

FACULDADE DE ADMINISTRAÇÃO E NEGÓCIOS DE SERGIPE - FANESE
NÚCLEO DE PÓS-GRADUAÇÃO E EXTENSÃO – NPGE
CURSO DE PÓS-GRADUAÇÃO “LATO SENSU”
ESPECIALIZAÇÃO EM SISTEMAS DE INFORMAÇÃO PARA WEB

PROCESSO ÁGIL DE DESENVOLVIMENTO DE SOFTWARE: UMA VISÃO GERAL
DA METODOLOGIA BASEADA NA MICROSOFT SOLUTIONS FRAMEWORK FOR
AGILE SOFTWARE DEVELOPMENT.

Wilson Queiroz da Silva Neto¹

Resumo

Este artigo analisou o conceito de metodologias ágeis, esclareceu o conceito da Microsoft Solutions Framework (MSF). Demonstrou, assim, sua praticidade e utilidade em empresas desenvolvedoras de softwares.

Após a coleta das informações anteriores, foram definidas as Estruturas de Soluções da Microsoft para Desenvolvimento de Software Ágil, contextualizando através de ilustrações e citações. Foram definidos os papéis delegados aos membros da equipe, com suas atividades e a interação com que cada um interage.

Palavras-chave: Equipe, MSF, Metodologias Ágeis.

Abstract

This project analyzed the concept of agile methodologies, explained the concept of Microsoft Solutions Framework (MSF). It was presented the practicality and utility in companies that develop softwares.

The Structures of Solutions of Microsoft for Development of Agile Softwares was defined after the collect of informations. It was defined the functions of the members of the group, theirs activities and interaction of each one.

Expressions-key: Team, SF, Agile Methodologies.

¹ BACHAREL em Sistemas de Informação
Certificado em ITIL V3.0
Web Developer
wilsonneto22@gmail.com

LISTA DE QUADROS

Quadro 1: Objetivos chave de qualidade no modelo time.....	8
Quadro 2: Especificação da Atividade.....	14
Quadro 3: Desvantagens entre metodologias.....	15

LISTA DE FIGURAS

Figura 1: União dos grupos e parceiros da Microsoft viabilizado pelo MSF.....	7
Figura 2: Modelo de time.....	9
Figura 3: Estrutura do metamodelo MSF 4.0.....	10
Figura 4: Fases de planejamento – práticas do XP.....	12
Figura 5: Principais processos de trabalho do FDD.....	13

1 Introdução

Com a crescente evolução do homem e da necessidade de buscar soluções para poder resolver problemas adversos num período de tempo cada vez menor é que softwares são criados, para assessorar o homem nas suas atividades profissionais. Por isso, cada vez mais crescem investimentos e novas empresas para a confecção de programas. Softwares esses que beneficiam, por exemplo, um setor de planejamento que, para viabilizar e facilitar seus processos administrativos utiliza-se de planilhas eletrônicas.

A partir desse crescimento exacerbado na construção de softwares foram aparecendo problemas no processo de desenvolvimento desses programas, entre estes os mais comuns são os de decisão de uma equipe responsável pelas etapas de construção, permissões, cronogramas, testes e manutenções.

Por causa dessas adversidades a Microsoft, com sua vasta experiência, começou a desenvolver um “manual” que ilustra algumas possíveis barreiras e/ou problemas que poderiam ocorrer na construção de softwares, ajudando empresas desenvolvedoras de sistemas de computadores.

Dessa evolução surge o Microsoft Solutions Framework (MSF), uma documentação que possui Estruturas de Soluções da Microsoft para auxiliar essencialmente no processo de construção de soluções. Essa tendência se desmembra em duas vertentes: MSF for CMMI Process Improvement, para empresas que buscam a certificação CMMI²; e o MSF for Agile Software Development, para empresas que procuram por uma abordagem ágil para desenvolvimento de softwares. Essa última será o tema do artigo onde serão detalhados os principais processos usados pela metodologia.

Este artigo apresenta: em seu capítulo inicial o conceito sobre metodologias ágeis para melhor explicitar o tema proposto. Mostra, no capítulo seguinte, o MSF for Agile Software Development com o conceito de MSF, além da estrutura do MSF e exemplos de metodologias no mercado.

² Capability Maturity Model Integration.

2 Metodologias Ágeis

A Metodologia Ágil tem por finalidade dar uma maior flexibilidade aos processos de planejamento e construção de softwares, uma vez que essa metodologia é um conjunto de coleções de práticas e métodos que agilizam os processos: da confecção à conclusão dos programas. Estas metodologias provêm de manifestos ágeis que defendem os seguintes requisitos:

- É mais importante entregar ao cliente o software funcionando, que apresentar documentações complexas e detalhadas;
- Oferecer mais importância para os indivíduos e interações da equipe, que processos e ferramentas utilizadas;
- Proporcionar um excelente acordo com o cliente, que negociar efetivamente o contrato;
- Adaptar mudanças no projeto, ao invés de seguir o projeto inicial até o fim para depois verificar possíveis mudanças.

Por sua vez,

Quando estudadas profundamente percebemos que, de fato, as principais mudanças sugeridas pelas novas metodologias em comparação as mais “velhinhas” são três:

1. Um foco maior em testes como agente “garantidor” de qualidade³;
2. Interações mais curtas visando ter respostas rápidas e proporcionar uma homologação constante;
3. Atitude (CAMARA, 2007).

Além disso, Ken Schwaber (apud Mendonça, 2006) define que a metodologia ágil é interativa, incremental, auto-organizante e emergente, ou seja, as equipes que testam a qualidade dos projetos empresariais são as que apresentam a grande responsabilidade de garantir a qualidade dos serviços prestados, através da comunicação entre os envolvidos no projeto, desde o gestor ao usuário final. Essa

³ Agente Garantidor de Qualidade são equipes que testam a qualidade dos projetos em questão.

interação, portanto, deve ser prática e o mais eficiente possível, na medida em que assegure a aquisição de confiança do usuário em relação à empresa.

Esses fatores contribuíram para o começo da procura de processos ágeis, uma vez que empresas sentiram a necessidade de resultados mais rápidos e flexíveis, em razão dos orçamentos apertados e da constante variação e necessidades do mercado. Porquanto, “[...] o desenvolvimento de software ágil permite que algumas dessas empresas emergentes obtenham mais por menos e iniciem projetos significativos com pequenas equipes e orçamentos curtos” (FILEV, 2007).

3 MSF for Agile Software Development

A partir da evolução do MSF, a Microsoft desenvolveu o MSF for Agile Software Development. Neste capítulo explicaremos melhor sobre esta vertente do MSF 4.0, discriminando conceitos, estrutura e a implantação numa empresa de software.

3.1 Conceito do Microsoft Solutions Framework (MSF)

Com o crescimento exacerbado da construção de programas, as empresas começaram a encarar problemas no processo de desenvolvimento dos softwares, como a decisão de uma equipe responsável pelas etapas de construção, permissões apropriadas para decisões autônomas, cronogramas de atividades e procedimentos futuros, testes nas linhas de códigos para a procura de possíveis erros, manutenções, comunicação entre membros das equipes envolvidas e os clientes, entre outros. Por causa dessas adversidades, a Microsoft começou a desenvolver um “manual” que ilustra algumas possíveis barreiras e/ou problemas que poderiam ocorrer na construção desses softwares, ajudando então empresas desenvolvedoras de sistemas de computadores a tornar seus sistemas computacionais mais organizados e, conseqüentemente, mais eficientes e lucrativos em inúmeros aspectos.

Dessa evolução, surge então o MSF que é uma documentação que possui estruturas de soluções da Microsoft para auxiliar essencialmente no processo de construção de soluções. Essa documentação tem como propriedade fundamental unir todos os setores da Microsoft numa relação prática entre eles.

Segundo a Microsoft (2004), o MSF alia recomendações de design e desenvolvimento de software à criação e implantação da infra-estrutura em um único ciclo de vida de projeto para orientar soluções de tecnologia de todos os tipos. O MSF

ajuda as organizações a alcançar um equilíbrio perfeito entre flexibilidade e a necessidade de cumprir objetivos e minimizar riscos.



Figura 1: União dos grupos e parceiros da Microsoft viabilizado pelo MSF
Fonte: Microsoft, 2004 (adaptada).

O MSF segue uma estrutura dividida em princípios, modelos, metas, objetivos e disciplinas. Quanto aos princípios, a documentação do MSF se baseia nos procedimentos básicos de Metodologias Ágeis, com foco no negócio em questão, comunicação eficiente, visão de projeto compartilhado, aumento do poder dos membros do time, esclarecimento das responsabilidades compartilhadas, agilidade, investimento em qualidade e aprendizagem com experiências progressas.

Inicialmente, o MSF busca entender as necessidades do cliente solicitante para realizar um projeto satisfatório para o mesmo. Além disso, o MSF tem a propriedade de recomendar a conversação entre os membros participantes da equipe responsável pelo projeto e com o cliente, valorizando a comunicação entre as partes interessadas e assim tornando o procedimento funcionalmente iterativo.

A visão de projeto compartilhado é uma filosofia que permite a realização de reuniões por parte da equipe para a discussão, identificação e resolução de itens conflitantes e visões enganosas em relação a decisões tomadas no início do projeto.

O MSF também proporciona um melhor esclarecimento das responsabilidades compartilhadas para que todo o time participe com suas idéias, decisões e suas conseqüências, aprendendo com isso e melhorando o seu relacionamento com os respectivos stakeholders⁴, já que estes possuem visões diferentes de gestores, analistas e programadores, mas ao mesmo tempo são importantes para o processo final.

⁴ Agente externo que possui grande influência no projeto.

Dessa forma, ao dividir as responsabilidades com os membros principais das equipes, estes aumentam seus poderes de decisão, o que significará uma maior rapidez no alcance de metas e entregas dos projetos. Isso tudo porque esses times permanecerão baseados em pares MSF, tomando decisões e direções quando necessário.

Por sua vez, La-Salette (2007) ilustra os objetivos chave de qualidade no modelo de time.

Objetivo chave de qualidade	Papéis do Modelo de Time
Entregar a solução dentro das restrições do projeto	Gerência de Programa
Construir a solução obedecendo às especificações	Desenvolvimento
Garantir que a solução só será entregue após a identificação de seus defeitos	Teste
Implantar a solução sem maiores impactos	Gerência de Liberação
Maximizar a performance do usuário	Experiência do Usuário
Satisfazer os clientes	Gerencia de Produto

Quadro 1: Objetivos chave de qualidade no modelo time
Fonte: La-Salette, 2007.

Faz-se mister citar, ainda, que “para os propósitos de manutenção de expansão depois da entrega do software, a melhor resposta é ter código auto-documentante e gerar quaisquer documentações suplementares automaticamente” (MENDONÇA, 2006, p.229). Pois este se configurará como um recurso facilitador para o usuário final, incorporando valor ao produto.

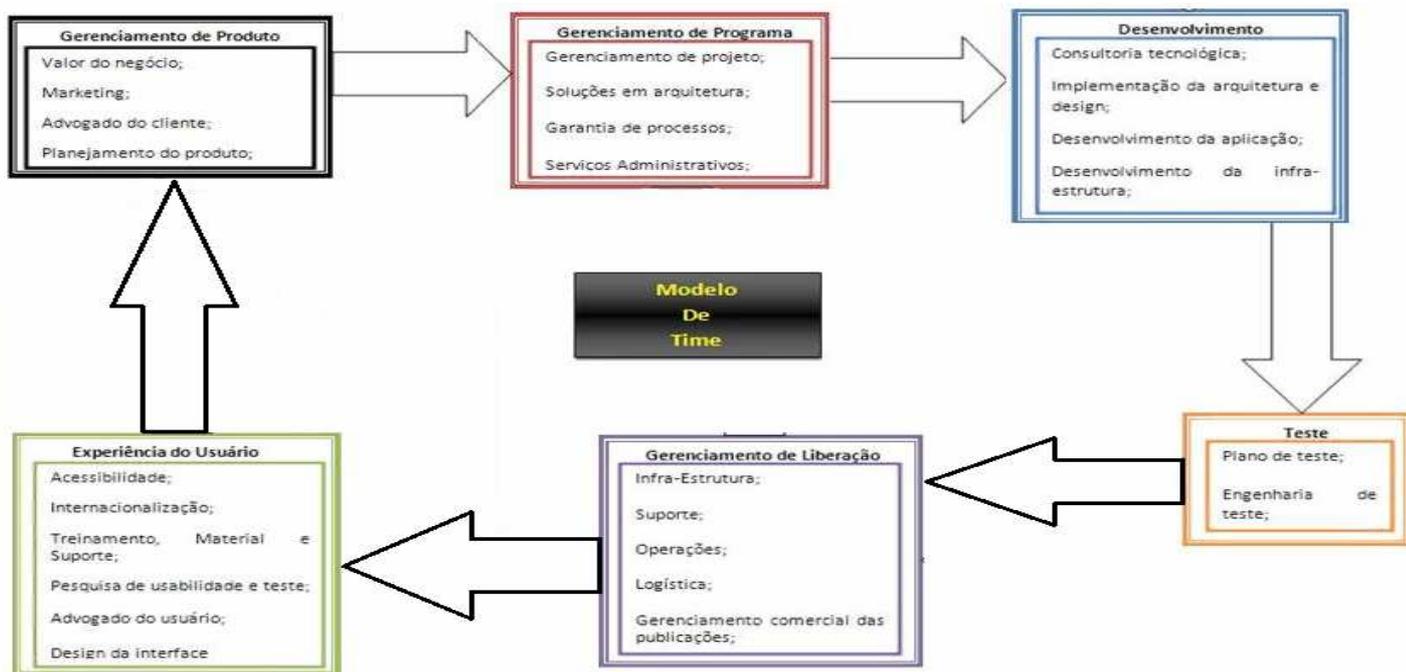


Figura 2: Modelo de time
 Fonte: Microsoft, 2004 (adaptada).

3.2 ESTRUTURA DO MSF FOR AGILE SOFTWARE DEVELOPMENT

A versão do MSF 4.0 possui um process template, o MSF for Agile Software Development que é uma implementação prescritiva do metamodelo do MSF 4.0. Esse metamodelo utiliza-se de técnicas adaptáveis e flexíveis para o movimento emergente dos desenvolvimentos de software ágeis.

O MSF for Agile Software Development permite maior agilidade no processo de desenvolvimento dos programas, revisão de melhorias menos formais, dispõe da habilidade individual dos desenvolvedores, fundamenta-se no manifesto ágil e dá ênfase ao foco no projeto. Além disso, apresenta maior importância aos clientes, o quê, segundo Hirota (2006) é um indicativo de que as pessoas são mais importantes que o processo.

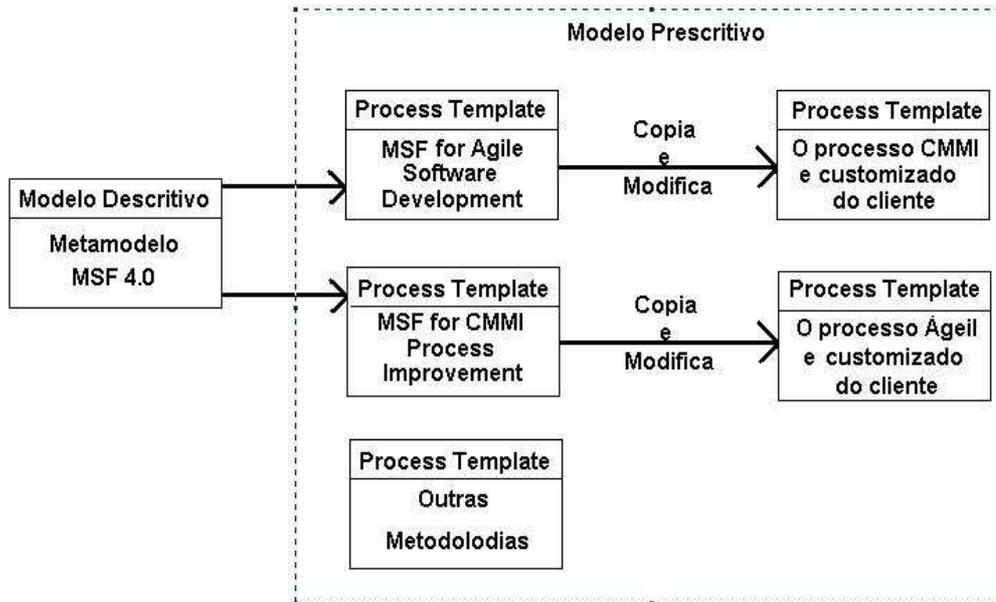


Figura 3: Estrutura do metamodelo MSF 4.0
 Fonte: Hundhausen, 2006, p.182 (adaptada).

A sua estrutura está dividida em seis etapas, que são: (i) papéis (roles), (ii) linhas de trabalhos (work streams), (iii) as disciplinas (disciplines), (iv) qualidades do serviço (qualities of service), (v) governança e trilhas (governance and tracks) e (vi) tipos de itens do trabalho (work item types).

3.2.1 Roles

Os papéis determinam uma ou mais responsabilidades aos membros da equipe do projeto, dividindo-os em analistas de negócios, gerentes de projetos, arquitetos, programadores, verificadores de erros e gerentes de liberações.

3.2.2 Work Streams

No Work Streams as linhas de trabalhos são caracterizadas por critérios de entradas, saídas e atividades. O MSF for Agile Software Development define as linhas de trabalhos através dos seguintes papéis: Business Analyst ou Analista de Negócios, Project Manager ou Gerente do Projeto, Architect ou Arquiteto, Developer ou Programador, Tester ou Controlador de Testes e Release Manager ou Gerente de Liberação.

3.2.3 Disciplines

As disciplinas são as áreas de perícia não técnica que planeja o projeto inteiro e pertencem a todos os membros da equipe, como no caso da gerência de projetos e da gerência de risco, ambas incluídas nas disciplinas do MSF 4.0.

3.2.4 Qualities of Service (QoS)

O MSF 4.0 organiza a qualidade do software em formulários, dividindo em segurança, desempenho e experiência do usuário. Todos na equipe devem estar conscientes das exigências de QoS para o projeto e usar estas exigências para o desenvolver, executar, testar e distribuir.

3.2.5 Governance and Tracks

A governança é responsável pelo controle do tempo e do investimento financeiro num projeto MSF, definindo seis pontos de verificação, cada um deles focalizando uma pergunta específica para ser respondida. Dá-se o nome de trilhas aos meios através dos quais passam as atividades e trabalhos e vão aos pontos de verificação. Essas trilhas têm interações diferentes das que funcionam dentro do ciclo do modelo de processo porque elas agrupam as atividades que conduzem aos pontos de verificação da governança. Cada uma dessas trilhas possui a capacidade de medir interações múltiplas, sobrepondo-se e interagindo de forma contínua no feedback de atividades de trilhas diferentes.

3.2.6 Work Item Types

Work Item Types é um registro da base de dados que o Visual Studio Team Foundation usa para designar atribuições e monitorar o estado do trabalho. Cada work item é analisado separadamente e inclui o cenários, qualidade de serviço (QoS), tarefa, bug e risco.

4 Metodologias Ágeis Existentes no mercado

O **Extreme Programming (XP)** é uma metodologia ágil para equipes pequenas e médias que desenvolvem software baseado em requisitos vagos e em constante mudança (BECK, 1999).

Essa metodologia utiliza como princípio a simplicidade, a comunicação, o feedback constante, o incentivo a coragem e testes automáticos.

Segundo Hilmer (2004), os planejamentos em release possuem as seguintes fases:

O CLEINTE DECIDE	OS PROGRAMADORES DECIDEM
<ul style="list-style-type: none"> • Escopo • Prioridade • Composição do Release • Data do Release 	<ul style="list-style-type: none"> • Estimativas • Conseqüências • Processo • Planejamento intra-release (o mais arriscado primeiro)

Figura 4: Fases de planejamento – práticas do XP
Fonte: Hilmer, 2004 (adaptação).

O **Feature Driven Development (FDD)** também é um exemplo de uma metodologia ágil, criado por Jeff de Luca e Peter Code e reconhecido em 1997. O FDD além de possuir um mecanismo mais preciso para acompanhamento do projeto, possui também maior número de passos e requisitos mais formais.

O FDD segue dois processos principais de desenvolvimento: (i) descobrir uma lista de características (features) para serem implementadas e depois realizar essa implementação; (ii) a exemplo do XP, o FDD precisa de uma participação ativa do cliente junto ao desenvolvedor, para que com isso os dois consigam traçar um plano com os requisitos necessários à satisfação do cliente.

Para que ocorra a implementação em pequenas interações as listas de características devem possuir unidades de trabalho para os programadores e que toda feature seja suficientemente pequena. Essa implementação ocorre, em primeiro plano, através da união de um conjunto de características catalogadas em um pacote de trabalho.

Essas interações apresentam: (i) uma primeira reunião para um pacote de trabalho, visando esclarecer detalhes incluídos nas características; (ii) projeto, onde se define as classes, métodos e documentações necessárias; (iii) o desenvolvimento,

para implementar as funcionalidades e os testes de unidades; (iv) revisão do projeto, revendo o projeto concluído para poder liberá-lo ou rejeitá-lo; (v) revisão dos códigos, quando procura-se encontrar pequenas falhas; (vi) fechamento de liberação (release), onde as características implementadas são liberadas em um build (configuração).

Hilmer (2004) ainda ilustra os 5 principais processos da seguinte forma:

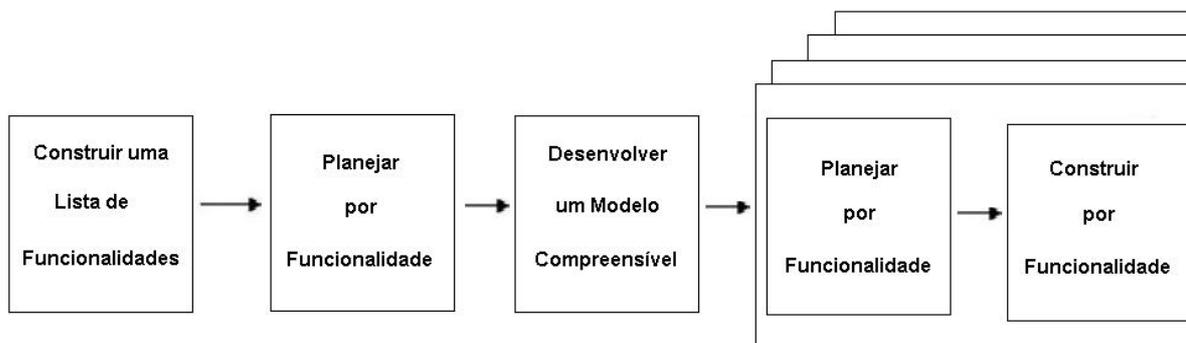


Figura 5: Principais processos de trabalho do FDD

Fonte: Hilmer, 2004 (adaptação).

O SCRUM é outro exemplo de metodologia ágil, que foi desenvolvido para criar software em empresas que possuem requisitos confusos e instáveis. Ele segue os princípios semelhantes aos do XP, com equipes pequenas, iterações curtas para uma melhor visibilidade no desenvolvimento e com funcionalidades desconhecidas ou instáveis.

Essa metodologia fornece um procedimento apropriado para um projeto ou desenvolvimento orientado a objetos, sendo dividido em sprints⁵ de 30 dias, com times de sete pessoas, no máximo, formados por: projetistas, programadores, engenheiros e gerentes de qualidades. Cada time criado fica sendo responsável pelas respectivas funcionalidades designadas no início de cada sprint.

Ao final do dia uma reunião é realizada com o time e a gerencia para mostrar as funcionalidades que foram concluídas e as que ficaram para o dia seguinte. Com isso a gerencia analisará os requisitos, aprovando-os ou não.

⁵

Pequenas metas definidas para ser concluídas imediatamente.

	Especificação da Atividade
XP	Clientes escrevem as user stories.
	Equipe técnica e clientes definem as user stories que serão desenvolvidas nas iterações. As iterações duram de 1 a 4 semanas.
	Implementação das user stories que fazem parte da iteração corrente por duplas de programadores.
	Os programadores executam os testes de unidade e os clientes executam os testes de aceitação.
	A integração acontece paralelamente ao desenvolvimento das user stories.
	O sistema é disponibilizado ao cliente para que o mesmo realize validações.
	Cliente satisfeito com o sistema.
FDD	Geração de artefatos para a documentação dos requisitos.
	As características são agrupadas, priorizadas e distribuídas aos responsáveis pelo seu desenvolvimento. As iterações duram no máximo 2 semanas.
	Análise da documentação existente, geração de Diag. de Seqüência da UML, refinamento do modelo gerado nas atividades anteriores e implementação das características que serão desenvolvidas durante a iteração corrente.
	Os testes e inspeções são executados pelos próprios programadores após a implementação.
	Atividade realizada após os testes no incremento.
	Esta atividade ocorre através das inspeções e dos testes de integração.
	O sistema é entregue após todos os conjuntos de características implementados.
SCRUM	O produto evolui em uma série de “Sprints” mensais
	Equipes se auto-organizam
	Os requerimentos são listados em um “Product Backlog”
	Não há prática de engenharia prescrita (o Scrum adequa-se a todas)
	Usa regras generativas na criação de um ambiente ágil para a entrega de projetos

Quadro 2: Especificação da Atividade

Fonte: Basto, Deters e Silva, 2008.

Segundo Fransson e Klercker (2005 apud LUIZ e AKIO, 2008, p. 25): “Metodologias Ágeis. Uma Visão Prática”, um estudo mais dirigido foi feito na Suécia, com o objetivo de mapear as pequenas e médias empresas de desenvolvimento de software do país e levantar o grau de satisfação de cada uma delas, com relação à metodologia de desenvolvimento que estão adotando. Partindo de um universo de

1016 empresas, os autores estipularam uma amostra relevante de 100 empresas e, destas, chegaram a 57 relevantes. Na figura 4, é indicado o número de empresas que adotam cada metodologia, descartados os valores menos representativos.

Desvantagens entre metodologias	
XP	<ul style="list-style-type: none"> • A documentação é quase totalmente inexistente. • Não há especificação do software. • Não há uma fase de design ou testes separados. • Há uma proibição explícita contra mudanças. • Não há revisões formais ou inspeções.
SCRUM	<ul style="list-style-type: none"> • Muitas vezes é prioritário agradar ao cliente em diversos aspectos, como na rapidez de entrega e cumprimento de prazos e no acompanhamento deste na evolução do projeto (o que implica que haja mudanças repentinas e inesperadas na sua evolução). • As equipas de desenvolvimento têm ainda que operar adaptativamente num ambiente complexo, usando processos imprecisos.
FDD	<ul style="list-style-type: none"> • O caso inicial de aplicação de FDD foi no desenvolvimento de uma aplicação bancária. • Há controvérsia sobre o tamanho mínimo de um time FDD. • Pouco poder de escalabilidade do time. • Ainda que o FDD pressuponha um ciclo completo de documentação, a especificação original da metodologia não comenta sobre sua aplicabilidade à manutenção de sistemas.

Quadro 3: Desvantagens entre metodologias

Fonte: Ferreira, Costa, Alonso, Alves e Nunes, 2005 (adaptada).

No mercado atual podemos utilizar qualquer uma das metodologias, pois a essência básica de ambas é a mesma. Primeiramente deveremos analisar a empresa contratante e a contratada para poder escolher qual metodologia se emprega para o cenário em que se encontra. Numa empresa desenvolvedora de software de pequeno porte, o ideal seria aplicar um dos três exemplos citados acima, pois com certeza um mesmo membro da equipe possui mais de um papel, podendo com isso agilizar no processo.

A MSF for Agile Software Development por sua vez é melhor utilizada para empresas de médio e grande porte, onde melhor é definido os papéis e responsabilidades de cada membro da equipe, que por sua vez terá autonomia para realizá-la, seguindo o passo a passo da estrutura do MSF.

5 Considerações Finais

As transformações pelas quais passam a sociedade impõem uma nova tendência nas empresas desenvolvedoras de software para a sua confecção. Os clientes estão ficando cada vez mais exigentes e participativos na hora de definir as atividades e exigir a flexibilidade de acrescentar alguma modificação futura.

As empresas começaram a implantar metodologias ágeis em sua filosofia para poder atender e mostrar o seu diferencial para os clientes, ganhando com isso uma maior confiança e adquirindo vantagens em relação a empresas concorrentes.

As metodologias ágeis contribuem para uma maior integração das equipes envolvidas no projeto, de modo que todos participam e não ficam presos com burocracias. Empresas de grande porte já começaram a quebrar o paradigma de que a metodologia não poderia ser implantada, pelo fato de possuírem filiais em outros países, pensando nas dificuldades que encontrariam com as diferentes culturas, economia e mão de obra qualificada.

A Microsoft criou uma documentação para servir de guia para as empresas desenvolvedoras de softwares, a MSF for Agile Software Development, a qual define os papéis de cada membro da equipe; como gerente de produtos, gerente de projetos, analista de negócios, entre outros, e suas respectivas atividades.

Neste projeto, foi abordada toda a estrutura do MSF, com suas evoluções, versões e foi mostrado como melhorar na hora de confeccionar os softwares, ajudando de maneira significativa os resultados finais e satisfação do cliente.

Com isso, é possível mostra agilidade com que empresas possam o mercado, visando atender aos clientes de forma mais ágil e com um melhor feedback por parte deste usuário, o qual começou a participar ativamente de determinados processos durante a confecção do software.

6. Referências Bibliográficas

BRESSANO, Alécio. Gerenciando projetos no VSTS com práticas do PMI. In: CÂMARA, Fábio (org.). **Visual studio team system rocks**. Florianópolis: Visual Books, 2006.

CÂMARA, Fábio. **O que há de novo no MSF**. Disponível em: <http://www.linhadecodigo.com.br/Artigo.aspx?id=1010>. Acesso em: 18/12/2010.

_____. **Os princípios de liderança do MSF**. Disponível em: <http://www.linhadecodigo.com.br/Artigo.aspx?id=1105>. Acesso em: 18/12/2010.

_____. **Processos Ágeis e MSF**. Disponível em: <http://www.linhadecodigo.com.br/Artigo.aspx?id=833>. Acesso em: 18/12/2010.

_____. Processos de software no TFS. Disponível em: <http://www.microsoft.com/brasil/msdn/Tecnologias/vsnet/ProcessosTFS.mspix>. Acesso em: 18/12/2010.

FILEV, Andrew. **Adotando e se beneficiando de processos ágeis no desenvolvimento de software no exterior**. Disponível em: http://www.microsoft.com/brasil/msdn/Tecnologias/arquitetura/Proc_Ageis.mspix#E6. Acesso em: 26/02/2011.

HUNDHAUSEN, Richard. **Visual Studio 2005 Team System**. Canadá: Microsoft, 2006.

IBM. **Comparando RUP X MSF**. Disponível em: http://www-306.ibm.com/e-business/br/campaign/rational/columns_articles_desenv_process_compMSF.shtml. Acessado em: 26/02/2011.

LA-SALETTE, Mauro. **Avaliando a Metodologia Pro.NET em relação ao MSF 4.0**. Disponível em: <http://www.cin.ufpe.br/~tg/2005-1/mscla.pdf>. Acessado em: 27/02/2011.

MENDONÇA, Clementino. Msf v4 e VSTS 2005. In: CÂMARA, Fábio (org.). **Visual studio team system rocks**. Florianópolis: Visual Books, 2006.

MICROSOFT, Brasil. **Guia de Distribuição**. Disponível em:

<https://www.microsoft.com/brasil/security/guidance/topics/wlans/ap03.aspx>. Acesso em 13/08/2007.

_____. **MOF: Uma Abordagem Executável e Prescritiva para o ITIL**. Disponível em:

<http://www.microsoft.com/brasil/technet/itsolutions/cits/mo/mof/mofitil.aspx>. Acesso em 13/08/2007.

MICROSOFT. **Team Foundation Process Guidance**. Disponível em:

[http://msdn2.microsoft.com/en-us/library/ms242865\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms242865(VS.80).aspx). Acesso em 01/03/2011.

BASTO, Priscila. In: INÊS, Janice, In: SILVA, Sandro. **Comparação Entre Os Processos Dos Métodos Ágeis: Xp, Scrum, Fdd E Asd Em Relação Ao Desenvolvimento Iterativo Incremental** . Disponível em:

<http://revista.ctai.senai.br/index.php/edicao01/article/viewDownloadInterstitial/21/18>.

Acesso em 01/05/2011.

FERREIRA, Décio. In: COSTA, Felipe, In: ALONSO, Felipe, In: ALVES, Pedro, In: NUNES, Tiago. **Scrum, Um Modelo Ágil para Gestão de Projectos de Software**.

Disponível em: http://paginas.fe.up.pt/~aaquiar/es/artigos%20finais/es_final_19.pdf.

Acesso em 02/05/2011

PASTORELLO, Thiago. **Microsoft Solutions Framework (MSF)**. Disponível em:

<http://www.linhadecodigo.com.br/Artigo.aspx?id=771> Acesso em 13/08/2007.