

Fatores (Des)Motivadores na Adoção de Metodologias Ágeis no Desenvolvimento de Sistemas de Informação

Fernando Kenji Kamei	Felipe Prata Lima	Jairo Barros Júnior	Marcilio F. Souza Júnior
Faculdade de Alagoas	CPMAT/UFAL	Maurício de Nassau	IF-AL
fkenjikamei@gmail.com	felipepratalima@gmail.com	jairobjunior@gmail.com	marcilio@cefet-al.br

RESUMO

As metodologias ágeis para desenvolvimento de sistemas são uma resposta às chamadas metodologias pesadas ou tradicionais. Métodos ágeis de engenharia de software têm emergido recentemente como uma nova e dinâmica forma de desenvolver sistemas de informação comparados aos métodos tradicionais. Entretanto, seu sucesso foi em grande parte anedótico e as investigações nesta temática ainda tem sido alvo de produções no meio acadêmico, tais como aquelas que buscam analisar e determinar os fatores críticos de sucesso de projetos de desenvolvimento ágil de sistemas com abordagem quantitativa. Neste contexto, este artigo busca elucidar fatores que motivam e/ou desmotivam as organizações e suas equipes que adotaram metodologias ágeis de desenvolvimento de sistemas em seus projetos. O procedimento metodológico utilizado foi uma pesquisa qualitativa e descritiva, baseada em uma pesquisa bibliográfica de artigos que descreveram suas experiências e lições aprendidas com esses métodos. Os fatores motivadores e desmotivadores levantados são analisados e discutidos, e outras diretrizes para a continuação dos estudos são apresentadas.

Palavras-Chave: Sistemas de Informação. Desenvolvimento de Sistemas de Informação. Metodologias Ágeis.

1. INTRODUÇÃO

Os sistemas de informações (SI) têm papel fundamental nas organizações. Através deles que um gestor consegue ter acesso com facilidade as informações de todos os aspectos de sua organização. A correta administração dessas informações é fundamental para o sucesso, pois, com base nelas os executivos podem decidir o rumo da organização.

Conforme Rezende (2003), as organizações beneficiam-se com os sistemas de informação à medida que podem: controlar suas operações; diminuir a carga de trabalho das pessoas; reduzir custos e desperdícios; aperfeiçoar a eficiência, eficácia, efetividade, qualidade e produtividade da organização; aumentar a segurança das ações; diminuir os erros; contribuir para a produção de bens e serviços; agregar valores ao produto; suportar decisões profícuas; oportunizar negócios ou atividades; e contribuir para sua inteligência organizacional.

Enquanto SI's são muito importantes em todas as facetas do mundo moderno, o desenvolvimento dos mesmos não vem sendo um processo perfeito. SI's são desenvolvidos e implantados em diversos tipos de organizações e setores (REZENDE, 2002; HEEKS, 2006; O'BRIEN, 2006; STAIR e REYNOLDS, 2006). Durante o desenvolvimento do SI, métodos são definidos pelas equipes de Tecnologia da Informação com o intuito de eliminar gastos com documentação excessiva e burocrática, enfatizando a comunicação, colaboração com o cliente e as atividades que trazem valor imediato na produção de sistemas com qualidade

(REZENDE, 2002). Por meio de um processo empírico, com ciclos constantes de inspeção e adaptação, a equipe trabalha sempre em um ambiente de melhoria contínua.

Desta forma, em desenvolvimento de SI, não é apenas a prática de programação que é necessária e fundamental para o desenvolvimento do mesmo. Organizar a forma pela qual as etapas (concepção, análise, projeto, desenvolvimento, testes, implantação) são abordadas exige que a equipe possua um bom conhecimento de todas as atividades que envolvam o ciclo como um todo. A análise de todo o processo, sendo este documentado ou não, depende do modelo de metodologia que será adotada pela organização. Existem vários modelos metodológicos de desenvolvimento que se diferem na sua forma estrutural, até na maneira de implantação do sistema na organização. Sobre engenharia de software, Rezende (2002) concluiu que é uma metodologia para o desenvolvimento de soluções em software, ou seja, um roteiro que pode utilizar diversas técnicas. A seqüência de passos preestabelecidos permite optar e variar de técnicas e ferramentas nas suas diversas fases.

Neste contexto, as metodologias ágeis para desenvolvimento de sistemas são uma resposta às chamadas metodologias pesadas ou tradicionais. Métodos ágeis de engenharia de software têm emergido recentemente com uma nova e dinâmica forma de desenvolver sistemas comparados aos métodos tradicionais. Entretanto, seu sucesso foi em grande parte anedótico e as investigações nesta temática ainda tem sido alvo de produções no meio acadêmico, tais como aquelas que buscam analisar e determinar os fatores críticos de sucesso de projetos de desenvolvimento ágil de sistemas com abordagem quantitativa (CAO, 2006; CHOW e CAO, 2008).

Em um levantamento recente da adoção de métodos ágeis nas empresas, Ambler (2009) apresenta que 69% das organizações têm adotado uma ou mais técnicas ágeis de desenvolvimento de sistemas, em um total de 642 respondentes. Em 268 empresas os times de desenvolvimento são compostos de seis a dez pessoas; 45,3% dos projetos em andamento adotaram princípios de agilidade; e 60% delas indicaram que as abordagens ágeis afetaram sua produtividade de forma significativa. Tais números corroboram com a preocupação da academia em continuar insistindo e explorando as questões que envolvem metodologias ágeis de desenvolvimento de sistemas de informação.

Face ao exposto, este artigo busca elucidar fatores que motivam e/ou desmotivam as organizações e suas equipes que adotaram metodologias ágeis de desenvolvimento de sistemas em seus projetos. Para tanto, uma pesquisa bibliográfica foi realizada, permitindo fazer uma comparação entre as metodologias adotadas como referência, apresentando as principais vantagens e desvantagens de cada uma. Contudo, o objetivo não é apenas comparar, mas definir os fatores/aspectos que motivam ou desmotivam a adoção de ágeis na revisão literária e na pesquisa bibliográfica desenvolvida.

O restante do trabalho está organizado do seguinte modo: a Seção 2 analisa e diferencia as metodologias de desenvolvimento de sistemas de informação utilizadas pela engenharia de software, destacando seu importante papel para o processo de construção de sistemas, classificadas como tradicionais e ágeis, tais como RUP (*Rational Unified Process*), SPICE, Scrum e XP (*Extreme Programming*); a Seção 3, por sua vez, apresenta os procedimentos metodológicos adotados neste trabalho; a Seção 4 explora os resultados da pesquisa com relação aos fatores que motivaram ou desmotivaram a adoção de métodos ágeis; por fim, a Seção 5 apresenta as considerações finais.

2. METODOLOGIAS PARA DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO

Metodologia de desenvolvimento de sistemas de informação significa o “como fazer”, ou seja, o roteiro para criação de um sistema que estabelece os processos e as técnicas a serem utilizadas na construção do mesmo. Desta forma, são todas as fases do processo, seus conjuntos de atividades e resultados, para a construção de um software (SOMMERVILLE, 2007).

Rezende (2002) descreve que para desenvolver um software, um processo ou uma metodologia são fundamentais, independente das técnicas, ferramentas ou notações de software. A necessidade de adoção de uma metodologia justifica-se pelo fato de quando bem implantada permite um melhor planejamento e controle dos projetos, aumenta a qualidade dos produtos e a produtividade dos profissionais, estabelece um processo uniforme e padronizado do projeto, fixa uma documentação do projeto, dá uma maior visibilidade dos envolvidos, facilita as atividades de mudanças dos sistemas.

Na literatura as metodologias de desenvolvimento de sistemas de informação são classificadas em tradicionais e ágeis. As metodologias tradicionais ou pesadas são caracterizadas por serem orientadas ao planejamento (SOMMERVILLE, 2007). Elas especificam diversos documentos que precisam ser gerados antes, durante e após cada fase do projeto. Com isso se gasta muito tempo com planejamento e documentação, e o surgimento da necessidade de mudanças podem vir a causar um grande impacto no projeto, é necessário revisar todo o planejamento e fazer alterações em uma grande quantidade de documentos.

O primeiro processo de desenvolvimento de sistemas publicado foi o modelo clássico ou seqüencial. Ele é dividido em etapas, seguidas rigidamente uma a uma, para qual cada etapa tem uma documentação final. Essas etapas (definição e análise de requerimentos do software; projeto software; implementação e testes; implantação e integração do produto) tem por objetivo definir previamente o que deverá ser feito em cada fase, mostrando os seus respectivos resultados.

Outro processo tradicional que se destacou foi o de prototipação, que como a maioria das abordagens de desenvolvimento inicia-se com um reconhecimento de requisitos. O processo é dividido em fases, sendo estas: coleta e refinamento de dados, projeto rápido, construção do protótipo, avaliação do protótipo pelo cliente, refinamento do protótipo (fase em qual se repete o processo desde a fase de projeto rápido) e engenharia do produto (REZENDE, 2002).

Na prototipação o desenvolvedor pode criar um modelo do software que será implementado, sendo este modelo em papel retratando a interação homem-máquina, as funções exigidas do software ou um programa que executa parte ou toda a função desejada do software, devendo este programa ser depois melhorado.

O modelo espiral é outra importante metodologia tradicional, que foi desenvolvido baseado nas melhores características do modelo cascata e prototipação. Dividido em quatro atividades: i) planejamento: onde há a coleta inicial de requisitos no início e a partir de novos planejamentos baseados em comentários do cliente; ii) análise de riscos: baseadas nos requisitos iniciais e depois na reação do cliente; iii) engenharia: que é o desenvolvimento do sistema desde o protótipo inicial até o sistema final; iv) avaliação do cliente: nível de satisfação do cliente quanto ao produto (PRESSMAN, 1995).

Outras metodologias mais modernas estão obtendo grande ascensão, geralmente baseadas nos modelos vistos acima, elas surgem cada vez mais com melhorias. A metodologia RUP (REZENDE, 2002) se destaca por ser um processo dinâmico e iterativo, que utiliza a

abordagem de orientação a objetos, possuindo então uma grande flexibilidade incremental para o desenvolvimento de sistemas. A iteração permite a acomodação de novos requerimentos ou mudanças táticas nos objetivos do projeto, através de refinamentos sucessivos, endereçando os itens de alto risco do projeto e permitindo a identificação e mitigação de riscos do projeto.

No entanto, a implementação da RUP como metodologia de desenvolvimento de sistemas tem como objetivo realizar uma grande documentação de todo o processo, visando uma melhor qualidade de desenvolvimento e a satisfação de toda a equipe envolvida (equipe multidisciplinar e cliente usuário), respeitando o planejamento e orçamentos previsíveis. A seção a seguir destaca as metodologias ágeis, que são voltadas para projetos em que os requisitos são vagos e que se modificam com frequência com equipes pequenas trabalhando em paridade (até 12 desenvolvedores) e desenvolvimento de sistemas incremental (assim como o RUP, o sistema é implementado no início do projeto e ao longo do tempo ganha novas funções).

2.1 METODOLOGIAS ÁGEIS

O movimento ágil tem seu marco inicial com o surgimento do Manifesto Ágil, que foi proposto no início de 2001, nos Estados Unidos, a partir da reunião de um grupo de especialistas em desenvolvimento de software. Cansados dos processos burocráticos dos métodos tradicionais, definiram valores e princípios que permitiria às equipes de desenvolvimento produzir e responder rapidamente as mudanças. O resultado foi o Manifesto Ágil (AGILE ALLIANCE, 2001).

Os princípios e valores propostos pelo manifesto ágil são:

“Estamos evidenciando maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através desse trabalho, passamos a valorizar:

Indivíduos e interações MAIS QUE processos e ferramentas

Software em funcionamento MAIS QUE documentação abrangente

Colaboração com o cliente MAIS QUE negociação de contratos

Responder a mudanças MAIS QUE seguir um plano

Ou seja, mesmo tendo valor os itens à direita, nós valorizamos mais os itens à esquerda.” (AGILE ALLIANCE, 2001).

Seguindo as idéias propostas no manifesto ágil, as metodologias ágeis trabalham para aumentar a produtividade no desenvolvimento do software através de atividades coerentes com seus valores. O objetivo é ter software funcionando o mais cedo possível, agregando o máximo de valor ao produto para o cliente em qualquer fase do processo.

As metodologias ágeis abraçam as mudanças como parte inseparável do software, acredita-se que o software nunca está completo. O desenvolvimento é voltado para atender os requisitos iniciais do cliente, concentrando o desenvolvimento na solução mais rápida e mais simples para o funcionamento do software e satisfação do cliente (FONSECA e CAMPOS, 2008).

Uma abordagem iterativa e incremental é aplicada para o desenvolvimento do software. O sistema é dividido em pequenas partes, que são desenvolvidas em ciclos de curta duração, e das quais resultam pequenas releases durante todo o projeto. No início de cada iteração são definidas quais funcionalidades serão implementadas pelos desenvolvedores, de acordo com a prioridade acordada com o cliente.

Durante cada iteração o cliente deve participar o máximo, como parte da equipe ou o mais presente possível, estabelecendo boa relação e comunicação com os desenvolvedores do projeto para esclarecer dúvidas e avaliar o trabalho que está sendo feito o mais cedo possível. Assim, no final de cada iteração o objetivo é ter todas as funcionalidades, inclusive as que foram implementadas nas iterações anteriores, funcionando corretamente, sem que nenhuma das outras funcionalidades tenha sido prejudicada no final da iteração (BANKI e TANAKA, 2008).

Existem diversas metodologias ágeis, com processos diferentes, porém preservando os princípios entre si: o envolvimento do cliente, a entrega incremental, a valorização das pessoas, a crença de que as mudanças são inevitáveis e a simplicidade no desenvolvimento. (SOMMERVILLE, 2007).

Cabe destacar que não existe nenhum método ou tecnologia que será sempre ideal para o desenvolvimento de qualquer software (BROOKS, 1987). Cada metodologia tem suas características e pontos forte, o engenheiro de software ou o gerente de projeto deve avaliar seu projeto (equipe, tempo, custo, etc.) antes de adotar uma determinada metodologia.

3. MÉTODO

Este trabalho consiste em uma pesquisa exploratória, sob a forma de revisão bibliográfica que tem como objetivo identificar, listar e discutir os diversos fatores motivadores e desmotivadores relacionados ao processo de adoção de métodos ágeis disponíveis na literatura e em estudos de caso.

A pergunta da pesquisa consistiu em saber *quais os fatores motivadores e desmotivadores na adoção de metodologias ágeis de desenvolvimento de sistemas de informação nas organizações.*

O procedimento metodológico utilizado foi uma pesquisa qualitativa e descritiva, baseada em uma pesquisa bibliográfica de artigos que *descreveram suas experiências e lições aprendidas com esses métodos.* Nesta seção é apresentada a forma de definição da amostra estudada bem como o referencial conceitual e os procedimentos utilizados para a análise dos artigos científicos.

A amostra foi composta por 72 artigos científicos, sendo 23 nacionais e 49 internacionais, publicados entre 2004 e 2009. Os meses da pesquisa foram os de maio, junho e julho do corrente ano.

As publicações internacionais também foram pesquisadas por entender que a análise deveria ter um cunho não apenas local, uma vez que a adoção de ágeis é relevante em todo o mundo. Assim, foram considerados os periódicos indexados pela biblioteca digital da ACM (*Association for Computing Machinery*), *ACM Digital Library*¹. Foi considerada tal biblioteca, por ser representativa e por divulgar o que vem sendo produzido em relação a métodos ágeis na área de Engenharia de Software.

O período de 5 anos foi escolhido devido ao fato de que no início dos anos 2000 os métodos ágeis ganharam espaço, e surgiram estudos sobre os casos de sucesso de sua adoção. Para tornar o mapeamento exequível, foram selecionados os artigos que possuíam explicitamente em suas palavras-chave: “métodos ágeis”, “ágil”, “desenvolvimento ágil”, “gerenciamento ágil de projetos”, “aplicação de ágil” (no caso dos nacionais), e “agile”, “extreme programming”, “agile survey”, “agile study case”, “agile development” (nos internacionais).

¹ Disponível em <http://portal.acm.org/dl.cfm>

Foram especialmente observadas as relações de fatores críticos de sucesso de adoção de ágeis em todos os materiais pesquisados. Em seguida, esses fatores foram tabulados, sendo efetuada uma comparação para verificar os fatores mais recorrentes e o grau de importância dado a eles.

4. RESULTADOS: FATORES MOTIVADORES E/OU DESMOTIVADORES NA ADOÇÃO DE METODOLOGIAS ÁGEIS

Com base na análise efetuada nos artigos pesquisados, foram encontrados distintos fatores motivacionais ao uso de métodos ágeis no desenvolvimento de software, classificados em diferentes categorias de análise, como apresentado na Tabela 1.

Tabela 1. Relação e quantitativo dos Fatores Motivadores encontrados nos artigos

ID	Fatores Motivadores	Nacionais	Internacionais	Total
M01	Aumento na Qualidade do Produto	13	24	37
M02	Melhora a comunicação entre a equipe	10	23	33
M03	Desenvolvimento Iterativo e Incremental	12	17	29
M04	Propriedade Coletiva - Programação em pares - Aprendizado colaborativo	10	17	27
M05	Rápido retorno de Investimento (ROI)	12	13	25
M06	Redução de Defeitos	8	15	23
M07	Cliente presente	3	19	22
M08	Escopo aberto (aberto a mudanças)	10	10	20
M09	Redução da Complexidade do Projeto	6	13	19
M10	Facilidade de Manutenção	8	10	18
M11	Maior produtividade da equipe	6	10	16
M12	Satisfação do cliente	2	10	12
M13	Aumento na Motivação da Equipe	2	10	12
M14	Integração contínua	4	8	12
M15	Redução do tempo de desenvolvimento	3	5	8
M16	Facilidade no Desenvolvimento	0	7	7
M17	Aumento de Confiança da Equipe	1	6	7
M18	Redução de Custos	2	3	5
M19	Redução de Riscos	1	4	5
M20	Coragem da equipe para mudanças	2	3	5
M21	Pouca documentação	3	2	5
M22	Adaptabilidade	1	3	4
M23	Experiência	0	3	3
	Total:	119	235	354

Alguns fatores obtiveram expressivas ocorrências, onde se conclui que são os principais fatores que motivam à adoção de metodologias ágeis nos projetos, a exemplo do fator *aumento da qualidade do produto* [M01]. Pois, todo projeto que pretende implantar uma metodologia, tem como finalidade a busca pela melhoria na qualidade de desenvolvimento e do produto final, e este foi o principal fator motivador ao uso de métodos ágeis como indicado na Tabela 1. Portanto, o aumento na qualidade pode ser entendido como o reflexo de outro fator, denominado *melhoria da comunicação entre a equipe* [M02], alcançado devido a algumas práticas ágeis, como as reuniões diárias e a *programação em pares* (*Pair*

Programming) [M04], que permitem criar um ambiente colaborativo na equipe, através do compartilhamento do conhecimento e de outro fator motivador, a *propriedade coletiva do código* [M04], que por sua vez, facilita na padronização de codificação. Essas práticas seguem um dos princípios do Manifesto Ágil que valorizam os indivíduos e interação, mais que processos e ferramentas (AGILE ALLIANCE, 2001).

O *desenvolvimento iterativo e incremental* [M03] também foi considerado um grande fator motivador, devido a sua entrega de software funcional ao cliente em curtos períodos, contribuindo assim para um *rápido retorno de investimento (ROI)* do projeto [M05], agregando mais valor ao negócio do cliente, e conseqüentemente eleva o índice de *satisfação do cliente* [M12] junto ao projeto. Este aumento na satisfação também é reflexo da mudança da visão proposta pelos métodos ágeis, que enxergam o cliente como um membro da equipe, estreitando bastante o canal de comunicação, obtendo sua efetiva participação no projeto. Essa estreita comunicação com o cliente tem propiciado o surgimento de dois outros fatores motivadores, como a *redução da complexidade do projeto* [M09] e *facilidade no desenvolvimento* [M16], uma vez que com o cliente mais próximo permite sanar as dúvidas e as correções das funcionalidades em tempo hábil.

A partir deste ambiente criado, faz surgir outros fatores, como o *aumento na motivação* [M13] e *aumento de confiança da equipe* [M17], que tem superado melhor os desafios, ganhando mais autonomia e *coragem para abraçar as mudanças* [M20] que ocorrem durante o projeto, permitindo ao fator motivador, *contrato de escopo aberto às mudanças* [M08] ter expressiva ocorrência. Pois, diferente de outras abordagens, como os métodos tradicionais, que trabalham com contrato de escopo fechado, onde qualquer mudança é um trabalho custoso ao projeto, os métodos ágeis compreendem que as mudanças são necessárias à satisfação do cliente junto ao produto desenvolvido.

E ainda, com a adoção das práticas ágeis, foi constatada na pesquisa a presença do fator motivador *aumento na produtividade da equipe* [M11], que além de facilitar o desenvolvimento, tem colaborado para a *redução de defeitos* [M06] e pela *facilidade de manutenção dos sistemas desenvolvidos* [M10]. Essas facilidades têm sido propiciadas através das práticas citadas anteriormente, juntamente com a utilização de testes unitários e de outro fator motivador, o uso da *integração contínua* [M14] proposta pela Extreme Programming (XP), que propiciam a *redução de riscos para o projeto* [M19].

Alguns fatores causaram certa surpresa, pelo baixo índice ocorrência, como a *redução do tempo de desenvolvimento* [M15], e a *pouca documentação* [M21] proposta pelo Manifesto Ágil, e fortemente pregada como um “slogan” para as metodologias ágeis.

Outros fatores motivadores foram encontrados, porém com uma menor intensidade, como a *redução dos custos* [M18], a prática de *adaptabilidade* [M22] do projeto, e a *experiência da equipe em práticas ágeis* [M23], sendo esses os aspectos com menores motivações ao uso de métodos ágeis.

Tabela 2. Relação e quantitativo dos Fatores Desmotivadores encontrados nos artigos

ID	Fatores Desmotivadores	Nacionais	Internacionais	Total
D01	Falta de experiência do time nas práticas ágeis	2	8	10
D02	Falta de Relatos comprovando sua Eficácia	2	7	9

D03	Falta de Referencial Bibliográfico	2	5	7
D04	Aumento no tempo de desenvolvimento	0	7	7
D05	Organizações mergulhadas em metodologias tradicionais	4	3	7
D06	Choque de personalidade entre membros do time	0	4	4
D07	Falta de Gerenciamento de Riscos para o Projeto	2	1	3
D08	Aumento dos custos	0	3	3
D09	Cliente não acostumado com Escopo Aberto	2	1	3
D10	Não apropriado a times grandes	1	2	3
D11	Pouca documentação	1	2	3
D12	Falta de precisão nas técnicas para o Planejamento	0	2	2
D13	Dificuldade para criação de Planejamento do Projeto	1	1	2
D14	Não apropriados a sistemas com segurança crítica	0	2	2
D15	Aumento na distração do time	0	2	2
D16	Dificuldade na Seleção da Metodologia Ágil	0	2	2
D17	O cliente sempre presente pode causar insatisfação devido ao cansaço e esgotamento	1	1	2
D18	Falta de consenso da comunidade sobre as práticas ágeis	1	1	2
D19	Equipe geograficamente separada	1	1	2
D20	A estrutura da empresa	2	0	2
D21	Falta de Entendimento do Cliente quanto as Práticas Ágeis	1	0	1
D22	Conhecimento Tácito	0	1	1
D23	Não apropriados a projetos grandes	0	1	1
D24	Não apropriados a sistemas críticos	0	1	1
D25	A comunicação não reflete no domínio	0	1	1
	Total:	23	59	82

Conforme esperado, na Tabela 2, apesar de 19% dos fatores serem desmotivadores, os resultados foram bastante distintos. O maior fator apresentado está relacionado com a *falta de experiência do time nas práticas ágeis* [D01], alguns trabalhos apresentam que *não existem relatos que comprovem a eficiência na adoção de métodos ágeis* [D02], mesmo com ferramentas de métricas, mas obteve ganhos significativos na comunicação, satisfação do cliente e na qualidade no produto. Os *métodos tradicionais* [D05] também podem ser considerados um fator de bastante impacto quando o intuito é em aplicar a metodologia ágil em uma organização, o *conhecimento tácito* [D22] foi citado como desmotivador pelo fato de que gerentes preferem visualizar os documentos das especificações do projeto em papeis, do que utilizar as simples formas de documentação do ágil.

Foi constatado que muitas pessoas confundem *pouca documentação* [D11] com nenhuma documentação, pois a documentação existe, e é feita utilizando simples características, como por exemplo: anotações, fotos e código fonte. A simplicidade da

documentação está relacionada com as constantes mudanças realizadas ao longo do projeto. A documentação em um projeto é de extrema importância, pois para que os novos desenvolvedores consigam ter o conhecimento necessário para ingressar no projeto e que consiga compreender o domínio, torna-se fundamental (TELES, 2006). A *estrutura da empresa, física e hierárquica* [D20], pode atrapalhar em uma eficiente utilização de algumas práticas da metodologia ágil, programação em pares é uma delas, onde é necessário que os programadores fiquem lado a lado, logo irão precisar de espaço físico. Já a estrutura hierárquica pode ser um pouco mais complicada, pois a organização precisa aprender a trabalhar utilizando o modelo horizontal de gerenciamento, eliminando algumas barreiras, aumentando a velocidade no trânsito da informação, dando mais agilidade. A programação em pares é uma prática que se pode tornar difícil de ser aplicada, não só pelo fato da estrutura física, mas também devido à grande chance de conflito através do *relacionamento humano* [D06]. Porém, o ganho na comunicação e no aprendizado é um excelente fator que motivada à sinergia entre a equipe.

A comunicação é fundamental em qualquer processo de desenvolvimento de software, foi citado que com a *participação intensa do cliente* [D17] pode haver um desgaste entre ambas as partes, além dos clientes não estarem acostumados com essa nova abordagem de trabalho. Porém, quando o cliente está presente, participando ativamente no projeto, as dúvidas tendem a ser eliminadas mais rápidas, diminuindo o tempo no desenvolvimento, pois o *feedback* é constante (BECK, 1999). Outro aspecto importante são os termos utilizados nas reuniões entre clientes e desenvolvedores, pois devem ser o mais simples possível e que o domínio reflita no código do *software*. Alguns trabalhos apresentam que a *distração do time é preocupante* [D15], mas para isso deve-se ter a figura do líder sempre presente, para que ele possa eliminar impedimentos do time, e garantir as melhores condições possíveis de trabalho, para que a equipe possa produzir bem e conduzir as solicitações dos clientes. O *escopo aberto* [D09] é ideal para eliminar desperdício, pois as partes do produto são desenvolvidas sob demanda e que agregue maior valor ao cliente, porém existem alguns clientes que ainda não estão acostumados. Apesar de ter apresentado com pouca frequência, e ser um fator de bastante importância, o *gerenciamento de risco* [D07] não é abordado na literatura, pôde-se observar que, para isso, a metodologia ágil é bastante utilizada em conjunto com outras práticas e métodos. A literatura menciona que, métodos ágeis não são adequados para projetos com *equipe grande e geograficamente separada* [D10][D19], no entanto houve poucos trabalhos que apontaram que tal fator torna-se desmotivador.

A *mudança de paradigma* [D21] na adoção de métodos ágeis é razoável, alguns dos fatos desmotivadores podem estar associado à *falta de experiência no gerenciamento* [D13], ocasionando um aumento no tempo de desenvolvimento e um elevado custo.

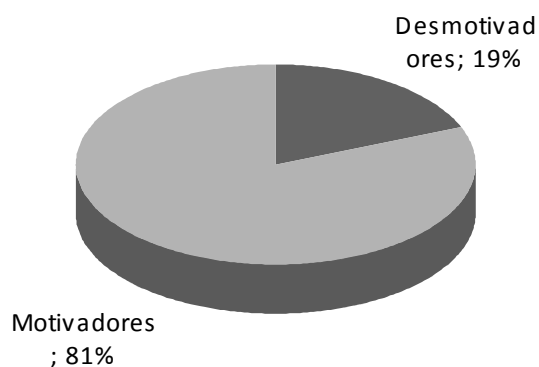


Figura 1. Fatores Motivadores / Desmotivadores

Pôde-se notar, na Figura 1, que os resultados dos fatores motivadores foram bem superiores aos desmotivadores, e que a maior parte dos fatores desmotivadores estão relacionados à *falta de experiência da equipe com a utilização de métodos ágeis* [D01]. Os resultados apontam que a comunicação durante o processo de desenvolvimento teve uma melhora significativa na qualidade do produto e na satisfação do cliente, reduzindo a complexidade do projeto.

5. CONSIDERAÇÕES FINAIS

Esta pesquisa tratou de um estudo preliminar sobre os fatores motivadores e desmotivadores na adoção de métodos ágeis de desenvolvimento de sistemas de informação. Mas ainda é possível efetuar uma busca e análise mais detalhada acerca das ações de engenharia de software das organizações. Isso pode ser complementado com questionários, solicitação de relatórios e entrevistas junto as equipes de TI.

Os resultados encontrados podem se tornar ainda mais consistentes e abrir espaço para novas questões de pesquisa, como uma análise mais aprofundada de cada um dos fatores. Enfim, ainda há muito espaço para exploração de ricas informações seguindo o viés dessa pesquisa.

6. REFERÊNCIAS

AGILE ALLIANCE. Manifesto for Agile Software Development. 2001. Disponível em <http://agilemanifesto.org/>. Acessado em agosto de 2009.

AMBLER, S.W. Results from Scott Ambler's Agile Adoption Survey February 2008. Disponível em <http://www.agilemodeling.com/surveys>. Acessado em junho de 2009.

BANKI, A.L.; TANAKA, S. Metodologias Ágeis: Uma Visão Prática. Engenharia de Software Magazine, Rio de Janeiro, ano 1, ed. 4, p. 22-29, 2008.

BECK, K. Extreme programming explained: embrace change. Boston: Addison-Wesley Professional, 1999.

BROOKS, F.P. No Silver Bullet Essence and Accidents of Software Engineering. Computer, v.20, n.4, p.10-19, 1987.

CAO, D. An Empirical Investigation of Critical Success Factors in Agile Software Development Projects. 2006. Doctoral Thesis – Capella University.

CHOW, T. and CAO, D.B. A survey study of critical success factors in agile software projects. Journal of Systems and Software, v.81, issue 6, p.961-971, 2008.

FONSECA, I.; CAMPOS, A. Por que SCRUM?. Engenharia de Software Magazine, Rio de Janeiro, ano 1, ed. 4, p.30-35, 2008.

HEEKS, R. Health information systems: failure, success and improvisation. International Journal of Medical Informatics, v.75, issue 2, p.125-137, 2006.

O'BRIEN, J.A. Sistemas de Informação e as decisões gerenciais na era da internet, 2ª edição. São Paulo: Saraiva, 2006.

PRESSMAN, R.S. Engenharia de software. Tradução José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995.

REZENDE, D.A. Engenharia de softwares e sistemas de informação, 2ª edição. Rio de Janeiro: Brasport, 2002.

_____, D.A. Planejamento de Sistemas de Informação e Informática. São Paulo: Atlas, 2003.

SOMMERVILLE, I. Engenharia de Software, 8ª edição. São Paulo: Pearson Addison-Wesley, 2007.

STAIR, R.M.; REYNOLDS, G.W. Princípios de Sistemas de Informação, 6ª edição. São Paulo: Thomson, 2006.

TELES, V.M. Extreme programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. São Paulo: Novatec Editora, 2006.