and community informatics. At NRC-IIT he combines his computer

science and social informatics research interests to develop information technologies for enhancing community life. This includes democratic participation, exchange of life-critical information, and culture. Dr McIver is a graduate of Morehouse College, Georgia Institute of Technology, and the University of Colorado at Boulder with BA, MS, and Ph.D. degrees, respectively, in computer science. He may be reached at Bill.McIver@nrc.gc.ca. See also the NRC-IIT Web site at <http://iit-iti.nrc.gc.ca> .

References

[Banker1993] Banker, R. D.; Datar, S. M.; Kemerer, C. F. & Zweig, D. Software complexity and maintenance costs. *Commun. ACM, ACM Press*, 1993, 36, 81-94.

[EnvironmentCanada2007] Environment Canada, WMO Volume A Report—Canada, REGION IV—NORTH AND CENTRAL AMERICA, Generated 2007-08-13 9:00 pm,

URL: http://www.climate.weatheroffice.ec.gc.ca/prods_servs/wmo_volumea_e.html

[FAA2006] Federal Aviation Administration, Passenger Boarding and All-Cargo Data, 2006. URL:http://www.faa.gov/airports_airtraffic/airports/planning_capacity/passenger_allcargo_stats/passenger/media/cy06_primary_np_comm.pdf

[FEDEX2007a] FedEx, FedEx Statistical Book, 2007. URL: http://www.fedex.com/us/investorrelations/downloads/sec/corp/current_stat_book.xls

[FEDEX2007b] FedEx, FedEx Express Facts, 2007. URL: http://www.fedex.com/us/about/today/companies/express/facts.html

[Hammer1978] Hammer, M. & McLeod, D. The semantic data model: a modelling mechanism for data base applications SIGMOD '78: Proceedings of the 1978 ACM SIGMOD international conference on management of data, ACM Press, 1978, 26-36.

[ITU2007] International Telecommunication Union, ICT Statistics, 2007. URL:http://www.itu.int/ITU-D/ict/statistics/ict/index.html

[Kent1978] Kent, W. *Data and Reality*, Amsterdam; New York: North Holland, 1978.

[LoC2007] Library of Congress, Fascinating Facts, 2007. URL: http://www.loc.gov/about/facts.html

[UPU2007] Universal Postal Union, Postal Statistics: Query Results, 2007 URL: http://www.upu.int/pls/ap/ssp_report.main?p_language=AN&p_choice=AGGREG

[WHO2007] World Health Organization, WHO Statistical Information System: Core Health Indicators, 2007.

URL: http://www.who.int/whosis/database/core/core_select.cfm

[Wittgenstein1922] Wittgenstein, L. *Tractatus Logico-Philosophicus,* C.K. Ogden, Trans., Routledge and Kegan Paul, 1922.

10. Opportunities in the network age

Editor: Leyland Pitt (Simon Fraser University, Canada)

Learning Objectives

- define a business opportunity, especially in a business to customer setting
- understand what is meant by the "new five forces" and how these forces generate business opportunities
- explain Moore's Law and its effects
- explain Metcalfe's Law, and how it operates in network settings
- understand "Coasian", or transaction cost economics, and how innovations in communication technologies affect the costs of firms and markets
- understand and explain the "flock-of-birds" and "fish-tank" phenomena

Introduction

US General Douglas MacArthur once said: "There is no security on this earth, there is only opportunity." An opportunity is simply an appropriate or favorable time or occasion. Alternatively it may be a situation or condition that is favorable for attainment of a goal. It might just be a good position to be in, a good chance, or a prospect. Opportunities abound in the network age, but many organizations are blinded to them because they seek security in environments that make them comfortable. They like dealing with familiar customers, and satisfying familiar needs with familiar offerings, and competing against easily recognizable rivals who do similar things that they do. Yet significant growth seldom comes from safe comfortable business environments—it usually comes from change. Business opportunities arise most often when customers don't have a choice.

According the "theory of competitive rationality", ideas developed by marketing professor Peter Dickson (1992), in most markets and industries there is a situation of over-supply, or simply, there is more capacity to produce the goods or services than there is a demand for them. There is more capacity to produce cars than there is a demand for them (witness the large number of cars on dealer lots), more capacity to produce life insurance than people who want it, and more capacity to produce clothes than there is a demand for them (as the stacked racks and shelves in clothing stores will attest to). Over-supply creates customer choice—without over-supply, customers have no alternatives. And how do customers behave when they are faced with choice? They exercise it. In doing so, they acquire information, they make all kinds of trade-offs, and they make decisions. In doing so, they learn, become smarter, and more sophisticated with regard to the particular product or service.

How do firms respond to sophisticated, smarter customers? They innovate—they try to offer customers something they haven't seen before, something new, that will get their attention and hopefully their spending. But the problem with innovations is that competitors copy them, and that imitation leads to…more oversupply. And so the cycle of competitive rationality keeps turning.

By now, some readers will have thought of current exceptions to the oversupply rule. There is no over-supply for example, of organs for transplantation. There is no over-supply of great works of art—Rembrandt, Vermeer and Van Gogh are deceased. There are many other situations where there is no over-supply, and of course this means that customers have no choice. Biotech firms are working hard to find answers to the organ donation problem, and museums strive to find solutions to the fact that lots of people want to see, appreciate and perhaps even own a great and famous picture. The point is simple: When customers don't have a choice, smart entrepreneurs see opportunities.

The network age—the age of the Internet, cellular phones, digital media, satellites and GPS systems—is spinning off opportunities faster than almost any age in history. In this chapter we consider some fundamental forces that may serve as guideposts to entrepreneurs in identifying, at best, opportunities that might be identified, or at least, recognize potential threats to existing firms in business to customer markets.

Traditional strategy and killer applications

A leading influence on strategic thinking throughout the 1980s and 1990s has been Harvard Business School strategy professor, Michael Porter. Practitioners, academics and consultants alike have used his well-known "five forces" model to evaluate industry attractiveness as strategic positioning. Porter's "5 Forces Model"2,3 is very orderly and structured—in fact, very applicable to the 1980s, which was a far less fragmented era than what the late 1990s, and especially the new millennium, have proven to be. It explains neatly why some industries are more attractive than others in a way that at least gave managers confidence in their judgment, even if it didn't make them feel better about being in a dead-loss market. Similarly, at least the forces were all about business and management issues—customers and suppliers, barriers to entry and substitute products, and of course, good old firm-to-firm competition itself. Small wonder then that practitioners, consultants and academics loved the approach, for it gave reasonable predictability in a reasonably predictable world.

The traditional view of strategy in organizations has been that it is possible to understand fully the environment in which the organization functions, and therefore to plan for the firm's future accordingly. This view might be acceptable when the environment changes slowly, and the future is reasonably predictable. It might even be gratifying when trends are linear. However, in recent years some observers have noted that the environment is changing so swiftly that it is not possible to strategically plan for the future (see for example, Downes and Mui4). As we saw in the first chapter, trends nowadays are usually paradoxical and contradictory rather than linear. The new environments that emerge, especially as the result of phenomenal changes in technology, have profound effects on society, and not just on the firms that operate within it.

If one were to study the occurrence of inventions in history, one phenomenon is particularly prominent—the rate at which technological change occurs over time. During the Middle Ages, for example, significant innovations appeared at a very slow rate-sometimes as infrequently as 200 or 300 years apart. During the time of the Renaissance, new technologies begin to emerge slightly more rapidly—for example the invention of movable type by Gutenberg. In the era of the Industrial Revolution, inventions begin to surface far more frequently, now at the rate of one every 5 or 10 years. Entering the 21st century, we begin to see innovations break the surface once every two years, or indeed every year. The kinds of innovations that we are talking about are not simple improvements—rather, we are referring to what have become known as "killer applications".

A killer application or "killer app"4 is not merely an innovation that improves the way something is done. It is not even something that merely changes a market or an industry—indeed, a killer application is one that changes the way society itself, works and functions. The automobile was a “killer app” because it didn’t just simply replace horse-drawn carriages, or alter the way people traveled—it transformed the way we live, shop, work and spend our leisure time. It also changed the appearance of the physical environment in most countries. In the past 10 or 15 years we have seen killer applications arise at the rate of more than one a year, and this frequency is increasing in an exponential fashion at the moment due to “spreading” technologies such as the Internet. So strategy that attempts to plan five years ahead is befuddled by the fact that society and the way the world works may indeed change at the rate of one or two killer applications a year. The more traditional strategic planning models such as those of Michael Porter are less effective at dealing with the kind of strategic planning problems that killer applications and rapid technological changes cause.

The five new forces

We need to develop a perspective on the new forces that impact on strategy and the way organizations deal with the future. One possibility is that, in the spirit of Porter’s 5 forces, we consider five new forces that will affect the way the business and management environment works. We also illustrate these forces and their effects, and how they spun off opportunities in business to customer situations for alert entrepreneurs as well as threats to the incumbents, using two cases: First, the music industry worldwide and second, the online betting exchange, Betfair.com. (These two industries are briefly described in the vignettes illustrated in Exhibit 54 and Exhibit 55). The effect of these forces on individuals and organizations is illustrated and summarized in Exhibit 54. In no particular order these forces are:

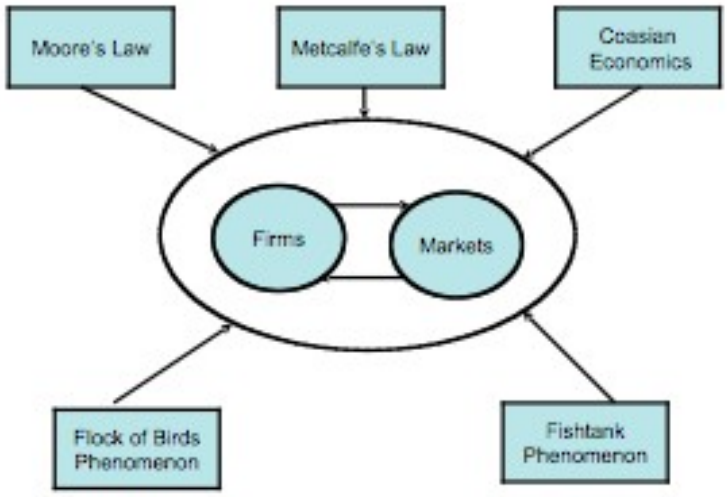


Exhibit 54: The new five forces

Moore’s Law

In 1965, an engineer at Fairchild Semiconductor named Gordon Moore noted that the number of transistors on a computer chip doubled roughly every 18 to 24 months. A corollary to "Moore's Law", as that observation came to be known, is that the speed of microprocessors, at a constant cost, also doubles every 18 to 24 months. Moore's Law has held up for more than 40 years. It worked in 1969 when Moore's start-up, Intel Corp., put its first processor

chip—the 4-bit, 104-KHz 4004—into a Japanese calculator. It worked at the return of the century for Intel's 32-bit, 450-MHz Pentium II processor, which had 7.5 million transistors and was 233,000 times faster than the 2,300-transistor 4004. And it works today, when Intel's Rio Rancho factory is expected to begin producing 45-nanometer chips—meaning they will have features as tiny as 45-billionths of a meter—in the second half of 2008. The transistors on such chips are so small that more than 30 million can fit onto the head of a pin. Intel says it will have a 1-billion-transistor powerhouse performing at 100,000 MIPS in 2011.

For users ranging from vast organizations to children playing computer games, it's been a fast, fun and mostly free ride. But can it last? Although observers have been saying for decades that exponential gains in chip performance would slow in a few years, experts today generally agree that Moore's Law will continue to govern the industry for another 10 years, at least. Moore's Law is illustrated graphically in Exhibit 55, which shows the increases.

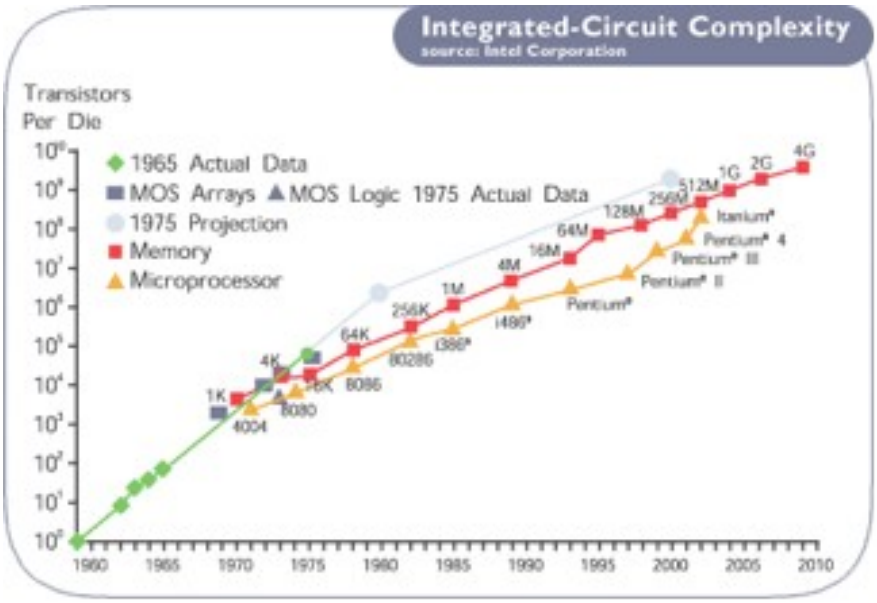


Exhibit 55: Moore's Law

The implications of Moore's Law are that computing power becomes ever faster, ever cheaper. This not only means that just about everyone can therefore have affordable access to powerful computing, but that the power of computers can be built into devices other than computers themselves. Moore's Law also drives convergence by placing computer chips into objects that previously had nothing to do with them—today there is more processing power in the average cellular telephone or digital television set than NASA had access to when Neil Armstrong landed on the moon in 1969. Already, computers are used in products as diverse as vehicles, surgical equipment and elevators, enabling these machines to operate more efficiently, predictably and more safely. We are beginning to see computer chips in disposable products such as packaging, as the costs continue to decline. Hitachi Ltd., a Japanese electronics maker, recently showed off radio frequency identification, or RFID, chips that are just 0.05 millimeters by 0.05 millimeters and look like bits of powder. They're thin enough to be embedded in a piece of paper, yet are able to store considerable amounts of information which can then be read and processed. Computers have become ubiquitous—they are everywhere, but we do not consciously think of them or notice them.

The primary question that Moore's Law should prompt in strategic planners is this: what will our industry or market be like when computers or chips are literally everywhere—in every product we make or part of every service

we deliver? Some managers may think this is silly, simply because it is difficult for them to imagine a computer or chip in their product or service. Yet there are countless products or services being delivered today that have computers as an integral part that the same reasoning would have applied to just twenty years ago: Hotel doors with chips in that facilitate card access and record entry and exit; microwave ovens; and digital television. In the recent past we have witnessed the demise of the VCR, as home owners turn to hard drives to record many hours of television entertainment. An 80 gigabyte hard drive recorder costs less than USD 300. In the lifetime of most readers of this book, there was a time when the combined computer storage of most countries did not reach 80 gigabytes.

Metcalfe's Law

How useful is a piece of technology? The answer depends entirely on how many other users of the technology there are and on how easily they can be interconnected. For example, the first organization with a facsimile machine had no one to fax to, and no one to receive faxes from! One telephone is useless; a few telephones have limited value. Many millions of telephones create a vast network. These effects are known as Metcalfe's law. Robert Metcalfe, founder of Novell, 3COM Corporation and the designer of the robust Ethernet protocol for computer networks observed that new technologies are valuable only if many people use them. Roughly, the usefulness, or utility of the network equals the square of the number of users, the function known as Metcalfe's Law. This is illustrated in the simple line graph in Exhibit 56.

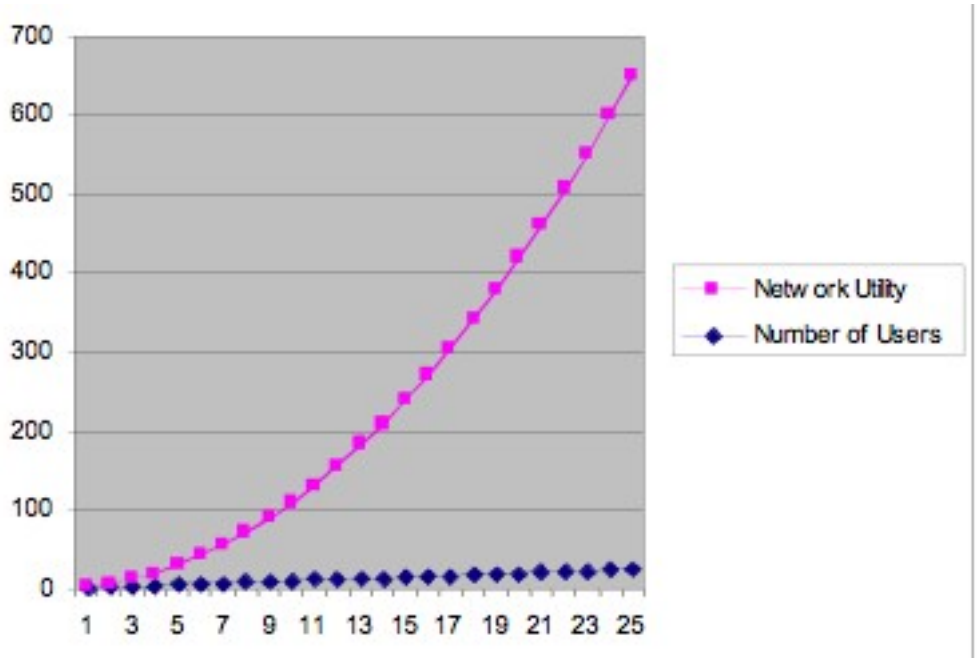


Exhibit 56.: Metcalfe's Law

The more people who use software, a network, a particular standard, a game, or a book, or indeed a language such as English, the more valuable it becomes and the more new users it will attract. This in turn increases both its utility and the speed of its adoption by still more users. The Internet is perhaps the best illustration of Metcalfe's law. While it began in the 1960s, it is only in the past dozen years that it has gained momentum—as more users joined the medium it became more useful to even more users, thus accelerating its growth. Now its potential to spread new ideas, products and services is awesome. Other good examples of Metcalfe's Law in recent years have been cellular telephones and the ubiquitous Palm digital assistant. In the case of the latter for example, most early adopters recommended the product to friends and colleagues, and the Palm succeeded not because of a large

advertising budget but because of word-of-mouth. The early adopters insisted that others buy the product not just because it was good, but because it increased the size of their network, which made their own Palms more useful. One of the key factors in this success was the Palm's ability to "beam" to other Palms via a built-in infrared device. Palm users were proud not to carry paper business cards, and preferred to beam their details to others.

Networks are important because they create short cuts. Anyone who is part of the network can contact anyone else who is part of it, and bypass more traditional channels and structures. This is important for planners who should consider what the effects of technology will be that enables their customers to talk to each other, suppliers to talk to each other, and customers to talk directly with suppliers, wherever in the world they may be. As networks grow, their utility increases, so Metcalfe's Law tells us—this is true for those who are part of the network, and for those who then choose to join it.

Coasian economics

Nobel Prize winner in economics, Ronald Coase made a discovery about market behavior that he published in a 1937 article entitled "The Nature of the Firm"5. Coase introduced the notion of "transaction costs"—a set of inefficiencies in the market that add or should be added to the price of a good or service in order to measure the performance of the market relative to the non-market behavior in firms. They include the costs of searching, contracting, and enforcing. Transaction cost economics gives us a way of explaining which activities a firm will choose to perform within its own hierarchy, and which it will rely on the market to perform for it. One important application of transaction costs economics has been as a useful way to explain the outsourcing decisions that many firms face—for example, whether the firm should do its own cleaning, catering or security, or pay someone else to do this.

The effect of communication technology on the size of firms in the past has been to make them larger. Communication technologies permits transaction costs to be lowered to the extent that firms are capable of subsuming many activities within themselves, and thus are able to operate as larger entities even across continents. This has permitted multi-nationals such as General Motors, Sony and Unilever to operate as global enterprises, essentially managed from a head office in Detroit, Tokyo or London, or wherever. Communication technology such as telephones, facsimile machines and telex machines enabled these operators to communicate as easily between Detroit and Sydney as between Detroit and Chicago. Smaller firms found this more difficult and more expensive. So, large firms brought more activities within the firm (or the "hierarchy" in transaction cost terms), for it was cheaper to do this than to rely on the market.

What strategic planners will overlook at their peril in the age of the Internet is that these same communication capabilities are now in the hands of individuals, who can pass messages round the world at as low a cost as the biggest players—essentially, for free. Free voice over Internet protocol services such as Skype allow individuals to talk for free, regardless of location or distance. They can also hold multi-user conferences, including live video, for free, and simultaneously transmit documents and images. The effect of the new communication technologies, accelerated by Moore's Law and Metcalfe's Law will be to reduce the costs of the hierarchy. But more especially, they will reduce the costs of the market itself. As the market becomes more efficient the size of firms might be considerably reduced. More pertinently, as the costs of communication in the market approach zero, so does the size of a firm, which can now rely on the market for most of the activities and functions that need to be performed. A very thorny strategic issue indeed!

A simple illustration of the effects of transaction cost reductions in an industry is shown in Exhibit 57. When banking is done across the counter with a teller, the average cost per transaction exceeds a dollar; when the same transaction is conducted online the costs reduce to nominal cents. Yet most banks charge their customers more for online banking! One wonders how long they will continue to do this, and how long customers will tolerate it? Already, alternative services are beginning to emerge that may prove more appealing to many customers.

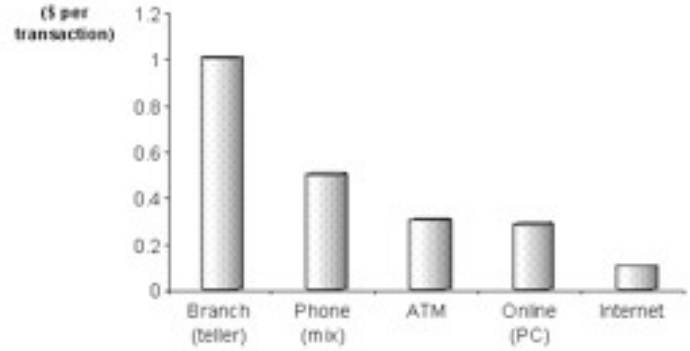


Exhibit 57: Reducing costs of banking transactions

There are many strategic questions that Coasian economics prompts in the age of the Internet. However, what should undoubtedly top the agendas of many strategic planners in this regard is the issue of what functions the Internet will permit them to outsource. Allied to this is the matter of responding to competitors who do not carry the burden of infrastructure normally borne by traditional firms, having relied on technology to effectively outsource infrastructure and functions to the market.

The flock-of-birds phenomenon

A feature of many of the new communication technologies has been the fact that in most cases they do not "belong" to any one institution, nor does any particular authority control them. The Internet is a case in point. Some have referred to this as the "flock-of-birds phenomenon"6. When one observes a flock of birds flying in formation, or fish schooling in a distinct pattern, one is tempted to speculate whether there is or could be a "bird in charge" or an "alpha fish". Naturalists will explain that flocking is a natural phenomenon and that there are indeed no "head" fishes or birds in charge. The network becomes the processor.

Humans have been conditioned to seek a controlling body or authority for nearly all of the phenomena that we experience in life, for that is indeed how most modern societies have been organized. The response to the questions: "Who's in charge?", "Who controls this?", or "Who owns it?" is generally, "some large firm", "the government", a government department or ruling institution. In the case of many of the phenomena of the observed on the Internet, there is indeed no one in charge. They are like giant flocks of birds or schools of fish. The response to questions such as: "Who owns them?" or "Who's in charge?" is either "We all do", or "No one does". These are great mechanisms for democracy, but their effects can also be anarchic, and society may have to develop new ways to deal with these liberating effects.

The effect of the flock-of-birds phenomenon is that access is equalized, unlike what occurs in traditional media. In a very real sense, no one has a "better right of access", and no one, even the largest corporation, can shout louder. The smallest player, the individual, has a right and opportunity to be seen and heard. Furthermore, many laws designed to regulate a physical world do not work as effectively when no one owns or controls the medium.

The fish-tank phenomenon

Moore's Law and Metcalfe's Law combine to give individuals inexpensive and easy access to new media such as the Internet. This means that any one can set up a website and theoretically at least, be seen by the world. As a result many have noticed the so-called “fish-tank phenomenon”6. The fish-tank phenomenon is named after the fact that in the early days of websites, people used to put a video camera on top of their tropical fish tank (or coffee percolator, for that matter), so that when you logged on to their site that is what you saw. This added to the clutter and junk on display—today there are hundreds of thousand of silly, futile, "junk" sites that only do something silly—let the viewer build a cow, tickle Elvis Presley’s tummy, cure their addiction to lip balm, or whatever. The question that this prompts is, wouldn't it be better if, rather than relying on individuals for their input on the Net, we depended instead on the considerable resources of large institutions and corporations?

The answer to this question really lies in another: what is more profound? And the answer to this second question is that, actually, it is the creative inputs of millions of individuals, all over the world who now have the ability to show us what they can do. In other words, the creative outputs of millions of individuals will beat the doings of large institutions in a great majority of cases. So, while we may see lots of junk, such as fish tanks, coffee percolators, and devices to tickle a long dead rock artist, every now and then some individual (probably a seventeen- year-old in his or her bedroom) is going to produce something so revolutionary that it will change our world. For strategists this means that many firms may find themselves threatened by small start-ups that were previously unable to get access to the market. No longer will it be good enough to merely observe one’s close and known competitors, in the future these competitors could be anyone and anywhere. They may be difficult to see before it is too late—they usually fly under the radar.

How the new five forces work in industries and markets

The music industry (see Exhibit 58) and Betfair (see Exhibit 59) represent classic cases of the five technology-related forces, which when working in concert, cause radical change in industries and markets. They also illustrate how astute entrepreneurs have utilized the five forces to uncover immensely valuable opportunities in business to customer systems. Observers of industries diverse as music, online betting, and telecommunications (e.g. Skype) will immediately recognize the generalizable parallels between these innovations and Betfair, and the applicability of the five forces. An examination and comparison of the music industry and Betfair enable an intriguing examination and extension of these forces.

Exhibit 58.: The global recorded music industry

For many years—indeed since Edison’s invention of the phonograph in 1877—the music recoding industry did not change that much. While technological changes did come to the market for recorded music over the years, in the form of improved recording techniques, hi-fidelity stereo, and the advent of the compact disk (CD), essentially the industry remained stable, with its structure largely unaffected by technological developments. Recording companies found and recorded talent and marketed it, and the products of the industry—essentially disks and cassette tapes—were distributed through record stores. Artists were remunerated in the form of royalties, retailers in the form of margins, and the record companies kept the rest.

The fundamental distribution issue of assortment was (and in many ways still is) perhaps the most significant dilemma in the market for recorded music. The structure of the industry and the way the product was produced held an inherent problem for both the retailer and the consumer. The retailer’s predicament is that of inventory—

the need to hold very large stocks of records, in order to provide a selection to customers, and to be able to make available to the customer the one that they will choose when they want it. This means that a lot of capital is tied up in stock, much of which moves slowly and often needs to be discounted in order to meet working capital requirements. The consumer is also in a quandary: Will the particular retailer stock the one album that they are looking for? And, will they be able to find it among the thousands of other items? Even once found, the consumer's problems are not over—there may be 12 songs on the album, and they may only really want three or four. But they are forced to purchase the entire album with all 12 songs.

Exhibit 59.: The market for sports betting

Players wishing to have a wager on a football game, or a flutter on a horse race, or to back up their opinions on a political election or the outcome of the Oscars, have until recently had few alternatives. In a majority of European countries, Canada, and most states in the USA their only option would have been to place a bet on a pari-mutuel or totalisator system, while in countries such as the United Kingdom and Australia, and in US states such as Nevada, they would also have access to licensed bookmakers. Both of these systems place the player at a significant disadvantage, not least of which is the fact that the "rake-off" or house percentage is considerable. This means that winners get paid well below the "true" odds against their choice. Furthermore, neither of the two systems allows a player to pick a "loser"—the player can only stake on a winning outcome. In simple terms a player cannot back team to lose—they can only back the other team to win or on a draw. A specific disadvantage of pari-mutuel systems is that subsequent weight of money for a player's choice will reduce the payoff, so that there is no opportunity to exercise any skill in the timing of a bet. Both systems profit not from the losers (as most inexpert gamblers believe) but from winners, by paying them out at less than true odds. In the case of pari-mutuels the house percentage is around 20 per cent, and even the most generous bookmakers make books that have an edge of around 14 per cent in their favor.

Betfair (www.betfair.com) which commenced trading in 2002 is the world's largest betting exchange. As a business, Betfair has no interest in the outcome of any event it makes available to gamblers. It simply provides a market for opinions and for trades to take place. It requires players to make and/or take fixed odds and all income is derived from a small percentage commission (ranging between two and five per cent, depending on a player's turnover) on a player's net winnings on an event. In general terms, the greater an event's turnover, the more revenue and profit Betfair generates, although Betfair's income is strictly a function of the total net winnings on an event, not turnover.

How Moore's law affects music and gambling

In the music industry, many consumers can now afford a relatively powerful computer, and use it as a device for recording and playing recorded sound. The cost of storage media has declined exponentially—multi-gigabyte hard drives that would have exceeded the storage capacity of entire countries just a few years ago are now so inexpensive that individuals can use them not just to store complete music collections, but to carry around in their pockets. The success of Apple's iPod is testimony to this. The hugely popular devices—some as large as 60 gigabytes in capacity—have become fashion icons. No one thinks of them as "computers" with "hard drives".

It would be fair to say that without the effects of Moore's Law, Betfair would not exist. On the one hand, just ten years ago, the computing power required to process, manage and store the millions of rapid transactions Betfair handles each day (Betfair processes more transactions each day than Visa) was simply unavailable. On the other, Betfair relies on the fact that its customers also have access to considerable computing power, in the form of affordable and easy to use laptop and desktop computers, and that these have access to the Internet. A generation ago, only pari-mutuels and the very largest bookmaking firms had access to computers, and the ordinary player had none.

Metcalfe's Law—networks in music and wagering

Music consumers find that purchasing or obtaining music online suits them far better than acquiring hard copy music products through traditional retail outlets. Napster, the first major advance in music downloads provided a free music sharing service that enabled consumers to choose and download only those songs they wanted. Its main problem was that it relied on one central exchange to facilitate the exchange of music, and at its peak its network consisted of more than 20 million users. The fact that it had a centralized server (or in network terms, one main hub) meant that it made an easy target for the organized music industry in its legal battle to close Napster down. Apple's iTunes service sells music (legally) under a similar system, and charges around USD 1 per song, grossing over USD 1 million daily.

However, users no longer need to rely on centralized distribution or even central servers for their music—the Internet is a huge network of distributed power, which connects anyone to anyone else in a very short time. Nowadays services such as Morpheus permit users to share millions of files (including movies and pictures as well as music) daily (although the legality of these exchanges is questionable). The first of these distributed network systems was Kazaa, the brainchild of Niklas Zennström and Janus Friis. After numerous legal battles with the recording industry, these entrepreneurs used their peer-to-peer technology to create Skype, the Internet telephony network. Nowadays Skype exceeds an average of 9 million daily users worldwide. The network effects are immense —users encourage friends and family to download Skype so that they can all engage in free communication. Skype was acquired by eBay in 2006 in a multi-billion dollar deal.

Metcalfe's Law is a very real factor explaining Betfair's success. The fact that more than fifty thousand players use the site to place and lay bets each day means that the likelihood of any individual player finding a match for what they want (assuming that what they are asking for is reasonable) is very high. This of course results in highly efficient and very liquid markets, which are to the advantage of all players.

Networks are important because they create short cuts. Anyone who is part of the network is by definition in contact with anyone else who is part of it, and can therefore bypass more traditional channels and structures (such as conventional bookmakers and totalisators in this case). As networks grow, their utility increases, so Metcalfe's Law tells us—this is true for those who are part of the network, and for those who then choose to join it. Neither traditional bookmakers nor totalisators enjoy the same network advantages as Betfair: bookmakers only have contact with players in their geographic vicinity or those with telephone access. Totalisators are on tracks only in some countries, where they will be affected by liquidity problems on quiet days, and even when they are not, they are restricted by local geography. Betfair faces none of these restrictions, and can enable the world to wager.[70]

70 While this is true in principle, in reality there are some exceptions. For example, wagering online with sites outside of its jurisdiction has been outlawed by the US government. While many other countries decry this as just another example of American government bullying and in blatant disregard of all of the initiatives on free global trade, the US government has been ruthless in pursuing this policy. It has arrested executives of foreign online wagering operations on entry into the USA;

Coasian economics: transaction costs in online music and wagering

Hierarchies such as large music companies and their distribution networks are no longer able to achieve the lowest transaction costs in the recording industry. The low costs of communication and distribution have made the market more efficient, and millions of individuals benefit as a result. After the invention of the tape recorder, copying music became technically feasible, but of course the network effects were very limited: the song had to be played while it was being copied, and distribution was limited to a few local friends who had tape players. Computers and the Internet nowadays mean that a song(s) can be copied very quickly, and distributed, theoretically (if not always legally) at least, to a multitude of other users in a few seconds.

When contrasted with traditional bookmaking firms, Betfair is a fine example of the potential of low cost, networked computing to make markets more efficient by lowering transaction costs. Traditional bookmakers need to be well informed (studying horse racing and sporting events carefully) in order to make odds, and need to monitor market changes constantly in order to avoid being taken advantage of or of being over-exposed. Betfair simply provides the platform for many individuals (theoretically, “firms of one”) to do this for themselves. These individuals do not have to rent space or equipment, advertise, or employ staff. With a click of a mouse they can essentially either do what Ladbrokes or William Hill does—or, if the fancy takes them, play their fancies.

The flock-of-birds phenomenon: lawlessness in music and gambling

The exchange and distribution of recorded music has become very difficult to legislate and control. While it eventually became possible for the Recording Industry Association of America (RIAA), after extensive legal and political efforts to take action against and eventually close Napster, it is unlikely that it can have the same results against millions of individuals all over the world. When it has attempted to pursue these, it has come out in publicity looking simply like a blundering bully. Also, just because it may succeed in one country (e.g. the USA) does not mean it will succeed in another with a different legal system. Technologies like Gnutella, Kazaa and Morpheus made this even more unlikely and complex. Commenting on the RIAA’s attempts to block Napster, Charles Nesson (cited in 7), professor at the Berkman Center for Internet and Society, Harvard Law School had this to say of Gnutella: “There is a generation of young people out there who have already learnt that music is something you get on the Net, rather than buy. The only way for the music industry to stop Gnutella is to turn off the Internet”. He added: “And, as no one owns it or controls it, that is impossible.”

The case of Betfair has seen some interesting reactions from incumbents and also governments at various levels. Traditional competitors have attempted to cry, “no fair”, as they struggle to contend with a player whose methods they do not fully understand. Sports bodies are anxious to understand the possible impact of a firm such as Betfair on the potential for financial abuses in their domains. Governments, while supposedly welcoming competition as a force in consumer interest, will immediately seek new ways to tax new entrants such as Betfair, for it seems like, yet is not like, traditional firms such as bookmakers and totalisators.

In the United Kingdom, Betfair has been the focus of lawsuits by the major British bookmaking firms such as Coral, William Hill, and Ladbrokes. They have charged that by allowing individuals to lay bets (rather than merely take them), they are acting as bookmakers, and are not licensed to do so. Graham Sharpe, spokesman for William Hill argues: "If you have a bet on an exchange, you don't know who it's with; if [the person] is offering extravagant odds, you don't know why."

---

these include Canadians and UK citizens.

The British courts have thrown out all the cases and rejected the bookmakers' arguments, and Betfair has been allowed to proceed with its business. While bookmakers might not like the idea of the site, it is evident that many of them are using it, either to buy back bets at advantageous prices, or to lay bets for which they might otherwise not have been able to find takers. The substantiation for this is the significant amounts of money that are available to be traded on many events.

In Australia, not only have bookmakers objected to the site and attempted to close it down through legal action, it has also been the focus of an aggressive advertising campaign by TABCORP (essentially a consortium of Totalisator operators), a listed company that is the world's fourth largest gambling and entertainment business. Its advertising has attempted to cast Betfair in a negative light by claiming that betting on exchanges encourages dishonesty in sporting events and racing, a similar argument used in the United Kingdom by bookmakers. Betfair has disabled certain features on its Australian site to comply with Australian gambling legislation (for example, Australian players are not able to bet "in the running"—that is, after an event has started or a race is running). Gambling is more a matter of state than federal legislation in Australia. While the legislation is not quite clear whether it is legal or not for Australians to bet on betting exchanges or online casinos outside of Australia, there is no legislation prohibiting players in one state betting on operations in another.

The real effect of the flock-of-birds phenomenon is that access is equalized by mechanisms such as Betfair when they operate online, unlike what occurs in traditional operations such as bookmakers and totalisators. In a very real sense, no one has a "better right of access", and no one, even the largest corporation, can shout louder. The smallest player, the individual, has a right and opportunity to be seen and heard. Furthermore, many laws designed to regulate a physical world do not work as effectively when no one owns or controls the medium. This has been apparent in the lack of success in the traditional incumbents' attempts to fight Betfair in courts of law.

It has also been demonstrated by Betfair being a better, not less effective, mechanism for the detection of dishonest dealings in sport. If a particular player has inside information on an event, there is nothing stopping them from exploiting this advantage by placing large cash bets either with bookmakers or totalisators, and reaping the benefits of this insider trading anonymously. Betfair is arguably in a better position to deal with this type of problem. For example, in July 2004, Betfair was dragged into the spotlight when it reported suspicious betting patterns on its exchange to the Jockey Club in the United Kingdom just before the Lingfield race in which leading jockey, Kieren Fallon, riding favorite Ballinger Ridge, lost to Rye after seemingly easing down before the finishing line. News of the World alleged Fallon had told an undercover journalist that Rye would win. No proof was found that the race was fixed, but a Betfair spokesperson was quoted as saying: "We are putting a searchlight on the sport and helping it clean up its act. There is a clear paper trail on our site that doesn't exist in high-street [betting] shops. We are entirely transparent. We have no vested interest in the outcome of a horse race."

As observed, there is also the issue of how governments will attempt to tax firms such as Betfair. Ideally, from the firm's point of view, tax would best be levied on net profits (which in simple terms would amount to net commissions on winnings less other direct costs). However, governments might not see it in that way, and might attempt to tax market makers such as this based on other measures, such as turnover. Such a measure could be very detrimental to the firm and its thousands of players, because turnover is not an accurate measure of financial performance.

The fish-tank phenomenon: the power of creative individuals in music and wagering

While there are a lot of junk as well as stupid schemes placed on the Internet, there are also an incredible number of good ideas developed by individuals (often a teenager in their bedroom) that can now finally see the light of day through this medium. In the latter half of 1998, the ground shifted in the music business when MP3 arrived on the Web. MP3 is short for Moving Picture Experts Group Audio Layer III or "MPEG3", and is a compression format that shrinks audio files with only a small sacrifice in sound quality. MP3 files can be compressed at different rates, but the more they are scrunched, the worse the sound quality. A standard MP3 compression is at a 10:1 ratio, and yields a file that is about 4 MB for a three-minute track. MP3 started life in the mid-1980s, at the Fraunhofer Institut in Erlangen, Germany, which began work on a high quality, low bit-rate audio coding with the help of Dieter Seitzer, a professor at the University of Erlangen. In 1989, the Fraunhofer Institut was granted a patent for MP3 in Germany and a few years later it was submitted to the International Standards Organization (ISO).

In 1997, a developer at Advanced Multimedia Products created the AMP MP3 Playback Engine (essentially a piece of software that plays MP3 recordings), which is regarded as the first serious MP3 player. Shortly after the AMP engine was made available on the Internet, two university students, Justin Frankel and Dmitry Boldyrev, took the AMP engine, added a Windows interface and dubbed it "Winamp". In 1998, when Winamp was offered up as a free music player, the MP3 craze began: music enthusiasts all over the world started MP3 hubs, offering copyrighted music for free. Before long, other programmers also began to create a whole toolset for MP3 enthusiasts. Search engines made it even easier to find the specific MP3 files people wanted, and portable Walkman-size players like the Rio let them take MP3 tracks on the road after first downloading them on to a computer hard drive and then transferring them across.

When Napster became available on the Internet in 1999, it allowed anyone with a connection to find and download just about any type of popular music they wanted, in minutes. By connecting users to other users' hard drives, Napster created a virtual community of music enthusiasts that has grown at an astonishing pace. Developed by a twenty-year-old student named Sean Fanning, Napster boasted some twenty million members throughout the world by the end of 2000.

The common thread through all of the above online music history is the absence of any major, for-profit recording company in any of the technological developments. Indeed, the only role played by any of the incumbents was that of stifling, or attempting to stifle, progress. The major innovations came from academic institutions, students, and penniless enthusiasts whose only real resources were talent, persistence, creativity.....and an Internet connection. Sean Fanning was a young, not very wealthy student, and not in the research department of a major recording company. A firm's next serious competitor might not be a multinational conglomerate, but an individual operating from home. This individual now has a unique mechanism for bringing good ideas to market.

Similarly, Andrew Black invented Betfair and changed sport wagering forever. He never worked for a government totalisator or pari-mutuel agency, nor was he availed of the significant resources of one of the major traditional bookmaking firms. He was simply a very talented individual, frustrated that trying to pick a winner was difficult enough without having the odds truly stacked against the player. There had to be a better way—and that way was through the Internet. Neither Sean Fanning nor Andrew Black had previous experience of the industries that they changed forever, and neither had worked for any of the established incumbents previously. The incumbents' only action was to try to smother the innovations that they saw as threats, rather than opportunities at

best, or at worst, the writing on the wall. Recent evidence is that their attempts at suppression have failed miserably.

Summary

In the future, firms may still need to consider Porter's 5 forces—but they will be a very different set of forces, if they are to uncover the opportunities that networks present in business to customer markets. There will be the technological effects of Moore's and Metcalfe's laws, hyper-accelerating change and spreading it like a deadly virus. There will be the contradictory effects of transaction cost economics, not only making firms smaller, but virtual too. There will be the societal effects of the flock-of-birds phenomenon bringing undreamed of democracy along with the threat of anarchy. And the fish-tank phenomenon brings access to all.

Michael Porter2 argues that his five forces determine the attractiveness of an industry, which in turn influences strongly the profitability of firms within that industry. The five new forces of the information age are more ethereal and impact on firms and industries in ways that are far less predictable or structured. To use the new five forces astutely the decision maker must depend on them not so much as guidelines and prescriptions, but as prods from behind to keep challenging oneself, one's firm and one's market. When this is done effectively, it is likely that many opportunities will raise their heads.

The technological forces of Moore's Law and Metcalfe's Law accelerate change not only within a firm, but also within industries and markets, and this acceleration tends to be exponential. The decision maker must consider what will happen when computer chips are not just in computers, but also in every device and product, and what will happen when these computers, like all computers, in their turn become part of an exponentially growing network.

Transaction cost economics, and technology's effects on the efficiency of firms and markets means that the manager must constantly reflect on what will happen to the shape and size of the firm. The decision maker must continually evaluate what activities the technology will allow to be performed in the market, and what functions may indeed be brought back within the firm itself. In channels of distribution, managers will have to observe the constant tussle between disintermediation and reintermediation. In the case of the former, many traditional intermediaries will disappear from channels as their roles are either usurped by technologies, or performed more efficiently by other channel members. Already the Internet is having a profound effect on institutions such as travel agents and financial brokers, and the long-term impact of online music on conventional record stores is obviously of great concern to those institutions. In example after example of reintermediation, we are seeing new intermediaries enter channels using technology to improve the channel's efficiency while taking a share of the margins available in the channel for themselves. Online consolidators such as Priceline.com in the travel industry, and Autobytel.com in the channel for new and used cars are prime examples of this.

The social forces of the flock-of-birds phenomenon and the fish-tank phenomenon will require managers to work in a new environment where control and governance are not as structured and clear as they have been throughout most of our lives. Managing in a world where significant issues are not really within the control of a government or a government department, or under the remit of a large organization will be a new and often scary experience for most executives. Not knowing where competition may come from, because it may not be upfront and visible will also require a constant revisiting of strategy. When competition comes head on, or at least from the side

or from behind, it can be seen, and dealt with, even if slowly. When competition has the potential to come from a computer in the bedroom of a seventeen-year-old in another country, life becomes less predictable.

Many managers may take cold comfort from an identification of the five new forces, and what they will do to the business environment. They are not neat and structured, like Porter's 5 Forces, nor do they seem to suggest much in terms of strategic direction, as do popular analysis tools such as the Boston Consulting Group grid. Much of the recent writing on strategy emphasizes the effects of these forces however, and suggests that conventional approaches to strategy will at least be insufficient, if not ineffective, for coping with corporate survival. A number of these authors (cf. 4; 9; 10; 11) offer perspectives that are worth considering. While, as would be expected, there is no absolute concurrence on their advice for strategy in the future, these authors do tend to agree on certain fundamentals.

In closing, it is worth summarizing some of these basics. First, change is too rapid for anyone anywhere to feel comfortable—success has an anesthetizing effect that becomes its own enemy. Second, it may be a good idea to continually seek ways of destroying one's own firm's value chain and putting oneself out of business—if one does not, someone else will in any case. Third, resources are increasingly less about tangible assets and more about knowledge and the ability to constantly innovate. Fourth, firms should constantly find and exploit ways to give the customer as much of an opportunity to do as much of the work as possible. Technology offers great opportunities in this regard. Strangely (and as we pointed out in the first chapter), customers do not want more service, they want less. They want control, and the power to solve their own problems, and victory will go to those players who find ways for them to do this well12. Finally, strategy is no longer long-term, as the half-life of ideas diminishes. The five-year plan or the long-term strategy is no longer viable, and the value of the annual strategic planning session is to be questioned. Strategy becomes incremental, rather than planned. It is revisited and revised not annually or even bi-annually, but monthly, probably weekly, and possibly, daily. It's not that strategy should be thrown out with the bath water. Rather, perhaps, managers should consider that the strategy needed for the 21st century might indeed be a new baby, born of five new forces in an age of convergence. And in an age of opportunity.

For many years—indeed since Edison's invention of the phonograph in 1877—the music recoding industry did not change that much. While technological changes did come to the market for recorded music over the years, in the form of improved recording techniques, hi-fidelity stereo, and the advent of the compact disk (CD), essentially the industry remained stable, with its structure largely unaffected by technological developments. Recording companies found and recorded talent and marketed it, and the products of the industry—essentially disks and cassette tapes—were distributed through record stores. Artists were remunerated in the form of royalties, retailers in the form of margins, and the record companies kept the rest.

The fundamental distribution issue of assortment was (and in many ways still is) perhaps the most significant dilemma in the market for recorded music. The structure of the industry and the way the product was produced held an inherent problem for both the retailer and the consumer. The retailer's predicament is that of inventory—the need to hold very large stocks of records, in order to provide a selection to customers, and to be able to make available to the customer the one that they will choose when they want it. This means that a lot of capital is tied up in stock, much of which moves slowly and often needs to be discounted in order to meet working capital requirements. The consumer is also in a quandary: will the particular retailer stock the one album that they are looking for? And, will they be able to find it among the thousands of other items? Even once found, the consumer's

problems are not over—there may be 12 songs on the album, and they may only really want three or four. But they are forced to purchase the entire album, with all 12 songs.

Players wishing to have a wager on a football game, or a flutter on a horse race, or to back up their opinions on a political election or the outcome of the Oscars, have until recently had few alternatives. In a majority of European countries, Canada, and most states in the USA their only option would have been to place a bet on a pari-mutuel or totalizator system, while in countries such as the United Kingdom,Australia and in US states such as Nevada, they would also have access to licensed bookmakers. Both of these systems place the player at a significant disadvantage, not least of which is the fact that the "rake-off" or house percentage is considerable. This means that winners get paid well below the "true" odds against their choice. Furthermore, neither of the two systems allows a player to pick a "loser"—the player can only stake on a winning outcome. In simple terms a player can't back team to lose—they can only back the other team to win or on a draw. A specific disadvantage of pari-mutuel systems is that subsequent weight of money for a player's choice will reduce the payoff, so that there is no opportunity to exercise any skill in the timing of a bet. Both systems profit not from the losers (as most inexpert gamblers believe) but from winners, by paying them out at less than true odds. In the case of pari-mutuels the house percentage is around 20 per cent, and even the most generous bookmakers make books that have an edge of around 14 per cent in their favor.

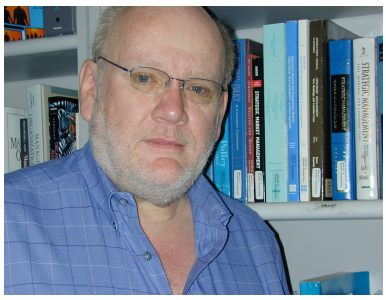
Betfair (www.betfair.com) which commenced trading in 2002 is the world's largest betting exchange. As a business, Betfair has no interest in the outcome of any event it makes available to gamblers. It simply provides a market for opinions and for trades to take place. It requires players to make and/or take fixed odds and all income is derived from a small percentage commission (ranging between two and five per cent, depending on a player's turnover) on a player's net winnings on an event. In general terms, the greater an event's turnover, the more revenue and profit Betfair generates, although Betfair's income is strictly a function of the total net winnings on an event, not turnover.

Exercises

1. Explain in your own words what is meant by a "killer app", and find some examples not discussed in the text.
2. Identify five more examples of the effects of Moore's Law in products or services not already discussed in the text.
3. What is mean by "transaction costs"—explain these by using practical examples from an existing organization.
4. It has been said by many observers that Google is the most important new firm for the new millennium. Is Google an example of the new five forces at work? Explain how the forces apply to Google.
5. Do a "new five forces" analysis for the organization you work in or one that you are very familiar with. List ways in which the forces apply to this organization, and how they represent either opportunities or threats.

Editor

Leyland Pitt



References

Dickson, P. R. (1992) Toward a General Theory of Competitive Rationality, Journal of Marketing, 56, 1, 69—84

Porter, M. E. (1998) Competitive Advantage: Creating and Sustaining Superior Performance, New York, NY: Free Press

Porter, M.E. (1980) Competitive Strategy: Techniques for Analyzing Industries and Competitors. New York, NY: Free Press.

Downes, L., and Mui, C. (1998) Unleashing the Killer App, Boston MA: Harvard Business School Press

Coase, R.H. (1937) The Nature of the Firm, Economica, 4, 386–405

"The Accidental Superhighway", (1995) The Economist, July 1, special supplement

For a full description see Pitt, L.F. (2001) "Total E-clipse: Five New Forces for Strategy in the Digital Age", Journal of General Management, 26, 4 (Summer), 1- 15

For a full description see Davies, M., Pitt, L.F., Shapiro, D., and Watson, R.T. (2005) Betfair.Com: Five Technology Forces Revolutionize Worldwide Wagering, European Management Journal, 23, 5 (October), 533-541

Kelly, K. (1998) New Rules for the New Economy: 10 Radical Strategies for a Connected World, London, UK: Fourth Estate

Shapiro, C. and Varian, H. R. (1998) Information Rules: A Strategic Guide to the Network Economy, Boston MA: Harvard Business School Press

Schwartz, E. (Lauren Marino, ed.) (1999) Digital Darwinism : Seven Breakthrough Business Strategies for Surviving in the Cutthroat Web Economy, New York, NY: Broadway Books

Berthon, P.R., Pitt, L.F., Katsikeas, C., and Berthon, J-P. (1999) "Virtual Services Go International: International Services in the Marketspace", Journal of International Marketing, 7, 3, 84-105

11. Opportunities in business to business systems

Editors: Maha Shakir (Zayed University, United Arab Emirates) and Guy Higgins (New York Institute of Technology-UAE Campus, United Arab Emirates)

Learning objectives

- describe what a B2B system is and why such systems are important in the modern economy
- define the concept of integration in terms of information systems and discuss why it is important to the modern business organization
- describe what an ERP (or enterprise) system is and why such systems are important in the modern economy
- define the concept of sSupply cChain in terms of external integration and discuss why it is important to the modern business organization
- describe several technologies (EDI, E- Marketplaces, Web Services) in current use in B2B systems
- discuss both the opportunities arising from the adoption of B2B systems and the challenges to their implementation

Introduction

Information systems provide many opportunities for businesses to improve the efficiency and effectiveness of their operations through the integration of business systems. While the majority of information systems in the past were focused on applications within the boundaries of the enterprise, the focus is gradually shifting outwards. Business-to-business (B2B) systems are a part of those systems that are applied to relationships outside of the boundaries of the enterprise.

Ongoing innovations in Web technologies are making the integration of business systems across companies (i.e. outside the boundaries of an individual company but forming its supply chain) technically possible and financially feasible. Managing the B2B aspects of these supply chain relationships is creating substantial opportunities, both in the streamlining of operations and in the development of new and innovative business delivery mechanisms. These opportunities are the focus of this chapter.

This chapter explores the importance of system integration (both internally and across the supply chain), provides a brief history of B2B systems, introduces the types of information technologies enabling these systems, and identifies the challenges to their adoption. New business models that are enabled utilizing these technologies are also discussed.

What is integration and why is it important?

Although this chapter is about B2B systems, throughout the chapter we will be discussing integration. But what is integration and why is it so important? Integration simply means causing or allowing things to work together

cooperatively. Any business organization consists of a set of diverse people and systems, but the underlying premise of that organization is that all of its diverse elements will work smoothly together in order to meet its organizational goals in an efficient and effective manner.

If you consider a very small company, say a company consisting of only one person, then all of the functions of that company are integrated because that one person performs all of the needed functions of the company. Refer to Thirkettle and Murch [1] for an insight into the challenges pertaining to managing information systems in a oneperson company. There a many examples of one person companies that are very successful because that one person is good at doing whatever it is that the company does. Unfortunately, few people can do everything needed by a company well. Even if they could, the quantity of output of a small company is necessarily limited. One of the tenets of the modern organization is that, as the organization grows in size, its members tend to specialize in specific functions. While this specialization allows each of those members to do their specific jobs better (as they are more focused on their individual tasks), it also decomposes what was originally a set of integrated tasks into a chain of separate, but related, functions. Thus, while functional specialization (i.e. decomposition) allows the various components of an organization to act more effectively, it increases the possibility of friction between those components. This friction would tend to reduce the efficiency of the organization. Making this chain of separate, but related, functions work cooperatively (or to reduce their friction) is the function of the executive management of the company. Finding ways to enable the people in an organization to cooperatively work together is a part of the study of Management. Finding ways to cause the information systems of an organization to cooperatively work together is a part of the study of Management Information Systems, or the topic of this book.

Unfortunately, the information systems of most organizations are not a set of programs that work together cooperatively; in other words, these systems are NOT integrated. Instead, in most organizations they are a collection of disparate systems which were individually developed over time to meet specific demands within the organization. This happened for many reasons. Certainly it is easier to develop an information system intended to provide for a limited number of demands rather than to provide for all of the needs of a company. Additionally, in the early days of IT the available technology was simply not robust enough to attempt what we now call integrated systems. Today, we call this collection of disparate systems, which are found in most companies, legacy systems. These systems are our legacy from those systems developers that went before us in the organization. Typically, these legacy systems do the jobs for which they were designed extremely well; after all, they have been used, tested, and improved over the years. Legacy systems were typically designed to support only one business function and are often called functional systems as they were not intended to be integrated with other systems within the organization. Any needed integration would be provided by people taking the output of one system and adding it as input to another system. For example, the accounting department usually has one or even a collection of legacy systems which support accounting operations. However, the same information would not necessarily automatically flow through this set of systems that support the accounting operations.

The problem with legacy systems is not with the jobs that they do, it is with their inability to cooperate with each other—they are not integrated. The lack of integrated systems requires that people work harder to provide the integration. They must manually enter one system's output as an input to another. This is the friction that we discussed earlier and it creates inefficiencies within the overall set of business systems of the organization. It also means that the reports that these systems generate only provide a snapshot of the results for the specific function(s) that the system was designed to support. This makes it extremely difficult to have up-to-date information about the

overall status of the organization. Having this up-to-date information, or in practical terms integrating the organization's information systems, has been the holy grail of executive managers and systems professionals for many years.

ERP systems: the means of internal integration

Integrating key business functions so that information can flow freely between them has become the job of a type of software application known as an enterprise resource planning (ERP) system or an enterprise system. Since the mid- to late-1990s, many organizations have been replacing their disparate legacy systems with these organization-wide ERP systems. Although originally focused on the internal functions of an organization, ERP systems are typically considered the most feasible solution for integrating business information systems within the organization. Therefore, a basic understanding of ERP systems is helpful in understanding the similar integration issues that organizations face nowadays in their pursuit of external business relationships.

An ERP is a packaged software application which includes a collection of software modules that are designed using best-practice business processes [2]. Each module may replace one, or even many, legacy systems in the company. The module meets the same demands met by the legacy systems that they replace, but they are also designed to integrate with the other modules with the ERP. An ERP is primarily responsible for managing the transactional processing operations of the business in its various areas (e.g. accounting, finance, human resources, marketing, manufacturing, logistics, sales, procurements, etc.). Each of these functional areas is usually supported by a software module (or modules) that is designed to integrate with the other modules; these packaged modules then need to be configured, and sometimes customized to meet the specific demands of the company implementing the ERP. ERP systems are available from vendors such as SAP, Oracle, and Microsoft.

Organizations have different drivers for implementing an ERP [3]. These drivers mostly fall into two main categories. The first is concerned with solving those existing business problems caused by inadequate IT infrastructure and disparate information systems (i.e. integrating existing operations). However, the second driver is related to improving future business operations. This could include support for future business flexibility and growth, reducing operational costs, supporting customer responsiveness, improving data visibility, and making better business decisions.

When implemented, an ERP often replaces the many existing legacy systems within the organization. Due to the integrated nature of ERP, its modules overcome many of the drawbacks of legacy systems and enable online integration not only within the same function but within and across the other functions of the business. As a result, ERP systems are considered good candidates for forming a technology platform to support the integration of other intra- and inter-organizational applications such as supply chain management (SCM), customer relationship management (CRM), and e-commerce (refer to Exhibit 60).

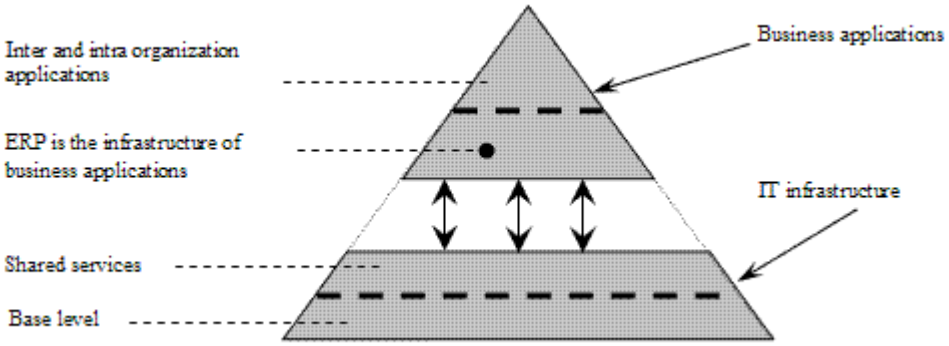


Exhibit 60: ERP as a platform for business applications (Adapted from Broadbent and Weill [4])

In the early-2000s, the high-end of the ERP market became saturated because most large organizations had already implemented an ERP. In response to the increased competition in the ERP applications market, ERP application vendors started including other applications as part of their ERP offerings. In order to achieve this, ES vendors built the new functionalities in-house, and/or acquired, or made partnerships with, specialized enterprise application vendors. ERP systems are gradually evolving to become inter-organizational and Internet-enabled. New modules are added to the product portfolio, such as supply chain management (SCM), customer relationship management (CRM), data warehousing, and business intelligence.

The supply chain: the focus of external integration

Every business is made up of a complex set of relationships between the business, as a producer of goods or services, and its suppliers and customers. This set of relationships is called a supply chain (refer to Exhibit 61).

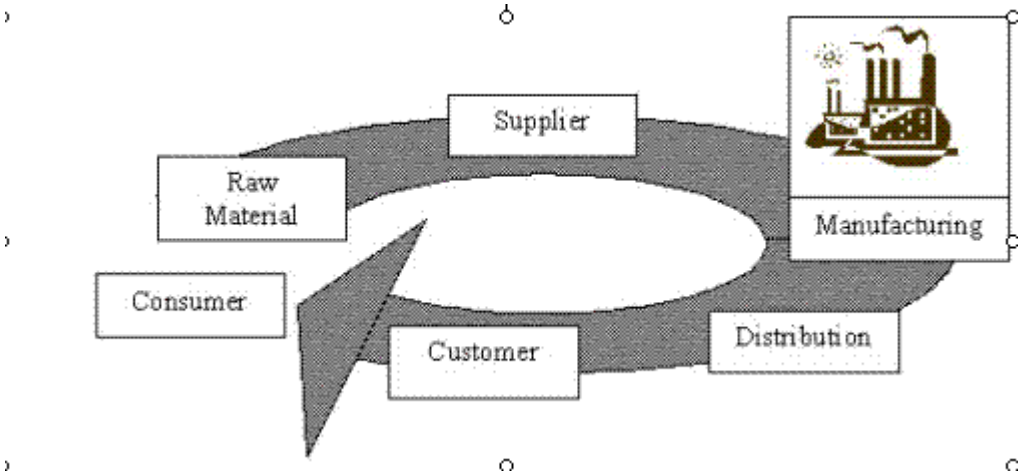


Exhibit 61: Supply Chain

It is from within its supply chain that a business must first be able to procure its raw materials and other necessary means of production from its suppliers. This requires that the business be able to communicate what it needs to those suppliers. Suppliers who have the materials that meet those needs must then offer them to the business. Then, following its evaluation of the available offers, the business must actually purchase the materials from its preferred supplier. This communication of needs, offers, and purchases results in much paperwork—requests for quotation (RFQs), quotations, purchase orders (POs), etc. The purchased materials must then be delivered to the business' place of production. Hence, more paperwork is generated by the actual delivery of the

purchased material—bills of lading, receipt documentation, invoices, and finally payment. All of this paperwork is both time consuming and expensive to all of the organizations involved.

The other end of a business' supply chain shows its relationships with its customers to whom it sells its products. Once the goods of the business have been produced, yet another set of paperwork is generated with sales orders from the customer, sales receipts to the customer, shipping documents and invoices to the customer, and finally, payment from the customer. There is additional expense in generating all of this paperwork, but paperwork is just another one of the costs of the production of the goods or services of the business.

Successful businesses generate wealth by identifying opportunities to produce desired goods and/or services at prices that will make it possible to sell those goods and/or services to enough customers to produce a profit. One way that businesses can increase their profits is to reduce the costs of production of their goods and/or services. Thus, businesses must constantly monitor and, when possible, improve their internal processes in order to identify opportunities to reduce their costs of production. This is the application of microeconomic, or transaction cost theory, analysis to the operations of the business. The application of transaction cost theory to reduce production costs was the impetus for first automation and, more recently, the implementation of ERP systems. But a business must also monitor and improve its external relationships, in both directions, within its supply chain. B2B systems include those efforts to monitor and improve the firmís external relationships and to integrate the processes involved in supporting these relationships.

History of B2B systems

Throughout most of our history, cities have been the generators of opportunity and wealth. In her seminal work of 1984, *Cities and the Wealth of Nations*, Jane Jacobs argued that “economic life depends on city economies;” because, wherever economic life is developing, the very process itself creates cities'[5]. Thus, economic activity requires the bringing together of the producers of goods and/or services, their suppliers, and their customers. Historically these groups are brought together in physical proximity—this is called agglomeration. This agglomerative process resulted in the development of cities throughout the world. This was true because with the limited communications and transportation systems of earlier times, businesses generally needed to be physically near to either their suppliers or their customers, and preferably near to both. But the last quarter of the twentieth century saw the decline of the industrial output of many of the world’s great cities and the transfer of this production to other parts of the world. This was possible as improvements in transportation and communication systems began to counter-balance the need to for businesses to be co-located in cities. However, the trend towards greater separation between suppliers, producers, and customers (also called globalization) has increased the volume of paperwork, and its attendant expense, needed to produce and sell goods and services. If you are importing raw materials from one country, producing your goods in another country, and selling your product around the world, you can no longer simply walk across the street to talk to your supplier about what you need or to deliver your product to your customer (refer to Exhibit 62).

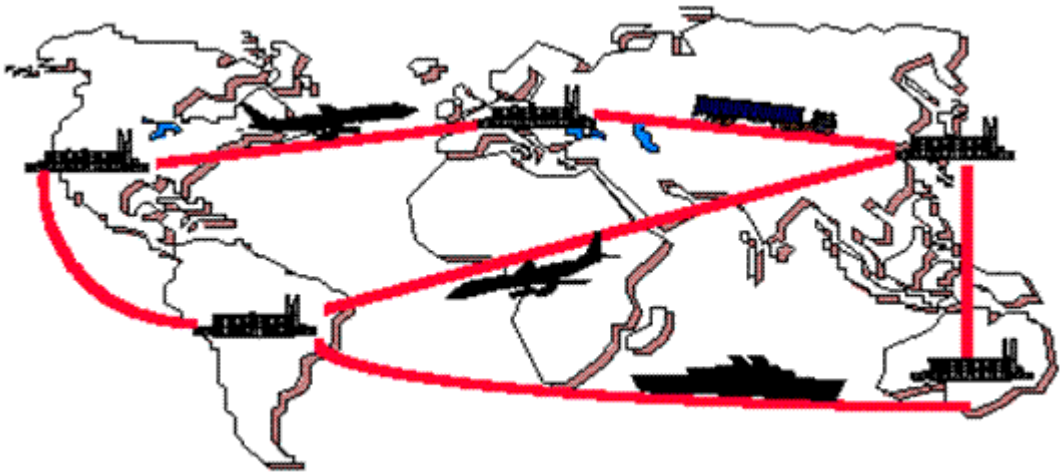


Exhibit 62: Global supply chain

Traditionally, the method of exchanging data between members of a supply chain has been through the use of paper-based systems. As discussed, paper-based systems are inefficient because they are both slow and expensive to maintain. B2B systems were first used to replace these inefficient paper-based systems with the electronically exchanged data. This was much faster, and because it involved less manual processing of that data, it was also much more cost-effective. Seen from an economic perspective, these early B2B systems improved a company's processes for ordering supplies and paying for the ordered supplies—this is a simple application of Transaction cost theory, where the efficiency of the individual transactions are improved in order to reduce the overall cost of production. But, in addition to efficiency improvements, modern B2B systems also provide increased opportunities for finding even better suppliers and for reaching out to more customers. Thus, modern B2B systems are also improving the effectiveness of a firm's supply chain.

The earliest B2B systems used proprietary systems which, while an improvement over the paper-based systems that they replaced, were still relatively expensive to operate. The advent of the Internet has led to new and improved technologies for B2B commerce that are less expensive and easier to implement. Thus, the current rapid growth in Internet-based B2B commerce is based on the twin facts that electronic exchange of data is more efficient than the traditional paper-based exchange and that Internet-based technologies are less expensive and simpler to implement. The arrival of Web services appears to promise an even closer integration of the corporate systems that run modern business organizations. This may call for new models of cooperation between members of a supply chain.

What is a B2B system?

As a part of the trend towards a more global economy, the final decades of the twentieth century saw the birth of electronic commerce (e-commerce). E-commerce, the buying and selling of goods and services over electronic channels, has become an exploding phenomenon. E-commerce can be divided into three major categories:

1. B2C—Business to Consumer: those aspects of e-commerce that involve Internet sales by businesses to consumers.
2. B2B—Business to Business: those aspects of e-commerce that involve the exchange of goods or services between companies over the Internet.

3. C2C—Consumer to Consumer: those aspects of e-commerce that involve Internet sales by consumers to consumers.

The B2B category involves the exchange of goods or services between companies with the receiving company intending to use the received good or service in the furtherance of its own production processes (not the final consumption of that good or service). Still, one must remember that all businesses are both the suppliers and customers of other businesses. B2B is thus involved in ALL aspects of the supply chain. Therefore, B2B systems are a part of the efforts of a company to monitor and improve its external relationships with the other businesses in its supply chain.

But e-commerce is not only about buying and selling in the global marketplace, it is about being aware of global opportunities and reducing the production and sales cost of the business. You may be more familiar with the B2C type of business, as you may have personally purchased something over the Internet. However, by far the greatest volume of e-commerce is found in B2B transactions. A recent study by the US Department of Commerce found that sales via e-commerce had a reported growth of 24.6 per cent for the year 2005 versus 2004 [6]. Another recent report from Forrester Research projects that the European Union's online B2B transactions will surge from the 2001 figure of 77 billion to 2.2 trillion in 2006—increasing from less than 1 per cent of total business trade to 22 per cent [7].

This is an information systems book; hence the B2B systems discussed in this chapter involve the combination of information technologies and the business processes that support the exchange of goods, services, information, or money between organizations. However, we must acknowledge those B2B systems that have existed long before computers or the Internet were invented. Such systems include those that were not originally technology-based, such as the postal system, the banking system, the accounting system, the legal system, the taxation system, etc. In their own times, these systems were considered exemplars of innovative means to facilitate the exchange of both information and goods between business entities. Nowadays, all of these systems utilize technology to a great extent. Other technology based-systems that fulfilled similar roles in the recent past include the telegraph, telephone, and television systems. Taking this perspective, we can say that B2B systems are not new but have evolved from the older manual or paper-based systems of the past. The main difference however is that information and communication technologies (ICT) are now key components of B2B systems. As a result, ICT have both accelerated the business processes they are supporting and reduced their costs.

B2B technologies

In this section we discuss three types of technologies that play an important role in B2B systems. Two of these, EDI and e-marketplaces, support inter-organizational integration. The third, Web services, is a new flexible and platform-independent method for integrating information systems. Similar to ERP systems, Web services can be used for both intra- and inter-organizational integration.

EDI

The history of electronic B2B systems starts with the electronic data interchange (EDI) systems which were first developed in the mid 1960s. Prior to the advent of EDI, when one company generated paperwork to be sent to another company, the receiving company would be required to retype that paperwork into the format that they wanted for their computer-based systems (refer to Exhibit 63). This would occur even if both the producing and the

receiving companies were using computer-based systems, because the means of transmitting of data between them was still that piece of paper that had been used for thousands of years for this purpose.

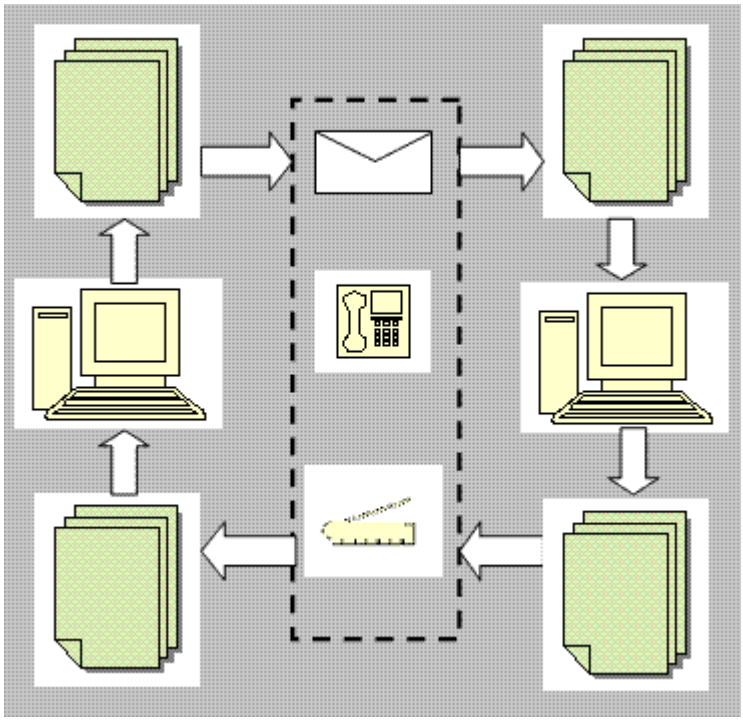


Exhibit 63: Traditional paper based systems

An EDI system facilitates a computer-to-computer exchange of data according to predefined standards. Thus, both the company generating the data and the company receiving the data must agree in advance as to the exact electronic format that the transmitted data will take. Only then, an electronic file can be exchanged between the two companies instead of paper. This type of data exchange greatly reduces the human labor, and therefore the cost, of exchanging data between companies. The typical use for EDI is in the procurement of goods and services. Most likely the exchange is between a big company and its suppliers (refer to Exhibit 64).

The advantage of EDI systems is twofold, efficiency and effectiveness. The first is associated with automation. As discussed elsewhere in this book, efficiency is a result of doing more with less. Electronic communications between the buyer and suppliers result in the elimination of paper-based documents and all the manual processes involved in handling, verifying, entering, sending, receiving, and recording these transactions, plus the reduction in time for completing many of the procurement processes involved. Additionally, accurate and timely information helps EDI partners improve the effectiveness of their collaborative business relationships, particularly when business processes are redesigned.

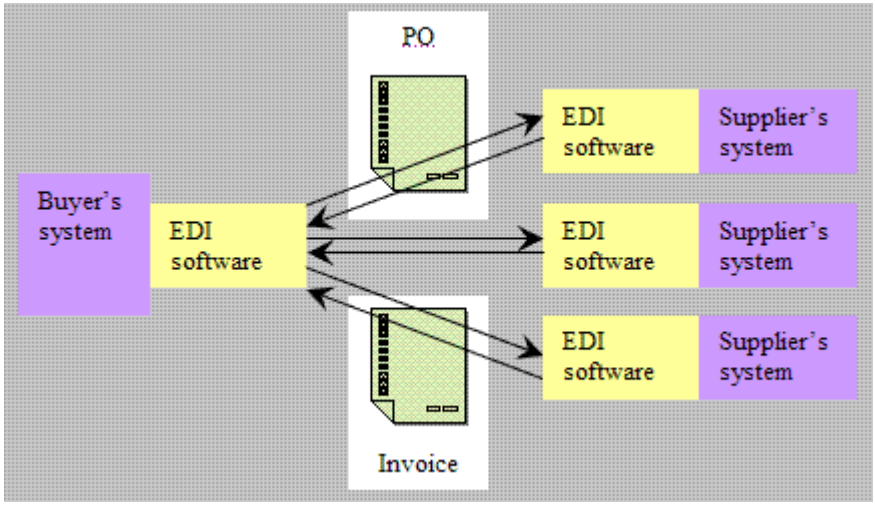


Exhibit 64: Traditional paper based systems

The main obstacle for implementing EDI is the setup cost required for both EDI technology and the changes required in business processes to utilize this technology effectively. These setup costs can be substantial. For this reason, it is more affordable for big businesses that have both the financial resources and IT expertise to facilitate its adoption. Nevertheless, there is little benefit to any business in using EDI if the majority of its small-and-medium supply chain partners are reluctant to invest in EDI. Consequently, the big business that is still committed to EDI may have to use a combination of strategies to encourage its adoption among suppliers[8, 9]. These include (a) providing financial subsidies, (b) providing technological assistance, and (c) enforcing mandatory use as a condition for doing business.

Even now when the Internet has become a universal platform that facilitates anytime-anywhere communication, the majority of e-commerce transactions are still conducted using EDI. Considering that big businesses are typically the initiators and key users of this technology, the significant number of transactions they generate may be responsible for such a result. This is expected to change in the future when small businesses become comfortable and more technologically capable with using e-commerce tools, particularly with the introduction of Internet-based technologies such as Web services and service oriented architecture (SOA).

E-marketplaces

The rise of the Internet in the late 1990s provided businesses with an easier means of providing for the electronic exchange of information pertaining to the procurement of goods or services between companies within their supply chains. By then, many organizations had already made substantial investments in ERP systems, and found that they are not using these systems to their full potential simply because ERP systems did not support external integration. By then, a substantial amount of business operations involved interacting with members of the supply chain outside of oneís own organization, particularly with suppliers. After establishing integration of their internal systems, organizations started exploring alternatives for inter-organizational integrations, and hence e-marketplaces came into existence. An e-marketplace is an Internet-based digital marketplace that provides a centralized point of access for many different buyers and sellers.

Many businesses already had an experience with digital marketplaces in the past through their EDI systems. The difference however is in accessibility. Internet-based technologies have made it possible for anyone, anywhere, and at anytime to communicate. In contrast, the older EDI technologies were proprietary, difficult to implement, and

not accessible to those outside of the closed network. The lowered adoption cost and reduced system complexity provided by these Internet-based technologies are quite attractive, particularly to small-and-medium size businesses. Since the value of these inter-organizational networks increases when more participants are added, many businesses started capitalizing on the idea of these linkages through the creation of e-marketplaces.

E-marketplaces can be classified according to ownership (private, shared, or public), industry (vertical vs horizontal), trading mechanism (catalogue, auction, or exchange) or service type (transactional vs collaborative) [10]. Transaction services support procurement activities where: (a) product specifications are explicit, (b) purchases are high-volume, (c) demand can be easily consolidated, and (d) common quality standards can easily be defined [11]. Many e-marketplaces, especially these that use catalogues, fall into these categories. Collaborative services support procurement activities on the other side of the continuum; that is where (a) product specifications are tacit, (b) purchases are low-volume, (c) demand is fragmented, and (d) there are sensitive quality requirements.

According to ownership, e-marketplaces can be classified as:

- Private— To establish own marketplace. This option is similar to the EDI model and is dually the most expensive and the highest risk. An example is SeaPort (http://www.seaport.navy.mil), which supports the contracting for professional support services for the US navy. Most often this type of marketplace is also classified as both vertical industry and relationship-oriented.
- Shared—To establish a trading consortium often with organizations in the same line of business (i.e. vertical industry). This latter option involves collaboration with others. Surprisingly, because it involves creating a network of organizations with similar interests, many members are existing competitors. An example of the trading consortium marketplace is Covisint (http://www.covisint.com) which is an automotive marketplace that was established in 2000. Covisint has since diversified into the healthcare industry. A shared marketplace serving organizations in different industries is called a horizontal marketplace. Often the marketplace supports the spot buying and selling of indirect goods or service (e.g. office supplies) and is transaction-oriented. An example of which is Tejari (http://www.tejari.com).
- Public (managed by an intermediary)—To join an independent marketplace. This is the low risk, least capital intensive option, particularly in the short term. The main difficulty however is to balance the needs of the diverse population of trading partners. An example of this is Ariba (http://www.ariba.com), which offers procurement solutions to different vertical industries.

Many of the advantages of EDI systems also apply to e-marketplaces. However, one key differentiator is that there is more than one way for an e-marketplace to be put together. The options discussed above provide several avenues for companies considering using Internet technologies for inter-organizational integration. As with the earlier EDI technologies, having a substantial number of participants is frequently the key to the sustainability of an e-marketplace. So, big businesses are still often the initiators for such B2B participation as they can be more easily successful in securing the necessary numbers of participants. As a result, e-marketplace setups still seem to favor value creation for the big over small- and medium-size businesses. For this reason, value creation is likely to be biased to the powerful side of the relationship (i.e. the big buyer or the big supplier). One new development in Web technologies, Web services, is empowering small- and medium-size businesses to take the lead in initiating B2B relationships and hence creating more value for the networks they form. Refer to Ray and Ray [12] for an insight into the value of using Web services to conduct B2B ecommerce by small-and-medium size businesses.

Web services

The explosion of the popularity of the Internet in recent years has greatly increased the demand for better means of electronic connection in the whole world of computers. As you now know, B2B is about using electronic means to connect businesses in order that they become more efficient and effective in the pursuit of their individual corporate goals. Moreover, with almost forty years of experience in electronic connections, users of B2B systems have become increasingly sophisticated in their expectations of these systems. These factors combine to form an imperative to connect people, information, and processes, and this imperative is changing the way that software in all areas is being developed.

Successful business systems increasingly require interoperability across platforms and a range of flexible services that can easily evolve over time. XML appears to be becoming the universal standard for representing and transmitting structured data that is independent of programming language, software platform, and hardware. The term Web services refers to new methods for integrating programs through standardized XML technologies with open, standardized interfaces that mask applications program interfaces (APIs).

Web services are based on four emerging Internet standards:

- XML schema
- Simple Object Access Protocol (Soap)
- Web Services Definition Language (WSDL)
- Universal Description, Discovery and Integration (UDDI)

Each of these emergent standards forms an important foundation piece for the provision of Web services. The XML schema provides the common syntax for representing data. The Simple Object Access Protocol (SOAP) provides the semantics for data exchange. The Web Services Description Language (WSDL) provides a mechanism to describe the capabilities of a Web service. Universal Description, Discovery and Integration (UDDI) is an XML-based registry that allows businesses worldwide to list themselves on the Internet. Registering allows them to streamline their online transactions by enabling other businesses to find them on the Web. But more importantly, using Web services should make the various company systems interoperable for e-commerce. UDDI is often compared to a telephone book's “yellow pages”; it allows businesses to list themselves by name, product, location, or the Web services they offer.

Web services enable computer systems on any platform to communicate over corporate intranets, extranets, and across the Internet. Therefore they have the potential to become the next major step in the ongoing evolution of the Internet. As such, Web services have the potential to redefine the world of B2B technologies. With the deployment of a platform using Web services, companies can have more opportunities to identify and communicate with existing and potential members of their supply chain, either suppliers or customers. Unlike earlier e-commerce technologies, the use of Web services allows for complete platform independence. Platform independence helps to reduce the integration problems between different corporate systems that have been the bane of B2B and other systems developers for the past forty years.

Challenges for B2B adoption

There are many challenges to the adoption of B2B systems regardless of the particular technology to be implemented; these can be categorized into [11, 13, 14]:

(a) economic feasibility challenges, (b) business environment challenges, (c) buyer challenges, and (d) supplier challenges.

Economic feasibility

The issues involving the economic feasibility of B2B adoption can be categorized as: (a) higher setup costs, (b) unsustainable growth, (c) unconvincing pricing, and (d) not enough profit.

High setup costs

Often the setup of a B2B system is an expensive endeavor. Many of the earlier B2B systems (e.g. the EDI technologies) were proprietary systems, meaning that they were custom-built, and thus there were higher upfront investment costs for each participating member in the EDI system. Telecommunications costs were also significant, particularly because many of these systems used dedicated communication lines. Total setup costs included technology systems setup as well as the cost of change in internal business processes and the education and training required to put these systems into operations. The cost barrier has been reduced significantly with the arrival of the Internet, however, cost remains an issue for many organizations. New models for B2B system delivery promise to be significantly cheaper as many Internet-enabled B2B systems are not technology dependant. This means that any business can implement the B2B solution irrespective of the type of IT system the business and its trading partners are using.

Unsustainable growth

While many B2B systems were profitable when they started, many failed to maintain growth for reasons such as:

- One or more members decided that the B2B technology was not suitable for themselves either because of cost or fit to business concerns.
- The B2B technology did not deliver what it promised.
- It was difficult to foster a collaborative spirit for doing business in a highly competitive market.

Unconvincing pricing

Pricing models for earlier B2B systems did not have a direct relationship with the value these systems delivered. Many B2B delivery modes were new so pricing was experimental. Many small- and medium- size business participants felt disadvantaged having to pay a substantial upfront setup cost and/or a high annual subscription cost. More recently, transaction-based pricing has become more popular as members pay according to volume of transactions they generate. However, this may only be possible when there is substantial volume to accommodate B2B system profitability.

Not enough profit

This is of particular significance to the B2B solution provider since not generating enough profit means going out of business. The cause for this can be any/or a combination of the challenges mentioned here.

Business environment

The issues involving the business environment for B2B adoption can be categorized as: (a) ineffective public infrastructure, and (b) restrictive regulations.

Ineffective public infrastructure

In many countries, particularly in the developing world, the public IT infrastructure is not well established. Problems included breakdown in service due to disruptions in telecommunications and electricity blackouts. One

means of resolving such problems can be in the ability of the B2B solution provider to identify these problems early on and to provide an alternative means to accomplish the task on hold. For example, an extension of time is granted to all bidders if a breakdown in service is identified at the time of bids closing. This extension might be made using other telecommunications means such as text messaging in order to reduce bidder anxiety.

Restrictive regulations from domestic governments

In many countries, the rules governing business communications and contracts using electronic means are yet to develop. This is a major obstacle that hinders B2B interaction because participants would have no protection if anything went wrong. Workarounds for this issue could be to get the B2B service endorsed by the government or to enroll a major governmental organization as one of the key participants.

Buyer Issues

The buyer issues involved in a B2B adoption can be categorized as: (a) organizational inertia, (b) buyer's fear about the capability of the B2B solution provider, and (c) buyer's fear about the integrity of the B2B solution provider.

Organizational inertia

Employees within the buyer organization may resist the move to B2B system for reasons such as: limited understanding of what the systems can so, inability to use the technology involved, or fear of exposing dishonest practice (particularly in procurement). This issue can be reduced through informing the buyer's community, providing education/training, and instilling rigorous ethical guidelines.

Buyer's fear about the capability of the B2B solution provider

Many of B2B solutions are new. As a result, many have undergone incomplete testing, or they have never been tested in a real-work environment. This issue is especially important when the buyer is required to invest in the B2B market setup. One way to resolve this issue can be to tie financial commitment to outcomes or deliverables. That is, the buyer pays for actual use and according to the services delivered.

Buyer's fear about the integrity of the B2B solution provider

Many of the interactions handled through B2B systems include sensitive data that is used to determine the award of contacts. The B2B solution provider needs to demonstrate compliance with various data management and security standards so that the system can be considered both trustworthy and reliable.

Supplier issues

The supplier issues involved in a B2B adoption can be categorized as:

- supplier's fears of competitive bidding
- few benefits to suppliers

Supplier's fears of competitive bidding

This is one of the key issues affecting supplier's adoption. One of the key benefits for using an electronic B2B system is to get the best price for the buyer. This is sometimes achieved to the disadvantage of the supplier (i.e. the supplier with the lowest bid wins). The effect of this issue can be softened when the buyer only invites qualified suppliers and includes criteria other than the price as part of bid evaluation.

Few benefits to suppliers

Many suppliers are reluctant to commit to using B2B systems when these systems are complex, expensive and promise little or no benefit. It is then vital for the buyer to consider criteria as ease-of-use, cost, and benefit to vendors as key when evaluating electronic B2B systems.

This section touched upon some of the common challenges to the adoption of electronic B2B systems. It also provided a few suggestions on how to deal with these challenges. The next section offers a few thoughts on the opportunities for the development of new business models that are enabled by the adoption of B2B systems. Some of these thoughts were developed with an understanding of the challenges discussed here.

Opportunities: New business models enabled by B2B systems

The basic business model, the supply chain, is not modified by the integration of B2B systems into a company's external operations. However, the actual implementation of that supply chain model is significantly changing. By providing increased opportunities for finding better suppliers and for reaching out to additional customers, B2B systems (along with modern transportation systems) are reducing the former need of businesses to be located in physical proximity to either its suppliers and/or its customers (i.e. agglomeration). This is increasing the trend towards globalization while providing for the more efficient and effective operation of businesses around the world. While a simple statement, this is a complex change in the way that businesses can operate. It has social and political implications far beyond the profit and loss calculations that are typically used to evaluate changes in business operations. Will new business models develop as a result of these changes? Stay tuned more to come.

Future of B2B systems

Many B2B systems were setup creating new, and sometimes, unconventional business models. These systems modified the conventional way businesses interacted. Some of these interactions (e.g. information sharing, online bidding, and auctions) affected the competitive position of the business. Hence, many participants agreed to come onboard, but their participation was minimal. With few active participants, these business networks did not produce the value they promised.

The trend towards more integrated systems, both intra- and inter-organization, is expected to not only continue, but to increase in pace. This trend towards integration, particularly inter-organizational integration, is the central concept of B2B systems. As more businesses accept the use of B2B systems, the users of these systems will become even more sophisticated in their expectations and the designers of these systems will provide better technologies to meet these increased expectations. Will this result in new ways of operating within the supply chain of the business? Certainly! Will this result in a new business model? Possibly! However, the supply chain model, although extremely simple, is very capable of representing any modification in the ways that business partners interact.

Editor

Maha Shakir

References

Thirkettle, F. and R. Murch, Business of art. 2006, Ivey. Retrieved January, 2007, from http://cases.ivey.uwo.ca/Cases/Pages/home.aspx?Mode=showproduct&prod=9B06E008| http://cases.ivey.uwo.ca/Cases/Pages/home.aspx?Mode=showproduct&prod=9B06E008 (http://cases.ivey.uwo.ca/Cases/Pages/home.aspx?Mode=showproduct&prod=9B06E008)

Davenport, T.H., Putting the Enterprise Into the Enterprise System. Harvard Business Review, 1998. 76(4): p. 121-131.

Ross, J.W., Surprising Facts about Implementing ERP, in IT Professional. 1999. p. 65-68.

Broadbent, M. and P. Weill, Management by Maxim: How Business and IT Managers Can Create IT Infrastructures. Sloan Management Review, 1997: p. 77-92.

Jacobs, J., Cities and the Wealth of Nations: Principles of Economic Life. 1985, New York: Vintage Books USA.

comScore E-Commerce Sales Data Accurately Predict U.S. Department of Commerce Estimates Weeks Ahead of Their Publication. 2006, PR Newswire. Retrieved January, 2007, from http://www.prnewswire.com/cgi-bin/stories.pl?ACCT=109&STORY=/www/story/02-23-2006/0004288403&EDATE=|http://www.prnewswire.com/cgi-bin/stories.pl?ACCT=109&STORY=/www/story/02-23-2006/0004288403&EDATE= (http://www.prnewswire.com/cgi-bin/stories.pl?ACCT=109&STORY=/www/story/02-23-2006/0004288403&EDATE=)

Greenspan, R., EU B2B Expected to Explode. 2002. Retrieved January, 2007, from http://www.clickz.com/showPage.html?page=1453831| http://www.clickz.com/showPage.html?page=1453831 (http://www.clickz.com/showPage.html?page=1453831)

Iacovou, C.L., I. Benbasat, and A.S. Dexter, Electronic data interchange and small organizations: Adoption and impact of technology. MIS Quarterly, 1995. 19(4): p. 465-485.

Wang, E.T.G. and A. Seidmann, Electronic data interchange: competitive externalities and strategic implementation policies. Management Science, 1995. 41(3): p. 401-418.

Christiaanse, E. and M.L. Markus. Participation in Collaboration Electronic Marketplaces. in Hawaii International Conference on System Sciences (HICSS). 2003.

Hsiao, R.-L. and T.S.H. Teo, Delivering on the Promise of E-Procurement. MIS Quarterly Executive, 2005. 4(3): p. 343-360.

Ray, A.W. and J.J. Ray, Strategic benefits to SMEs from third party web services: An action research analysis. Journal of Strategic Information Systems, 2006. 15(4): p. 273-291.

Day, G.S., A.J. Fein, and G. Ruppersberger, Shakeouts in Digital Markets: Lessons from B2B exchanges.California Management Review, 2003. 45(2): p. 131-150.

Wise, R. and D. Morrison, Beyond the Exchange: The Future of B2B. Harvard Business Review, 2000. 78(6): p. 86-96.

Glossary

Business-to-business (B2B):	B2B refers to the transactions (purchases of equipment, supplies, services, *etc.*) that occur between businesses as opposed to between businesses and consumers.
E-commerce:	E-commerce is the buying and selling of goods and services over the Internet, especially over the World Wide Web.
E-marketplace:	The E-marketplace is the electronic community of buyers and suppliers that integrates buyers' procurement systems with suppliers' fulfillment systems thereby creating a single standard process for transacting business electronically.
Enterprise resource planning (ERP):	ERP systems (also called enterprise systems) are information systems that are intended to integrate all of an organization's data and processes into a single uniform information system.
Enterprise system:	Enterprise systems (also called ERP systems) are information systems that are intended to integrate all of an organization's data and processes into a single uniform information system.
Globalization:	The concept of Globalization is the acknowledgment of the increasing interconnectedness of people and places that were formerly seen as unconnected because of the physical distances between them. Their new interconnectedness is a result of the recent advances in transportation, communication, and information technologies.
Integration:	Organizations integrate the functions of many different departments in order to produce the output of the organization; however, in the past the information systems that served these different departments were unable to share data between systems——they were unintegrated. In IT the term Integration refers to the trend of designing information systems that are able to share data across functional boundaries. See "ERP" and "Enterprise systems".
Legacy systems:	A Legacy System is simply an older information system that has not been

Business-to-business (B2B):	B2B refers to the transactions (purchases of equipment, supplies, services, *etc.*) that occur between businesses as opposed to between businesses and consumers. replaced by a newer system. Legacy systems are frequently unintegrated and the trend towards integration is seeing many legacy systems replaced by more integrated or even enterprise systems; however, most legacy systems actually perform the functions for which they were designed quite well, it is simply that they were not designed to integrate with other systems.
Supply chain:	Supply Chain is a concept that sees the various organizations, people, resources, and information involved in bringing products to their ultimate consumers as a continuous stream (or chain) where resources and information flow through to that final purchase. Supply chain sees all of the involved organizations as connected in a network whose purpose is to deliver the final product.
Transaction cost theory:	Transaction Cost Theory is one of the earliest attempts to theoretically define the business organization in relation to the marketplace. Because an organization cannot control its interactions with the marketplace (leaving continual uncertainty as to costs), it attempts to minimize these interactions by bringing many otherwise market transactions inside the functions of the organization. Bringing these "transactions" inside the firm rationalizes the costs of the firm.

12. Opportunities in peer-to-peer systems

Editors: Janis L. Gogan and James L. Linderman (Bentley College, USA)

Introduction and overview

Many highly effective managers have proven to be rather ineffective in predicting the diffusion and impacts of new information technologies. For example, IBM founder Tom Watson, Sr. initially foresaw a very small market for mainframe computers, and Digital Equipment Corporation founder Ken Olsen famously remarked that personal computers were not likely to catch on, except with electronics hobbyists. More recently, one IT application that caught many managers by surprise in the early years of the twenty-first century is peer-to-peer (or P2P) file sharing. Although the basic elements of P2P were invented back in the 1960's, a confluence of enabling technologies and applications caused P2P file sharing and other P2P Internet applications (including some in which "P2P" stands for "person-to-person" rather than "peer-to-peer") grew rapidly in the late 1990's. In some domains P2P has not had much impact, but in others –especially file sharing in the music industry and person-to-person auctions in electronic commerce—P2P has had a tremendous impact and continues to generate great turmoil (Stainaker, 2008).

This chapter is organized as follows:

In section 2 we place P2P in its historical context by describing how the evolution of computing, networking, and software technologies led to P2P applications. By understanding the evolution of information technologies, you will be better prepared to capitalize on as yet unforeseen capabilities and applications when they become available during the course of your career.

In section 3 we explain how P2P file sharing works today. This section helps you understand how P2P differs from other approaches to making data and information products available to individual users. We then discuss the early impacts of some specific P2P file-sharing applications on individuals, business organizations, and institutions such as governmental and educational organizations. We explore how P2P file-sharing has brought both intended and unintended benefits to individuals and organizations, while also ushering in significant technical, legal, and ethical challenges to individuals and organizations.

In section 4 we examine other P2P Internet applications – such as person-to-person sales on eBay and person-to-person video creation and sharing on YouTube-- and compare them with their traditional counterparts for communication, commerce, and collaboration, in terms of advantages and disadvantages.

Lastly, in section 5 we discuss P2P in light of the Law of the Double-Edged Sword—why every P2P application brings both opportunities and challenges. Neither P2P nor any other existing or emerging information technology is immune from this law. Effective managers and entrepreneurs recognize this, plan accordingly, and are ready to change those plans as unforeseen opportunities or challenges arise.

P2P in its historical context

In order to understand the recent impact of P2P applications, it is helpful to understand the evolution of computing and networking devices and data management and file sharing applications over the last several decades. This section briefly reviews these topics.

The earliest computers (1940's and 1950's) were just that—devices that performed numerical computations. Computations were strictly numeric; many years would have to pass before digitized data included audio or visual information. Forget sounds and colors, music and graphics, never mind pictures and video clips. Even alphanumeric characters were unusual and—if present—in a generic device-specific mono-spaced font. Many early computers had no keyboards, printing devices or alphanumeric display monitors, instead relying on binary input toggles and binary output light arrays. These computers were designed to process numbers, provide small amounts of numeric results, and little else. By contrast, in today's world the numeric processing is confined largely to research, science and engineering applications, whereas the average computer user deals with almost every form of data other than numbers. When computer manufacturers boast billions of arithmetic computations per second, many users wonder why bother, since arithmetic is the last thing on their minds; they want music, graphics and video—preferably in combination. Yet, today the binary number system remains the means of representing all data, including numbers, words, music, graphics and video. A great deal of computational power is essential to quickly interpret, record and re-construct such data. Some computer tasks that we take for granted, such as re-sizing and re-locating monitor windows, demand prodigious amounts of numerical computations to accomplish. The take-away lesson here is that computers still compute, even when our inputs, files and outputs are non-numeric. Furthermore, the non-numeric applications that are in demand today were all but impossible just a few decades ago.

Computers were introduced into large business organizations in the 1950's, and the first applications automated simple accounting tasks. At first, the transaction data associated with an application were linked directly to the application that created it. Most businesses created a data processing organization to build and run these applications. Timesharing techniques made it possible for individual users to view data on computer screens ("dumb terminals").

By the time the Internet (actually, ARPANET, the Internet's predecessor) came along in the sixties, data representation had advanced beyond numbers to include alphanumeric text files. The storage representation for files had also come a long way from the hole-punched cards and paper tapes of early computers to magnetic tapes and disk drives. The original rationale for ARPANET was to enable long-distance transmission of files. ARPANET's developers would have been astounded to think that one day such information interchange would (1) be available for mass usage by non-technical citizenry and (2) be bi-directional and highly interactive in real time. Keep in mind that digital file sharing involving the average individual was all but impossible (let alone affordable) only a few decades ago.

In the sixties, the key Internet protocols – TCP and IP—were developed to support communication among the US armed forces, other governmental agencies, and a small number of defense contractors. TCP/IP was designed so that if an enemy were to destroy one or more network nodes, communication would still be possible (as a network-of-networks, the Internet does not have a central point of failure). A group of computer science professionals, the so-called Network Working Group, developed various networking tools. As this group worked, they recorded their deliberations via memos called Requests for Comment (RFC). RFC 1, written by Stephen Crocker on April 7, 1969

discussed approaches for transferring files from one computer to another over a network (see http://tools.ietf.org/html/rfc1).

In 1965 Intel co-founder Gordon Moore foresaw a long-term trend of dramatic improvements in the integrated circuit (the foundation technology for computing devices). As predicted by "Moore's Law," in the sixties and seventies there were dramatic price/performance improvements in computer hardware. While business computers in the sixties were huge mainframes managed by a data processing (DP) organization and used for routine transaction processing, by the seventies many companies used less expensive mini-computers for less routine applications in design, engineering, manufacturing, and marketing. Some of these applications processed a greater variety of data types, including text, graphs, and some images. Many organizations invested in database management systems, which enable data to be shared among many software applications. Management Information Systems (MIS) departments were formed to oversee data administration and create various reports based on the information stored in these databases. Despite receiving very fat stacks of paper reports, some managers complained that the particular information they needed was not quite reflected in them! Subsequently, software was created to allow individual users to make queries of these databases and to generate their own reports.

In the seventies, File Transfer Protocol (FTP) was developed to make it easier for people to share files of various sizes containing various data types. An FTP server "listens" for requests coming from other computers. When such a request is detected, the server then establishes a connection with the requesting computer and subsequently receives a file and instructions on where the file is to be delivered. Then, in a process that is similar to how Electronic Data Interchange (EDI) works, the server forwards the file to the Internet address specified by the sender.

The microcomputer was also invented in the seventies, but it really burst on the business scene in the early eighties (the IBM PC, introduced in August 1981, dominated the early days of the end-user computing revolution). The first "killer application" was an electronic spreadsheet (VisiCalc, which later lost ground to a product called Lotus 1-2-3, which in term was supplanted by Microsoft Excel). To share spreadsheet or word processing files, a user might save a file to a floppy disk and walk over to another user (via "sneakernet") who would insert it into their disk drive. It did not take long for managers to realize that this was not an efficient way to work within groups or across business units, so local area networks (LANs) were developed to share files on businesses campuses. Companies used wide-area networks (WANs, which were leased or owned) to send some well-structured transaction data to their business partners, replacing paper documents such as purchase orders and invoices. Much effort went into standardizing business messages so they could be communicated using EDI protocols such as X.12 and EDIFACT.

By the eighties, the Internet had broadened to include many armed services personnel, as well as defense contractors, researchers and computer science professionals. Another key development in the late eighties was client/server computing. With the advent of PCs, companies retired many "dumb" terminals. Also, managers noted that while it made sense to give many PC users their own word processing and perhaps spreadsheet software, it was too expensive to provide individual copies of other software that would only be used occasionally. Instead, users would gain access to that software via the company network. This client/server computing idea evolved further, so that some business applications might entail having some code and/or data residing on a user's desktop "client" machine and other code and/or data residing on a server machine. A "fat" client would have quite a lot of software

and/or data but occasionally rely on a server, while a "thin" client would rely heavily on software and data provided by one or more servers.

FTP was a useful mechanism for sharing files in the seventies and eighties (see RFCs 114, 454 and 765). However, many users' connections were over slow telephone lines, which limited both the size of files that could be sent and the speed of transmission. And, the U.S. National Science Foundation, which maintained several key elements of the Internet, decreed that use was restricted to defense- and research-related activities. So, researchers, armed services personnel and defense contractors were able to FTP files to one another, but most other PC users did not have access to this tool.

Although the average Internet user may regard this history as quaint and irrelevant, many real advantages and disadvantages of the Internet in general, and P2P applications in particular, relate directly to (1) the ability to digitize all forms of data using binary numbers, and (2) the nature of high-speed, long-distance computer-mediated interchange of information. As just one example of what is both an opportunity and a risk of the ability to digitize information, this means data can rapidly be captured, stored, copied, and distributed—nice for many applications, but not so nice if the capture/copying is unauthorized and the distribution illegal or unethical and nearly undetectable. For an example of what is both an opportunity and a risk of long-distance computer-mediated communications, consider the long-distance aspect—nice for reaching a remote audience, but not so nice if fraud or deception is involved and the damage is done in a "hit and run" high-speed manner with little legal recourse due to conflicts in geographical jurisdiction. As will be developed throughout this chapter, any Internet application can be used or abused.

In the nineties, several elements combined to turn the Internet into a global platform for collaboration. Prices dropped to levels that enabled very rapid growth of sales of PCs to businesses, educational institutions, and home users. The Internet expanded and the National Science Foundation dropped its restrictive Acceptable Use Policy, opening the door to private and commercial use. E-mail (which was previously used primarily by scientists or by very large organizations) now became commonplace. Furthermore, the key technologies of the World Wide Web — (hypertext markup language) html, uniform resource locators (URLs), and web browsers, made it easy for users to gain access to data in a huge variety of forms (numbers, words, images, music, video) stored in files all over the world. File transfer protocol (FTP) remained a useful mechanism for transferring large files (however, FTP transfers require the use of an available FTP server, which is a limiting factor).

Digital music or video files do not play very well over slow or unreliable Internet connections. Also, such files are more useful offline, especially for users who want access to the music or videos when they are not sitting at a desktop computer. So, PC users sought easy ways to locate and download specific types of files (especially music), which they could store both on their computer and on other devices (such as iPods or cell phones). This became a major impetus for the development of some P2P file sharing services.

A second motivator for P2P applications was the observation that personal computers typically sit idle for long periods of time (such as when you are asleep) and thus are not used to their capacity. Since PCs were being connected to the Internet in rapidly increasing numbers, some people reasoned that there ought to be a way to capitalize on all that collective excess capacity. These two needs led to several new varieties of peer-to-peer file sharing.

Peer-to-peer file sharing

Previously we noted the rise of client/server computing in organizations, including the use of "fat" clients containing many applications and some data, and "thin" clients which do not store data and contain so little application software that they are similar to a dumb terminal (yet much nicer to work with, since a "thin" client would normally have a web browser and might support both audio and video). Some industry leaders, such as Oracle CEO Larry Ellison, predicted in the nineties that the PC would become obsolete because networks would have such robust bandwidth and speed that a user could work online with applications in such a transparent way that they would be barely aware that the data and applications were not stored on their "thin" client device. Others took a different view. They reasoned that since many users make regular use of a few key applications (such as word processing), there would continue to be a need (whether real or perceived) for those applications to reside directly on the user's PC (so, it would be a "fat" client). Since, thanks to Moore's Law, those devices were getting ever more powerful yet less expensive, most users' PCs would have a lot of excess computing capacity. Instead of relying on a client/server architecture to serve up applications via the Internet, why not find ways to tap into the excess storage and processing capacity of the many PCs already connected to the net?

When FTP is used to share files, an FTP server must (at least temporarily) store the file before it can be forwarded to its destination. This is not necessary in a peer-to-peer network, in which resources (data and applications) are distributed throughout "peer" computers, instead of being concentrated in server computers. Once you have obtained a particular file from another user, your machine can provide it to other users on the network. Any peer computer can at different times act as either a client or a server. And, some services are designed to ensure that every participant is both a "taker" and a "giver"; users who prevent access to files residing on their computers are eventually barred from further use of the network (a strategy known as "give to get").

Early P2P applications worked as follows: Users install a P2P software application on their machines. When a user wants a particular file, he issues a request for that file, and agent software then searches the Internet for machines containing the P2P software. On each such machine found, the agent then searches for the requested file. This process worked pretty well at first, since computers operate at the speed of light and the random path of searches meant that different user PCs were tapped at different times. However, it was inherently inefficient, since it meant that if multiple users sought the same file, each user's agent would go through the same process of hunting up machines and then checking the contents of each machine it encountered until it found the desired file. So, very soon new P2P application software was developed that would create one or more indexes, which were stored on one or more computers. Most P2P software today involves creating at least one index, such as a directory of machines containing the P2P software. This index can be kept on every participant's machine or, as with the Gnutella P2P service, the index can be kept on a small number of machines that act as directory servers. Upon receiving a user request, the agent first examines the directory on one of these machines, and then uses that information to find those machines.

Some P2P services also create an index of music files stored on specific users' machines. Since the directory/index itself is not a large file (in comparison with the music file itself), most experts do not consider this approach to be a violation of the basic P2P principle. Music and video files that are to be shared are distributed across users rather than stored in a few centralized servers. Music files are large; when they are stored centrally a server can easily become paralyzed by multiple simultaneous user requests. In contrast, when these files are distributed across the Internet, no one user's machine is heavily burdened. Also, to reduce burden on individual

users, some services have different users' machines provide portions of large files rather than the entire multi-megabyte file. Thanks to meta-data, which describes where each block fits, these portions are then re-assembled at the recipient's machine.

In 1999, while a student at Northeastern University in Boston, Sean Fanning created the Napster P2P music file-sharing service. Napster caught on so quickly that soon record labels and some artists (such as Madonna and members of the Metallica band) complained that it was being used to illegally share copyrighted songs. By December 1999 the Recording Industry Association of America (RIAA) had already filed a lawsuit against the fledgling company, leading to its shutdown in 2001 (subsequently another company bought the Napster name and logo and it offers a fee-based file sharing service). In the meantime, other Napster-like services, including Grokster, Kazaa, BitTorrent, eDonkey and others were coming online, so P2P file sharing continued to be easy and ubiquitous. Many people have debated whether these services are unethical. U.S. courts have ruled that if copyrighted material is exchanged over such a service, it can be legally shut down.

Although P2P file sharing eliminates the bottlenecks that arise when many users seek files stored on few servers, it can nevertheless increase network congestion when users are concentrated geographically, such as on college campuses (Dzubeck, 2005). Both colleges and Internet service providers have complained about this problem. To reduce their internal network traffic, as well as out of respect for intellectual property rights and to avoid lawsuits, some U.S. colleges have banned their students from using their college e-mail accounts and other college-provided network services for file sharing. However, critics have noted that some P2P file sharing is perfectly legal and that an outright ban unfairly penalizes those users who act in an ethical and responsible fashion (see for example Navin, 2006).

After Napster was shut down, another P2P player that emerged was Kazaa, a music file-sharing service founded by Janus Friis and Niklas Zennstrom, who subsequently sold parts of the company to Sharman Networks. They then formed an Internet telephony business, Skype, which also was based on P2P. eBay saw enough potential in Friis and Zennstrom's P2P technology that they bought the company for $2.6 billion in 2005. In 2006, Friis and Zennstrom announced that they were working on a venture dubbed the Venice Project, in which P2P technology will provide the foundation for a licensed video and film-sharing service (Green, 2006; Rosenbush, 2006). Clearly, these two visionaries believe that the ability of P2P to spread the processing load across participating computers represents a viable business opportunity in several domains.

With the resolution of some of the lawsuits, Napster re-emerged as a legitimate fee-based file-sharing service, although many reports indicate that they are struggling to reach profitability. BitTorrent, another popular file sharing service, was also re-launched and now offers ways for paying customers to legitimately pay for and obtain music, film, and video files.

The P2P controversy continues. *Business Week* reported in February 2007 that "illegal files still account for an estimated 90% of the music download market." (Holahan, 2007). However, some music labels have stated that they no longer plan to attempt to block illegal downloading, and some companies, such as Skyrider and Internet MediaWorks, offer services that attach advertisements to shared files (Myer, 2007).

P2P networks are reportedly targets for distributed denial-of-service attacks, worms, and other malware that can affect users' computers and their ability to share files. "Bad guys" recognize the enormous popularity of file sharing and launch opportunistic attacks aimed to affect large numbers of users. Other bad guys use "botnets" to

rapidly reach large numbers of users' machines, leading to the emergence of an underground market for advertising bounties. Bots are programmed to look like P2P users' agents, which in turn can fool advertising network services into believing that large numbers of real users have viewed various advertisements (companies usually pay advertising networks based on the number of "impressions," or user hits; Hines, 2007).

Meanwhile, all the publicity about Napster got many IT managers thinking about the possibilities for using peer-to-peer technologies to improve corporate IT productivity. One observer (Kharif, 2001) summed up the perceived potential:

"For big companies, the promise of P2P is threefold. By using idle PCs, P2P connections could tap into cheap, underutilized processing power and bandwidth using a technique called distributed processing. By linking users directly, P2P could create easier collaboration, allowing the rapid formation of work groups that sidestep traditional barriers such as firewalls and restricted intranets. Finally, P2P could make information more accessible throughout an enterprise by opening up the desktops of individual employees – allowing staffers to search each other's virtual desk drawers more freely."

For the distributed processing benefit, companies looked to examples such as the SETI project (Search for ExtraTerrestrial Intelligence), in which millions of volunteers agreed to download software that enabled their machines to analyze telescope data when they would otherwise be sitting idle. SETI has since inspired many other examples of this type of volunteer distributed computing (see http://boinc.berkeley.edu/volunteer.php), such as for doing gene sequencing, protein analysis, analyzing clinical trials data, and stress-testing of eCommerce sites. While these distributed computing examples are a P2P application, they are not really a P2P file-sharing application, since (with SETI, for example) a central server sends out the telescope data, and peers do not directly communicate with one another.

An example of an initiative aimed at using P2P technologies to both enable collaboration and to make information available throughout an enterprise is a product offered by Groove Networks, a company founded in 1997 by Ray Ozzie, a renowned computer scientist who was the architect of Lotus Notes. Groove users can easily share in the preparation of various documents, and files stored on any Groove user's machine can be available to all other Groove users. This is an interesting example, because it represents a departure from how companies have been approaching enterprise software in recent years. ERP software relies on a centralized (or virtually centralized) database of business and financial transactions, and many early knowledge management systems took the same approach by creating centralized (or virtually centralized) knowledge repositories. Groove, in contrast, does not attempt to centralize data and documents. In 2005 Microsoft announced that it was acquiring Groove; subsequently Ray Ozzie became the chief software architect for Microsoft. Microsoft's acquisition of Groove, and the appointment of Ozzie to his influential post are two indicators that this software giant sees potential in P2P file sharing and in distributed computing in general.

Although in the late nineties there was much excitement about P2P's potential for both consumer and enterprise applications, a dramatic and widespread U.S. technology downturn in spring 2000 put such ideas on hold for several years, and many managers adopted a conservative wait-and-see attitude to spending plans for all new technologies. After robust growth from 2005 to 2007, a stock market meltdown in January 2008 led analysts to predict another U.S. technology downturn. As to P2P in particular, many managers also preferred to wait until some of the lawsuits were resolved.

There continues to be strong interest in understanding how P2P impacts network traffic, and how this approach can be harnessed to utilize excess capacity on user's PCs. MIT's *Technology Review* magazine reported in 2007 that "TV shows, YouTube clips, animations, and other video applications already account for more than 60 percent of Internet traffic" and this number was predicted to rise to as high as 98 percent by 2010 (Roush, 2007). Both this piece and another study reported in 2007 that P2P connections can help alleviate network congestion when files are served up from nodes distributed all over the world, instead of concentrated in a few servers.

In an interesting twist, researchers used P2P technology to gather more than six million data points per day for two years, yielding a map of the Internet's structure. They found that although a huge volume of Internet traffic worldwide today passes through about 80 key nodes, many other nodes are well connected with other peer computers and could thus bypass these key nodes if necessary. Researchers are working to develop new networking tools to distribute network loads more effectively (Carni, et al., 2007; Graham-Rowe, 2007). For example, a system called Chunkyspread developed by Cornell University professor Paul Francis, reduces the need to broadcast metadata about files, by assigning "slices" of files to each participating user. As explained in *Technology Review*: "A slice consists of the nth bit of every block – for example, the fifth bit in every block of 20 bits. Alice's PC might obtain a commitment from Bob's PC to send bit five from every block it possesses, from Carol's PC to send bit six, and so forth. Once these commitments are made, no more metadata need change hands, saving bandwidth." (Roush, 2007).

In recent years there has been an increase in reports of government organizations—such as the U.S. Department of Homeland Security –investing in P2P applications. And, another acclaimed computer scientist – Alan Kay, who invented several key personal computer innovations (including the graphical user interface) during his years at Xerox PARC—is touting the expected benefits of a new software developer kit for collaborative applications, Croquet, which will be based on P2P principles (see http://croquetconsortium.org).

There have also been reports of P2P problems. For example, in June 2007 pharmaceutical giant Pfizer stated that the spouse of an employee had inadvertently leaked personal information on more than 17,000 employees stored on a company computer. The individual had included the directory in which the file was stored as an allowable source for a music-sharing application (Leyden, 2007).

Other P2P Internet applications: communication, commerce, and collaboration

In our discussion of P2P file sharing applications in Section 3, we noted that "P2P" stands for "peer-to-peer," referring to an architecture of computer "peers" (as opposed to "clients" or "servers"). Of course, in Section 3 many of the examples also involved person-to-person file sharing (after all, college students are sharing music files with one another; they just happen to be using their computers to do so, in a peer-to-peer computer network architecture). In this section, we broaden our discussion to include person-to-person Internet applications that do not necessarily involve a peer-to-peer network from a computational perspective. In other words, while some of the applications which we discuss next could be designed using peer-to-peer tools and principles, others are based on client/server tools and principles.

So, in this section P2P stands for "person-to-person." We classify person-to-person applications according to what we call participation: the number of people who normally participate in the application as suppliers (senders) or consumers (receivers) of information:

- One-to-One participation implies communication between exactly two persons

- One-to-Many participation implies communication between one person and many others
- Many-to-Many participation implies communication between many persons

A secondary categorization is that of *symmetry:* whether participants have equivalent/interchangeable roles in the application (for example, VoIP telephone communications, in which participants are both senders and receivers of information), or reciprocal/unique roles (for example, e-mail, in which one participant is a sender only and the other is a receiver only).

In the following subsections we discuss a few applications, and compare them against traditional means of performing similar communication or collaboration tasks (for example, how Internet telephony bypasses conventional telephones). We discuss advantages and disadvantages of these applications, and precautions appropriate to their use.

One-to-one P2P applications

One-to-One participation implies communication between exactly two persons, who normally know one another well enough to initiate communication by invoking each other's address (be it a postal address, telephone number, Web URL or e-mail address). Participants' prior knowledge of one another affects their interpersonal trust, which in turn presumably affects both their willingness to use the communication tool as well as the content of their communication. Two familiar pre-Internet examples of one-to-one communications are land line telephone calls and written mail. One-to-one Internet examples are e-mail (reciprocal), VoIP (equivalent) and Instant Messaging (both reciprocal and equivalent).

Two traditional One-to-One examples differ in terms of symmetry:

Written mail is *reciprocal* in that one party is a sender and the other is a receiver.

- Land line or cellular telephone calls and text messages are *equivalent* in symmetry, in that participants act as both senders and receivers of information.
- Each of these traditional examples can be compared with similar one-to-one Internet applications, as well as one-to-one applications delivered via other network technologies:
- Like traditional written mail, communication using e-mail is *reciprocal* in that one party is a sender and the other is a receiver.
- Like traditional telephone calls, communication using Internet telephony (VoIP, using services such as Skype) is *equivalent* in symmetry, in that participants act as both senders and receivers.
- Like traditional telephone calls, instant messaging is reciprocal in that at a given moment, one party is sender and the other is a receiver. However, it is also *equivalent* in that these role-changes occur on a moment-by-moment basis, giving an experience that is more like a telephone call in its immediacy.
- One-to-One P2P Internet applications (such as e-mail, VoIP, and IM) offer both advantages and disadvantages over their traditional predecessors (mail and phone).
- Some users view e-mail as "free" because someone else (such as an employer, school or advertiser) pays for the service. Most U.S. schools provide complimentary e-mail accounts to registered students, for example. Many (not yet all) e-mail users also feel it is far easier to compose and send an e-mail than to hunt down

pen, stationery, envelope, and postage stamp, use legible handwriting to write the letter, and post it. E-mail has additional advantages: near-instant delivery, ability to append photos, documents, and other digital attachments, ease of reply, inexpensive storage, and the option to send the same message to multiple recipients at widely dispersed locations.[71] Some users nevertheless do feel that traditional written mail has its advantages. For example, because the sender takes time to carefully think through and phrase a paper letter, some feel that better communication results. Others feel that the time investment (in good language and personal if not good penmanship) itself is meaningful to the recipient and that personal touches (enclosed mementos, applied fragrances, etc.) transform written letters into cherished keepsakes, which is rarely the case for e-mails.

- VoIP is often (although not always) considerably less expensive than long-distance telephone calls (this depends on the price of the Internet access and the volume and average length of the calls in comparison with your phone company's pricing plans). On the other hand, many VoIP users mention dropped connections and poor clarity as irritating problems as compared with traditional land lines (on those aspects, VoIP is more comparable to cell phones, at least in the U.S. where users continue to complain about dropped or unclear calls).
- Like e-mail, IM is viewed as "free" if Internet access is provided by a third party for free. Also, a given user may be view it as "free" if the user accepts their Internet access bill as a fixed expense and sees IM as an unexpected or "bonus" application.
- Traditional and Internet One-to-One applications can also be compared in terms of security risks.
- Traditional mail is insecure in that it can be intercepted before it reaches the sender or after the letter has been discarded. The U.S. Postal Service is generally reliable, but there are occasional reports of mail that is stolen before it reaches its destination (theft of elderly citizens' Social Security checks, for example, is so common that the U.S. Social Security Administration now urges retirees to arrange for direct electronic deposits into their bank accounts). There are many reports of identity theft based on "dumpster diving" (thieves rummaging through discarded garbage to unearth credit card account numbers and such), which has given rise to robust sales of inexpensive document shredders for home use. Also, because a sender can easily disguise the letter's origin, there are risks of mail fraud, hate mail, and unsolicited "junk" mail.
- E-mail is vulnerable to risks that mirror traditional mail risks. A sender needs to know the address of the intended recipient, but if so inclined the sender can disguise his or her e-mail address by routing the message through an Internet "anonymizer" service, which gives rise to threats of fraud, spam, and hate mail. So threats to e-mail privacy include unauthorized interception of the e-mail before it reaches the recipient or unauthorized access after the recipient has read it. Also, e-mail has given rise to other privacy issues. For example, many users do not understand that when they hit the "delete" key, the e-mail has not actually been destroyed. Also, many employees do not realize that (at least in the U.S.) the employer is legally allowed to look at their messages and can use them as to find evidence of offensive language, harassment, revelation of company secrets, and other offenses which can lead to the employee's dismissal.

71 We consider e-mail to be a 1-1 application (rather than 1-M) despite the option of multiple recipients, which is merely a convenience equivalent of sending multiple 1-1 communications.

- A traditional phone conversation is subject to the threat of external parties listening in via wiretapping or overhearing a telephone conversation (or at least one half of the conversation) when it takes place in a public location. Unauthorized recording is also a problem. For example, a participant or an eavesdropper (including law enforcement officials, presumably with authorization) could secretly record the conversation without one or both parties' permission, giving rise to subsequent issues related to access to the information that was exchanged and/or actions subsequently taken as a result of the conversation.
- Unauthorized interception or recording of VoIP conversations gives rise to similar issues as in the traditional telephone application.
- Instant Messaging is similar to VoIP except that the communication is text based rather than voice based.

Thus, One-to-One P2P applications give rise to similar risks as their pre-Internet predecessors, and they also give rise to new risks. Personal communications are almost always intended (or assumed) to be private, so the most obvious threat is compromised privacy through interception before, during, or after communication events. Whereas there are usually governmental prohibitions and sanctions for tapping telephone conversations or violating post office regulations, Internet communications are both less secure and less regulated. Because it is inexpensive to copy, transmit, and store information in digital form, it is also easy for unauthorized users and thieves to gain access to it.

Note also that both traditional and Internet one-to-one applications continue to evolve, including taking on the characteristics of one-to-many applications. For example, as telephone networks "go digital," new threats have arisen. Identity-blocking technologies and agent software can help voice "spammers" flood phone lines with incoming calls. This flooding tactic was reportedly used as a "dirty trick" in the United States by some computer-savvy people who sent false messages about a candidate to voters just before a presidential election to dissuade voters from supporting that candidate.

One-to-many P2P applications

One-to-Many participation implies communication or commerce between one person and many others, who may or may not know either the one person or the many others.

A traditional example is a community bulletin board, in which an individual may post an announcement (perhaps for an upcoming event, or a lost and found notice), an offer of services (tutoring, babysitting, gardening, etc.) or a For Sale sign. Reciprocal communication is involved, since one individual posts the information in hopes that many individuals will respond to it. One problem with bulletin boards occurs when individuals post offensive information; the remedy is that a sponsor (or concerned citizen) simply removes the posting). Another problem is that individuals may fail to remove a notice when it is no longer useful (because the event has already occurred or the lost item found and returned, for example). Also, the authenticity of a posting, including the reliability of the individual who posted it, may be beyond the control of the organization providing the bulletin board.

Another traditional one-to-many example is a yard sale (which in different regions of the U.S. is called a garage sale, tag sale, or other name). An individual (personally or acting as an agent for members of his/her household) places items (furniture, tableware, clothing, books, toys, etc.) on display in their yard, and invites passersby to purchase them. The relationship is reciprocal because one seller supplies items in the hope that many individuals will make purchases. The prices charged might be pre-determined, negotiable, or subject to an auction. Items that are not acquired by the end of the sale can be put aside for another sale or disposed of in some other way.

These traditional one-to-many examples can be compared with similar P2P Internet applications:

Like a community bulletin board, Internet bulletin boards require an individual to initiate a posting, and let users decide for themselves if and how to respond. Organizational sponsors usually monitor postings for offensive content, and the principle of *caveat emptor* "let the buyer (or responder) beware" usually applies. Aside from these similarities, there are significant differences between old fashioned physical bulletin boards and the Internet variety. For one, Internet bulletin boards usually have no theoretical space limitations, although sponsors may establish prioritization rules for message deletion or visual placement. Also, except when restricted to an organizational intranet or extranet, Internet bulletin boards are potentially accessible to all Internet users which can complicate the identification and authentication of both posters and responders. Even intranet and extranet bulletin boards are far more readily accessible to authorized users, because users' physical location is irrelevant in cyberspace. Lastly, even when a sponsor monitors postings for content and attempts to control the identification and authentication of those who post notices, significant damage can be done quickly if monitoring does not take place in real time.

- Like a yard sale, on Internet-based person-to-person marketplaces such as eBay or Craig's List individuals put objects up for sale and negotiate with potential buyers. Site sponsors monitor activity to some extent, but the principle of "let the buyer beware" still applies. Ironically, the increased anonymity on the Internet (and long distance nature of the transactions) introduces the reciprocal principle of "let the seller beware"; that is, the buyer may be using someone else's identity, acting as a shill bidder to bring up the prices, or otherwise acting fraudulently. Such actions occur far less frequently during traditional yard sales, especially when sellers require payment in cash. Also, in the traditional yard sale transactions are normally consummated on the spot; the buyer pays for and receives the items concurrently. Also yard sales are usually subject to geographical jurisdiction, whereby a dishonest seller can be traced to a location, identified, and prosecuted under the laws of the jurisdiction; Internet sales transcend geographical jurisdiction, severely complicating tracing or identification, let alone prosecution.

The One-to-Many P2P Internet applications discussed here (Internet bulletin boards and P2P marketplaces) have clear advantages over their traditional counterparts. Internet bulletin boards are dramatically more accessible to both suppliers and consumers of information—so long as all parties have Internet access. Constraints of time and distance are all but eliminated. Internet P2P marketplaces are accessible to a far wider audience of potential buyers. Not only does this increase the likelihood of a sale, but it also can facilitate buyer competition (auctions) and therefore higher profitability. Still, the "old fashioned" yard sale does have a few advantages. The seller can deal on a more secure cash basis, and buyers get the goods immediately without having to wait or pay for subsequent delivery. Also the potential buyer gets to see, touch, and potentially try on or test items, and there is some (albeit limited) recourse if fraud is involved.

Many-to-many P2P applications

Many-to-Many participation implies communication between or collaboration among many persons and many others, who may or may not know one another well. A few "traditional" examples are:

A meeting involving participants who share a common interest (such as belonging to the same political party). For example, in the U.S., the Democrat and Republican political parties have conventions in which representatives from the 50 states meet to decide on their parties' platforms and to nominate presidential candidates. Such

meetings normally take place in one location at the same time. There is normally a schedule associated with the meeting, and a record of what people say, decisions taken, etc. might be captured in the form of minutes prepared by an observer. Participation is *equivalent*, with most participants acting as both suppliers and consumers of information. Facility and regulatory constraints (such as fire or other safety rules) normally place restrictions on the number of participants.

- In another many-to-many example, individuals with a common interest come together to produce a document, such when a fund-raising group produces a cookbook of members' favorite recipes. Such a publication involves *reciprocal* communication, in that many participants contribute and many participants consume the information. The consumers are rarely identified or tracked.
- A poll is also a many-to-many example, in that many individuals' views on an issue (a political issue, for example, or student evaluations of a teacher) are captured and then shared with many other issues. Here again the communication is reciprocal in that many participants supply and many participants consume the information. A consumer survey of products is quite similar.

In all of the above traditional many-to-many examples above, there is a third role besides "provider" and "consumer" of information. That role is that of sponsor, organizer or mediator. Some entity (an individual or a relatively small group) organizes the logistics of a meeting. A potentially different entity edits and potentially a different entity publishes a group publication. Yet another entity may conduct a poll and/or analyze the polling data.

Many-to-many Internet P2P applications have rather similar characteristics as their traditional counterparts, as well as offering new capabilities:

- The use of the Internet as a P2P platform for political conventions and other meetings of large or small groups of people allows participation by home-bound or disabled individuals and those who are unable or unwilling to travel to a meeting location. It also provides the ability to capture a complete record of an entire meeting.
- The use of the Internet as a P2P platform for the production of group documents such as cookbooks offers advantages of wider and timelier access. Both the number of contributors and the number of information consumers is likely to be significantly larger. Also, since the publication is not static, if a consumer uncovers a mistake in a recipe, a corrected version can be easily posted. This is why wikis (such as Wikipedia) are such powerful and popular P2P applications. A wiki allows the rapid evolution of useful information, in a significantly compressed time frame, for larger groups of both contributors and consumers.
- Similarly, the Internet offers a powerful platform for polling or rating systems that capture citizens,' affinity group members' or consumers' views on issues or products. On the Internet, the reduced costs of capturing polling and rating data, analyzing it and reporting on it (even in real time), increases the number of views that can be captured and the ability to track the evolution of opinions over narrower slices of time.

Many-to-Many P2P Internet applications have advantages over their traditional counterparts. Internet-hosted meetings eliminate distance constraints, enable time shifting (that is, the meeting can be held in synchronous or asynchronous mode), and facilitate record keeping (including capturing a record of the entire meeting). There is also the possibility of admitting many more participants, potentially in different categories such as disabled or

otherwise home-bound citizens, anonymous attendees, and non-contributing observers. Some feel that a disadvantage is the reduction in the personal touch of face-to-face communication (such as the ability to slap a participant on the back or shake his or her hand). Security is a great challenge in order to avoid breaches of privacy, unauthorized participation, and unauthorized recording or subsequent use of the meeting record.

The use of Internet-based P2P collaboration systems such as wikis for shared authorship offer the potential advantages of capturing ideas and information from many more contributors while reaching a potentially wider audience of consumers. However, some people feel that such systems face steeper challenges compared with their traditional counterparts in ensuring the competence and reliability of the contributors and the information that they provide.

P2P and the Law of the Double-Edged Sword

Our discussion of P2P applications illustrates an important phenomenon that applies across the full spectrum of IT applications. Every IT application brings challenges and risks along with benefits. We call this the Law of the Double-Edged Sword. Note that peer-to-peer file sharing, by enabling every PC on a network to act as both client and server, brought the benefit of avoiding bottlenecks (which occur when excess demand for music or video files strains servers located centrally or on only a few nodes), and introduced an easy, non-commercial approach to file sharing. However, it also introduced ways for users to violate intellectual property rights by essentially stealing music and video files – an ethical breach. While P2P is a great way to share open-source music or video files, as well as to increase the portability and usability of music files that are legitimately owned, it is also a way to avoid paying for music and videos, depriving musicians, studios, and others of their rightful share of profits. Reports of hacker attacks and worms that propagate across these P2P networks again illustrate the Law of the Double-Edged Sword.

It seems that every day, somebody announces a new IT application. In the eighties, the personal computer was a new phenomenon which brought such new capabilities as word processing, spreadsheets, and end-user databases, as well as new risks, such as flawed decision-making based on poorly designed spreadsheet models, inconsistencies in data that were not centrally managed, and viruses that spread from one PC to another. In the nineties many companies invested in enterprise resource planning (ERP) software in hopes of speeding up business transactions and improving the transparency of business processes. Along with these benefits came new challenges, including less flexible business processes (since now the process had to conform to the software), issues related to heavy reliance on the ERP vendors, and other challenges that helped confirm the Law of the Double-Edged Sword. Some new IT application categories in recent years include wikis, blogs, mashups, and widgets. Each of these has already given rise to stories of success, failure, and challenge. Next year other applications – including new P2P applications for one-to-one, one-to-many, and many-to-many file sharing, communication, commerce, and collaboration—will inevitably be added to the list.

Every category of IT applications brings both opportunities and challenges for individual users and for corporations, consistent with the Law of the Double-Edged Sword. Thus, employees, managers and citizens need to realistically assess each new application that comes along. Sometimes people focus only on the good side, and fail to adequately prepare for the inevitable challenges. Other people focus only on the risks, and fail to harness the potential of the new technology. Either approach falls short of effectively managing IT. Only by simultaneously considering both sides of the Law of the Double-Edged Sword can managers expect their organizations to realize the awesome potential of new technologies. With that in mind, we offer a set of P2P precautions below. Some

precautions concerning P2P Internet applications – such as *caveat emptor*—mirror those for their traditional counterparts, but in addition, other precautionary measures are also needed. For example:

- Internet service providers should offer clear and user-friendly options for secure composition, transmission, and reception of e-mail messages.
- Providers of VoIP, e-mail, and IM systems (which support the reception, storage, backup, retrieval, and eventual "deletion" of messages) must ensure secure handling of messages and communication events.
- End users of P2P applications should learn about the limits of these applications to protect them against such dangers as identify theft, fraud, communications falling into unauthorized hands, and illegal, improper, or dangerous attachments (e.g. viruses).
- It is important for end users to recognize that the Internet's global reach amplifies the number of adverse events that can occur, and that the "bad guys'" ability to conceal their true identity creates an especially problematic risk.
- End users as well as system providers and other intermediaries need to recognize that even when "bad guys" are caught and sanctioned (such as by blacklisting or blocking access) severe damage may already have been done. Furthermore, since laws vary around the world there may be little if any legal recourse. And, since it is relatively easy to adopt a new online disguise, individuals who fabricate or misrepresent information often return to attack again.
- End users should also learn about the limits of P2P applications to protect them against problems resulting from gullible reliance on misinformation. For instance, many college professors do not allow their students to use Wikipedia as a reference source, since there is only weak assurance that the information posted there is correct.
- End users should also learn about and adhere to ethical principles in the use of these applications, including not sending or forwarding offensive or nuisance messages, not violating intellectual property rights, and (if called for) the reporting such violations to appropriate authorities.
- Employers and other organization which regulate who is allowed to use their communication systems and under what conditions should clearly communicate their policies and procedures and commit to appropriate education and enforcement.
- Organizations and individuals who host bulletin board or auction Internet sites should adopt rigorous policies and procedures—including sanctions such as blacklisting—and clearly communicate them to all participants. The policies and procedures should be strictly enforced.

In April 2005, on the fortieth anniversary of his influential article in *Electronics Magazine*, Gordon Moore predicted that the ability to pack more and more transistors onto a chip (Moore's Law) will continue to hold true for at least a few more decades (Niccolai, 2005). Networking technologies (including computing devices like routers as well as software tools and protocols that help networks work well) have also improved dramatically in the last few decades. In contrast, tools and methods for developing software applications improved at a far slower rate in the eighties and nineties, but the pace has accelerated recently and, with the advent of web services, is expected to continue to gain. Taken together, improvements in foundation technologies and methodologies for computing,

networking and software development will give rise to many new IT applications – including P2P applications—in the coming years, just as these foundation technologies and methodologies gave rise to peer-to-peer file sharing and a host of person-to-person applications for communication, collaboration, and commerce. Smart managers and effective companies will harness these new applications in the service of key business processes and competitive initiatives, while ineffective companies and managers will fail to recognize or capitalize on their potential.

Exercises

1. Given the IT trends discussed in this and other chapters, what scenarios do you foresee for P2P in the coming decade?
2. Does your school have a P2P file sharing policy? Should it? If so, what should it say? If not, why not?
3. The concept of intellectual property involves the right of an individual or organization to profit in some way for the effort invested. Even where profit is not an issue, the concept disallows unauthorized resale and/or tampering with the material. Given the ease with which information can be copied, modified and redistributed over the Internet, should traditional copyright protections still apply, and if so, what jurisdictional recourse is available once national boundaries are crossed?
4. Suppose you were the chief information officer for your school. If a worm propagated through your school's systems as a result of illegal P2P file sharing, what actions (if any) would you take against the students who were participating in the problematic file sharing services?
5. What are some examples of wikis that would be particularly useful in your job or to support your personal interests? Who would be the best contributors to their development? Do the advantages of having access to reliable information in wikis outweigh the dangers of contamination with unreliable information?
6. What are other examples of one-to-one, one-to-many and many-to-many person-to-person systems? What are the benefits and challenges that each application brings?

References

Carmi, Shai; Havlin, Shlomo; Kirkpatrick, Scott; Shavitt, Yuvall; Shir, Eran. A model of Internet topology using k-shell decomposition. Proceedings of the National Academy of Sciences of the United States of America 104 (27): 11150-11154, July 3, 2007.

Dzubeck, Frank. Get ready for corporate P2P apps. NetworkWorld, April 11, 2005. Online at http://www.networkworld.com/columnists/2005041105dzubeck.html.

Graham-Rowe, Duncan. Mapping the Internet. MIT Technology Review, June 19, 2007.

Green, Heather with Lacy, Sarah and Rosenbush, Steve. What comes after YouTube. BusinessWeek, October 30, 2006.

Holahan, Catherine. Advertising to the file-sharing crowd. BusinessWeek, February 26, 2007. http://www.businessweek.com/print/technology/content/feb2007/tc20070226_793620.htm

Kharif, Olga. Waiting for the killer apps. BusinessWeek, August 1, 2001.

Leyden, John. Pfizer worker data leaked via P2P. The Register, June 14, 2007. www.theregister.co.uk2007/06/14/pfizer_p2p_data_leak/print.html

Myer, Michael. Free music downloads, lawsuit not included. Business 2.0 Magazine, April 23, 2007.

Navin, Ashwin. The P2P mistake at Ohio University. C/net news.com, May 7, 2006.

http://news.com.com/2102-1027_3-6181676.html

Niccolai, James. Gordon Moore looks back – and forward. PC World April 18, 2006. Online at www.pcworld.com/article/id,120429/

Planner, Eric. Music industry reaches deal with file sharing site. International Herald Tribune, July 27, 2006

Rosenbush, Steve. Kazaa, Skype, and now "The Venice Project." BusinessWeek, July 24, 2006.

Roush, Wade. Peering into video's future. MIT Technology Review, March-April, 2007.

Stainaker, Stan. Here comes the P2P economy. In: Harvard Business Review Breakthrough Ideas for 2008, Harvard Business School no. R0802a, February 1, 2008.

13. Opportunities for new organizational forms

Editor: Antony Bryant (Leeds Metropolitan University, United Kingdom)

Reviewer: Geoffrey Dick (University of New South Wales, Australia)

Learning objectives

- In this chapter you will be introduced to the following concepts:-
  - Formal & Informal Organizations
  - Horizontal & Vertical Division of Labour
  - Management & Specialization
  - Organizations and the impact of Information & Communications Technology [ICT]
  - Globalization & the Global Economy
  - Outsourcing and Off-shoring and associated business models
  - The Open Source model
  - Linux, Wikipedia and the Wiki model
  - You will also be introduced to the ideas of:-
  - Peter Drucker
  - Henri Fayol
  - Chester Barnard
  - Henry Mintzberg
  - Eric Raymond

Introduction

This chapter discusses the concept of organizations against the context of Information Systems [IS], and then develops this to indicate the ways in which Information & Communications Technology [ICT] makes possible new forms and conceptions of organizations in the 21st century.

At certain points in this chapter you will be asked one or two questions, and also to complete one or two simple tasks. Please give some thought to answering these before continuing to read the material. Also please note down your answers so that you can develop them as further questions are asked, and you are requested to complete some tasks.

What is an organization?

You probably have some idea what an organization is, and can think of some examples; but can you define an organization? Here are some tasks to help you.

1. list some of the organizations you have encountered in any way over the last few years (5 or 6 will be adequate)
2. try to explain some of the features that they have in common
3. then try to produce a definition of an organization, using the examples and features from your list.

The sort of answers you might have given

- a company you have worked for or currently work for
- a company someone in your family works for
- a local sports team
- a government department
- local government
- a religious organization
- radio or TV station
- a group to which you belong that has some common interest – sports activity, political interest, hobby, etc.
- You probably did not find it hard to list some examples. People know what organizations are, and if asked to give examples most people could readily produce a list of five, six or more. Your list probably includes organizations such as where you were educated, where you are currently and/or were previously employed, together with certain government and financial examples. You might also mention organizations connected to some sport or recreation that you follow as a spectator, or that you participate in more actively; and some of these may be less formally recognized than others.

The list might also include large commercial companies (local, national, multinational), government departments, churches or other religious institutions, universities, colleges, labour units, football teams, social clubs, orchestras, and charities.

But you may have had more difficulty in dealing with the second task.

You may well have hesitated when you tried to list what these all have in common. Some of them are what might be termed ‘formal’ organizations with legal charters, definite structures, recognizable characteristics and locations. Others are far less formal, with little or no existence other than through the activities of members.

This difficulty has long been an issue in the study of organizations, and many theorists have argued that it is easier to describe specific organizations than it is to offer a single definition. Peter Drucker, one of the key management thinkers of the postwar period, has argued that we now live in a ‘society of organizations’. But in several of his key works he avoids the issue of defining the term ‘organization’.

Chester Barnard, an early writer on management, did offer a definition of a formal organization in his work on ‘Functions of the Executive’ (1938). He defined a formal organization as a ‘system of consciously coordinated

activities of two or more persons'. This is a very wide-reaching definition and would include two people planning to visit somewhere together, as well as the United Nations or the World Health Organization.

We can expand on this slightly to offer a minimal definition of an organization as something that requires at least two people, who acknowledge each other as members, and have at least one shared objective or common purpose, deliberately working together to attain that objective.

A quick search on the Internet produces the following variations.

- A group of people who work together wordnet.princeton.edu/perl/webwn (http://wordnet.princeton.edu/perl/webwn)
- Basically, an organization is a group of people intentionally organized to accomplish an overall, common goal or set of goals. Business organizations can range in size from two people to tens of thousands. www.managementhelp.org/org_thry/org_defn.htm (http://www.managementhelp.org/org%20thry/org%20defn.htm)
- An organization is a formal group of people with one or more shared goals. This topic is a broad one. en.wikipedia.org/wiki/Organization (http://www.managementhelp.org/org_thry/org_defn.htm)

In order for an organization to exist over any significant period of time – from a few months to many decades – it will also need some resources drawn from its environment. Small organizations may need no more than the time and effort of its members. Larger ones will need a range of resources, particularly financial and material ones. One way to visualize this is to think of an organization as a system – which at the very simplest level can be depicted as a system, with a boundary, an input and an output (and feedback).

add diagram here

The inputs can include time, skills, effort, raw materials, finance, components.

The outputs can include products, services, increased skills, profits, achievement of the primary objective.

Let's think of one of the simplest possibilities; my friend and I decide to go on a walking tour for our vacation. We meet a few times to consider possible places to go, and we eventually agree on a location and a date on which we will start on our tour. We make all the arrangements for travel, accommodation and so on. We start our walk and complete our tour. For the time from which we first decided to go on our tour, to the time when we completed it, we could be considered to be an organization: But a very small and an informal one. If we no longer saw each other afterwards – perhaps we had some major disagreement during the vacation – then our particular organization would simply cease to exist.

Now consider if, after we had completed our tour, we decided that we could book similar tours for our friends so that they would not have to do all the arranging themselves – and they would pay us a small commission for doing this. After a few months we find that we are spending all our time on these activities, not just for our friends but also for others who have heard about our service. In fact we have to employ three other people, and rent a small office, and we can only do this after we have been to a bank to arrange credit to make the down payments and pay the first few months' wages of our new colleagues. After a further period we start to book other sorts of tour – cycling, climbing, sports holidays and many other types. By this time we have become not only a larger organization, but a more complex and formal one.

In fact some other important changes have happened with our organization. We are now using far more resources, including

- other people;
- financial resources – such as the bank loan;
- office space – including overheads such as lighting, energy for heating or air-conditioning, etc;
- phones, computers and printers.

We also have more established links with our environment. For instance we may negotiate favourable terms with hotels or travel agencies for our bookings. We may set up an agreement with an insurance company to cover our bookings and our employees. We will want to ensure that we have relevant sources of information so that we can make and confirm our bookings, contact potential customers and suppliers, and so on.

We have also become more formal and have a structure of some sort

- our employees should have a legal contract of employment detailing their conditions and responsibilities amongst other things;
- we may have decided that one of the three is the office manager, and so is responsible for managing the other two, as well as reporting to the two founders – this would be a form of vertical division of labour;
- we may have divided responsibilities between the employees and ourselves, so that some of us deal with the walking tours, and the others deal with cycling, climbing and so on – this is a form of horizontal division of labour.

Our organization will now have to become a legal entity, which in most countries will mean that it has to be registered and be accountable in some sense. It is also operating as part of an environment which will include customers, suppliers and competitors. This means the organization is dependent on its environment; the economy and market, competitors, government and legal factors, and other external factors and forces. A very large organization will be able to influence its environment to an extent, while smaller organizations will have to adapt to environmental factors if they are to survive. Microsoft and Sony can exert enormous pressure on their environment: our small travel company cannot.

Management and the Division of Labour

Note that a distinction has been made between horizontal and vertical division of labour. The horizontal division of labour refers to the ways in which a large project or task can be split into several smaller tasks. The classic example of this is to be found in Adam Smith's 'An Inquiry into the Wealth of Nations', published in 1776. He noted that if a group of workers wanted to produce metal pins, then they would be far more efficient – producing far more pins – if each person undertook a specialized sub-task than if each person attempted to complete the entire process from beginning to end. A more modern example would be a highly automated factory producing computers or high-definition TVs, with each unit moving slowly along an assembly line with individual workers completing short and highly specific tasks.

The vertical division of labour, on the other hand, implies a hierarchy of command as opposed to merely a differentiation and specialization of tasks. So within the factory context mentioned earlier there may be a supervisor who has to oversee the activities of other workers – as the term 'oversee' implies, the supervisor's

position is considered to be above that of the other workers. Similarly a production manager may be put in charge of several supervisors. The traditional organization chart is a model of this hierarchy, illustrating the vertical division of labour.

In many cases it can be argued that this vertical division also implies specialization of tasks – the supervisor or manager needs to be skilled in dealing with people, monitoring activities, anticipating problems and difficulties, and generally ensuring that things run smoothly and efficiently. This is the point at which 'management' as a specific skill becomes apparent. In a small scale organization management might be carried out by one or more of one's colleagues, who essentially are performing the same tasks as everyone else, but in addition are overseeing the work of others. So in the case of our fictitious travel company, the office manager works alongside the other two employees, carrying out similar tasks plus some of the management tasks mentioned earlier.

Once an organization grows to any appreciable size, management becomes a specialized task in itself. People who manage large departments in government, or who run large private companies will spend their entire time 'managing'. On a farm or in a factory the manager – or managers – will spend most of their time 'managing' rather than working on the farm itself or on the factory-floor. In a large school the head teacher will often do no teaching at all, since all the available time will be taken up with 'managing' the school itself. Now this raises the question – 'What does a manager actually do?'

What Do Managers Do?

Imagine that you are successful in a job application for the post of manager in our small travel organization – now grown to 15 employees. What sorts of responsibilities and activities would be expected of you? How do you expect you would spend your average day at work?

Now repeat the exercise; but this time assume you are to take up the post of manager in a large chain of travel companies, with about 50 shops. Each shop employs 10 staff, both full-time and part-time. You work in the Head Office which employs 80 administrators and support staff. The company is itself part of a larger group with shops and offices in many different countries.

What aspects are common to the two situations? What aspects are different? Try to note down some ideas before reading the next section.

The classical view of management is derived from the work of early theorists such as Henri Fayol (1841-1925). Fayol defined the five functions of management as:

- planning
- organizing
- co-ordinating
- deciding
- controlling

Fayol's experience as head of a coal mine in France led him to identify these functions or activities, and they are related to the ideas discussed earlier particularly those concerned with the two forms of the division of labour.

Using the list you prepared earlier, can you match your ideas against the five functions given by Fayol? It may not always be easy to select just one function. Did you find that some of the tasks you specified do not fit with any of Fayol's five functions?

It is important to understand that although Fayol's ideas might seem fairly obvious and conventional now, they were not really taken up until late in the 20th century. The concept of management as something distinctive did not really become widespread and important until after World War II. Peter Drucker, generally regarded as the most influential management guru of the last 35 years, argues that it was really only after World War II that what we now consider to be the essential aspects of management came into being. After World War II, management became a key focus for 'big business' and the private sector. In particular the practice of management was encouraged in a systematic manner by the head of General Motors, Alfred P Sloan Jr. (1875-1966). Sloan developed a systematic approach to management of large corporations. Many common ideas such as developing business objectives, formulating business strategies and strategic planning were started by Sloan. At this time the first multi-national organizations appeared, including the Unilever Companies that merged Dutch and English organizations. So it can be argued that the core concepts underlying modern management were first formulated in the late 1940s; but it should also be noted that some key features such as leadership, influence and power are far older, dating back at least to the 16th century and the publication of Niccolo Machiavelli's book *The Prince*.

Drucker extends Fayol's ideas by proposing three 'dimensions of management', each of which requires a particular task; each is 'equally important but essentially different':

1. 'to think through and define the specific purpose and mission of the institution, whether business enterprise, hospital or university';
2. 'to make work productive and the worker achieving';
3. 'to manage social impacts and social responsibilities' (p36).

Many might question whether modern management actually encourages the third set of tasks. Some would argue that it certainly does not; some that it should not; and others that it should, but does not.

A more important point is that models such as those put forward by Fayol, Drucker and many other theorists often rely on highly idealized views of the daily routines of organizational reality. This was noted in particular by Henry Mintzberg who demonstrated the discrepancy between what many managers said they did, and what they actually do. He argued that the view of managers as rational decision-makers and planners was at best only partially true. More importantly he identified a series of roles that managers undertake in the course of their activities. These were grouped under the headings – interpersonal, informational, decisional. Each one subdivided as follows:-

- Interpersonal
    - figurehead
- leader
- liaison
- Informational
    - monitor

- disseminator
- spokesperson
- Decisional
    - entrepreneur
- disturbance handler
- resource allocator
- negotiator

The details of Mintzberg's ideas, and the general issues of management are beyond the scope of this chapter; but it should be noted that the concepts of organization, division of labour, and management are closely related to one another, although their actual realization in practice will depend on many factors, including economic, cultural and social ones; but more particularly for the purposes of this chapter the impact of technology.

Development of information and communication technology

Information and communications technology (ICT) changes the extent to which organizations have to be located in a particular place, with all – or almost all – the important functions occurring face-to-face in a specific location. Early factories had to be built close to sources of power and if possible key raw materials and workers. Inside the factories the workers themselves were closely watched and monitored. As technologies have developed many of these issues have become less important. When people talk about the 'global economy' they often have in mind the ways in which processes of production and manufacture can now extend across continents and time-zones in ways that were not possible in the early days of industrialization

As the impact of technological development is realized what was previously assumed to be the only way of doing something is seen as just one possibility amongst many. At the same time the core aspects become more prominent. An example of this, and one relevant to this chapter, is 'office automation'. This became a popular idea in the 1980s as computer technology became more widely available and affordable – particularly with the appearance of the personal computer [PC]. This information technology [IT] readily lent itself to many of the tasks associated with the office – typing letters could be done more efficiently and effectively with word-processing software; filing could be done using database software if the material was in electronic form; calculations and estimating could be accomplished with spreadsheets; some document handling could be accomplished using fax machines. The technology was seen as a solution to many problems faced by organizations – small and large – such as delays in sending out letters and invoices, losing important documents, staff shortages, and so on. This led some people to conclude that the office in its previous form would disappear. Why maintain a special space and group of staff, when all the key functions could be carried out by technology? This proved to be as false and unfounded as the idea of 'the paperless office'. The outcome of office automation was a better understanding of the role and nature of 'the office' in organizational life. It was not simply somewhere that letters got typed, and papers got filed; but the site of many other activities, many of which were essential to the smooth running of the organization. The office was a space where people met colleagues and engaged in informal discussions, where rumours and gossip were exchanged, and so on. The office was not simply a place but the location for a whole range of processes and interactions. The introduction of IT and other technology changed people's ideas about what actually went on at people's place of work, in many cases altering or challenging long-held the assumptions.

The impact of IT since the 1980s on administrative and secretarial type activities has led to a dramatic restructuring in many organizations. Some of the core functions remain, but with an altered emphasis as a result of incorporation of new technology and new ideas about the role and purpose of these functions and activities.

A similar pattern has come about in many other aspects of contemporary organizations. An increasing range of organizational activities are now bound up with ICT. This has meant that many aspects regarded as essential to the smooth operation of an organization have become topics for discussion and re-evaluation. In some cases these reconsiderations apply to specific types of organization; but in many cases they have a far wider and more general scope.

The general issues develop from the potential for technologies, particularly but not only ICT, to allow a far wider range of options for an organization to exist and function. An early example was the way in which factories reduced the levels of their inventories – i.e. the raw materials or components needed to produce their finished product. The earliest pioneers of this were in Japan, where manufacturers were encouraged to aim for 'zero inventory'. In other words goods delivered to the manufacturing site were not booked in to stock areas, waiting to be used at a later date – and so consuming space, effort and money. Instead the delivered items were immediately sent to production and manufacture on a 'just in time' [JIT] basis. For JIT to work in practice, there needs to be a fairly accurate model of material requirements, together with reliable and responsive communications links between the manufacturer, suppliers and transport (what is now often referred to as 'logistics'). Without these just-in-time is in danger of becoming just-too-late. But if such facilities can be assured then the potential cost savings and efficiency gains can become a reality.

In the 1970s when these ideas were in their infancy the relevant technology was fairly low-level compared to what is now potentially available – although not universally widespread. The Internet, particularly email, real-time communications, mobile phones, on-line tracking, developments in e-commerce and the like have all had an impact. Initially this impact was centred on the supply and production part of the value chain, but more recently it has also affected the consumer. This has been described by Sviokla and Rayport as the move from 'market-place to market-space'.

Sviokla and Rayport argue that value for consumers is created by three components which are usually found together, but which with the development of the internet and e-commerce have become distinguishable. Moreover organizations can position themselves to focus on one or two, rather than all three. The three components are; content – what is offered; context – the form in which it is offered; infrastructure – how it is delivered or distributed.

They offer as an example a newspaper. Until recently the first two value components were tightly bound together. The organization that produced the paper was also responsible for printing multiple copies and delivering them somewhere from where they could be sold to the readers. In some cases the final part of the logistics – the delivery – involved it arriving at the consumer's house in a more-or-less readable condition. With the advent of the Internet, email, RSS feeds and a whole host of other alerting and delivery possibilities all three components have become far more flexible. A consumer can still purchase a newspaper in the traditional manner, but there are also other options including paying for an on-line service by an internet service provider, receiving an email with the document attached ready for printing, headlines and extracts sent to one's mobile phone and so on. Here again is

an example of the way in which ICT and related technologies dismantle existing structures and open up new possibilities.

An interim summary

So as we move towards the latter part of the first decade of the 21st century we are all bound up with a movement that has, amongst other effects, resulted in a dismantling of the ways in which organizations need to operate. They no longer need to be situated in a single location, their routine operations can be tightly linked to other organizations, and aspects of their routine existence such as division of labour and management are open to several possible alternative forms.

In manufacture this can be most widely understood as an increasing number of manufacturing processes are dismantled and spread across the globe in what is one of the primary and most visible forms of 'globalization'. The global economy is a complex concept, but one of its key characteristics is the way in which a finished product available for sale in a shop in, for example, the USA or Western Europe has passed through a series of stages of manufacture and packing that may have taken in factories and warehouses in Asia, Africa, and Eastern Europe.

At an abstract level these sorts of developments have been widely discussed and investigated under such headings as 'the virtual organization' and 'computer supported cooperative work' [CSCW] – both terms originating in the 1980s. In the commercial domain virtual organizations and CSCW have taken on the form of 'out-sourcing' or 'off-shoring' both or goods and services. Some people have welcomed these developments on the grounds that they offer employment and development opportunities (both individually and more generally) to areas that previously have been deprived and under-developed. Others criticize such moves as simply perpetuating dependency and under-development, since these strategies are all-too-often driven by the aim of cutting costs and so involve child labour, very low wages, and poor and dangerous working conditions. A further criticism from developed countries is that such practices move jobs away from developed economies, and so deprive employment opportunities to lower-skilled people in those countries.

Virtual organizations and CSCW have also had an impact in the non-commercial sectors – affecting NGOs, civil society organizations [CSOs], community organizing and many other forms of collective activity. These issues will be discussed further in the later sections of this chapter. For the moment it needs to be understood that all the developments discussed so far lead to the necessity to rethink the 'value chain', the role of management, organizational forms, the division of labour, the nature of competition and cooperation.

New Models of Organization

In response to the developments discussed above people have started to question traditional models of the organization, management and associated concepts. Two concepts in particular are worthy of further analysis - the virtual organization, and the open source model: Each in its own way demonstrating opportunities for new organizational forms.

Virtual Organizations – Outsourcing and Off-shoring

A virtual organization is one that exists very much along the lines of the earlier example of two friends getting together with a very specific and limited objective; usually a single project. To an extent virtual organizations have always existed, but only with the advent of the internet have they really flourished. Virtual organizations are virtual in that they do not exist as organizations in the formal sense, but they do exist in the sense of having an on-line existence. In many cases commercial virtual organizations – or virtual enterprises – were the most visible examples

of this trend in the 1990s. In these cases several already existing commercial organizations would collaborate on a specific project, often an exploratory one or one in which some new idea or product required development and testing. These early forms of virtual organization were largely established as loosely linked alliances; with little or no formal structure or hierarchy. In terms of division of labour, there was certainly a horizontal form, but little or no vertical form. The emphasis was on trust and collaboration, rather than command and control.

In the period since the 1990s many organizations have emerged that are part-virtual, part-actual; making use of models centred on a strategy of 'out-sourcing' or 'off-shoring'. These organizations grow from existing, traditional forms, taking advantage of the significant developments in ICT that allow cheap, fast, and reliable forms of communication and monitoring. A company that previously manufactured something from start to finish in one location may now be dismantled so that some of the tasks are completed elsewhere – in a different state or even in a different continent. This may be done for a variety of reasons; perhaps the initial raw materials or components are more readily available, and cheaper, elsewhere: Or a particular task or process is done more efficiently and effectively by a specialist company. In many cases out-sourcing or off-shoring is seen as a way in which commercial organizations can reduce their labour costs, locating some or all of their core labour-intensive processes where the workforce is less expensive to employ and retain. Thus some aspects of the organization's activities become virtual, with links, collaborations and associations often being temporary or on an 'as needed' basis. For instance Wal-Mart, Nike and Dell have grown using a strategy of being 'highly decentralized' or 'highly distributed'; they exist physically and over time in the sense that they have headquarters, offices, depots and so on, but in other regards many of their operations are 'virtual'.

At one extreme outsourcing simply becomes a form of extension of a traditional organization. All the key aspects remain, but the core processes are undertaken in a slightly modified form. So the chain of activities and processes leading to the finished product are dispersed across the globe. Nike was one of the first companies to use this as a central part of their organizational operation, locating almost all of the production of their sports and leisure wear to countries outside the US. The actual number of people directly employed by Nike was always very small given the size and turnover of the company; with most of these employees being based in the United States.

Dell adopted a customer-directed business model, essentially reducing its inventory costs by assembling computers on demand, and only accepting components for delivery and payment when actually needed for assembly. This allowed the company to take payment for its finished products before they were actually assembled, so avoiding the costs of paying for the components in advance and then waiting for the orders. They were able to do this because they quickly saw the ways in which the internet could be used to re-engineer many of the core organizational and management processes. Thus they were able to achieve co-ordination of the entire process from manufacture to assembly to delivery. They could tie-in suppliers so that supplies were only delivered as needed. Moreover, since Dell only sold directly to customers, they were able to use this relationship to promote follow-on sales to their customer base.

The ways in which an organization such as Dell or Nike operates can be outlined using the concepts introduced earlier. In terms of the horizontal division of labour, the range of activities and processes involved occur along the same lines but can now be located where raw materials, labour costs, economies of scale, specialized skills or other factors can be optimized. The internet offers the rapid communication infrastructure for co-ordinating and controlling these activities. Companies that have pioneered and taken advantage of these potentials include Toyota and Wal-Mart. In recent times Wal-Mart and Nike, amongst others, have been heavily criticized for the ways in

which they have implemented their particular forms of outsourcing. These criticisms include using child labour and other forms of non-unionized cheap labour particularly in third world countries, flouting environmental legislation, and selling at below-cost prices in order to drive other, smaller companies out of business. These criticisms and rebuttals from Nike, Wal-Mart and others are all well documented on various websites – Wikipedia is a good place to start if you wish to look into these issues in further detail.

In this chapter these arguments can be put to one side. Whatever the rights and wrongs of the ways in which these commercial companies have taken advantage of technological advances, they clearly demonstrate the potential for ICT to facilitate new forms of organization. This can be seen by a fairly simple reconsideration of Fayol's five functions of management and how they have been affected by ICT;

- Planning; this can now be undertaken on a more-or-less continuous basis. Increasingly organizations are focusing on far shorter-term planning for at least two reasons
    - the global context is changing so quickly that long-term planning will be at best only of limited value
    - communicating planning decisions can be done quickly and effectively, and so can any new plans or changes in direction. It should be noted, however, that this does not necessarily mean that planning is any more accurate or effective than it was previously – the concluding section of this chapter considers this in more detail.
- Organizing; the ways in which tasks, people and groups need to be structured can now be established and then changed quickly and easily. As will be explained briefly below, organizations can now function as flexible networks, altering their patterns of communication and interaction to take best account of changing circumstances, priorities and opportunities.
- Co-ordinating; The model of Just-in-Time production can now be extended to services as well as manufacturing activities, since ICT allows fast and efficient communication. The models used by Toyota, Wal-Mart and many other companies rely on co-ordination of a whole range of core and support activities dispersed across the globe. In recent years these models have themselves come in for intense criticism, and some of the pioneering organizations such as Wal-Mart and Dell have sought to make significant changes in the ways in which they organize and co-ordinate their core processes – see the concluding section.
- Deciding; decision-making is often thought to rely on information to the extent that more information is likely to produce better – i.e. more accurate – decisions. This has led to a demand for real-time updating of many aspects of an organization's activities. This does not always lead to better decisions.
- Controlling; Based on the points about the other four functions, the ability to control widely dispersed but potentially well co-ordinated activities should be well established. ICT certainly allows this, although other – non-technical – issues can over-ride this.

Open source and related models

Wal-Mart and Nike can be seen as examples of firms that have established themselves using novel forms of organization and co-ordination, but still centred on what has been termed a command-and-control model. This model is most often associated with military organizations and also with governmental ones, where there is a centralized structure and all communications and decision-making goes through this centre. One of the weaknesses of such an approach is that if anything happens to the central command, or to its ability to communicate with the

periphery, then the entire organization can be prevented from operating. In the 1960s the US Department of Defense identified precisely this weakness in its reliance on a centralized structure, and so encouraged the development of what became known as ARPANET – a network of inter-related computer-based communication sites. The technology at the heart of ARPANET became the basis for what we now know as the internet – a network of networks. It is this model and also this technology that has prompted and promoted new forms of organization that have moved away from – or beyond – the command-and-control structure.

The most visible form of this model is probably Wikipedia and the various other forms of Wiki now available. But one of the key building blocks was the open source software movement. We now all rely on software in a whole host of various guises. Software is produced by people who often identify themselves as software engineers, and in the earliest days (1950s) of software development large-scale software was often seen in terms of a major construction project requiring a command-and-control approach to its project management. But since at least the 1970s there have been software engineers who have sought a different approach. This was outlined by F.P. Brooks in 1986 when he argued that instead of thinking about software as something to be built, it might be more useful to think in terms of growing or cultivating software. Large complex software systems could then be seen as things that grew, incrementally in stages, rather than as one-off large-scale constructions. By arguing in this fashion Brooks was countering the view of software development as a large-scale engineering project – almost inevitably managed in a command-and-control manner; instead seeing it as far more untidy and disorganized.

This idea of cultivating software was linked with the idea of developers contributing their efforts for the common good rather than on the basis of some commercial contract or agreement. The idea might not have developed much beyond the confines of this very specialized group without the contribution of Linus Torvalds. Torvalds at the time (1991) was a graduate student who had developed his own operating system, Linux, as part of his studies. He allowed anyone who wished to do so to contribute to this system, revising and enhancing it. From this basis the Linux operating system has flourished into the open source model of development and collaboration. If Torvalds was the initial driving force behind the model, Eric Raymond must be regarded as its chief advocate, supplying the open source manifesto in his paper contrasting the cathedral with the bazaar.

For Raymond the image of a cathedral is a model that conjures up a vast, complex structure developed by people with near-magical skills and powers; Raymond even calls them wizards. These people work in 'splendid isolation' developing a product that needs to be completed and fully guaranteed or secured prior to its 'release'. Raymond contrasts this to 'a great babbling bazaar of differing agendas and approaches … out of which a coherent and stable system could seemingly emerge only by a succession of miracles'. Raymond specifically centres his writings on software development, more specifically on software debugging – the process of locating and fixing problems in software-based systems: A process that is truly endless in all commercial systems. The cathedral model relies on a small group of proficient developers working in splendid isolation, only releasing their software to users after extensive and thorough testing – all of which takes time and effort. In stark contrast stands the bazaar-like model, whereby disparate groups and individuals with differing agendas and approaches somehow produce a coherent and stable outcome.

The Linux philosophy is encapsulated in Linus Torvalds' philosophy as stated by Raymond – 'release early and release often; delegate everything you can, be open to the point of promiscuity'. The result ought to be chaotic and anarchic, a hotchpotch of different versions of software, proliferating to the consternation of developers and the despair of users and customers. Yet precisely the opposite has occurred. Linux has survived, thrived and continues

to flourish. Moreover the model has been used and adopted with regard to other activities, although Raymond and many others associated with the open source software community are very hesitant about such claims and developments.

For the purposes of this chapter the bazaar model indicates that it might be possible to undertake even fairly complex development projects without having to resort to command-and-control policies; instead relying largely on voluntary and collaborative involvement. To an extent in such cases there is a level of control and management, but it is carried out in a more distributed and semi-autonomous fashion. People take on tasks and responsibilities because they feel motivated to contribute and exchange their views, their ideas and their efforts. The development of open source software – specifically Linux – has come about on a model based on several principles that at first sight ought not to prove effective or successful; but they have, and they do. The question at this point is can such principles of operation be applied to organizations in 21st century?

One of the key issues, and one that comes through very clearly in Raymond's description of his own participation in the open source community, is that involvement must at least owe some of its initial impetus to the motivation and enthusiasm of the participant. Peter Drucker is credited with the statement that 'in the knowledge society we are all volunteers'. This might appear strange to most of the working population who are certainly not engaged in their particular employment voluntarily. In fact Drucker's saying is more insightful in its complete form: 'Everyone in the knowledge economy is a volunteer, but we have only trained our managers to manage conscripts'. For our present purposes the key points are developed by Drucker himself in an article for Forbes Magazine in the late 1990s.

What motivates workers -- especially knowledge workers -- is what motivates volunteers. Volunteers, we know, have to get more satisfaction from their work than paid employees precisely because they do not get a pay check. They need, above all, challenge. They need to know the organization's mission and to believe in it. They need continuous training. They need to see results. Implicit in this is that employees have to be managed as associates, partners -- and not in name only. The definition of a partnership is that all partners are equal. It is also the definition of a partnership that partners cannot be ordered about. They have to be persuaded. (Drucker, 1998)

This is clearly at the opposite end of the spectrum from commercial organizations such as Dell, Nike and Wal-Mart. Perhaps these companies would like to think that concepts of partnership, challenge and the like ought to motivate workers; but it is all too obvious that this has little relevance to the vast majority of people's experience.

On the other hand Raymond's characterization of the open source model demonstrates many of the features to which Drucker refers. The participants are motivated by the challenge, and the capacity to act as partners. A similar motivational mix can be demonstrated by many organizations operating in the voluntary and civil society sectors – including many NGOs.

It should be noted that this model is really only possible given the existence of the internet. Without such resilient, extensive, and virtually effortless communication the open source community simply would not have been able to develop as far as it did. The arrival of Linux coincided with the point at which the Internet, as a key component of everyday life for a significant and rapidly growing proportion of people, was taking up its central role as the communications technology par excellence. The bazaar model, with its 'great babbling … of differing agendas and approaches', demands constantly available and extensive means of communication and co-ordination. Raymond recognizes this in asserting that: 'Provided the development coordinator has a communications medium

at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one.' The importance of the internet is not that it allows individuals to talk to each other on an individual basis, but that it affords a forum for exchange and co-operation; with both asynchronous and synchronous interactions.

It should also be noted that this grouping might include co-developers, suppliers, users and advisers. In fact anyone who feels that might have some thing to contribute to the specific task or project. In this case the organization is virtual not merely in the sense of being on-line and taking advantage of ICT, but in the sense of not really existing. Outsourcing and other forms of virtual organization still have an existence as organizations, but at this end of the scale a bazaar-like organization may well have only a fleeting existence for as long as the project or task is relevant.

A further look at Fayol's five functions indicates how the open source model is distinctive – at least in its most extreme form.

- Planning; any planning is restricted to the extremely short-term.
- Organizing; this is most definitely accomplished in an informal and bottom-up fashion – similar in some regards to the very small, informal organization mentioned at the start of this chapter.
- Co-ordinating; again this is done on the smallest of scales, based on direct interaction with other members.
- Deciding; there is no single, central decision-making; individuals make decisions and these then contribute to the overall development of the loose alliance,
- Controlling; there is no single point of control.
- Three of Raymond's maxims, taken together, summarize the position. 'The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.' [#11] 'Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.' [#12] And even more insightfully he states that – 'Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.' [#8]

Taken together these offer an outline of a form of organization that is best seen as a loose alliance of interested and motivated people, acting together in ways that may achieve both expected and unexpected purposes and results.

An example of this model, with which you may be familiar, is the Wiki, particularly in the form of Wikipedia. The term Wiki seems to have several meanings and derivations. The word itself means 'quick' or 'fast' in Hawaiian, and the slogan WikiWiki is apparently used by the shuttle bus company at Honolulu International Airport. It is also claimed that Wiki is an acronym for 'What I Know Is'. Hence the term has come to denote collaborative efforts where people come together to pool their knowledge and expertise with a minimum of fuss and formality. In many regards the Wiki principles are more easily understood from stating what the Wiki movement is not, rather than what the Wiki movement actually is. Hence the following headings from the Wikipedia entry on Wikipedia itself;

- What Wikipedia is not
    - Wikipedia is not a paper encyclopedia
- Wikipedia is not a dictionary

- Wikipedia is not a publisher of original thought
- Wikipedia is not a soapbox
- Wikipedia is not a mirror or a repository of links, images, or media files
- Wikipedia is not a free host, blog, or webspace provider
- Wikipedia is not an indiscriminate collection of information
- Wikipedia is not a crystal ball
- Wikipedia is not censored for the protection of minors
- What the Wikipedia community is not
    - Wikipedia is not a battleground
- Wikipedia is not an experiment in anarchy
- Wikipedia is not a democracy
- Wikipedia is not a bureaucracy

In fact the Wikipedia organizational model relies on a sufficient number of people feeling motivated and enthused to contribute and participate; exactly the same prerequisites that Drucker identifies for volunteers or associates, and that Raymond describes for the Linux participants. Moreover the Wiki philosophy is best seen as cultivation as opposed to construction. The Wiki model and the open source model share a view of organization that is chaotic and disordered, but also more extensive, more accessible, more visible, and more speedily updated and corrected than standard command-and-control centralized organizations.

Many of you will be familiar with Wikipedia, but there are numerous other forms of 'on-line bazaar', with people contributing voluntarily - you can probably name several, and may well already be signed up to sites such as MySpace and Facebook.

Critique and Conclusions

The range of possible opportunities for new organizational forms includes both those devised around the business models of Dell, Nike, and Wal-Mart amongst others, and the open source and Wiki approaches. The criticisms of Wal-Mart's approach have already been hinted at, and in recent years the ability of these approaches to continue to deliver benefits, returns and profits in a consistent and reliable manner has been brought into question. Only recently Dell has gone through a dip in performance and a crisis in its organizational structure and business model. Similarly Wal-Mart has begun to change its model of operations and focus less on cost-savings and more on customers.

On the other hand the bazaar model is also open to criticism and re-evaluation. Raymond himself makes a key point about the bazaar model: 'one cannot code from the ground up in bazaar style'. In other words, for the bazaar-model to work something, perhaps more cathedral-like, must already be in existence. This can be illustrated with regard to the development of Wikipedia and Wikis in general. They could only develop once the internet was a reliable and accessible reality. Moreover in the years since they first emerged the Wiki model has undergone various developments which take it away from total reliance and commitment to a bazaar of multiple babbling agendas. There is now a form of hierarchy within the Wikipedia community, effectively a horizontal and vertical division of

labour, although it is far less formal and more fluid. Similarly developing features may be found within the open source community.

So it is important to note that these new 'opportunities' require careful thought and reflection. This is not to imply that the command-and-control model is immune from precisely the same criticisms. On the contrary, developments such as outsourcing and open source ventures demonstrate that there are good grounds to challenge the general arguments that justify the need for command-and-control management. Raymond goes even further in stressing that the overheads required for these forms of management cannot be justified; they do not even deliver what they are meant to do.

In the discussion of office automation it was noted that technological advances often result in undermining what was previously assumed to be the only way of doing something; opening out other possibilities or opportunities: So too with the development of ICT in general and the internet in particular with regard to organizations and management. The context now is that there needs to be recognition that the functions identified by Fayol, and the roles described by Mintzberg and other management and organizational theorists, all need to be re-evaluated in this new context. Some indication of this has already been given with regard to Fayol's functions.

The opportunities for new organizational forms made possible by ICT provide a basis for innovation and also for re-evaluation of traditional and accepted ideas about organization and management. Raymond, despite arguing in some places against the application of the open source, bazaar-like model to contexts other than software development, clearly sees the open source experience as offering a glimpse of a new way of organizing. He ends his classic paper with a quote from the 19th century Russian anarchist Pyotr Alexeyvich Kropotkin:

> *Having been brought up in a serf-owner's family, I entered active life, like all young men of my time, with a great deal of confidence in the necessity of commanding, ordering, scolding, punishing and the like. But when, at an early stage, I had to manage serious enterprises and to deal with [free] men, and when each mistake would lead at once to heavy consequences, I began to appreciate the difference between acting on the principle of command and discipline and acting on the principle of common understanding. The former works admirably in a military parade, but it is worth nothing where real life is concerned, and the aim can be achieved only through the severe effort of many converging wills.*

What ICT offers are various ways in which command-and-control models might be realized on a global scale, but also a host of possible alternatives and opportunities.

Summary

This chapter has sought to outline some central issues for an understanding of the ways in which organizations have developed, and the impact that developments in ICT in particular have had; causing people to investigate and question existing ideas and assumptions, challenging these with innovative ideas. The net result is that to some extent the nature of an organization is now more clearly understood, while in other respects the range of possibilities for organizational development and continuity has grown.

Chapter editor

Antony Bryant

References

14. Information systems security

Editor: Gurpreet Dhillon (Virginia Commonwealth University, VA, USA)

Learning objectives

•

Background

The security of information systems has always held a relevant position in CIO's agendas. However, in the past years, information security related issues have been brought to the public fore as a consequence of media attention to such incidents as the collapse of Barings Bank, Enron and WorldCom, and the security lapses at ChoicePoint, Bank of America, T-Mobile, and LexisNexis. The consolidation of IS security as an important topic in today's business world results from the interaction of several technological and social factors.

First has been the increased dependence of individuals, organizations, and societies on information and communication technologies. As individuals, we rely on these technologies to execute a wide spectrum of tasks, from communicating with other people, improving our job performance, accessing various sources of information, booking flights, or buying a book. For organizations, information and communication technologies are not only a major component of basic operational systems and an enabler of productivity improvements, but also a means for gaining competitive advantage, developing new businesses, and promoting new management practices. As a society, it is enough to consider the role played by these technologies in the working of critical infrastructures, from transportation to energy supply and financial services, as well as in the provision of public services to citizens and companies.

Second, resulting from the exploitation of information and communication technologies' capabilities in the business arena, the whole business model for many organizations has been transformed. In the past, companies could rely on staying in a particular geographical area to conduct their activity. However, developments such as global scale interconnectivity, distributed processing, explosive growth of the Internet, open architectures, liberalization of telecommunication markets, and e-commerce diffusion have dramatically changed the business landscape. Today, employees experience increasing mobility, which results in the need to access information by diverse means, in different situations, and from distinct places, many of these outside their own organization. As an implication of this increasing location independence, companies are finding themselves to be strategically disadvantaged if they are confined to a particular place.

Advances in information technologies and the changing boundaries of the firm have stressed the importance of information. It is information that helps companies realize their objectives, keeping them in touch with their environment, serving as an instrument of communication, helping managers to take adequate decisions and providing support for the exchange of employee knowledge (Chokron and Reix 1987).

In the past, information to a large extent was confined to a particular location and it was relatively easy to preserve its confidentiality, i.e. restricting access to those authorized. Because information was usually processed in a central location, it was also possible, to a reasonable level of certainty, to preserve its integrity, i.e. ensuring that its content and form were not subject to unauthorized modification, as well as maintaining the availability of information and related resources, i.e. preventing their unauthorized withholding.

Maintaining confidentiality, integrity, and availability were the three main goal of managing security. Today, considering the transformed nature of organizations and the expanded scope of information processing, managing information security is not just restricted to preserving confidentiality, integrity, and availability. The emphasis should move to establishing responsibility, integrity of people, trustworthiness, and ethicality (Dhillon and Backhouse 2000)

Yet, the protection of IS assets is not an easy task. There is often a lack of a unique and well-defined purpose to protecting these assets. Besides the preservation of confidentiality, integrity, and availability, among other immediate goals that an organization may pursue in securing its information system may be the maintenance of privacy of the data related to their employees, customers, and partners; the minimization of the effects resulting from the dependence on non trustable or non reliable systems and entities, and the resilience to technological malfunctions (Neumann 1994b).

The achievement of many of these goals raises significant difficulties for organizations because they may be conflicting. An organization must be closed to intrusions, fraud, and other security breaches, and at the same time it needs to remain open in order to share information with its partners and customers (Erwin 2002). As well firms face –the continuous development of new forms of attacks, the discovery of new vulnerabilities in technologies and business processes, and the increased need for organizational flexibility.

In order to manage IS security, organizations have to encompass a broad set of factors, ranging from the technical ones, to the consideration of the business environment, organizational culture, expectations and obligations of different roles, meanings of different actions, and related patterns of behavior. This means that IS security can be understood in terms of "minimizing risks arising because of inconsistent and incoherent behavior with respect to the information handling activities of organizations" (Dhillon 1997, p. 1). These inconsistencies and incoherencies in behavior may lead to the occurrence of adverse events. Besides losses from natural causes, such as fires and floods, the majority of adverse events can be traced back to deliberate or non-deliberate inappropriate behavior of individuals, whether in the form of human error, systems analysis and design faults, violations of safeguards by trusted personnel, system intruders or *malware*, such as viruses, worms and Trojan horses (OTA 1994).

In order to prevent, detect, and react to the occurrence of these events, organizations may apply a set of measures usually know as security controls. Because information handling in an organization can be undertaken at three levels – technical, formal, informal (Liebenau and Backhouse 1990) – information systems security can be achieved only by coordinating and maintaining the integrity of operations within and between those three levels (Dhillon 2007). This implies that an organization should adopt a holistic posture in managing IS security, namely by implementing a set of security controls that as a whole support the integrity of the organization's IS.

At the technical level, an organization may adopt security controls such as anti-virus software, firewalls, intrusion detection systems, access control devices, and cryptographic controls.

To be effective, the deployment of technical controls requires adequate organizational support. Consequently, formal controls need to be put in place. These controls take the form of rule-based formal structures that assist on determining how specific responsibilities are allocated and define the consequences of misinterpretation of data and misapplication of rules. Security policies, structures of responsibility and contingency plans are examples of formal controls.

The previous two levels need to conform to the normative schemes prevalent in the organization. At the informal level, measures such as awareness programs, adoption of good management practices, and development of a security culture that fosters the protection of information assets are illustrative of security controls.

In recent years organizations have fallen short of developing adequate security controls to deal with information security problems. Various studies have reported significant losses in explicitly reported security breaches (Garg 2003; Gordon et al. 2006) and as a consequence of computer crimes because of violation of safeguards by internal employees of organizations (Dhillon 1999a). Not only are organizations suffering from a 'policy vacuum' to deal with information security problems, as well authorities have been experiencing a certain inability to establish an adequate basis to deal with such cyber crimes.

Consider the case of Randal Schwartz, a well known programmer and author of programming books, in which it was difficult to establish whether illicit use of computers by Schwartz amounted to a computer crime (Dhillon and Phukan 2000). In 1995, Schwartz was brought to trial for illegally bypassing computer security in order to gain access to a password file while working as a consultant for Intel. According to Schwartz, he was only trying to show that Intel employees were selecting weak passwords that could be easily guessed by crackers who then could compromise information security. Schwartz was convicted on three felony counts, but in 2007 his arrest and conviction records were sealed through an expungement action.

Advances in information technologies have introduced another kind of problem for organizations which many classify as 'input crimes' (Dhillon 1999b). In one case, a former employee of a wholesaler was convicted under the UK Computer Misuse Act when he obtained for himself a 70 percent discount when the regular staff of the wholesaler was otherwise engaged.

Given the increased dependence of companies on information systems, one would assume that most firms would have well established contingency and disaster recovery plans. Unfortunately research seems to suggest otherwise (Adam and Haslam 2001). Many managers tend to think that contingency and disaster recovery planning is an irrelevant issue and hence prefer to concentrate on projects that generate direct revenues.

It emerges from the prior discussion that IS security management is a complex task that poses a number of challenges for maintaining the integrity of information handling activities in an organization. The challenges

In a climate where incidents of computer crime, information security problems, and IS enabled frauds have been on the increase, any attempt to deal with the problem demands an adequate understanding of the four challenges that organizations must confront, namely

- Establishing good management practices in a geographically dispersed environment and yet being able to control organizational operations.
- Establishing security policies and procedures that adequately reflect the organizational context and new business processes.

- Establishing correct structures of responsibility, given the complex structuring of organizations and information processing activities.
- Establishing appropriate contingency plans.

Several authors such as Dhillon (1997), Dhillon et al. (2004) and Siponen (2001) have noted the There is a major problem in managing information security, especially with respect to regulating the behavior of internal employees (Dhillon 1997; Dhillon et al. 2004; and Siponen 2001). Internal employees frequently subvert existing controls to gain an undue advantage because either an opportunity exists or they are disgruntled (Audit Commission 1994; Backhouse and Dhillon 1995). This problem gets compounded even further when an organization is geographically dispersed, and it becomes difficult to institute the necessary formal controls. This was evidenced in the case of Nick Leeson, who brought about the downfall of Barings Bank in Singapore. Barings collapsed because by its reliance on information technology for IS security, Leeson was able to successfully conceal the positions and losses from the Barings management, internal and external auditors, regulatory bodies in Singapore, and the Bank of England. Leeson's case is illustrative of breaches of control, trust, confidence, and deviations from conventional accounting methods or expectations.

The management of Barings had confessed in internal memos that clearly its systems and controls were distinctly weak. However, there was nothing new in this confession, and it has long been known that lapses in applying internal and external controls are perhaps the primary reason for breaches in information security (Audit Commission 1990; 1994). Failure of management to curtail Leeson's sole responsibilities, which empowered him to create an environment conducive to crime, lack of independent monitoring and control of risk, communication breakdown between managers, and the believe that IS can overcome basic communication problems in organizations were other reasons that created an opportunity for Leeson to deceive many.

There is also the challenge of establishing appropriate security policies and procedures that adequately reflect the organizational context and new business processes. This challenge is present at two levels. First, at an internal organizational level, businesses are finding increasingly difficult to develop and implement appropriate security policies. Second, at a broad contextual level, it is becoming less effective to rely on traditional legal policies to regulate behavior.

At an internal organizational level, there is a problem with respect to establishing security policies. This problem stems directly form a general lack of awareness within organizations that such a need exists. Based on a longitudinal study of information security problems within the health services sector and the local government councils, Dhillon (1997) advances two reasons that explain this state of affairs. One of the reasons is the lack of commitment from top management in the security policy formulation process. The other reason is that security policies are conceived in a formal-rational manner. Indeed, the assessment of security problems is characterized as 'acontextual' and the organizational responses to address the security issues are at the best superficial.

At a broad contextual level, although a number of regulations have been enacted in recent years, sometimes their nature and scope seems to be at odds with the reality. Clearly there are a number of situations where it is important to institute punitive social controls in order to curtail criminal activities and in some cases to recover stolen money or goods. There are perhaps a number of other computer crimes where severe punitive control may not be the best option. In many cases monetary gain is not the prime motive, but the intellectual challenge of

tearing apart computer systems. In such cases it would perhaps be counter-productive to institute severe punitive controls.

Another challenge in managing information system security is the establishment of correct structures of authority and responsibility. The inability to understand the nature and scope of such structures within organizations or to specify new ones aligned with organizational routines and goals are a source of information security problems. One example of this kind of problems comes from Daiwa Bank. When this Japanese bank fell short of understanding the patterns of behavior expected of businesses operating out of the USA and allowed Japanese normative structures to dominate, it resulted in a bond trader, Toshihide Iguchi, accruing losses to the tune of $1.1 billion. At the same time, it also allowed Iguchi to engage in at least 30,000 illicit trades. The drama ended in Iguchi being prosecuted and Daiwa's charter to conduct business in the USA being suspended.

Situations such as the one illustrated by Daiwa pose a challenging issue of managing access to information processing facilities. It is insufficient to merely stating 'read only' or 'write only' accesses according to an organization's hierarchical structure, especially in light of the transformation in organizational forms. Modern enterprises are in a constant state of 'schizoid incoherence,' and there are very short periods of stability in organizational forms (Dhillon and Orton 2000). This is especially true for businesses structured in a 'networked' or 'virtual' manner. As a consequence of the evolving nature of organizational forms, the applicability of formal methods for instituting access control is open to debate.

The last challenge concerns dealing with contingency plans, namely information technology disaster recovery plans and policies. These plans have a central place in today's technological dependent business world. However, their success is not only a function of the ability of an organization to recover its technical infrastructure capability, but also of its capacity to replicate and apply business process knowledge and to reshape communications circuits between key organizational members. In other words, organizations need sound competencies in business continuity management.

Often, disasters occur because of staff complacency . An illustrative case is the disabling of Northwest Airlines' backup system. The investigation of this incident showed that a sub-contractor laying new lines in Eagan, Minnesota bored through a cluster of cables cutting 244 fiber optic and copper telecommunications lines. Airline passengers nationwide were stranded since those communications lines linked the Northwest's Minneapolis-St. Paul hub to the rest of the nation. Situations similar to the Northwest Airlines incident are usually prevented by the use of redundant lines, but apparently that airline's redundant communication lines ran alongside those used for backing up its system (Lehman 2000).

In a 1996 survey on business continuity practices conducted by IBM, 293 of the 300 surveyed companies had suffered security incidents in the previous year (IBM 1996). The estimation of loss of system capability due to these incidents was calculated as 500,000 man-hours. This study suggested that 89 percent of the responding organizations believed their computer systems to be critical. Nearly 25 percent of the companies stored 60 percent of their data in PCs and 76 percent were not aware of the cost of back up. A study of Irish experiences in disaster recovery planning presents a similar scenario (Adam and Haslam's 2001). Even after highly publicized terrorist attacks, recurrent distributed denial-of-service attacks, and of the forecasts about climate change, a quarter of U.K. companies do not store backup data off-site, two-fifths have no recovery plan in place and of those that do, less than half of the plans had been tested within the last year (ISBS 2006). Overcoming this challenge gets even more

complex when one observes that today's professionals, knowledge workers, can leave an organization anytime, taking with them their, and the firm's, means of production (Drucker 2001).

The Principles

Having sketched the background for information systems security and advanced the main security challenges confronting organizations, how should organizations proceed in the complex task of protecting their information assets?

The solution to the pressing problems of managing information security lies in shifting emphasis from technology to organizational and social process. Although this orientation has been defended by many, in practice the design of over-formalized, acontextual, ahistorical and reactive security solutions still dominates. Many solutions don't fit. because there is inadequate consideration of information security issues.

Although there is no magic bullet to solve IS security challenges, this section presents a set of fundamental principles necessary for managing current information security issues. This management framework is composed of six principles, which are classified into three classes, namely:

- Managing the informal aspects of IS security
- Managing the formal aspects of IS security
- Managing the technical aspects of IS security

Following a brief description of each class, each principle is elaborated and suggestions regarding its applicability advanced.

Principles for Managing the Informal Aspects

In the final analysis, the security of an information system is dependent on the people that form that system. People design, implement, apply, and execute security measures (Schultz et al. 2001). In the same vein, people access, use, manage, and maintain the IS resources of an organization (Henry 2004). As a consequence, the security culture shared by organizational members plays a critical role in ensuring IS security. Central to developing and fostering a security culture is the need to understand context. Research has shown the importance of the broader social and organizational issues that influence the management of information security.

An analysis of prescription fraud in the British National Health Services (Pouloudi 2001), suggests that by carefully interpreting issues and concerns of the various stakeholders, it is possible to understand the interaction between technical and social aspects of an IS implementation, thus facilitating fraud prevention. Similarly, an evaluation of the unethical computer use practices by Joseph Jett at Kidder Peabody & Co (Dhillon and Backhouse 1996), shows that it is important to create a culture of trust, responsibility, and accountability. It is evident that organizations need to develop a focus on the pragmatic aspects in managing IS security.

Principle 1: Education, training and awareness, although important, are not sufficient for managing information security. A focus on developing a security culture goes a long way in developing and sustaining a secure environment.

Education, training and awareness have long been suggested as important measures for improving the IS security level of an organization. However, unless or until an effort to inculcate a security culture exists, the desired organizational integrity will not be achieved. Clearly, issues such as lack of human centered security controls

(Hitchings 1994), mismatch between the needs and goals of the organization, poor quality of management and inadequate management communication (Dhillon 1997), can be considered as precursors of an unethical environment, thus endangering the health of an organization and making its information systems vulnerable to abuse or misuse.

Although managers are aware of the potential problems related with a disaster, they tend to be rather complacent in taking any proactive steps (Adam and Haslam (2001). Such an attitude can be explained considering the relative degree of importance placed on revenue generation. Hence, while automating business processes and pursuing optimal solutions, backup and recovery issues are often overlooked. Failing to recognize that organizational processes such as communications, decision making, change, and power are culturally ingrained is an attitude that can lead to problems in IS security .

To minimize the potential adverse events arising because of inability to appreciate human and social factors, normative or informal controls should be established (Dhillon 1999a). These security measures should instill and sustain a security culture and contribute to the protection of information assets.

Besides personal factors, work situations and opportunities available leverage the engagement in computer crimes (Backhouse and Dhillon 1995). Monitoring employee behavior is an essential step to maintain the integrity of an IS. Although such monitoring may be formal and rule based, informal monitoring, comprising the interpretation of behavioral changes and the identification of personal and group conflicts, can play an important role in establishing appropriate checks and balances. In the end, what an organization should seek is the establishment of an ethical environment among collaborators.

Principle 2: Responsibility, integrity, trust, and ethicality are the cornerstones for maintaining a secure environment.

In the beginning of this chapter, it was argued that the traditional three tenets for managing information security – confidentiality, integrity and availability – were too restrictive to develop secure environments in current organizations. Although this set of fundamentals was enough when organizations were structured hierarchically, its application falls short in networked organizations. This situation becomes clear as we consider the following facts. Confidentiality mostly concerns restricting data access to those who are authorized. However, information and communications technologies developments are pulling in the opposite direction, aiming at making data accessible to the many, not the few. This trend gets stressed if we consider the new configurations organizations are adopting, characterized by less authoritarian structures, more informality, fewer rules, and increased empowerment. Conventionally, integrity regards the maintenance of the values of the data stored and communicated. Equally important, however, is the way those values are interpreted. A secure organization not only needs to ensure that data do not suffer unauthorized modification, but also to guarantee that data get interpreted according to the prevailing norms of the organization, something that has been termed “the maintenance of interpretation integrity” (Dhillon and Backhouse 2000, p. 127). Although availability may be less controversial than the previous two tenets, the reality is that system failure is an organizational security issue.

In face of this new organizational paradigm, characterized by loosely coupled organic networks, cooperation instead of autonomy and control, intense sharing of information and high level of interpersonal and inter-organizational connectivity, a new set of fundamentals is required. In response to this quest, Dhillon and Backhouse (2000) suggest the RITE (responsibility, integrity, trust, and ethicality) principles. These principles

were inspired by an earlier period when extensive reliance on technology for close supervision and control of dispersed activities was virtually non-existent. The RITE principles are:

Responsibility

In a boundary diffused organization, members need to know their respective roles and responsibilities. This knowledge should not be seen as static, and it should enable organizational members to deal with new developments that require ad hoc responsibilities not anticipated in the company's organizational chart or formal procedures.

Integrity

Integrity is an important precondition for creating a secure environment, and personnel integrity is fundamental to ensuring the sustainability of that environment. Personnel integrity should be a requirement of membership in an organization. Prospective employees references should be properly checked, and once they are inside the organization processes should maintain and strengthen their personal integrity. As previously observed, the majority of security breaches come from existing employees. Individuals might be the target of pressures, and they might be subject to different kinds of problems, such as marital, financial, and medical.

Trust

Modern organizations are shifting their emphasis from external control and supervision to self-control and responsibility. In a physically dispersed organization, close supervision of employees is less viable, so trust must act as the glue between organizational nodes. Organizations need to set up mutual systems of trust, where members are trusted to act according to company norms and accepted patterns of behavior.

Ethicality

A formalized set of rules suit foreseen and predictable circumstances. However, new and dynamic situations create difficulties. In many situations there simply are no established rules for action. The way forward is to ensure members will act according to a set of working norms embedded in ethical standards. The difficulty that organizations nowadays experience is where do new and existing members get the ethics needed to shape informal norms and behavior. In recent years, the lowering of ethical standards in business has led to an increase in the number of frauds. Without a strong ethical foundation and in the absence of a supportive environment, organizations will confront serious IS security issues.

Principles for Managing the Formal Aspects

Organizational theorists have suggested that the formalization of organizational tasks through division of labor and coordination of efforts is an answer to the increased complexity within organizations (Mintzberg 1983). With the advent of computerization, technology has been used to automate many of the formal activities. This process involves deciding which aspects should be automated and which should be left alone (Liebenau and Backhouse 1990). Therefore, it is relevant to understand the nature and scope of formal rule based systems and the interrelationships between those systems and the design of information security in an organization. The following two principles should be considered when instituting IS security formal controls.

Principle 3: Establishing a boundary between what can be formalized and what should be norm based is the basis for establishing appropriate control measures.

The establishment of formalized rules is one step that could assist in managing IS security. An example of such formalized rules are the security policies that assist in clarifying bureaucratic functions in order to minimize

ambiguities and conflicting interpretations within organizations. The definition of security controls at the formal level of an organization should however take into consideration that the possibility of over-formalization. Management's inability to balance the rule and norm based aspects of work are a source of security problems. In order to prevent the misinterpretation of data and the misapplication of rules, formal rules and procedures need to be in place, and applied with other existing controls and their contexts. If formal rules are primarily designed as isolated and disconnected solutions for specific problems, they will have dysfunctional effects.

Although security policies are perceived as essential for expressing rules of conduct, the success of their application is a function of their integration with the organization's strategic vision. If, as in the past, enterprises keep formulating security policies based on checklists, following a rationale of identifying specific security responses to specific conditions, they will not be able to draw a line between formal rule based systems and pragmatic responses.

To design a well-balanced set of controls in a highly integral business environment, information security management needs to be on top management's agenda. Only then it will be possible to shift attention to the creation of a security vision and strategy where appropriate consideration is given to the threats and vulnerabilities of the business process architecture and of the technological infrastructure. When this state is reached, security considerations will acquire a strategic nature and will demand attention in order to serve as a business enabler, namely by maintaining the consistency and coherence of organizational operations. In this framework, security policies will tend to assume the role of functional strategies.

Principle 4: Rules for managing information security have little relevance unless they are contextualized.

An implication from the previous principle is that exclusive reliance on either the rules or norms will not provide enough protection for IS assets. If rules for managing information security are applied without due appreciation of context, the outcomes may be detrimental to the security of the company. Only by conducting a thorough security evaluation will the organization be able to design an integrated set of technical, formal, and informal controls. This evaluation will review the current security controls, taking into consideration the context in which each of the projected controls will be implemented and ponder on how different controls should be integrated.

The context dependence of security controls application may be appreciated by considering two formal controls, namely security policies and structures of responsibility and authority. The formulation of a security policy should result from the application of sound business judgment to the value ascribed to the data and the risks associated with its acquisition, storage, recovery, management, manipulation, communication, and interpretation. Because each organization is different, the content and form of a security policy is case specific, and it is difficult to draw any generalization. This suggests that a situational centered approach should be preferred when managing IS security controls.

The second illustrative control: structures of responsibility and authority (Backhouse and Dhillon 1996). The adoption of appropriate structures is an important step in establishing good management practices and to assist in the prevention of computer crime and of communication breakdowns within organizations. The concept of structures of responsibility and authority provides an effective means to identify the responsible agents in the formal and informal organizational environments and to determine what behaviors those agents perform. By facilitating the understanding of the ranges of conduct open to responsible agents, the influences they are subjected

to, the manner in which they make sense of the occurrence of events and the communications in which they participate, structures of responsibility and authority create a means to manage the formal aspects of IS security.

In order to benefit from the application of such framework, an organization needs to go beyond the sole concern of specifying an appropriate organizational structure, since this attitude usually results in a skewed emphasis towards formal specification. The most important step to solve the problems when establishing structures of responsibility and authority is the capacity to understand the underlying patterns of behavior of organizational members. The goal of developing and designing secure environments will only be successful if the context that shapes those attributes is taken into account.

Principles for Managing the Technical Aspects

From the previous discussion, it should be apparent that the security of the technical infrastructure is a function of the effectiveness of formal and informal organizational arrangements. Exclusive reliance on technical controls will not be enough to create a secure environment. Traditionally organizations have been conceived as purposeful systems, where security has not been considered part of the 'useful system' designed for the purposeful activities (Longley 1991). Actually, IS security management has always been considered as an activity that aims to warranty that the useful activities of an organization will continue to be performed and harmful incidents avoided. However, IS security management should be perceived as a key enabler in the smooth running of the business processes of an organization (Dhillon 1997), by the development of security visions, strategies, and cultures.

Of course from a holistic point of view, besides focusing on formalized rule structures and establishing an adequate understanding of behavioral practices, an organization also needs to develop and implement appropriate technical controls. These are vital measures, especially concerning who accesses the technical systems and what they are allowed to do once admitted. Two fundamental principles should be considered for adequately managing the technical aspects of information systems security. These follow.

Principle 5: In managing the security of technical systems a rationally planned grandiose strategy will fall short of achieving the purpose.

Many organizations focus on formulating security strategies, policies and procedures, and then hope their implementation will make them more secure. Although strategies, policies and procedures are important components of the organizational security effort, an exclusive emphasis on this top-down stepwise effort may be counterproductive. There are two main reasons for this argument.

First, there is the possibility that an exercise of rationally planned strategy may not necessarily consider the context where that strategy is being formulated and will be implemented. Several studies have provided supported for the multi-faceted nature of formulating strategy, where intended strategies interact with emergent strategies, and where formulation and implementation co-exist and do not always follow each other in the expected order (Mintzberg 1994).

Second, the fast changing pace of information technologies and the dynamic nature of businesses raises considerable obstacles to the formulation of grandiose strategies and waiting for them to play out. In the past, where a hierarchy was the dominant organizational structure and stability was the norm, it made sense to formulate strategies and policies and then proceed with their implementation, allowing the time for organizations to adapt to them. Back then, a rationally planned approach for information security formulation and implementation could have sufficed. Nowadays, with the emergence of new technologies, constant innovation and transformed structure

and business processes, context is a determining factor for maintaining organizational integrity of networked and virtual enterprises.

Principle 6: Formal models for maintaining the confidentiality, integrity and availability (CIA) of information cannot be applied to commercial organizations on a grand scale. Micro-management for achieving CIA is the way forward.

Confidentiality, integrity, and availability are key attributes of information systems security. From a technical perspective, security can only be achieved if these three aspects have been clearly understood. One key ingredient in the design of technical controls is to apply formal models of security. Examples of these models are the Bell La Padula and Denning's Models for confidentiality of access control; and Rushby's Separation Model and Biba's Model for integrity. Any formal model is an abstraction of reality and its adequacy and preciseness are crucial for determining model's usefulness.

To a large extent, the previously mentioned models have proved valid and complete. However, their validity exists not because of their mathematical correction, but because the reality they are mapping is well defined, namely the military organization. To a large extent, the military environment is characterized by a culture of trust among its members and a system of clear roles, lines of authority and responsibilities. As far as the organization works according to the stated security policy, the models successfully adhere to reality. However, the transferability of these formal models to a different reality, particularly the commercial one, calls in question the maintenance of their completeness and validity.

The first shortcoming is that organizational reality is not the same for all enterprises. Therefore, the stated security policy for one organization, might be radically different from that of the other because of environmental differences. Second, a model conceived for information security within a military organization may not necessarily be valid and applicable for a commercial enterprise. Consequently, any attempt to use models based on the military' situation may prove inadequate in a commercial setting, together with the possibility of such application generating a false sense of security. The way forward for achieving confidentiality, integrity, and availability is to create newer models for particular aspects of the business for which information security needs to be designed. This requires the development of micro-strategies for unit or functional levels.

Conclusion

This chapter has sketched the background for IS security, presented the major challenges that organizations need to address when establishing a secure environment and presented six principles for managing IS security in modern enterprises. The discussion has essentially focused on three core concepts: the technical, formal and informal aspects of IS security.

IS security has always remained an elusive goal and it is rather difficult to deal with security. As a concluding remark, no one approach is adequate in managing the security of an organization and clearly a more holistic approach is needed. Continued and new research and practitioners efforts are needed to help addressing issues and concerns in the complex and engaging field of IS security.

Chapter editor

Gurpreet Dhillon

References

Adam, F., & Haslam, J. A. (2001). A Study of the Irish Experience with Disaster Recovery Planning: High Levels of Awareness May Not Suffice. In G. Dhillon (Ed.), Information Security Management: Global Challenges in the New Millennium . Hershey, PA: Idea Group Publishing.

Audit Commission. (1990). Survey of Computer Fraud & Abuse: The Audit Commission for Local Authorities and the National Health Service in England and Wales.

Audit Commission. (1994). Opportunity Makes a Thief. Analysis of Computer Abuse: The Audit Commission for Local Authorities and the National Health Service in England and Wales.

Backhouse, J., & Dhillon, G. (1995). Managing Computer Crime: A Research Outlook. Computers & Security, 14(7), 645–651.

Backhouse, J., & Dhillon, G. (1996). Structures of Responsibility and Security of Information Systems. European Journal of Information Systems, 5(1), 2–9.

Baskerville, R. (1992). The Developmental Duality of Information Systems Security. Journal of Management Systems, 4 (1), 1–12.

Chokron, M., & Reix, R. (1987). Planification des Systèmes d'Information et Stratégie de l'Enterprise. Révue Française de Gestion, Janvier/Fevrier, 12–17.

Dhillon, G. (1997). Managing Information System Security. London: Macmillan.

Dhillon, G. (1999a). Computer Crime: Interpreting Violation of Safeguards by Trusted Personnel. In M. Khosrowpour (Ed.), Managing Information Technology Resources in Organizations in the New Millennium. Hershey: Idea Group Publishing.

Dhillon, G. (1999b). Managing and Controlling Computer Misuse. Information Management & Computer Security, 7(5), 171–175.

Dhillon, G. (2007). Principles of Information Systems Security: Text and Cases. Hoboken, NJ: John Wiley & Sons.

Dhillon, G., & Backhouse, J. (1996). Risks in the Use of Information Technology Within Organizations. International Journal of Information Management, 16(1), 65–74.

Dhillon, G., & Backhouse, J. (2000). Information System Security Management in the New Millennium. Communications of the ACM, 43(7), 125–128.

Dhillon, G., & Orton, J. D. (2000). Schizoid Incoherence and Strategic Management of New Organizational Forms. Paper presented at the International Academy of Business Disciplines, March 30-April 2, Las Vegas.

Dhillon, G., & Phukan, S. (2000). Analyzing Myth and Reality of Computer Crimes. Paper presented at the BITWorld Conference, Mexico City, Mexico.

Dhillon, G., Silva, L. & Backhouse, J. (2004). Computer Crime at CEFORMA: A Case Study, International Journal of Information Management, 24(6), 551-561.

Drucker, P.F. (2001). Management Challenges for the 21st Century. New York: Harper Collins.

Erwin, D. G. (2002). Understanding Risk (or the Bombastic Prose and Soapbox Oratory of a 25-Year Veteran of the Computer Security Wars). Information Systems Security, 10(6), 14–17.

Garg, A. (2003). The Cost of Information Security Breaches. The SGV Review, 33–40.

Gordon, L. A., Loeb, M. P., Lucyshyn W. and Richardson, R. (2006). CSI/FBI Eleventh Annual Computer Crime and Security Survey. Computer Security Institute.

Henry, K. (2004). The Human Side of Information Security. In H. F. Tipton & M. Krause (Eds.), Information Security Management Handbook (Fifth ed.). Boca Raton: Auerbach.

Hitchings, J. (1994). The Need for a New Approach to Information Security. Paper presented at the 10th International Conference on Information Security (IFIP Sec '94), 23-27 May, Curacao, NA.

IBM (1996). A Risk too Far?, April, IBM, London.

ISBS (2006). DTI Information Security Breaches Survey 2006 – Technical Report, Department of Trade and Industry, UK.

Lehman, D. (2000). Cable cuts ground Northwest flights. Computer World.

Liebenau, J., & Backhouse, J. (1990). Understanding Information. London: Macmillan.

Longley, D. (1991). Formal Methods of Secure Systems. In W. Caelli, D. Longley, & M. Shain (Eds.), Information Security Handbook. New York: Stockton Press.

Mintzberg, H. (1983). Structures in Fives: Designing Effective Organizations. Englewood Cliffs, NJ: Prentice-Hall.

Mintzberg, H. (1994). The Rise and Fall of Strategic Planning. New York: The Free Press.

Neumann, P. G. (1994b). Inside Risks — Technology, Laws, and Society. Communications of the ACM, 37(3), 138.

OTA, (1994). Information Security and Privacy in Network Environments. Office of Technology Assessment.

Pouloudi, A. (2001). Addressing Prescription Fraud in the British National Health Service: Technological and Social Considerations. In G. Dhillon (Ed.), Information Security Management: Global Challenges in the New Millennium. Hershey, PA: Idea Group Publishing.

Schultz, E. E., Proctor, R. W., Lien, M.-C., & Salvendy, G. (2001). Usability and Security: An Appraisal of Usability Issues in Information Security Methods. Computers & Security, 20(7), 620–634.

Siponen, M. (2001). An Analysis of the Recent IS Security Development Approaches: Descriptive and Prescriptive Implications. In G. Dhillon (Ed.), Information Security Management: Global Challenges in the Next Millennium. Hershey, PA: Idea Group Publishing.

15. Avoiding information systems failures

Editor: John Beachboard (Idaho State University, Pocatello, USA)

Contributors: Nelson Massad and Charmaine Barreto (Florida Atlantic University, USA), Alma Cole, Steven Hernandez, Mike Mellor and Moses Okeyo (Idaho State University, USA)

Reviewer: Geoffrey Dick (University of New South Wales, Australia)

Learning objectives

- Be able to explain why business managers need to understand the consequences of information system (IS) failure and actively participate in planning to avoid such failures.
- Be able to identify the components comprising an information technology (IT) and explain how IT infrastructure is necessary to support the delivery of IS-enabled business services.
- To recognize that IS system failures are not limited to the loss or destruction of information systems, but includes breaches of information confidentiality, integrity and availability.
- To be briefly identify and explain how managerial, operational and technical controls are used to minimize the probability of system failure and to minimize adverse consequences resulting from those failures which do inevitably occur.

Introduction

Avoiding information system (IS) failures; isn't that the job of the information technology (IT) staff? Why should I care? What is an IS failure anyway? This chapter is intended to help you understand what IS failures are, what tends to cause IS failures, and how to minimize the probability of experiencing IS failures. While the concept of IS failures is presented in more detail later in this chapter, we need simple definition of the concept of IS failure to understand the motivation for including this chapter in a textbook for business students. An **IS failure** occurs when an **information system**, that combination of computer hardware, software, data, and processes designed to support some type of organizational activity, fails to meet the organization's requirements. We will see that defining IS failures is a bit more complicated than defining failure for other organizational assets such as a motor vehicle failure. That is because an organization not only requires a system to be available to meet its requirements, but in many cases also requires the information system to ensure that the information stored is accurate and cannot be accessed by unauthorized individuals. Before getting into the details of this subject, we want you to understand why we, the authors of this text, think this subject is important for all organizational managers, not just IT professionals.

A considerable portion of this text is devoted to helping you understand how information technology can be used to benefit your business or organization and consequently this chapter does not explicitly discuss information system success. In this chapter we will look at the dark side of information systems (IS) and discuss the

implications of system failure and what organizations can do to reduce although not eliminate the probability of their information systems failing. Upon the completion of this chapter, we want you to appreciate that while an organization's professional IT staff may be responsible for implementing the majority of technical details associated with avoiding information system failures, organizational management must be involved in:

- Understanding the benefits derived from information systems and assessing the consequences to the organization of IS failure,
- Prioritizing IS investments required to improve system reliability and security, and
- Ensuring that policies are in place across the organization so that all organizational members recognize their individual and organizational responsibilities for maintaining the availability, integrity and security of those information systems on which the organization depends.

Management must recognize the potential downside of relying on information systems. The greater the benefits derived from information systems, the greater the potential for loss resulting from system failure. While the IT staff should play a critical role in helping organizational management to understand potential weaknesses in its information systems, the organizational management must be prepared to assess operational, financial and even legal implications of system failure. That is, what are the consequences to an organization if particular a particular IS service (e.g., email, production scheduling, point of sale, transportation scheduling) fails? Furthermore, do the consequences vary if multiple services fail individually or in combination?

IS failures can have financial, legal and moral consequences. Consider, if there were no adverse consequences resulting from an IS failure, then one must wonder why the information system exists. *Perhaps the most critical takeaway from this chapter is that organizational managers -- clearly top management, but operational and staff management as well -- must be involved in assessing the consequences of system failure and ensuring that their organizations have properly invested in reducing the probability of experiencing serious IS failures. That is, a*fter understanding the consequences of IS failure, organizational management must actively participate in the development of appropriate policies, procedures, training and technical safeguards to reduce the probability of sustaining unacceptable IS failures.

Managing the delivery of IS services: The role of IT infrastructure

Before launching into a discussion of system failures and management techniques intended to avoid them, we want to introduce some terms intended to give us a common set of definitions and a common understanding of how IT supports the delivery of IS services. In truth, organizational managers are generally more interested in the actual delivery of IS services than the performance of their information systems. However, the reliable delivery IS services is dependent upon the reliability of supporting information systems. Perhaps it is a problem within all disciplines, but the IT field seems particularly prone to assigning varying definitions to common terms. Unsurprisingly, the lack of universally accepted definitions can result in miscommunication. Given this somewhat chaotic state of affairs, we do not claim that the definitions and usage employed in this chapter are authoritative. However, they do reflect the understandings of important IS researchers and practitioners. We provide this cautionary note because we recognize the terms can be differently understood and that that assuming a common understanding of particular terms often leads to miscommunication.

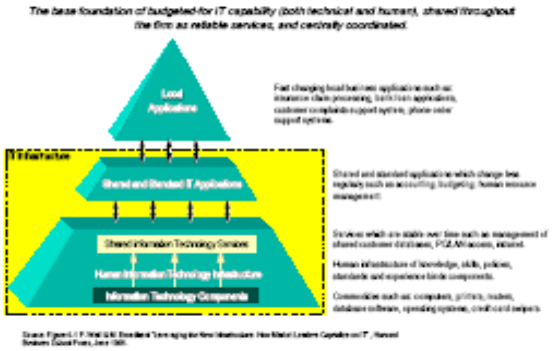
Key Concepts: Distinguishing between IT infrastructure and applications

In this section, we rely heavily on the work of Weill and Broadbent (1998) in their study of IT infrastructure and its relationship to business performance. While their research primarily focuses on the competitive advantages derived from effective IT infrastructure investment, their conceptualization of IT infrastructure, as distinguished from business applications, can be usefully applied within a variety of organizations. Just as with virtually any financial investment, they view IT investments as consisting of numerous individual investment decisions "... each with different objectives -- each with different risk-return profiles to be balanced to meet the goals of the firm" (p. 24). In evaluating potential IT investments, they find it useful to distinguish between IT infrastructure and IT applications. **Business applications** represent the software developed specifically to automate or inform business activities. For example, for airlines, the ticket sales and seat reservation system is considered a critical business application of IT. As you have seen in the many of the preceding chapters, there is a tremendous variety of applications that an organization can adopt to achieve its objectives. Throughout the chapter, when we refer to IS services, we are generally speaking of the services provided by business applications. However, what is relevant to our discussion here is that since business applications tend to directly support the needs of the organization, it is typically not too difficult to obtain management involvement in making investment decisions relevant to purchase or development of IT applications.

In contrast, **IT infrastructure** represents a foundation or platform which is needed to support the business applications. Weill and Broadbent conceptualize IT infrastructure as including:

- IT Components: the computers (desktops, servers, specialized storage devices), system software, and networking hardware, software and communications links,
- Human IT Infrastructure: the human resources required to configure, operate and maintain the IT components and applications,
- Shared IT Services and Applications: an array of shared IT services oriented more toward enabling the organization to function more effectively, but not directly related to the support of specific business processes, e.g., email, SPAM filtering, even widely adopted Enterprise Resource Planning (ERP) Systems.

Exhibit 15.2: The structure of IT infrastructure. Source: Figure 4-1 P. Weill & M. Broadbent, *Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology*, Harvard Business School Press, June 1998.



We have taken the space to distinguish between IT applications and IT infrastructure, because organizational management has too often failed to appreciate the critical relationship that exists between IT infrastructure investment and the performance, reliability and security of its business applications. Organizational management is primarily concerned with the failure of the business applications on which they have grown to depend. It is important to recognize the extent to which these business applications depend upon a secure and reliable IT infrastructure.

Information Technology Infrastructure Library (ITIL): An IT management framework

For managers to effectively address the issue of information systems failures, they should have a general understanding of what constitutes effective IT management practices and processes. ITIL provides a comprehensive framework of IT "best management" practices developed by the Office of Government Commerce (OGC) of the United Kingdom. The ITIL framework intentionally emphasizes the critical role of people and processes relative to technology in delivering delivery of high-quality IT services. While a thorough introduction to the ITIL framework lies well beyond the scope of this chapter (OGC presents the framework in multiple book-length publications), the framework provides several principles particularly relevant to our discussion of avoiding system failures. The ITIL framework is built around the core process of IT service delivery and management. ITIL promotes business driven identification of **Service Level Requirements** (SLRs) to be incorporated into **Service Level Agreements** (SLAs). SLRs are a set of operational requirements for individual IT services. Typically SLRs include a specification of service availability (the time of day that the service should be accessible and the percentage of downtime acceptable during those times the service is expected to be available), and service response times (how fast the automated service is provided once the action has been initiated by the system user). The

operational requirements represent a clear articulation of the organization's IT service needs to the IT service activity.

Consider the example of an information system supporting a bank's tellers. If the teller's terminal fails, that teller can no longer perform his or her primary functions of accepting and disbursing money. If the system to which the teller terminal is connected fails, then all of the tellers are unable to perform this activity. The banks customers cannon deposit or withdraw their money. The number and duration of system failures can be used to calculate the overall system availability. Sometimes an information system does not fail completely. Perhaps one component fails and the system runs slowly. Or perhaps, so many users are accessing the system and it becomes overloaded and slows down. A bank customer may be willing to wait 1 minute or two minutes for a withdrawal to be completed, but what if it took 15 minutes. The lines would probably grow very long the the bank's customers would be dissatisfied. No information system is 100% reliable. Accordingly, the bank owners or managers must decide what level of reliability and performance they need to maintain the quality of service expected by the bank's customers. Note also, that different information systems may have different performance standards. In the bank example, the failure of one teller's terminal would not be as serious as the failure of the server system to which all of the teller terminals are connected. You should be able to explain why this is so.

ITIL calls for the specification of service requirements into SLAs. The SLA essentially represents a contract between the organization and its IT service provider specifying the type and quality of IT services to be provided. As a contractual-like document, SLAs should specify user responsibilities as well as those of the IT activity. SLAs should specify service availability, reliability, performance and security in terms and measures that organizational users are capable of understanding. For example:

- Service availability of 99.5% during business hours and 95% availability on nights and weekends -- excluding scheduled outages for required maintenance and upgrades,
- Transaction response times, database queries to complete in less than 5 seconds during peak usage hours and in less than 2 seconds during non-peak hours,
- Telephone hold times for service desk support are not to exceed 2 minutes
- Replies to emailed service request should be within two business hours from time submitted.

The development of SLRs and negotiation of SLAs provides a basis for reaching an understanding among system owners, system users and service providers. For our purposes, the process provides a business driven approach to establishing realistic criteria for judging success or failure in the delivery of IT services. The ITIL documentation goes into great depth describing the numerous IT management processes that can contribute to fulfilling the service delivery obligations specified in an organization's SLRs and SLAs. Later in the chapter, we will be discussing ITIL most closely associated with avoiding and recovering from systems failure.

Defining information system failure: Confidentiality, integrity and availability

Information systems are somewhat unique with respect to the specification of failure conditions relative to other organizational assets. Typically, an asset failure can be described in terms of availability. That is, if the organization relies on a truck for transporting goods or a drill press for manufacturing products, failure occurs when the asset is broken or stolen. The asset is simply not available to support its intended use. Information systems, however, are a bit trickier in that they may well be present and appear to be running, when they are in fact in a failure mode.

Unlike tangible assets, information does not necessarily disappear when it has been stolen. If an organization holds confidential information, perhaps a list of potential clients or information describing a new manufacturing process, the information may be downloaded by an unauthorized individual but remain available to the organization. Exposure of information to unauthorized personnel constitutes a breach of confidentiality irrespective of whether the information is actually lost during the breach. That is, the breach of information **confidentiality**, the exposure of information to unauthorized persons, may constitute an IS failure.

Another type of system failure occurs when the **integrity** of the information can no longer be trusted. That is, rather than an unauthorized exposure of information, there are unauthorized changes to the information. A bank may be perfectly willing to allow its customers to log on and check their account balances but it certainly does not want to permit customers to adjust their account balances without ensuring that funds have actually been deposited. A business website containing documentation about how to configure or repair its products might suffer serious financial harm if an intruder were able to modify those instructions leading customers to mis-configure or even ruin the product they have purchased.

Finally, denial of access to the information or information service represents another type of information failure. Access denial is referred to as a breach of **availability** and constitutes another type of system failure. Failure of a payroll system resulting in a delay of depositing pay to employee accounts can result in serious hardship. But there can be even more serious consequences of system failures. If a doctor is prevented from accessing the results of diagnostic tests, a patient may suffer or might even die. A commercial website might lose important sales if it were to fail for an extended time.

We see then that defining failure for information systems can be more complicated than one might at first expect. IS failures may be reflected in loss of information confidentiality, loss of information integrity or an inability to access the information or automated service, i.e., a loss of system availability. Organizational management must work with its IT professionals to understand the types of failures that can occur and to assess adverse consequences should failures do occur. While a variety of techniques to minimize the probability of experiencing system failures are discussed in following sections of this chapter, all organizations must recognize that some failures will inevitably occur and should establish recovery procedures to minimize adverse consequences when they do.

Potential causes of systems failure

Now that we have described a variety of ways in which information systems can fail and recognize the potential consequences these various failures can hold for the organization, we want to get a better understanding of why or how failures occur. It is only through understanding potential causes of systems failure that we are able to take appropriate action to avoid them.

There are a wide variety of potential **threats** to an organization's information systems. **Threats** are any person, object or circumstance that has the potential for causing an IS failure. that Exhaustive threat lists are difficult, if not impossible, to create and security professionals often use threat categories in organizing their analysis of threats. Each category could broken into additional categories, as we have done with the category of human threats, depending on the level of detail desired. A representative set of categories follows:

- Human: Human threats are perhaps the most complicated in that the category includes such a wide variety of behaviors. To illustrate how the degree of detail may vary, some relevant subcategories include:

- Accidental behavior by organizational members
- Accidental behavior by technical support personnel
- Accidental behavior by organizational clients and other individuals that have authorized access to the information or information service
- Malicious behavior by organizational insider
- Malicious behavior by organizational outsider (malicious behaviors can be further broken out to include: theft, sabotage, extortion).
- Natural: Flood, fire, tornado, ice storm, earthquake, flu pandemic
- Environmental: Utility failure, chemical spill, gas line explosion.
- Technical: Hardware or software failure (whether maliciously intended or through normal ware and tear), perimeter defense failures (faulty closed circuit TV, key-code access system, fire alarm)
- Operational: A faulty process that unintentionally compromises information confidentiality, integrity or availability. For example, an operational procedure that allows application programmers to upgrade software programs without testing or notifying system operators may result in prolonged outages.

Upon reviewing the many potential causes of system failure, it becomes apparent that the use of information technology to support critical needs, while extremely beneficial, can be fraught with peril. Certainly, we do not mean to discourage the use of information technology in this chapter, but we intend to emphasize that careful information system planning, implementation and operation are required to minimize the probability of system failure as well as minimize the adverse consequences resulting from those failures which will inevitably occur.

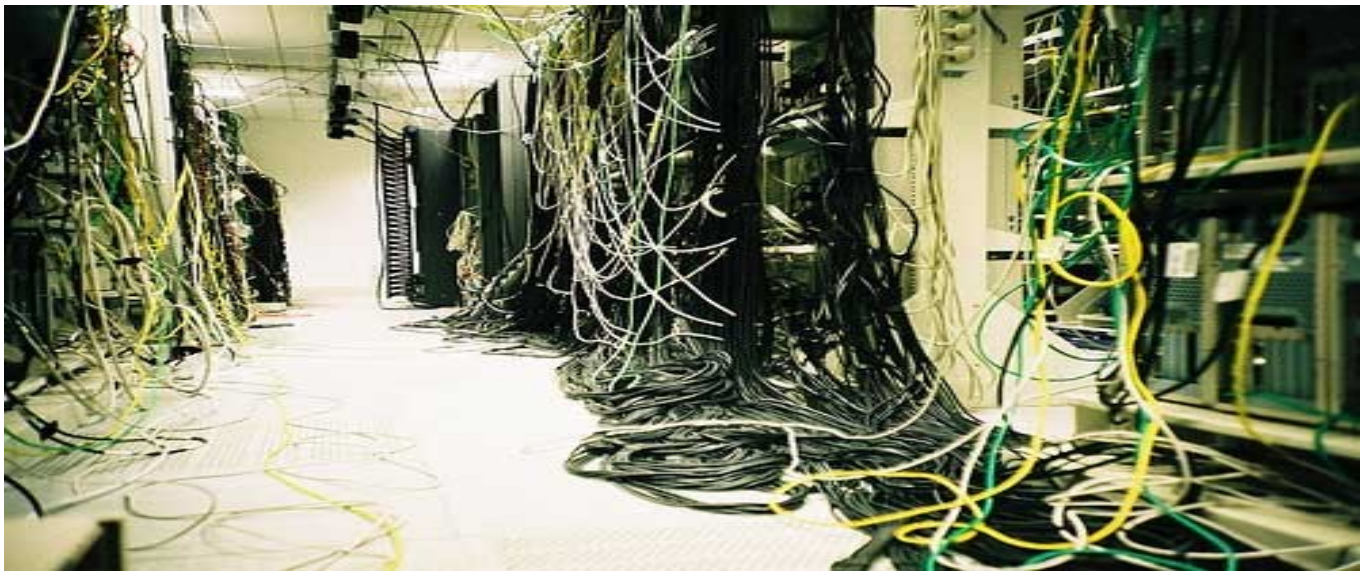


Exhibit 15.2: A rats nest of cables in an anonymous business's server room. This is not exactly accidental and not exactly malicious. However, it does exemplify how human actions can potentially contribute to system failures. Permission for use of this photo granted by the photographer, Cormac Phelan.

Mitigating risk: Reducing the probability of system failure

Risk Mitigation refers to the actions designed to counter identified threats. These actions are also referred to as controls and as with information system threats, there are numerous frameworks for categorizing the various controls intended to avoid system failure or compromise. A framework that we have found to be both comprehensive and comprehensible divides mitigation controls into three broad categories:

1. **Management controls**: managerial processes which identify organizational requirements for system confidentiality, integrity and availability and establish the various management controls intended to ensure that those requirements are satisfied.
2. **Operational controls**: include day-to-day processes more directly associated with the actual delivery of the information services.
3. **Technical controls**: technical capabilities incorporated into the IT infrastructure specifically to support increased confidentiality, integrity and availability of information services.

The remaining sections of this chapter present a general overview of managerial, operational controls and a subset of technical controls (technology investments associated primarily with improving system availability in the face of non-malicious threats.

Mitigating risks with management controls

Management controls include management activities related establishment of information system requirements and control processes intended to ensure that those requirements are met. Critical information assurance management controls include:

1. Creation of policies, procedures, standards and training requirements directly relating to the improvement of information system confidentiality, integrity and availability.
2. Performance of risk analyses to evaluate risk potential of new information systems and re-evaluate risks associated with existing business applications and IT infrastructure.
3. Management of information system change

The following sections provide a general overview of each of these three important information assurance management controls.

Information Assurance Policies, Procedures, Standards and Education

The overall objective of an information assurance program is to protect the confidentiality, integrity and availability of organizational information and IT-enabled services. Fundamental to the establishment of an effective information assurance program is the organization's establishment of appropriate information assurance policies, procedures and standards.

Policies are high-level statements communicating an organization's goals, objectives, and the general means for their accomplishment. The creation of information assurance policies may be driven by the need to comply with laws and regulations or simply reflect executive management's analysis of the organization's information assurance requirements. There can actually be a hierarchy of policies with each lower layer providing increasing degrees of specificity, but still recognizable as policies by their focus on "know what" content rather than "know how." Because policies tend to be formulated in general terms, organizations will generally develop procedures and standards that

more specifically elaborate what needs to be done. Policies might be used to identify information assets meriting special safeguards (e.g., client lists, product designs, market analysis), delineating information related roles and responsibilities (e.g., establishing a Chief Security Officer position) specifying the establishment and performance of information assurance related tasks or processes (e.g., organizational policy might dictate the establishment and conduct risk assessment and change management processes described below).

Standards can be thought of as a specific class of policies and represent mandatory rules (e.g., ensure desk is cleared of working papers before leaving worksite for the day), technical choices (e.g., all desktop systems connecting to the organizational network will have a particular anti-virus program loaded), or some combination of the two (e.g., the signature file for the anti-virus software is to be updated on a daily basis). The delineation of standards and policies can be fuzzy. A policy might dictate that servers containing confidential information reside behind a network firewall. A standard might specify the type of firewall to be used and even specify the configuration of the firewall. But all of that information might reside in a single policy document. Finally, an organization might specify procedures that spell out the specific activities or steps required to conform with designated policies and procedures. The **procedures** constitute the instructions for performance of policy- or standard-related tasks. The formal definitions matter less than the way the terms are actually employed with any given organization. The important point to understand is that the formulation of policies, procedures and standards constitute important elements of an organization's information assurance program and an organization's ability to avoid system failures.

There are extensive guidelines governing the development of effective policies, procedures and standards, and the reader is encouraged to consult such guidance if he or she becomes directly involved in the process of writing policies and procedures. However, we think it useful to briefly describe criteria for judging the effectiveness of information assurance policies. Good policies should:

- Good policies have the support of upper management. One can hardly imagine a factor more likely to undermine policy compliance within an organization than the realization that upper levels of management do not care about the policy, are unwilling to provide resources required to implement the policies or have no intention of conforming to the policies in their own behavior.

- Good policies are clear, concise and well written. Every attempt must be made to reduce ambiguity by selecting appropriate language, identifying a clear scope to which the policy applies and ensuring the policies are consistent with other organizational policies and practices. Organizational members cannot comply with policies if they cannot understand them and ambiguity may encourage the development of undesirable policy interpretations.

- Good policies will clearly delineate responsibilities and identify the resources required to support their implementation. If one commonly hears the phrases, "it's not my job" or "I don't have the resources" with respect to policy compliance, problems with compliance likely exist.

- Good policies are living documents. It seems that the only constant in today's world is change. Policies can quickly become outdated. Out-of-date policies lead to two problems. First, the policies gradually become inadequate as organizational requirements change over time and as well as due to changes in the types of risks present in the organization's environment. Second, as policies become increasingly inaccurate and irrelevant to the organization's needs, there is a natural tendency for the policies to be ignored.

• Good policies specify enforcement provisions and a process for handling policy exceptions. If there are no adverse consequences associated with policy non-compliance, then compliance will likely suffer. As it is difficult if not impossible to anticipate every contingency in the formulation of policies, long term compliance will be enhanced by specifically including provisions for requesting policy exceptions.

Finally, it is difficult to overestimate the importance of education and training in establishing effective policy compliance. The effectiveness of policies, procedures and standards are seriously undermined if organizational users are able to claim ignorance of their existence. This is particularly true with respect to compliance with specific standards and procedures. Education and training requirements will vary depending on the job responsibilities. Employees who deal with confidential information may require guidance concerning legitimate use of the information. IT professionals may require specialized training in order properly configure and employ technology used to increase reliability and security of information services. In short, the establishment of a comprehensive information assurance training program constitutes a critical a critical management risk mitigation control.

Risk assessment: Evaluating new systems and old

In an effort to speed delivery and reduce costs associated with the delivery of information services, many organizations short-change the planning and design phases of their information system projects. However, the consequences of adopting such a strategy often result in the delivery of services that do not adequately meet organizational requirements and may well end up increasing lifecycle system costs. The organization certainly leaves itself open to future problems if requirements for information confidentiality, integrity and availability are specified for the original system design.

IT professionals widely recognize that it is much more effective to design security and reliability directly into their systems from the outset than to try and add such capabilities after-the-fact. Consequently, the conduct of a risk assessment is essential in the planning of any major new information system or upgrade of existing capabilities.

A **risk assessment** essentially consists of:

• Clearly identifying organizational information assets, the data and information systems on which the organization depends

• Understanding vulnerabilities, the susceptibility of the asset to breakdown or malicious attack, associated with identified assets

• Identifying threats, object, person or incident capable of exploiting identified vulnerabilities.

An analysis system risks, that is, the probability of threats being realized, is performed to determine the probabilities of loss. Based on expected losses, the organization is better able to determine which countermeasures or controls are appropriate to its needs.

During the planning stage, organizations need to estimate the consequences of service failure, including how the consequences vary as a function of the duration of service failure, and the various threats capable of exploiting identified vulnerabilities. The participation of organizational management is critical to this process because they should best able to evaluate the consequences of system failure and determine the level of investment warranted to minimize adverse consequences.

IT and and security specialists can be expected to also play an important role by helping organizational managers to understand vulnerabilities, threats, and even probabilities associated with various threats.

As entire books have been dedicated to the subject, we do not attempt to provide a thorough treatment of risk assessments here. However, we do think it useful to include a brief discussion of a few representative issues that are usefully considered during the planning phase of an information system.

Information system planning necessarily focuses on IT solutions to meet identified requirements and minimizing system non-availability. IT solutions might include the purchase of redundant servers, tape backup systems, network firewalls and the like. These technology investments may represent warranted investments and we do not discount such recommendations. However, in considering overall systems availability and security, the physical location of the IT and information assets and the environmental systems on which they depend should also be carefully considered. For example, it is not uncommon to place computer centers in the basements of multi-story buildings, even if those buildings are located in known floodplains. Computers do not tolerate water well, and since water tends to seek the lowest levels within a building, a basement computer facility represent a risk that might be been easily avoided.

Power, air conditioning, external communications links all represent potential points of failure for computer systems. The likelihood of such events must be considered in the selection of information services on which an organization is to depend. In many areas, commercial power and communications are unreliable. Accordingly, managers must consider the probability and length of service outages and include additional investments, e.g., for uninterruptible power supplies capable of conditioning the power and generating backup power if commercial services are disrupted. When planning for the provision of IT-enabled services, organizational managers must realistically appraise the constraints and limitations imposed by the organization's environment.

In short, effective IT planning should incorporate a rigorous assessment of threats and the inclusion of appropriate safeguards and countermeasures within the overall design of proposed information systems.

Managing change and system configurations

A widely cited Gartner research report concludes that "80 percent of mission-critical application service downtime is directly caused by people or processes failures. The other 20 percent is caused by technology or environmental failure or a disaster" (http://www.gartner.com/5_about/press_releases/2002_03/pr20020325a.jsp). Often these failures result from the modification of software, loading a software patches to fix a security flaw or add some new functionality or the mis-configuration of critical servers or network devices. For example, important information services may be cutoff by mis-configuring security or communications devices such as network firewalls or routers.

IT management best practices such as those provided in ITIL emphasize the importance of change management. While often derided by IT practitioners as consisting of unnecessarily bureaucratic procedures that actually impede the practitioner's ability to quickly respond to customer requests, **change management** processes is intended to ensure that system changes are properly authorized, prioritized and tested, and that all interested parties are informed regarding proposed changes. Element of an effective change management process include:

- Selection of the appropriate and qualified staff to participate on the change management team.
- Establishment of formal change request and tracking system.

- Regular scheduling of change management team meetings.
- A formal means of ensuring that approved changes, including their implementation schedules) are communicated relevant stakeholders.
- A formal means, such as regularly scheduled system audits, to ensure that change management practices are being followed.

When system outages cost $20,000 a minute (see mini-case insert) the need to invest in a disciplined change management system becomes much clearer. Organization must assess the consequences of particular system failures to determine the level of investment in change management that warranted for the particular system. While recognizing that highly formalized procedures can pose an unacceptable burden on small- and medium-sized organizations, these organizations are still likely to benefit from managing change.

Consequences of an Untested Software Upgrade

A firm conducting much of its business over the Internet suffered a IS service failure during its peak sales season just before Christmas. The failure, the firm's web servers began to lockup so tightly that they could not even reboot themselves. This particular firm conducts almost 80% of its annual business in the weeks preceding Christmas. Losses were estimated at approximately $20,000 a minute. The cause of the failure was a supposedly "minor" software upgrade that a programmer made just prior to departing on holiday. It took the firm over 24 hours to discover the changes that had been made and even more time to back out the change to restore service. The programmer, once contacted, insisted that it was "inconceivable" that his change would have caused this outage (Behr, Kim and Spafford, 2004).

Before leaving the topic of change management, it is also useful to introduce the closely related topic of configuration management. While the term is sometimes used interchangeably with change management, configuration management refers specifically to implementation of a database that records critical elements of the IT infrastructure and applications necessary for the provision of IT services. The database should not simply be viewed as an IT inventory. Properly conceived and implemented, a configuration management database (CMDB) will include information documenting movement, maintenance, and problems experienced with various configuration items. Configuration items, those elements under configuration management and recorded in the CMDB, can include policies and procedures, human resources in addition to the hardware and software one would typically expect to find in an asset inventory database. Configuration management provides a necessary foundation for an effective change management process and as we shall see below contributes to the effectiveness of multiple service, infrastructure and security management processes.

Mitigating risks with operational controls

Even the authors sometimes wonder about the true distinction between management and operational controls. The easiest way to think about it is that management control functions are performed by managers and operational controls are performed by operators. However, if you look at real organizations, the distinctions between operations staff and management may not be all that clear. Nonetheless, we will use the categories because that is they do

reflect the terminologies that one commonly finds in both the trade and academic literature. Three operational controls commonly associated with maintaining system availability are:

- System monitoring and incident response
- Performing system backups
- Planning for disaster recovery

The careful reader will have noticed that these processes do not really help avoid system failures. Good catch! We hope that you will have noted that it is impossible to totally avoid system failures. Despite an organization's best efforts, sometimes things just go wrong. These processes are primarily intended to minimize the adverse consequences that can result if and when things do go wrong.

System monitoring and incident management

Organizations of almost any size will have some type of help desk function. When a user has a problem, they call for help and sooner or later they usually get it. Larger organizations may include another activity called an operations control center. In some organizations the help desk and operations control center are part of the same activity, i.e., the help desk may also serve as an operations center as well as providing user assistance. In other organizations the two functions are managed separately. The important point here is that the functions are being accomplished, not how they are organized.

We are concerned primarily with what are normally considered to be operations center functions. Those two functions are system monitoring and incident response. Quite logically, we cannot expect someone to respond to an incident before he or she has discovered that it has occurred. Detecting incidents is the function of **system monitoring. Incidents** can be defined as any event that impacts the confidentiality, integrity or availability of an IT-enabled information service. As we have discussed above, there are a lot of potential threats to an information system. When a threat occurs, it is an incident. An operator may back up old data over new data, a lightning strike may lead to a power surge that destroys a critical piece of equipment, a hacker may break into the system and steal or modify data. The list of possible incidents is limitless.

The challenge to maintain systems availability is to be able to detect incidents and respond to them quickly and effectively. While we are not trying to turn the readers of this text into ops center personnel, organizational managers should have knowledge of the system monitoring and incident response functions if their organizations significantly depend upon IT-enabled services. That is, organizational managers should ensure that the system monitoring and incident response functions are aligned to the operational needs of the organization.

First, the organization needs to decide whether it is willing to invest in proactive system monitoring as opposed to reactive system monitoring. Reactive monitoring simply means that the activity will be attempting to detect incidents as soon as possible after they occur and then respond to them. At its most basic level, there is virtually no monitoring going on at all. When someone calls to complain, the incident is recorded and the response is initiated. But there are also specific computer applications that can be used to monitor systems so that operations center personnel may receive an alarm that a component has failed before any user has detected that failure. These are typically referred to as systems or network management applications. These applications will monitor designated services or system components. When the component stops or fails to perform correctly, the application will initiate

some type of alarm or other notification. Some systems management applications have been designed to send email or telephone personnel to advise them that an alarm has occurred.

Proactive monitoring is watching for indications that a failure may occur. For example, if a hard drive, a computer component on which information is stored, becomes full, an important application might fail. With proactive monitoring, the systems management application monitors hard disk usage and sends an alarm when it reaches 80% of capacity. The IT support activity can either clear off some data or install a larger hard disk before a failure actually occurs. That is proactive system monitoring.

The key to both reactive and proactive system monitoring is to understand the system baseline. Operations center personnel want to have a very accurate understanding of how the system is configured and behaves when everything is working properly. Monitoring then becomes largely a function of detecting deviations from the system baseline. For example, imagine that an organization typically used very little Internet connectivity during the evening hours. The network administrator notices that for the last three nights that there has been a lot of network utilization starting at 2 am. It might be legitimate traffic; perhaps the organization had decided to back up its data to an offsite location during the early morning hours so that it would not interfere with normal system use. However, it could mean that an intruder had compromised the organizational system and was copying confidential data or perhaps using the organization's computers to launch SPAM out onto the Internet. Under reactive monitoring, the organization may not respond. After all, nothing appears broken and no one has complained. A proactive monitoring system will detect the incidents, the early morning network use, and investigates the cause of that traffic so that an appropriate response can be taken.

This example leads us to the second operations center function, incident management. **Incident management** refers to the actions that an organization takes in response to detected incidents. Upon incident detection, the first action is to minimize or contain damage resulting from the incident. The second response is to restore the service. Depending upon the organization and the type of incident, a range of other responses may be appropriate. There may be a need to provide notifications to key personnel. If there are multiple incidents occurring at the same time, a prioritization scheme may be required to determine which incidents are likely to cause the greatest damage to the organization. Effective organizations have a method of documenting incidents, such as a trouble ticketing system, and will perform after-action-analysis of incidents to determine if there are recurring patterns of incidents occurring.

Given that there are a near infinite number of possible incidents, there is also a near infinite number of possible responses. If a circuit to the Internet fails, the operations center will typically look to see if there is a problem with the organization's equipment. If the organization cannot isolate the problem to its equipment, then the appropriate response is to notify the telephone company or Internet service provider.

As indicated in the introduction to this section, there are a variety of ways that this function may be organized. Many organizations have adopted a three-tiered response model. The three-tiered response model reflects the observation that some operational personnel are quite knowledgeable while others are less so. The more knowledgeable, highly skilled personnel are paid more while less knowledgeable personnel are paid less. Organizations have learned that it is economically beneficial to have the low skilled personnel working on the easer incidents while the high-skilled personnel work on the more difficult incidents.

The tiered response model supports this objective. Less skilled individuals serve as the first responders. They record the incident information and resolve as many incidents as they are capable of resolving. Tougher incidents get passed to a second tier of fairly skilled personnel for resolution. Hopefully most of the incidents can be resolved in the first two tiers. However, sometimes things are really complicated and you need to bring in the really experienced professionals to resolve them. Some organizations will have these highly skilled individuals on their staffs. But other might rely on outside personnel to provide this third level of support. These individuals tend to be very expensive and an organization does not want them to spend their time working on easier problems that less-skilled, lower-paid staff or capable of handling.

Before leaving the topic of system monitoring and incident response, there is one other subject that merits discussion. We previously mentioned the ITIL framework as providing best management practices for managing IT operations. The authors of the ITIL have found it useful to distinguish between incident management and problem management. The incident response process just described falls under the category of incident management. Problem management is a bit trickier. If incidents are events that result in system failures, what are problems? Under ITIL, problems are the causes that underlie incidents. To explain, let us go back and reconsider the lightening strike that fried some of our equipment. The incident was the equipment failure. The equipment could no longer fulfill its intended function because of surge of electricity melted critical circuitry.

The problem, then, might be defined as the fact that lightening is a recurring problem in that particular geographic area. The organization may stock spare supplies of equipment and recover the service fairly quickly. However, services will still be disrupted and replacing the equipment could prove expensive depending upon how often lightening can be expected to strike. Or the problem might even be more broadly defined as an unstable power supply. Lightening may be one cause of power fluctuations, but there might be a variety of reasons that power might fluctuate. **Problem management** attempts to broadly identify the underlying causes of the incidents that are occurring, e.g. unstable power, and determine how that problem can best be managed. While replacing fried equipment can resolve the incident, other measures are required if the organization wants to avoid service failures resulting from unstable power. The examination of outages over a period of time to see how many result from unstable power is an example of using a trend analysis. When properly conducted, the combination of documenting incidents, conducting trend analyses, and resolving identified problems can greatly increase the availability of system services, reduce the number of incidents that must be fielded and may allow for reductions in the number of IT support staff.

Organizational managers might start to think that this discussion is getting technical and that the IT personnel should be taking care of these problems. Certainly, an organization should staff its operations center with technically competent IT staff. However, as was discussed above, organization management needs to determine what level of risk it is able to tolerate and ultimately determine the capabilities of system monitoring and incident response functions.

System backups

A recurring theme in this textbook has been the idea that value resides in the information and technology-enabled services rather than in the information technology itself. On a very fundamental level, this means that if your information technology (e.g., an application or web server) is damaged or destroyed, the organization can purchase a replacement. However, if the data on that server is destroyed, it may be much more expensive to replace

than the technology. Furthermore, the data may irreplaceable. Depending on the value of the data, the data's loss may result in business failure or a prolonged disruption of services provided by non-profit organizations.

The point is that information stored on an organization's automated system can be very valuable and merits organizational investment to ensure that it is safeguarded. Consequently, it is important that organizational managers understand system backup technology to the extent that they can confirm that their organizational data is safe guarded. Making a **system backup** entails saving a copy of data stored on the system's hard drives to a different data storage facility. A system backup can be made to hard drives on a different system or made to some other type of storage media such as digital tape or CD-ROMs. There are different types of systems backups that can be made.

Nairobi Fire and the Loss of Irreplaceable Documents

March 2, 2004. "The blaze started at 0200 local time (2300 GMT on Monday), sending sheets of orange flames into the night sky.... Council fire engines were unable to cope on their own and several ran out of water as no working hydrants could be found near the city hall.... Among the thousands of documents destroyed were maps showing the routes of new road bypasses (BBC, 2004 at http://news.bbc.co.uk/2/hi/africa/3524855.stm). Computers as well as paper may be lost in fires; both can constitute invaluable information resources of the enterprise. While insurance claims may help to restore the building, important documents, including contractual records and city planning drawings, were lost forever in the Nairobi City Council fire. These types of losses can be prevented by having having tested data recovery capabilities in place.

While the technical details of designing and maintaining a robust data storage infrastructure can be daunting, the fundamental principles for implementing storage backups are relatively simple. First among these is developing an understanding that there are different types of system backups. It is not sufficient to learn from your IT support staff that backups are being performed; managers need to ensure that the required types of backups are being performed. Different guides will approach the issue of backup types in varying levels of details.

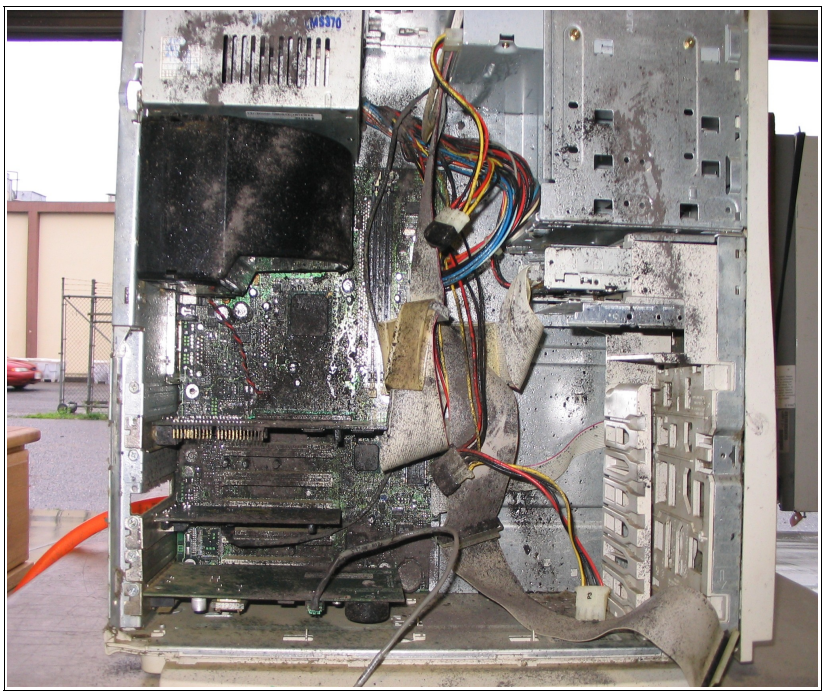
There are different types of data or information stored on information systems. We discuss just two categories while acknowledging that backup planners will often use a more detailed set of categories. The distinction that we wish to emphasize is between the software programs (we include the operating system and business application software in this category) and the actual data or information that is manipulated by the software. The rationale for this distinction is that one generally expects the operating system and programs to remain fairly stable, i.e., not change too much. The data, on the other hand, likely changes, or grows, daily. An organization does not want to repeatedly backup data that are not changing. This would pose an unnecessary expense. Plus, an operating system or application is lost, it still should be possible to rebuild the system by reloading the required operating system and applications. However, if operational data is lost, it is lost for good, if no backup exists.

We do not mean to imply that organizations are not concerned with backing up their software programs. They are. If the software programs and data have been backed up, an organization will be able to more quickly restore service than if it had to rebuild the system by reloading and reconfiguring the entire operating system and applications programs. Consequently, organizations do need to backup their entire systems, but not necessarily as often as they need to backup their data.

Accordingly, there are two major types of system backups (actually there are more; if interested see Wikipedia article on backups). There are **full system backups**, where all the software and data residing on the system are backed up. Then there are **incremental system backups**, where only the data that has changed since the last full backup or last incremental backup is saved. The advantage of this approach is to reduce the amount of time required to accomplish the backup. This is a particularly important issue on systems which must be taken out of operational status while performing the backup. Restoration is accomplished by restoring the last full backup and then applying the required incremental backups.

Another backup technique which has proven useful is the use of multiple copies of backups. Maintaining multiple backup copies provides several benefits. First, if data becomes corrupted and is accidentally backed up, it is possible to overwrite the good data. By maintaining multiple copies of backups, it may be possible to detect the data corruption before all versions have been overwritten, allowing recovery from an uncorrupted backup. Second, backup media occasionally fails. If there is only one backup, the restoration fails and the organization is out of luck. The probability of three or more sets of media failing simultaneously is extremely small. Consequently, even if some data is lost due to the need to restore from an older backup, the organization should still be able to recover the majority of its data. Finally, it is common to keep the most recent backup in general proximity to the devices on which it normally resides. The availability of the backup minimizes restoration time in the event of common failures. However, some failures are catastrophic and destroying not only the IT equipment, but the entire facility in which the equipment resides, e.g., flood or fire. In such cases the backup data is destroyed with the equipment and recovery becomes impossible, even when the equipment and facilities are replaced. Consequently, best practices dictates that at least one version of the backups be maintained off site to preclude catastrophic loss of all of the organization's data.

Exhibit 15.3.: Smoke damaged interior of computer. The system does not need to be completely destroyed by fire in order for data to be lost. In this case, however, a specialty firm was able to restore the system. Permission for use of this photo granted by the firm, Newlifeserviceco.



There are a variety of technologies that can be used to support the system backup function. They vary in terms of price, capacity [the amount of data that can be backed up], performance [the speed at which data can be backed up], reliability [the probability that the backup technology or media will fail] and overall functionality [special features included make the backup process easier to perform]. For our purposes, it is only necessary to recognize that the price of the backup escalates with speed and capacity. Thus, organizational management needs to understand the value of its information and information services and the consequences of system and data service interruption or loss so that it can make appropriate investment decisions in the purchase of backup system hardware and software.

There is one last critical principle that management truly needs to understand in it evaluation of its system backup process. That is, the organization does not truly have a backup plan if it is not willing to invest the time and resources to test its data restoration capabilities. More than a few organizations having made the investment in backup technology and have experienced the unpleasant surprise learning that their backup process was faulty. The takeaway is to insist that recovery processes be practiced regularly and particularly when significant system changes are implemented. A backup system that has not been tested should not be considered a backup system.

Planning for disaster recovery and business continuity

Some incidents are bigger than others and constitute catastrophic failures or disasters. Quite often these are associated with natural disasters, such has flood, earthquake or hurricane, but catastrophic failures may result from any number of causes. A major industrial accident might require vacating an organization's premises; a flu epidemic might incapacitate a significant percentage of an organization's staff. And of course, there is the chance of intentional sabotage. The issues of disaster recovery and business continuity range far beyond the domain of IT planning. However, to the extent an organization depends upon IT support to support core business processes, IT considerations must be fully integrated into an organization's overall disaster recovery and business continuity planning.

A **disaster recovery plan** is intended to provide detailed guidance concerning the actions to be taken in the event of that a disaster occurs. Disaster recovery plans may be written to address a wide variety of crises. Here, disaster recovery planning is discussed with reference to the restoration services disrupted by severely damaged IT facilities and services. In contrast, **business continuity plans** are more broadly concerned with ensuring that essential organizational functions can continue to be performed in the event of any circumstance that massively disrupts the normal operations of an organization. The two functions are closely related. To the extent an organization depends upon IT services to support core functions, then an effective business continuity plan necessarily ensures provision are made for restoration of essential IT services.

Key elements of both sets of plans require clearly established priorities, delegation of responsibilities (including contingency delegations should primary assignees prove unable to accomplish assigned tasks, and pre-staging of minimum essential infrastructure and assets to continue operations. For example, an organization may have designated a relocation point from which to continue its operations. However, if required phone lines are not available, business operations will remain disrupted until such time as new lines can be ordered and installed.

No organization can be expected to fully reconstitute itself in the face of a catastrophic loss of infrastructure or personnel. The key to business continuity planning is to identify the minimum essential functions that must be available if the organization is to survive (from a business perspective) or meet mandatory obligations (from a non-profit or government agency perspective).

From an IT perspective, disaster recovery and business continuity usually require designation of some type of back up facilities. As one might guess, the costs of alternative facilities can vary considerably. Maintaining dedicated facilities that replicate existing operational infrastructure is quite expensive. Accordingly, organizations employ a variety of techniques to manage the costs of disaster recovery and business continuity. The most expensive capability is the maintenance of hot backup site. In a **hot backup site**, essential systems have been duplicated at the alternative facility and are fully configured to pickup operations should the primary site fail. Given advances in communications technology, some companies continuously replicate their data to the alternative site thus minimizing the potential for data loss and the time it takes to restore service.

A **warm backup site** has many of the capabilities of a hot site, but its systems are not fully configured. While servers, workstations and communications facilities are in place, the organization will typically need to load its applications and data to make the facility operational. This can take from hours to days depending on the number, size and complexity of services to be supported. While still expensive, the maintenance costs of a warm site can be considerably less than those of a hot site. Some organizations maintain a **cold backup site**, essentially an empty facility in which an organization can reconstitute its system. There is essentially no hardware pre-installed; organizations will have to either relocate or purchase required equipment then install and configure it before operations can be resumed. Obviously, a lower cost alternative, but one that will likely result in a prolonged outage should the organization's primary facility becomes unavailable.

To reduce the cost of maintaining a hot site or a warm site, organizations may also contract with a service to provide contingency services. For a recurring fee, the service bureau agrees to maintain required equipment and facilities for the contracting organization to move into should the need arise. The service company can offer favorable prices by offering its facilities to multiple organizations or businesses - under the assumption that not all of its clients will require the use of its facilities at the same time.

We close this section with the same admonition provided at the close of the section on system backups. Disaster recovery and business continuity plans are not all that beneficial if the organization never tries to exercise them. While expensive, realistic test scenarios must be run to ensure that important details have not been overlooked. In truth, it is unlikely that an organization can adequately anticipate and plan for all possible contingencies. Yet those that do invest in contingency planning and in testing their plans are far more likely to survive the consequences of catastrophic disasters.

Mitigating risks with technical controls

The final set of controls associated with the avoidance of systems failures is related to technology. That is, organizations often make additional investments in their IT infrastructure with the explicit goal of avoiding or at least minimizing the consequences of information system failure.

There are numerous technical controls associated with information security. In this section, we introduce a range of infrastructure investments intended to improve the overall reliability of infrastructure and consequently, the reliability of organizational IT-enabled information services. These include:

- Redundant critical components (equipment, communications, etc.)
- Power conditioning and backup power
- System backup capabilities
- Network and system monitoring tools

Organization requiring high levels of availability will often find themselves needing to buy redundant hardware. In simplest terms, redundancy occurs when you buy to pieces of hardware to perform a function when one device is capable of performing the function adequately. Normally this is considered a bad thing to do. However, if the possibility exists of that component failing and disrupting a critical information service, then purchasing and installing the second device may be quite reasonable. The basic philosophy of operational redundancy is best illustrated with some simple probability calculations.

For example, assume an important device, say a hard disk used for storing data, has I device has a 1% probability of failing and disrupting service in any given year. The organization must determine if it is willing to accept the possibility of data loss (the data added since the last time the disk was backed up) and service disruption. If the 1% probability seems too high, the organization can purchase a second drive which we also assume has a 1% annual probability of failing. Now, what is the probability of the organization experiencing the data loss and service failure? Assuming drive failures are independent events, the probability of both drives failing simultaneously is: .01% or 1 in a thousand. This represents a considerable improvement over the 1 in 100 chance of experiencing a single drive failure. In complex environments, the probability calculations can become quite complex but the general logic remains the same. Investing in redundancy decreases the probability of systems failure.

Management must consider all potential causes of failure in prioritizing its investments in technical controls. In areas where power stability is a problem, power conditioning and the provisioning of on-site generators to provide backup power are critically important. Computer equipment is not just susceptible to power outages. Voltage fluctuations, drops as well as surges, can damage expensive equipment. Many organizations find it prudent to purchase uninterruptible power supplies (UPSs), essentially large batteries with regulators, to condition the power to avoid such damage. Where prolonged power outages occur, or where even short disruptions of service cannot be

tolerated, organizations will find it necessary to provide backup generators to ensure that at least the most critical systems remain operating. As with all technical investments, the cost of systems varies greatly.

We have already discussed the importance of backing up system data and will not elaborate much beyond that here. With respect to technical investments, organizational managers must recognize that purchasing some type of backup system is probably essential. However, there is a wide range of backup systems. As indicated above, managers will want to assess the consequences of data loss and service disruption in determining how sophisticated, really meaning how expensive, a backup system is required.

Likewise with systems and network management applications, we have addressed the usage of these tools in the discussion of systems monitoring ans incident response. A wide range of tools exist and the technology in this area is continuously improving. Early versions of these tools were largely limited the reporting of component failures. Later generations of system management tools started to provide more information about the health and status of the IS components they were designed to monitor. The latest generation of tools has begun to more directly monitor the health and performance of actual information services rather than monitoring the technical components which comprise the services. By monitoring services rather than technical components, these tools monitor what organizations are really interested in, service quality. The most advanced systems and network applications are even capable of conducting rudimentary forms of causal analysis such that when service outages or performance degradations are detected, the applications are able to combine service and component analysis to determine the specific causes of the observed problem.

It is unrealistic to expect that organizational managers to be involved in making the technical choices required in the acquisition of technical controls. However, management does need to understand that such technical choices are required if the organization's IT infrastructure is going to support the quality of service delivery that the organization requires. Better managers will be able to engage their IT staff in a conversation in which the relationship between recommended technical controls and service quality can be explained.

Summary

It seems odd to write a chapter summary when all of the sections in this chapter constitute summaries of concepts and issues commonly presented in much greater depth. However, that is the nature of a textbook and it is appropriate that we reemphasize those details that are most salient to future managers concerned with the avoidance of information systems failures.

As indicated in the introduction, perhaps the most important point to takeaway from this chapter is that organizational management should be involved assessing the consequences of systems failures and deciding an appropriate level of investment to minimize the potential failures. As with many management prescriptions, the advice is far easier to understand than to execute.

The chapter began with an introduction of some important IT terms: IT business applications and IT infrastructure. These terms provide a useful way of thinking about how information systems are constituted within an organization and a language for communicating with IT professionals. The IT Infrastructure Library (ITIL) was briefly described to introduce the concepts of service level requirements in service level agreements. Service level agreements, when properly negotiated provide a solid foundation for planning and implementing appropriate IT solutions that are effectively aligned with the organization's needs. Thus the negotiation of a level agreement provides a means for understanding and documenting the organization's tolerance for system failure.

The chapter then explores the nature of information system failure. Information systems are unique among organizational assets in the variety of possible failure modes. Failure may occur where the confidentiality of information stored within a system is breached, irrespective of whether the system is physically damaged or information destroyed. Additionally, organizations must be concerned with the integrity or accuracy of its information as well as ensuring that the information or IT enabled services remain available to meet its operational needs.

The chapter introduces risk management as a disciplined means for organizations to identify their information system assets, the vulnerabilities associated with those assets, and potential threats (the means that exploit vulnerabilities). Various qualitative and quantitative methods can be used to analyze these data in order to determine the selection and implementation of countermeasures appropriate for organizational needs. A more detailed discussion of risk and the risk assessment process is provided elsewhere in the book. While organization management may rely on IT and security professionals for information regarding relevant vulnerabilities, threats and countermeasures, the identification, valuation of assets and the determination of consequences in the event of system failure, most appropriately lies on their shoulders.

After presenting these foundational concepts, the bulk of the briefly introduces organizational activities associated with mitigating risks. Risk mitigation efforts are broadly categorized as following within two categories: management controls and operational controls. Management controls include:

- Establishing appropriate information assurance policies, procedures, standards and education,
- Incorporating risk management concerns into the organization's information system planning and design processes, and
- Establishing of formal change and configuration management processes.

Operational controls include:

- Establishing information system monitoring and incident response capabilities,
- Performing system backups, and
- Planning for disaster recovery and business continuity.

Selected technical controls, representing additional IT infrastructure investments, were briefly described as well.

In closing, we note that managers, at least within larger organizations, will rely on IT and security professionals to assist in the analysis of risk and the design and implementation of countermeasures. However, non-IT managers, particularly those working at executive levels, need a basic understanding of the concepts presented in this chapter to ensure that their organizations are properly protected. Depending on the organization's specific circumstances, there may be strong financial, legal and moral obligations to avoid information systems failures.

Chapter editor

John C. Beachboard joined the Computer Information Systems faculty at Idaho State University in 2001. He completed the Ph. D. in Information Transfer and the M.S. in Information Resources Management at the School of Information Studies, Syracuse University. He holds an M.S. in Business Administration from Boston University and a B.S. in Public Administration from the University of Arizona. Dr. Beachboard has taught graduate courses in research methods, project management, and IT use in business, and undergraduate courses in IT management,

systems architectures, information assurance and networking. He has held staff and management positions developing, implementing and operating information and telecommunications systems for the Department of Defense. He is keenly interested in the development and effective implementation of information technology service management practices within private and public sectors and information technology adoption within the health-care industry.

References

Alberts, C., & Dorofee, A. (2002). *Managing information security risks: The OCTAVE(sm) approach.* Boston, MA: Addison-Wesley.

Behr, K., Kim, G. & Spafford, G. (2004). *The visible ops handbook: Starting ITIL in 4 practical steps* . Eugene, OR: Information Technology Process Institute.

McCumber, J. (2005). *Assessing and managing security risk in IT systems: A structured methodology*. Boca Raton, FL: Auerbach Publications.

Microsoft (2006). *Security risk management guide*. *http://www.microsoft.com/technet/security/guidance/complianceandpolicies/secrisk/)*

National Institute of Standards and Technology. (2002). *Risk management guide for information technology systems*. *Special Publication 800-30*. Washington, DC: U.S. GPO. (http://csrc.nist.gov/publications/nistpubs/800-34/sp800-30.pdf)

National Institute of Standards and Technology. (2002). *Contingency planning guide for information technology systems*. *Special Publication 800-34*. Washington, DC: U.S. GPO. (http://csrc.nist.gov/publications/nistpubs/800-34/sp800-34.pdf)

Office of Government and Commerce. (2007). I*ntroduction to the ITIL service lifecycle (ITIL Version 3). United Kingdom: The Stationary Office.*

Weill, P., & Vitale, M. (2002). What IT infrastructure capabilities are needed to implement E-business models? *MIS Quarterly Executive, 1*(1), 17-34.

Weill, P., & Broadbent, M. (1998). *Leveraging the new infrastructure: How market leaders capitalize on information technology*. Boston, MA: Harvard Business School Press.

http://en.wikipedia.org/wiki/ITIL

http://en.wikipedia.org/wiki/Risk_management

http://en.wikipedia.org/wiki/Backup

http://en.wikipedia.org/wiki/Disaster_recovery

Glossary

Availability Denial of access to the information or information service; an IS failure mode.

Business application The software developed to automate or inform specific business activities or processes.

Business continuity plan Provides detailed guidance for reestablishing critical organizational functions or processes the event of any circumstance that massively disrupts the normal operations of an organization.

Change management Processes intended to ensure that system changes are properly authorized, prioritized and tested, and that all interested parties are informed regarding proposed changes.

Cold [backup] site A backup facility that contains required floorspace, electrical power and environmental controls required to support necessary information systems, but the hardware, software and communications to establish those systems are not installed.

Confidentiality The exposure of information to unauthorized persons; an IS failure mode.

Disaster recovery plan Provides detailed guidance concerning the actions to be taken to restore operations of information system facilities in the event of catastrophic failures, e.g., flood, fire or prolonged power failure.

Full system backups All the software and data residing on the system is backed up.

Hot [backup] site Fully operational site where essential systems, hardware and systems software, have been duplicated; data and application software data may need to be installed on these backup systems before operations can be resumed.

Incidents Any event that impacts the confidentiality, integrity or availability of an IT-enabled information service.

Incident management The processes that an organization has in place to respond to detected incidents.

Incremental system backups Only data that has changed since the last full backup or last incremental backup is backed up.

Information system failure When an information system fails to meet the organization's requirements.

Information technology infrastructure A foundation or platform , consisting of hardware, software, communications and the human resources, which is needed to support business applications.

Information Technology Infrastructure Library A comprehensive framework of IT "best management" practices developed by the Office of Government Commerce (OGC)of the United Kingdom.

Integrity Accidental or unauthorized modification information; an IS failure mode.

Management controls: managerial processes which identify organizational requirements for system confidentiality, integrity and availability and establish the various management controls intended to ensure that those requirements are satisfied.

Operational controls: include day-to-day processes more directly associated with the actual delivery of the information services.

Policies High-level statements communicating an organization's goals, objectives, and the general means for their accomplishment.

Problem management The processes to broadly identify and resolve the underlying causes of incidents that are occurring.

Procedures Instructions for performing policy- or standard-related tasks.

Risk assessment An analysis of organizational assets in terms of vulnerabilities and threats capable of exploiting those vulnerabilities performed as a critical initial step in determining which risk mitigation techniques that an organization should adopt.

Risk Mitigation Actions designed to counter identified IS threats or causes of IS failure.

Service Level Agreements Essentially, a negotiated contract between the organization and its IT service provider specifying the type and quality of IT services to be provided; the formality of this agreement may be determined to the extent within an in-house or external organization provides the IT services.

Service Level Requirements A set of operational specifications to assess the quality of individual IT services, typically stated in terms of availability, performance and security requirements.

Standards A specific class of policies representing rules, technical choices or a combination of the two.

System backup Saving a copy of data stored on a system's hard drives to a different data storage facility.

S**ystem monitoring** Process for detecting incidents or events indicating system failure or an increased potential for information failure; system monitoring can be performed reactively, monitoring for actual system failures, or proactively, monitoring for indications that a system failure is likely to occur.

Technical controls: technical capabilities incorporated into the IT infrastructure specifically to support increased confidentiality, integrity and availability of information services.

Threats Any person, object or circumstance that has the potential for causing an IS failure.

Warm [backup] site Information systems, e.g., servers, workstations and basic communications facilities, exist but are not necessarily configured for operation or have required data on-site to restore services; additional hardware and communications capabilities may need to be installed.

16. Green IS: Building Sustainable Business Practices

Editor: Richard T. Watson (University of Georgia, USA)

Contributors: Marie-Claude Boudreau, Adela Chen, Mark Huber (University of Georgia, USA)

Reviewer: Geoff Dick (University of New South Wales, Australia)

Learning objectives

- Understand the need for sustainability
- Know the difference between green Information Systems (IS) and green Information Technology (IT)
- Use the u-factors to analyze how IS can support a green physical system
- Apply a framework to identify opportunities for green IS and green IT
- Understand the need to align corporate and IS green strategies
- Understand three different approaches to ecological thinking

Sustainability

A global UN survey to determine the issues dominating the future identified sustainable economic development as the preeminent issue. The report notes, 'Never before has world opinion been so united on a single goal as it is on achieving sustainable development'. The current trend in our consumption of the earth's resources is unsustainable and is creating major environment problems. Climate change, resource depletion, loss of biodiversity, and air pollution have a major impact on many citizens and the earth, and we need to change our current behavior. Our present use of the earth's finite resources cannot be maintained. We need to move to sustainable development, which 'meets the needs of the present without compromising the ability of future generations to meet their own needs' (Brundtland, 1987, p. 8).

The environmental burden is a function of population, wealth, and technology and controlling the first two factors is extremely challenging. The larger the population, the more impact it has upon the earth. In addition, the vast majority of people aspire to affluent lifestyles, and wealthier people consume far more resources than less affluent people. Technology is both a cause of the environmental burden and also a potential solution.

Technology such as coal-fired power stations provides the electricity we need to support an affluent lifestyle, but at the same time it creates carbon emissions that contribute to global warming. Alternatively, renewable energy technologies based on wind and solar, for example, are possible solutions for sustainability, though each has negative consequences as well (e.g., the energy and materials required to construct wind turbines or solar panels). In the IT space, the disposal of equipment is a major environmental problem because of the toxic products in computers and displays. However, IS has been the major contributor to productivity growth in many countries over

the last half century. We will need IT to run the information systems that will support sustainable business practices.

Technology is an important means by which we can address our global problem. Leveraging technologies to produce goods and services that are environmentally friendlier is a momentous endeavor, and may in fact constitute 'one of the biggest opportunities in the history of commerce' (Hart, 1997).

Many business leaders are linking sustainability to their corporate strategy. They recognize that they have key responsibility to participate in solving this critical global problem and that their customers expect them to provide green products and services. Sustainability requires sustainable business practices because of the dominant role of corporations in the global economy, and IS will be a major element in the transition to a sustainable economy (Esty & Winston, 2006)

The need for green IS and green IT

The IT industry, often at the forefront of managerial practice, is an active player in supporting sustainable economic development. CIOs have identified Green IT as one of the most important strategic technologies for 2008. We carefully distinguish between green IS and green IT. There is a key difference.

- An information technology (IT) transmits, processes, or stores information.
- An information system (IS) is an integrated and cooperating set of software using information technologies to support individual, group, organizational, or societal goals.

Green IT is mainly focused on energy efficiency and equipment utilization. It addresses issues such as

- Designing energy efficient chips and disk drives
- Replacing personal computers with energy efficient thin clients
- Use of virtualization software to run multiple operating systems on one server
- Reducing the energy consumption of data centers
- Using renewable energy sources to power data centers
- Reducing electronic waste from obsolete computing equipment
- Promoting telecommuting and remote computer administration to reduce transportation emissions

Green IS, in contrast, refers to the design and implementation of information systems that contribute to sustainable business processes. Green IS, for example, helps an organization to

- Reduce transportation costs with a fleet management system and dynamic routing of vehicles to avoid traffic congestion and minimize energy consumption
- Support team work and meetings when employees are distributed throughout the world, and thus reduce the impact of air travel. IS can move remote working beyond telecommuting to include systems that support collaboration, group document management, cooperative knowledge management, and so forth.
- Track environmental information (such as toxicity, energy used, water used, etc.) about the creation of products, their components, and the fulfillment of services
- Monitor a firm's operational emissions and waste products to manage them more effectively

- Provides information to consumers so they can make green choices more conveniently and effectively.

Green IS has a greater potential than green IT because it tackles a much larger problem. It can make entire systems more sustainable compared to reducing the energy required to operate information technologies.

Green IS, and sustainable development, should not been seen as a cost of doing business. Rather, they are opportunities for organizations to improve productivity, reduce costs, and enhance profitability. Poor environmental practices result in many forms of waste. Unused resources, energy inefficiency, noise, heat, and emissions are all waste products that subtract from economic efficiency. Less waste means a more efficient enterprise. Firms that actively pursue green IS to create sustainable business practices are doing the right thing for their community, customers, investors, and future generations.

Managers seeking to create sustainable organizations and green IS should find frameworks very useful for thinking about problems, brainstorming solutions, and planning implementation of innovations. Hence, it is important to provide some frameworks for assisting in the development of green IS. We start by recognizing the four fundamental drives of information systems.

The information drives

We are addicted to information. People in affluent societies surround themselves with information appliances, such as cell phones, music players, and navigational systems. In the developing economies, nearly everyone can see the value of a cell phone and aspires to own one. Humans' inner need for information leads them to seek information systems that provide ubiquity (e.g., cell phones), uniqueness (e.g., navigation systems), unison (e.g., synchronized calendars), and universal services (e.g., high functionality smart phones) (Junglas & Watson, 2006). Satisfying these four information drives is a key ingredient in creating a successful IS, and we also believe critical to designing sustainable business practices. Only recently have we had the IT to fulfill these intrinsic human information drives, mainly because of the advent of network technologies, such as the Internet, WiFi, GPS, and mobile phone systems.

Table 1: The information drives and their physical counterparts

	Informational	**Physical**
U-construct	**The drive to …**	**The drive to …**
Ubiquity	have access to information unconstrained by time and space	have ready availability of a desired resource
Uniqueness	know precisely the characteristics and location of a person or entity	have the capability to tailor precisely the use of a physical resource to one's unique needs
Unison	have information consistency	have procedural consistency
Universality	overcome the friction of information systems' incompatibilities	overcome the friction of physical differences

If we are to change industry and society in the direction of greater ecological sustainability, we need to understand how to satisfy the four information drives. First, let's clearly define each of the drives from both a physical and informational perspective.

Ubiquity

Ubiquity, in an informational sense, is *'access to information unconstrained by time and space'*. This means, for instance, that I want to be able to use my cell phone to call anyone no matter where I am in the world, or that I

expect to be able find a WiFi connection in a hotel room or coffee shop so I can access the Internet. The worldwide popularity of cell phones is clear evidence of the strength of this drive.

In a physical sense, ubiquity is *the ready availability of a desired resource*. While we might expect that information and communication service should be accessible nearly everywhere, our expectations of physical resources are tempered by experience and reality. If we are to build a sustainable society, there needs to be a certain density of critical physical resources for them to be generally useful (e.g., the frequency of buses and placement of bus stops will affect patronage).

An appropriate IS can enhance physical ubiquity by supplying customers with information about the physical system. For example, people using a public transit system would find it very convenient to know the location of the nearest bus stop, their distance from it, the arrival time of next bus, and whether seats are available. Ubiquitous information access could be used to increase the utilization of many physical assets and thus contribute to sustainability.

Uniqueness

Uniqueness, from an information point of view, means '*knowing precisely the characteristics and location of a person or entity*'. We can use a GPS[72] to find out where we are. Companies are using RFID[73] tags for identifying products (so they can look up a database to find out their characteristics) and scanners to track their movement (so they know where they are). Some people embed RFID tags in their car so they can find it if it is stolen.

Physically, uniqueness is *the capability to tailor precisely the use of a physical resource to one's unique needs*. People often prefer using a personal car to taking public transportation because a car can get you can from A to B exactly the way you want to go. To provide higher levels of physical uniqueness for public systems, we need to support them with information systems so that consumers can more readily match available resources to needs (e.g., how to use a public transit system to get from the airport to a hotel).

Some luxury cars now offer the capacity to remember a particular person's setting for the driver's seat, external mirrors, favorite radio, station and so forth. Each time the person hops into the driver seat, she can select her unique identifier (e.g., driver number), and her predefined preferences are automatically set. This example illustrates how an IS, the car's preference memory system, supports tailoring physical resources to a person's unique needs.

Unison

In an information sense, unison is '*information consistency*'. People want a single source of accurate data. Corporations talk of the 'single view of the customer', which means an integrated database that contains a single entry for each customer. Too often, organizations have built functional systems that serve the needs of different sectors of the organization (e.g., marketing or production), and consequently, they can have duplicate, not necessarily identical, information about customers, suppliers, and so forth in different databases. At a personal level, we prefer to have a single set of browser bookmarks that are available on whichever computer we use to access the Web.[74]

72 Global Positioning System.
73 Radio frequency identification.
74 There are plug-ins for Firefox that support bookmarks unison.

Physically, unison is *procedural consistency*. This refers to a procedure for accessing or using a physical resource that has little variation across access points. A city transit system, for instance, might have the same process for buying tickets for bus, train, and water transport. Thus, consumers have to learn only one convenient procedure. Because personal time is a scarce resource for many people, procedural consistency is desirable and reassuring.

Information systems can made procedures simple and familiar. They can provide easy to use interfaces that hide procedural complexity and integrate information across physical systems. For example, an Australian arriving at Paris' Charles de Gaulle airport can use a familiar ATM-like kiosk with commands in English that accepts credit cards to purchase a ticket for travel within Paris that covers use of three transit systems. English and the use of a credit card with an ATM are all familiar procedures and highly consistent across many such encounters.

Universality

Universality, on the information side, is the drive to *'overcome the friction of information systems' incompatibilities'*. The universality drive surfaces in standards (e.g., XML[75]), currency unions (e.g., the euro), and multi-functional smart phones (e.g., one that includes a phone, GPS, camera, PDA, media player). An outstanding example of universality in action is the metric system of measurement.[76] The vast majority of countries follow this standard system. Imagine the difficulty of trading if there were multiple measurement systems, which there were before standardization.

Physically, universality is sought to *overcome the friction of physical differences*. Travelers often take along universal adapters because of regional differences in electrical outlet connections. The chargers for different brands of laptop computers are usually incompatible because of different types of connectors. Fortunately, we have standards, such as USB,[77] that facilitate the transfer of data between computers.

Information systems can help the transition between physical systems. For example, some power supplies for laptop computers can sense the characteristics of the power supply (i.e., voltage and frequency) and transform the input to that required by the computer. The sensor is a simple information system. Physical payment systems are a major form of inefficiency (e.g., national currencies), and a smart card based information system can lubricate payment by storing multiple currencies and dynamically converting between them when payments are made.

If a system is to serve its customers, then it should satisfy the four u-drives from both a physical and informational sense. Given the flexibility of information, it is much easier to provide high levels of the informational u-drives and, in so doing, increase the utility of the physical drives. An example illustrates the symbiotic[78] relationship that can be established between physical and informational systems to reduce carbon emissions..

Vélib

Vélib (a short form of Vélo Liberté, i.e., Bicycle Freedom) is the world's largest public self-service bicycle rental system.[79] The mayor of Paris launched the project in July 2007 to reduce the number of cars on the French capital's

75 XML is a language for data exchange that makes it simpler for the exchange of data by creating industry standard descriptions of items such as a credit card statement, an airline reservation, and medical record.
76 Also known as the International System of Units (SI) ('Système International d'Unités' in French, hence 'SI').
77 Universal serial bus (note the use of universal)
78 Symbiosis describes the interaction between two different organisms living in close physical association, and it this case we are applying this concept to two systems, the physical and informational, that can be designed to interact with each other.
79 http://www.velib.paris.fr/

roads. The intention is to enable Paris' citizen to use bicycles, instead of cars, for short trips (typically less than 30 minutes) within Paris. Vélib started with 10,648 bicycles and 750 stations and by early 2008 there were 20,000 bikes and 1,450 stations. The project is a great success and recorded 2 million trips on the first 40 days. Its operation is simple; subscribers go to a station, identify themselves, and rent a bicycle. The first 30 minutes of rental are free, with an incremental cost thereafter. Subscriptions are available for one day (€1), for one week (€5), or for a year (€29), with a security deposit of €150.

Vélib's information system is state-of-the-art for bicycle-sharing programs. Each station has a computer terminal ('borne'), from which an individual can purchase a subscription, recharge an account (for one-year subscriptions), determine the number of available bicycles at nearby stations (useful if the current location is empty), or see the state of that person's account. Stations consist of a series 'bornettes', with each bornette consisting of a bicycle stand, locking mechanism, and a swiper to read the subscriber's information. All bikes are uniquely identified with an RFID tag.

We now examine how the informational and physical, respectively u-factors [shouldn't we stick to u-drives?] reinforce each other to contribute to Vélib's success. Informational **ubiquity** is high because members can determine bicycles availability for any given station from any device that can connect to the Internet (e.g., a cell phone) or from another station. On the physical side, ubiquity is also high because bicycle stations are 300 meters apart, more than four times the density of Métro stations, which is very impressive because Paris's subway stations are the most closely spaced of any such mass transit system. Bicycles and customers are **uniquely** identified. Consequently, the beginning and ending stations of every ride can be tracked, and this information can be used to decide the placement and size of stations and when to move bicycles between them. Renters can tailor their ride to their personal needs quite accurately (at most 300 meters from the desired destination). A single integrated database is used to keep complete records of stations, bikes, customers and so forth. Furthermore, there is a standard process for rental (i.e., the same type of rental and payment procedures for all stations). Because the informational and physical aspects are both highly consistent , the system is high on the two types of **unison**. Informational **universality** is high, because payment and subscription systems work across all bicycle stations. Vélib terminals accept all traditional forms of payments (bank card, credit card, and cash), including the Moneo card, an electronic purse system for cashless small purchases. The Navigo smartcard, which works with the entire Parisian public transportation system, also gives access to Vélib for one-year subscribers. In the physical domain, universality is high because of the uniformity of bicycles, which reduces human learning and gives economies of scale in manufacturing and maintenance.

Systems design

The Vélib case illustrates the importance of co-designing physical and informational systems. In this case, the IS increases the convenience of Vélib. Customers can quickly find the nearest free bike, pay for a rental rapidly and simply, and be on their way. Vélib's managers can monitor the demand at each station and increase or decrease the number of bikes to maximize utilization. Information adds value to physical systems and in so doing increasing their potential patronage. High quality public systems, from both in a physical and informational sense are required to create a sustainable society.

A frameworks of sustainability options

There are three types of sustainability goals (Hart, 1997). The first goal is to prevent pollution by minimizing the level of emissions, effluents, and wastes. The second and higher level goal is product stewardship, where one

focuses on both reducing pollution and also minimizing the adverse environmental effects associated with the full life cycle of a product. This is also known as the 'cradle-to-cradle' approach, where the end state of a product is involved in the beginning of another. The third and ultimate goal is the use of clean technology that creates no harmful emissions or waste.

The three goals can apply at three different levels: individual, organizational, and societal. The combination framework (Table 2) can be used to identify opportunities to deploy IS or IT to improve sustainability. We now discuss each of the cells.

Table 2: Green IS and IT opportunities

	Individual	**Organizational**	**Societal**
Pollution Prevention	flexible printing capabilities automated energy conservation system	thin client virtualization telecommuting	electronic exchange of information congestion systems
Product Stewardship	recycling	reuse components recycle computers	governmental policies societal norms
Clean Technology	paperless interaction	video conferencing collaboration tools	open source Smart homes and appliances e-commerce vs. traditional commerce

Individual pollution prevention

There are many actions that you can take to reduce the IT impact on pollution. For example, you can turn off you computer when you will not use it for some hours. You might practice shutting it down when you go to bed. You could print on both sides of a sheet of paper (i.e., duplex) or turn on the energy conservation preferences for your operating system (see Exhibit 1) so that your computer will go to sleep after a certain period of inactivity. Flexible printing capabilities exist for most operating systems; yet, they are rarely activated. It is estimated that applying energy settings, such as 'sleep when inactive', can reduce greenhouse gas emissions at a rate equal to taking more than 8,000 passenger cars off the road for an entire year, or conserving 16 million liters of gasoline.[80]

Individual product stewardship

In additions to using energy more efficiently, you can play a significant role in product stewardship, such as in recycling used electronic products. For example, in our city there is a group[81] that takes unwanted computers, refurbishes them, installs Linux and OpenOffice, and gives them to charitable organizations. Organizations are relying on you to support cradle-to-cradle manufacturing. When you decide to dispose of an electronic product, check its manufacturer's web site for recycling options and procedures.

80http://www.techworld.com/green-it/features/index.cfm?featureid=3496
81 http://freeitathens.com/

Information systems can facilitate product stewardship by providing information and creating networks to support recycling. Started in 2003, the Freecycle Network,[82] promotes waste reduction by providing individuals and non-profits an electronic forum to recycle unwanted items. As they say, 'one person's trash can be another's treasure'. The Freecycle concept has since spread to over 75 countries and includes millions of members. Freecycle claims to keep over 275 metric tons of goods per day out of landfills.

Individual adoption of cleaner technology

Many of us find it difficult change established habits. Substitution is the simplest change. For example substituting check writing by paying bills online is a relatively easy change with a positive impact on the environment. It is faster and more convenient, and adds up: If every US home received and paid its bills online, annual greenhouse gas emissions would drop by 1.9 million metric tons, and waste would be reduced by nearly 1.45 million metric tons per a year.[83] UNESCO reports that of the average 1,510 sheets of paper produced per person in the world per year, at least half of this sheets goes through printers and copiers to produce office documents. A single tree produces about 80,500 sheets of paper.[84] Electronic media can be more environmentally friendly than paper. Acquiring news, music, movies, and books in electronic format is now possible because of the technological infrastructure and information systems in place. E-books (such as Sony's eBook reader) can reduce paper consumption. While the earlier e-readers are quite expensive, expect the cost to decline with time and volume.

Many cleaner technologies rely on an IS. The iPod (for music and movies) is backed by a sophisticated IS called iTunes. Amazon's Kindle (for books, magazine, and newspapers) is supported by a similar IS. The Toyota Prius, the world's most popular hybrid car, contains multiple computer chips to run its many information systems. It needs an IS to decide when the run the gas and electric engines, when to charge the battery, and what information to display to the driver.

Organizational pollution prevention

Organizations can redesign their IT infrastructure to make it more energy efficient. A thin client, a lean PC that relies on a central server for disk storage and applications processing, uses less energy than a regular PC. Verizon, for example, reduced energy consumption by 30 percent by replacing personal computers in its call center with thin clients.[85] Germany's Fraunhofer Institute reports that, when comparing thin clients to personal computers, energy consumption is at least twice as low, even when factoring in the additional energy and cooling power required by the server associated with the thin clients. In addition to the reduction of emissions, e-waste is also reduced by switching to thin clients. A thin client contains significantly fewer components and has a longer life expectancy than a regular PC.

Virtualization, software running on a virtual foundation rather than the physical hardware, has become a popular energy saver. Server virtualization (the most common form of virtualization) makes the physical resource (i.e., the server) function as multiple logical resources (e.g., running multiple operating systems). Virtualization means doing more work with fewer resources, which in turn frees up data center space and lowers energy bills. Virtualization has existed in the computer industry for decades, but it is now getting a lot of attention because of its

82 http://www.freecycle.org/
83 http://www.time.com/time/specials/2007/environment/article/0,28804,1602354_1603074_1603109,00.html
84 http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/print.htm
85 http://www.businessweek.com/technology/content/may2007/tc20070514_656985.htm?campaign_id=rss_tech

capacity to reduce energy consumption by increasing the utilization of excess or idle capacity possessed by existing hardware.

Organizational product stewardship

Sustainability requires that we develop extensive recycling systems and we change our behavior to think of recycling as the first step when we dispose of items we no long want. In many cases, electronic goods are not recycled because organizations have not created the procedures and information systems to facilitate recycling. Most organizations have complex information systems for manufacturing, distribution, and sales to get their products into the homes of consumer, but few go the complete cycle. That is, they don't consider that they are responsible for taking back products that their customers no long want. A few forward looking companies, however, have created such full cycle systems. Dell, for example, allows its customers to recycle their old printers (if they buy a new one) by simply providing the new printer's service tag and scheduling a pick-up; customers accomplish both activities online via Dells's web site. Dell has created a simple and convenient process, assisted with an IS to track the movement of the products to be recycled. It can also gain by recycling some of the returned products, or parts of them, where this is possible.

Organizationally cleaner technology

Humans prefer a face-to-face meeting over a telephone conversation on many occasions because of the richness of the interaction. Face-to-face meetings, however, can consume considerable energy when the attendees are scattered across the globe. Video conferencing is a good alternative, particularly with today's high quality systems. Video conferencing can transcend distance to replicate face-to-face communication. In the era of globalization and global climate change, organizations need to substitute cleaner technologies, such as video conferencing and electronic collaboration tools, to bridge the distance when a meeting's participants or a work team are scattered across different cities, countries, and continents. Electronic distributed meetings support communication without the carbon footprint of travel.

Organizations can also used cleaner technology (e.g., solar or hydro power) to run their data centers. Data center energy consumption is one of the most important green technological concerns because power and cooling account for up to 40 percent of a data center's costs.[86] Google, Yahoo, and Microsoft have located some of their data centers in the Pacific Northwest of the U.S., close to cheap hydro-electric power. Some organizations are looking at solar power for their data centers.

Societal pollution prevention

Countries and economic regions can reduce pollution by encouraging a shift to technologies that produce less emissions. In the case of IS, the energy cost of exchanging data can be significantly reduced by moving from the postal system to electronic networks. Electronic Data Interchange (EDI), for example, supports the majority of electronic commerce transactions. Depending on which standard is in use (ANSI ASC X12 in North America and Edifact elsewhere), structured information can be interchanged between and within organizations, governments, and other groups. In a similar way, XML supports the electronic exchange of information through an open standard. Both of these technologies can reduce the use and manipulation of physical administrative documents (e.g., invoices, sales orders, etc.), and thus minimize pollution. They also, as is the case with most efforts to increase sustainability, greatly reduce an organization's administrative costs.

86 http://www.cioinsight.com/c/a/Trends/The-Greening-of-the-CIO/1/

Traffic congestion is a major issue for most large cities. It wastes energy and increases pollution. Cities, such as Singapore and London, now levy fees for the use of particular city roads, with the help of information system, to reduce congestion. The US Department of Transportation employs the intelligent transportation system (ITS) to optimize public transportation by reducing congestion, improving road safety, and enhancing productivity. ITS is built upon a broad range of wireless and wire line communications-based information and electronics technologies. The US federal government is fostering widespread deployment of the system by integrating it into the transportation system's infrastructure. Current applications of ITS system include computer aided dispatching of vehicles, automatic vehicle location for public buses, and electronic toll booths that do not require driver to stop, and electronic freeway surveillance.

Societal product stewardship

Governments can play an active role in encouraging, and where necessary forcing, organizations to become better product stewards. Legislation is being used to make recycling of electronic products mandatory. The California state government, for instance, introduced an Electronic Waste Recycling Fee in 2004 on all new monitors and televisions sold. California's Electronic Waste Recycling Act mandates that retailers collect a set recycling fee and pass it on to the Board of Equalization. British Columbia in Canada has a similar policy. In 2003, the European Union enacted the Waste Electrical and Electronic Equipment Directive (WEEE Directive), which has become European Law, setting collection, recycling, and recovery targets for all types of electrical goods.

Societal cleaner technology

An information society that consumes (e.g., downloading movies via the Internet rather than renting from a local store) and exchanges information electronically (e.g., emailing rather than posting a letter) is cleaner than a society in which information exchange is based on paper and the postal system. An information society can also organize for the production and distribution of electronic goods to be cleaner. The open source model is a very good example of a cleaner form of production and distribution. Software is developed without requiring the physical presence of workers in the same physical space, that is, an office building and its significant infrastructure and the environmental costs of daily commutes. Moreover, once developed, open source software can freely flow across borders at electronic speed, without the need for wasteful packaging and retail store shelf space. The footprint associated with both production and distribution can be much lower for information products.

Beyond information products, the information age needs to find many other ways in which it can deploy IS to minimize society's ecological footprint. We need a generation of innovation to create a sustainable society, and much of this innovation will involve IT and IS is a variety of ways.

Organizational perspectives

Organizations are the major force for innovation in most societies, and corporations in particular are major change agents. As a result, we further examine some frameworks for promoting thinking about organizational sustainability.

Strategic alignment

Nearly all major enterprises establish a corporate strategy that guides their major actions and set directions for the future. To achieve societal sustainability, we need the great bulk of major corporations to incorporate sustainability as part of their corporate strategy. As most of the major firms are global, we turn to a global strategic framework as the foundation for discussing how enterprises can approach integration of sustainability into their

corporate strategy. Corporations who move faster and more effectively than those in their industry to create sustainable business practices should gain a competitive advantage. Eliminating waste increases profitability, and organizations need to learn how to operate in a world in which emission constraints are a part of doing business. Strategic issues can be addressed by from the perspective of aggregation, adaptation, and arbitrage, the AAA triangle (Ghemawat, 2007).

Aggregation

Organizations strategically seek economies of scale by **aggregating** development and production processes. The intention is to reduce costs by combing activities into optimal units for efficiency. From a sustainability angle, organizations also want to aggregate activities to reduce emissions and waste.

Wal-Mart's green supply chain

Through emphasizing a green supply chain, Wal-Mart plans to create less waste. It has taken strategic action to minimizing packaging. The idea is to reduce the size of products to save energy, shipping costs, and shelf space. It wants vendors to think 'small and mighty' by aggregating goods in the minimal space. It has, for instance, convinced vendors to replace bulky plastic jugs with condensed, slimmed-down containers for liquid laundry detergents. Toilet paper manufacturers have compacted their products so that a greater quantity can fit in a given volume. Through such initiatives, Wal-Mart aligns one of its key goals, low cost leadership in retailing, with sustainability because its movement in the direction of sustainable business practices reduce emissions, wastes, and costs.

IS can be used to measure and monitor the costs, emissions, and waste of each phase of a supply chain and packaging alternatives. It is also a tool for coordinating and aggregating the many activities in a supply chain to minimize overall emissions. From an IT perspective, aggregation describes actions such as locating servers in a single data center to reduce energy costs. Virtualization can be thought of as aggregating several software systems onto one server to increase utilization and lower costs.

Adaptation

Adaptation defines an organization's efforts to maximize its local relevance by being responsive to local stakeholders' needs and desires. Again, this can be done by exploiting in the power of IT and IS strategically. From an environmental perspective, this means adopting specific environmental initiatives that reduce emissions and wastes in the communities in which the organization operates.

Everything and Everybody Connected to the Network

Sun Microsystems Inc. is a global provider of network computing infrastructure solutions. Its corporate vision of 'Everything and Everybody Connected to the Network' is reflected in its desire to let 'everyone take part in opportunities and contribute to solutions regardless of their geographic location or economic situation' (company web-site). In this spirit, Sun created the 'Open Work' initiative, which consists of a solution suite of products, policies, and support tools that enable Sun employees to work effectively wherever their work takes them, may this be at the office, at home, on the road, or in drop-in centers. The company has about 43 percent of its workforce participating in

this program, utilizing its 115 flexible office locations worldwide. Through its initiative, Sun thus fosters the use of cleaner technology through a program that is in alignment with its strategic orientation.

Sun Microsystem's "Open Work" initiative, expands the definition of local and enhances its employees' abilities to work locally while competing globally. At the same time, reducing the need for employees to travel to work locations and reducing Sun's employees' overall carbon footprint.

Arbitrage

Arbitrage, the third global strategy, is the exploitation of differences between different markets. Considering the etymology of the word arbitrage, its French origin defines it as 'rendering judgment'. The underlying idea of this strategy is to achieve absolute economies through judging (and selecting) the very best alternatives. In the context of an environmental IT and IS initiative, this can viewed as achieving the most environmentally friendly product by selecting the least polluting vendors.

Cradle-to-cradle design at Herman Miller

Herman Miller, the international designer, manufacturer, and distributor of furnishings and interior products, follows an arbitrage strategy. Given that sustainability is one of its core values, Herman Miller has taken many green initiatives to distinguish itself from its competitors. One of these is the development of a cradle-to-cradle design into its products, such that all constituent components in a given product can be put back into service. This initiative led to the creation of an IS (the Design for Environment system, DfE), which allows Herman Miller to assess the extent to which a final product meets the goal of the cradle-to-cradle ideal, that is, made from 100 percent biological or technical nutrients. With DfE, Herman Miller can thus assess the components it acquires from its suppliers, and if a component does not meet its cradle-to-cradle metric, Herman Miller either helps the supplier to make needed formulation changes in its component, or, if the supplier is unwilling or incapable of making such changes, seeks an alternative supplier. In other words, Herman Miller is changing its supply chain to include only the suppliers that can contribute to the achievement of its sustainability goal. Thus, Herman Miller's goal of product stewardship is consistent with its values and is addressed by an arbitrage strategy.

In summary, the most successful sustainability initiatives will be those carried on by organizations aligning their green IS initiatives with their overall strategies, in ways that will achieve their business goals while simultaneously reinforcing their environmental goals, i.e., the reduction of pollution, product stewardship, or cleaner technology.

Three approaches to ecological thinking

Organizations strive to sustain their existence, and the notion of corporate sustainability incorporates ecological thinking and three different approaches to it: eco-efficiency, eco-equity, and eco-effectiveness. We now discuss each of these idea

Eco-efficiency

Eco-efficiency combines traditional efficiency goals with ecological considerations, and is defined as, 'the delivery of competitively-priced goods and services that satisfy human needs and bring quality of life, while

progressively reducing ecological impacts and resource intensity throughout the life-cycle to a level at least in line with the earth's carrying capacity' (DeSimone & Popoff, 2000, p. 47) Quite simply, it means consuming non-renewable materials more productively. Under eco-efficiency, financial goals still remain foremost in management's mind, but it should be mindful of the need to pursue sustainable practices where they do not interfere with financial considerations.

All waste products are a cost that a company has to bear, unless it can externalize them and make the community pay. For example, a carpet manufacture that has chemicals left over after production is legally required to dispose of these in a in a manner that does not damage the environment. This can be a costly process, and the carpet manufacturer would be more profitable if he did not have to dispose of these chemicals. The ecological approach is to switch to chemicals that don't harm the environment and thus avoid high disposal costs, or even better, the firm finds a way that requires no chemicals and thus avoid the costs of buying the chemicals.

Unfortunately, in too many cases industry passes on the costs of its eco-inefficiency to the community. It 'externalizes' its costs. A company that pollutes a stream with its waste products forces society to deal with the costs of environmental degradation. If the company were forced to bear the full cost of its polluting activities, it would have a strong incentive to be eco-efficient.

Eco-equity

Eco-equity aims for the fair distribution of natural resources between current and future generations. One group in society should not consume so much that it denies other members of its generation their fair share of that resource. Similarly, one generation should not over consume a resource to the point that it is unavailable or degraded for a future generation.

There is limited knowledge of the Earth's total stock of critical resources such as oil and water. Before we can start to implement eco-equity as a societal goal, we first need to know what resources we have and how rapidly they are being consumed. We need a data base for the full range of global resources with details of available stocks and depletion rates. Then, we need to develop methods for determining equitable distributions between and across generations. We cannot achieve eco-equity if we have insufficient data to determine what is equitable.

Eco-effectiveness

Eco-effectiveness means that we end current practices that result in ecological degradation. We need to mimic nature and create ongoing healthy systems where the waste products of one process become inputs to other processes. For example, a tree's dead leaves become food for insects and nutrients for the soil. Natural systems have had millions of years to evolve. Initially, waste products might have remained unused for eons until a species learned, or evolved, that could use the waste as its food. An eco-effective conversion cycle creates minimal waste.

In the industrial world, we can use information systems to accelerate the matching between output and input. We need to create information markets where the producers of the waste from one process can find a buyer, who will use the waste as input to another process. Eco-effectiveness requires a highly efficient information exchange so that the waste market continually clears. Attaining eco-effectiveness means that toxic dumps are no longer required and that landfills are a historical oddity that reminds future generations of their profligate forebears.

Summary

Sustainable development 'meets the needs of the present without compromising the ability of future generations to meet their own needs'. Information systems, as the major force driving productivity growth in the last half

century, should have a critical role in creating sustainable business systems. Green IT is mainly focused on energy efficiency and equipment utilization. Green IS refers to the design and implementation of information systems that contribute to sustainable business processes. There are several frameworks for identifying Green IS opportunities. First, the information drives (ubiquity, uniqueness, unison, and universality). Second, sustainability options (pollution prevention, product stewardship, and clean technology) by action levels (individual, organization, and societal). Third, strategic alignment (aggregation, adaptation, and arbitrage) of IS with the enterprise. Fourth, ecological thinking (eco-efficiency, eco-equity, and eco-effectiveness).

Exercises

1. What could your university do to increase its sustainability? How might students help?
2. What personal actions could you take to reduce energy consumption? What behaviors are you likely to change on an ongoing basis?
3. Thinking of a business process with which you are familiar, such as a stock ordering system, using the U information drives, outline how IS might improve sustainability in that process?
4. Using the U drives model, evaluate the public transport system in your city or town. How well does it meet the four information drives? How might information systems be improved to increase utilization of the transport system.

References

Brundtland, G. H. (1987). Our Common Future: Report of the World Commission on Environment and Development. Oxford: Oxford University Press.

DeSimone, L. D., & Popoff, F. (2000). Eco-Efficiency: The Business Link to Sustainable Development. MIT Press.

Esty, D. C., & Winston, A. S. (2006). Green to Gold: How Smart Companies Use Environmental Strategy to Innovate, Create Value, and Build Competitive Advantage (1st ed., p. 384). Yale University Press.

Ghemawat, P. (2007). Managing differences: the central challenge of global strategy. Harvard Business Review, 85(3), 58-68, 140.

Hart, S. L. (1997). Beyond greening: Strategies for a sustainable world. Harvard Business Review, 75(1), 66-76.

Junglas, I. A., & Watson, R. T. (2006). The U-Constructs: Four Information Drives. Communications of AIS, 17, 569-92.

17. Moving forward as a systems innovator

Editor: Paul Bauer (University of Denver, Denver, USA)

Learning objectives

•

This chapter is about creating the software-based products and services of the future (and what significant ones won't be)?

Moving Forward as a Systems Innovator

In the little over a half century since the first commercial computer was introduced, there has been phenomenal improvement in the performance to price ratio of these devices, not to mention the significant reduction in size. In the summer before graduate school, I worked for a company which made me responsible for running a critical production computer which had 4 kilo-words of core memory, a 30 kilobyte drum, 1-inch magnetic tape drives, a paper tape reader, a crude assembler, no compiler, and required 1000 square feet of floor space. The machine was reported to have cost over $1 million. The cell phone in your pocket today dwarfs that early computer in processing power by several orders of magnitude. A friend sent me an email just last week saying that he had found a 500 gigabyte hard drive on sale for $100. His comment was that less than fifteen years ago when he was working on a project to provide movies on demand over a cable network, he priced out terabyte hard drives at $10 million each. Again we see five orders of magnitude improvement in performance to price in less than 15 years. A very relevant question for a systems innovator is what key improvements are to come? And, of course, the answer to that question depends on what systems innovators do in the years to come. To explore this topic, we will look at the promise of information technology, the challenges we face in realizing the promise, and some of the resources available to help guide the systems innovator.

The Promise of Information Technology

The Industrial Revolution significantly improved mankind's living standard by replacing muscle power with mechanical devices driven by chemical, electrical, and other forms of energy. The promise of information systems, already realized to a great extent, is to produce a similar transformation not only in our mental capabilities but more importantly in our ability to communicate. Communications is at the heart of commerce-we tend to not do business with people we don't know or cannot trust. And as the pendulum of commerce has swung from one-on-one interaction with artisans and craftsmen to mass production of essential and even luxury goods, and now back again toward mass customization-mass producing goods which meet the unique requirements of each customer-information about markets, market segments, and market segments of one has become vitally important. Information technology not only gives us the opportunity to capture the required data, but to use them effectively in dealing with these diverse market populations. How many individual customer preferences can you hold in your head? Writing them down on paper significantly increases this number but also increases the work to organize

them to find or share a specific one when needed. Information systems allow us to both increase the number indefinitely and retrieve quickly a specific one as needed.

Computers are actually very dumb devices capable of dealing with only ones and zeroes in extremely logical ways. Business people, indeed most people, don't normally think that way. So initially a lot of energy went into trying to "think like a computer" to get them to produce results of value, at the expense of focusing on user requirements. This led to a schism between the techies and the tycoons; IT folks were seen as more interested in the technology than in the business objectives. As computers have gotten easier to program, we have made progress in closing that gap. Although in many companies today, there is still a process which creates a strategic plan for the company or division, then appliqués onto it an "IS strategy or plan." The opportunity here is to recognize that information technology, while becoming in one sense a commodity like electricity or water, in another sense will never become a commodity because it enables one to generate, gather and use information in unique ways. So instead of thinking of how IS can support the company strategy, leading edge companies are building their strategies around what information can be obtained and how they can use that information for competitive advantage. Their business models are built on capturing, creating, and effectively using information. And the capabilities of information systems, both extant and envisioned, are an integral part of that business model and competitive strategy.

In a CAIS working paper, Vasant Dhar [footnotes shown as such] identifies three invariant concepts upon which thinking about future business models and industry structure can be based:

1. rendering of things as information; for example, a bank balance rather than physical money as an indication of wealth.
2. exponential growth of hardware power, bandwidth, storage and the accompanying miniaturization of IT-based devices; and
3. sustained increase in programmability through modular software.

The consequences of these invariants are of substantive and lasting importance:

4. digitization facilitates the separation of information from artifacts which alters the fundamental economics of a number of industries, such as music, film and publishing.
5. IT infrastructures are becoming more powerful and more accessible; high speed digital connections now reach a large percentage of businesses and residences.
6. the importance and variety of "spaces of interactions" in society that are mediated by IT are growing; and finally
7. more data about these spaces of interaction are made available as is the ability to process these data intelligently.

"These consequences suggest a future for business that is inextricably intertwined with information technology." For the systems innovator of the future, this is a good starting point.

The Promise of Business

Business likewise is undergoing fundamental shifts with a new emphasis on sustainability and a triple bottom line. The concept of sustainability often carries overtones of "environmentally friendly" or "green", but actually

deals more broadly with the ability of a company to meet its goals today as well as position itself to meet them in the future. Couple environmental concerns with economic and social concerns, and you have the triple bottom line on which many companies today are reporting. Information technology again not only can be a catalyst and vehicle of effective execution, but it raises many relevant concerns of its own in this area. As John Thakara points out in "In the Bubble"[footnotes shown as such], "it takes 1.7 kilograms of materials to make a microchip with 32 megabytes of random-access memory-a total of 630 times the mass of the final product. The 'fab' of a basic memory chip, and running it for the typical life span of a computer, eats up eight hundred times the chip's weight in fossil fuel. Thousands of potentially toxic chemicals are used in the manufacturing process"¦ The amount of waste matter generated in the manufacture of a single laptop computer is close to four thousand times its weight on your lap. Fifteen to nineteen tons of energy and materials are consumed in the fabrication of one desktop computer. To compound matters: As well as being resource-greedy to make, information technology devices also have notoriously short lives. The average compact disc is used precisely once in its life, and every gram of material that goes into the production and consumption of a computer ends up rather quickly as either an emission or solid waste. In theory, electronic products have technical services lives on the magnitude of thirty years, but thanks to ever-shorter innovation cycles, many devices are disposed of after a few years or months." So while information technology makes more plausible achieving the promise of sustainability in business, it also adds significantly to the challenges for the systems innovator.

The Challenges of Information Technology

Rapidly changing technology, alluded to above, is a key challenge for the systems innovator. Most people today are familiar with Moore's Law: early in the life cycle of the large scale integrated circuit, Gordon Moore predicted that the density of components on a chip would double every 18 months. For over thirty years, the semiconductor industry has made good on that prediction. Think about that for a minute: if you double something every 18 months, in 30 years you would have doubled the amount 20 times, or produced 220 ~ 106 (210 = 1024 ~ 103), or a million-fold increase. So if at the start there were 1000 components on an integrated circuit (which is the order of magnitude achieved on the first memory chips in 1971), then today a chip should have a 1000 x 106 or a billion components. Intel announced in 2004 an SRAM chip with over a half billion transistors using 65 nanometer line widths.

The kind of challenge this presents can be seen in operating system programming. Early operating systems were very tightly coded because memory was scarce and cycle times relatively long. With today's billion transistor processors operating at gigahertz cycle times, that is no longer the case; Microsoft's Windows XP operating system reportedly has over 35 million lines of source code. And that, contend some, is inherently a problem since one software bug per 1000 lines of code is thought to be the currently achievable quality level. Further, such operating systems can take tens of seconds to boot up while they run though files for literally thousands of drivers for which no device is attached because it was easier to program that way. Even though it appears we will soon reach the physical limit of Moore's Law, the rapid pace of technology change will probably not slow as new techniques are introduced to compensate.

Rapidly changing technology leads directly to another challenge for the systems innovator: growing expectations. From Andrew Tanenbaum's web site (http://www.cs.vu.nl/~ast/reliable-os/): "TVs don't have reset buttons. Stereos don't have reset buttons. Cars don't have reset buttons. They are full of software but don't need them. Computers need reset buttons because their software crashes a lot. I know that computer software is different

from car software, but users just want them both to work and don't want lectures why they should expect cars to work and computers not to work." Tanenbaum further observes that TVs, stereos, and cars don't take 30 to 50 seconds to boot up. They start immediately when turned on. Users aren't interested in the gee whiz factors of the technology. They just want it to do useful things for them. And given the increases in computer power, they expect more and better useful things.

A third challenge for the systems innovator is managing the "soft side" of technology innovation. In "The Inmates Are Running the Asylum" Alan Cooper [footnotes shown as such] tells us that programming is such a difficult and absorbing task, that the creation of software is so all-consuming that programmers must immerse themselves in an alien thought process which supersedes the demands of the user. The goals of the programmer and the goals of the user are different, and the latter usually loses out. And that is tragic because when we let our products frustrate, cost, confuse, irritate and maim us, we are not taking advantage of the real promise of software based products: "to be the most human, powerful, and pleasurable creations ever imagined."

Some Resources for Systems Innovators

Fortunately significant progress has been made over the last fifty years, and there are known resources the systems innovator can draw on.

The whole process of innovation has been studied extensively (see, for example, Christensen, The Innovator's Dilemma, The Innovator's Solution, and Seeing What's Next; or Geoffrey Moore's Crossing the Chasm, Inside the Tornado, and Dealing with Darwin.) New product development and project management, two specific pieces of the innovation process, have also been widely studied and documented. The Project Management Institute (www.pmi.org (http://www.pmi.org)) has certification levels, training materials to help achieve these certifications, and periodic meetings of local chapters to hone and maintain skill sets. Similarly the Product Management and Development Association (www.pdma.org (http://www.pdma.org)) is a loose federation of professionals involved in the new product development and delivery process who share experiences and insights. Both organizations have extensive libraries which capture many of the successes and failures of the past, so that we can learn from others' mistakes rather than repeat them all ourselves.

Scenario planning is another indispensable tool for the systems innovator. Originally pioneered by Royal Dutch Shell before the energy crisis of the early 1970s, this technique is used by businesses today to plan their strategies. Since the most important technology related events in business are often disruptive events, linear projections of past performance are of little value in "seeing the future". With scenario planning, one envisions several futures-a desirable one, a troubled one and one somewhere in between. Assumptions leading to each of these possible futures are then captured and tracked, along with contingency plans related to each assumption. This puts the company in a position to quickly take advantage of opportunities and sidestep pitfalls. Since complex IT systems often require long lead times, combining scenario planning with modular, agile development techniques helps systems innovators be more responsive to business needs.

An increased emphasis on innovation in business has led to greater focus on design. Software developers have been doing design, either explicitly implicitly or explicitly, for over fifty years. What's new is the recognition, not only on the part of IT personnel, but managers in general that design does not apply uniquely to software, but to the whole business process in general. Rather than design software to meet current needs, companies on the leading edge are redesigning processes to take advantage of the opportunities software systems provide. And this design is a

joint activity of both business and technical managers, a process of give and take of equals with a common purpose- to create a more customer responsive business. Lego provides a recent example. When Lego sold its first robot kit, someone reverse engineered and published on the web the software which drove the microprocessor brain of the robot. The company's initial reaction was to sue for patent infringement. They then reconsidered the free publicity they got from this individual's efforts, and invited more customers to participate in the design of a new kit, the Santa Fe diesel engine. Two hundred volunteers contributed freely of their time, talent, and ideas. Lego planned a production run of 10,000 kits, and had all of them sold via word of mouth advertising of the two hundred participants before they even finished the production run. This kind of design cooperation in the "interaction space" provided by web technologies contributes immeasurable business value to the company and product satisfaction to the consumer.

And last but not least, professional organizations provide support for systems innovators. The Association for Computing Machinery (www.acm.org (http://www.acm.org)), which is actually an association of professionals, not of machines, provides resources that advance computing as a science and profession. The Association for Information Systems (http://plone.aisnet.org/) "is a professional organization whose purpose is to serve as the premier global organization for academics specializing in Information Systems." It does this through Special Interest Groups (SIGs), conferences and publications. The IEEE Computer Society's (http://www.computer.org/portal/site/ieeecs/index.jsp) "vision is to be the leading provider of technical information, community services, and personalized services to the world's computing professionals." Participation in these professional societies can provide the system innovator a network of like-minded individuals to help him or her learn, grow, and succeed.

Chapter editor

Paul Bauer