

**UNIVERSIDADE FUMEC  
FACULDADE DE CIÊNCIAS EMPRESARIAIS - FACE**

**BÁRBARA LOBATO QUEIROZ DRUMMOND DINIZ**

**GESTÃO DE PROJETOS EM METODOLOGIA ÁGEIS:  
Uma abordagem no *Scrum***

**BELO HORIZONTE**

**2010**

**BÁRBARA LOBATO QUEIROZ DRUMMOND DINIZ**

**GESTÃO DE PROJETOS EM METODOLOGIAS ÁGEIS:**

**Uma abordagem no *Scrum***

Monografia realizada na Universidade FUMEC, no curso de Ciência da Computação, apresentado à disciplina Trabalho de Conclusão de Curso.

Orientadores:

Professor Osvaldo Manoel Corrêa

Professor Ricardo Terra Nunes Bueno Villela

**BELO HORIZONTE**

**2010**

## **RESUMO**

A crescente demanda por sistemas e alta velocidade com que seus requisitos evoluem têm evidenciado a necessidade de um desenvolvimento de software com flexibilidade. Cada vez mais, as empresas procuram desenvolver projetos com sucesso e, portanto, diminuir o grande índice de fracassos que existe atualmente em projetos tradicionais. Os métodos ágeis surgiram para representar um conjunto de valores, princípios e práticas que auxiliam a equipe a fazer entregas rápidas, focando no valor de negócio. Este trabalho tem o intuito de apresentar o resultado do uso de metodologias ágeis e gerenciamento de tradicional de projetos em um mesmo projeto. A implementação de uma metodologia ágil em um ambiente tradicional é bastante complicada e requer muito comprometimento de todos os envolvidos para que seja efetuada com sucesso. De forma sucinta, foi possível comprovar que o gerenciamento de projetos pode ser usado em projetos baseados em agilidade, contanto que sejam feitas adaptações em ambas metodologias.

**PALAVRAS-CHAVE:** Gerência de projetos, metodologias ágeis, manifesto ágil, PMBOK, Scrum.

## ***ABSTRACT***

*The increasing demand for high-speed systems and the speed with their requirements evolve have highlighted the need for software development with flexibility. Increasingly, companies wish to develop successful projects to reduce the high rate of failures that currently exist in traditional designs. Agile methods have been created to point a set of values, principles and practices that help the team to make quick deliveries, focusing on business value. This work aims to present the results of using agile methods and traditional project management into a same project. The implementation of the agile methodology in a traditional environment is quite complicated and requires much commitment from all stakeholders to be successfully made. Succinctly, we could demonstrate that project management can be used in agile projects, since adjustments are made in both methodologies.*

*Keywords: Project Management, agile methods, agile manifesto, PMBOK, Scrum.*

## LISTA DE FIGURAS

Figura 1 - Causas de falhas de projetos.....	10
Figura 2 - Ciclo de vida genérico.....	13
Figura 3 - Nível de interação dos processos ao longo do tempo.....	14
Figura 4 - Exemplo de um <i>Sprint Burndown Chart</i> .....	28
Figura 5 - Exemplo de um <i>Release Burndown</i> .....	29
Figura 6 - Exemplo de um <i>Scrum Board</i> .....	29
Figura 7 - Ciclo do <i>Scrum</i> .....	31
Figura 8 - Fluxo geral de um projeto agile.....	37
Figura 9 - Fases do <i>Agile Project Management</i> .....	37

## **LISTA DE SIGLAS**

BPM – Business Process Management

DSDM – Dynamic Systems Development Method

EAP – Estrutura Analítica do Projeto

FDD – Feature Driven Development

MSF – Microsoft Solutions Framework

PMBOK – Process Management Body of Knowledge

PMI – Project Management Institute

XP – Extreme Programming

# SUMÁRIO

<b>INTRODUÇÃO .....</b>	<b>7</b>
<b>1. GESTÃO DE PROJETOS .....</b>	<b>9</b>
1.1 HISTÓRICO.....	9
1.2 CICLO DE VIDA DE UM PROJETO.....	11
1.3 O PAPEL DO GERENTE DE PROJETOS .....	13
1.4 PMBOK POR ÁREA DE CONHECIMENTO .....	13
1.4.1 Gerenciamento da Integração .....	15
1.4.2 Gerenciamento de Escopo.....	15
1.4.3 Gerenciamento de Tempo.....	16
1.4.4 Gerenciamento de Custos .....	16
1.4.5 Gerenciamento de Qualidade .....	17
1.4.6 Gerenciamento de Recursos Humanos .....	17
1.4.7 Gerenciamento de Comunicações.....	18
1.4.8 Gerenciamento de Riscos .....	18
1.4.9 Gerenciamento de Aquisições .....	19
<b>2. METODOLOGIAS ÁGEIS .....</b>	<b>21</b>
2.1 APRESENTAÇÃO GERAL.....	22
2.2 SCRUM .....	25
2.2.1 Atribuição de funções.....	25
2.2.2 Artefatos .....	26
2.2.3 Fluxo do Scrum.....	30
2.3 OUTRAS METODOLOGIAS ÁGEIS .....	31
<b>3. GESTÃO DE PROJETOS EM SCRUM .....</b>	<b>33</b>
3.1 GERÊNCIA ÁGIL X TRADICIONAL.....	33
3.2 CICLO DE VIDA DE UM GERENCIAMENTO DE PROJETO ÁGIL.....	35
3.3 ESTUDO DE CASO .....	37
<b>CONCLUSÃO.....</b>	<b>40</b>
<b>REFERÊNCIAS .....</b>	<b>42</b>

## INTRODUÇÃO

O desenvolvimento de software é um tema preocupante em muitas organizações em que muitos projetos fracassam devido à grande complexidade e desorganização em seu desenvolvimento. Segundo *CHAOS*<sup>1</sup> (2004), apud Alexsandro (2010):

Os resultados mostram que 31,1% dos projetos serão cancelados antes de serem finalizados, e que 52,7% custarão 189% do orçamento original, sem contar a perda de oportunidade que é inestimável. Somente 16,2% dos projetos são entregues no prazo e dentro do orçamento. Além de ser um valor muito baixo, nas grandes corporações a situação é ainda pior, pois apenas 9% de seus projetos são entregues no prazo e dentro do orçamento, mas com apenas 42% das *features* e funções originalmente propostas. Este número é mais animador nas pequenas empresas, onde 78,4% dos projetos são entregues, com 74,2% das *features* e funções originalmente propostas.

A gerência de projetos vem de encontro a essa preocupação, pois aplica conhecimentos, ferramentas e técnicas que podem fazer com que projetos complexos e desafiadores tenham uma chance maior de sucesso, ou seja, estar dentro do prazo, custo e expectativa do cliente.

Junto com a preocupação das empresas em diminuir o número de falhas, veio também o receio da queda de qualidade devido às alterações constantes no código desses sistemas de software. Para sanar essas preocupações surgiram as metodologias ágeis como uma solução para as modificações contínuas em seu código sem impacto à qualidade.

Desta forma, este trabalho visa apresentar e analisar a integração entre essas duas áreas da informática – gestão de projetos e metodologias ágeis – e, assim, verificar se é possível um bom relacionamento entre as práticas do *PMBOK*<sup>2</sup> e de agilidade.

No capítulo introdutório é feita a conceituação de gestão de projetos, abordando seu histórico, seu ciclo de vida e seus processos essenciais. No segundo capítulo, as metodologias ágeis são apresentadas através de conceito, histórico e características. O *Scrum* tem um foco especial, pois o capítulo posterior faz a integração dessa metodologia com a gestão de projetos. Portanto, ele será

---

<sup>1</sup> *CHAOS* relatório sobre o desenvolvimento de software feito pelo Standish Group, que é um grupo realiza pesquisas e sondagens no mercado de tecnologia da informação para utilizadores e fornecedores desta. (Nota da autora)

<sup>2</sup> *PMBOK* (*Project Management Body of Knowledge*) apresenta a visão geral de conhecimentos em Gestão de Projetos. (Nota da autora)

analisado por completo, partindo do seu conceito, traçando um histórico, demonstrando suas propriedades e as principais atribuições, artefatos, fluxo. Após a apresentação dessa metodologia, outras são apresentadas superficialmente.

O terceiro capítulo busca unir o conceito de gestão de projetos à metodologia *Scrum*, mostrando como acontece a integração da gestão de projetos - baseada no *PMBOK* - e as metodologias ágeis. Por fim, será feito um estudo de caso visando apresentar a aplicabilidade real dessa integração.

## 1. GESTÃO DE PROJETOS

Para o *Project Management Guide of Knowledge (PMBOK)*, projeto é “um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo”. Temporário porque todo projeto tem início e fim bem determinados e exclusivo, pois o resultado é algo diferente sempre, mesmo que seja pequena a diferença. (*PMBOK*, 2004)

O problema dos projetos que atinge uma grande parte das organizações e que é relatado pelo *CHAOS Report (CHAOS, 2001)*, é o alto índice de fracassos em projetos, principalmente os de tecnologia da informação, devido a sua grande complexidade, constante desenvolvimento de novas tecnologias e rotineiras mudanças por parte do cliente. Com o intuito de diminuir essa preocupação, surgiu o gerenciamento de projetos, que tem conseguido conscientizar cada vez mais as empresas de sua importância tanto profissional quanto organizacional.

O gerenciamento de projetos existe desde muito tempo, na época da Guerra Fria, no período militar, já era possível identificar como eram organizados alguns projetos. Após essa época, surgiram outras formas de gerenciamento e, hoje em dia, a apresentada pelo *Project Management Institute (PMI)* tem tido ampla aceitação internacional (*VIEIRA, 2003*).

Esse capítulo apresenta uma base para a gerência de projetos bem sucedida. Na seção 1.1, é observado o surgimento do gerenciamento de projetos, um breve histórico de como essa prática se iniciou, apresentando a entidade de gestão de projetos mais conhecida internacionalmente, responsável por credenciar os atuais gerentes de projetos e também uma breve abordagem sobre a sua evolução. Na seção 1.2, é apresentado o ciclo de vida de um projeto e de suas fases. Na seção 1.3, o papel do gerente de projetos e, por fim, na seção 1.4, é apresentado como são baseados os nove grupos de processos do *PMBOK*.

### 1.1 Histórico

Como foi dito anteriormente, o *CHAOS Report* relatou o elevado índice de projetos que não conseguem cumprir com seu objetivo e, por isso, fracassam.

Ainda relata-se que 74% dos projetos falham e, dessa porcentagem, considera-se fora do prazo, sem atender às expectativas, acima do orçamento (budget) e 28% são abortados. (CHAOS, 2001) A FIG. 1 ilustra um gráfico de pizza com os valores supra-citados:

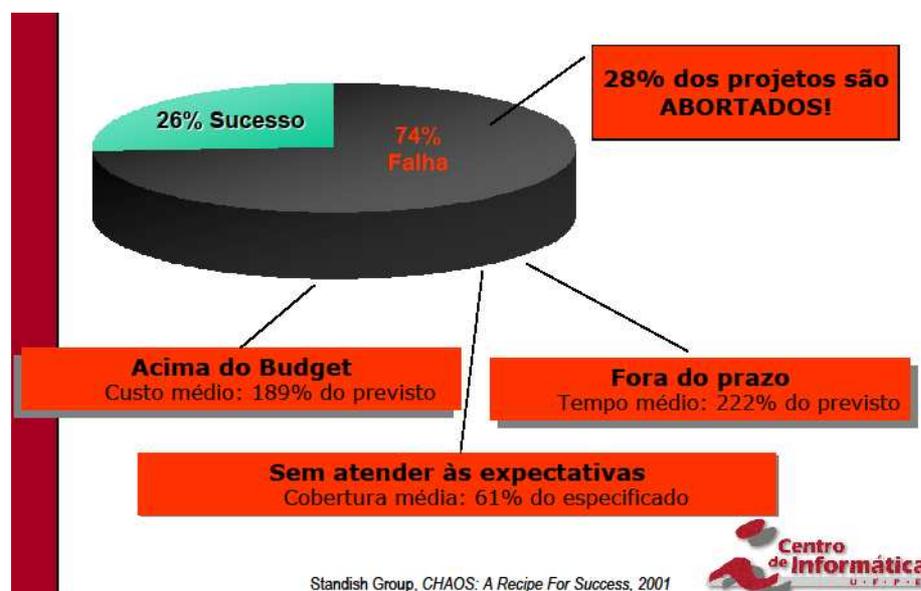


Figura 1 - Causas de falhas de projetos

Fonte: CHAOS(2001) apud Hermano Perreli(2004)

Além do estudo feito pelo Standish Group, uma outra pesquisa relatou que, nove dos principais motivos associados aos desastres de software são falhas de gerenciamento de projetos. (JONES, 1996)

Como dito anteriormente, o gerenciamento de projetos já pode ser observado há muito tempo, porém há pouco vem conseguindo obter destaque nas empresas. Em 1870, o primeiro movimento observado para prática de conceitos de gerenciamento foi feito pela Central Pacific Railroad 3 que construiu uma estrada de ferro transcontinental. Porém, antes disso, Frederick Taylor iniciou seus estudos sobre as sequências de trabalho e seu sócio Henry Gantt estudou detalhadamente a ordem de operações. Tais estudos foram aplicados também na construção de navios durante a Segunda Guerra Mundial. (SAGRES, 2007)

Em 1969, cinco profissionais da área de gerenciamento de projetos se reuniram na Pensilvânia para discutir as melhores práticas e, com isso, Jim Snyder fundou o Project Management Institute (PMI). Atualmente, esse instituto conta com mais de trezentos mil associados, em mais de cento e sessenta países e está

presente nos principais setores como a tecnologia da informação, serviços, financeiro, seguro, saúde, engenharia e construção, defesa aeroespacial e muitos outros. (PMI, 2009)

O conceito do gerenciamento de projetos é “a aplicação de conhecimento, de habilidades, de ferramentas e técnicas a uma ampla gama de atividades para atender aos requisitos de um determinado projeto”. (PMI, 2009)

O gerenciamento de projetos está em constante evolução, o que faz com que essa área ganhe cada vez mais espaço dentro das organizações. A gestão tradicional acontecia verticalmente, os funcionários eram organizados em cadeias de cima para baixo. No entanto, hoje tem-se consciência de que para um projeto de sucesso, é preciso muito planejamento e coordenação. Devido a isso, a nova gestão acontece horizontalmente, que acarreta produtividade e eficiência pela maior interação entre gerentes e subordinados.

## 1.2 Ciclo de vida de um projeto

O ciclo de vida de um projeto é o conjunto de fases do projeto, ou seja, uma metodologia que organiza o projeto por um conjunto de fases. Ele conecta o início de um projeto ao seu final e é através dele que o gerente de projetos conhece a extensão de um projeto e faz um estudo de sua viabilidade. (VARGAS, 2007, p.10)

O ciclo de vida permite que em um projeto seja elaborado seu estudo analítico (EAP), seja feita uma definição de como os gestores desenvolverão seus processos e perceber uma série de similaridades entre os projetos, independente de seu contexto ou área de atuação. (VARGAS, 2007, p.10)

Para o acompanhamento de um projeto, o ciclo de vida é fundamental para avaliar o seu progresso, se suas características tendem a mudar com a conclusão das fases e se a incerteza tende a diminuir com o seu desenvolvimento. No início do ciclo, o esforço gasto é praticamente zero, na fase de desenvolvimento chega ao máximo e retorna ao zero novamente no final. O esforço é medido pela quantidade de pessoas, o dispêndio de trabalho, dinheiro, quantidade de horas extras, preocupação e complicação.

O ciclo de vida representa trabalhos técnicos a serem realizados em cada fase, suas entregas, quem estará envolvido e como cada fase deve ser

controlada e aprovada. (PMBOK, 2004)

As fases de um projeto dependem de sua natureza, cada uma é caracterizada pela entrega ou finalização de um trabalho. Nela podem ser observadas as atividades a serem realizadas e quem estará envolvido em seu desenvolvimento. As fases de um ciclo são (VARGAS, 2007):

1. **Iniciação:** é a fase inicial do projeto, em que identificam-se as necessidades, estabelece-se a missão e o objetivo do projeto e faz-se uma identificação das melhores estratégias para guiar seu desenvolvimento.
2. **Planejamento:** nessa fase acontece um detalhamento do que será feito no projeto, como cronogramas, interdependência entre atividades, alocação de recurso, análise de custos. Existem também os planos auxiliares, responsáveis pela comunicação, qualidade, riscos, aquisições e alocação de recursos humanos.
3. **Execução:** materialização das fases anteriores. Os erros são identificados para que não aconteçam em projetos futuros. Fase responsável pelo consumo da maior parte do orçamento e do esforço previstos.
4. **Monitoramento e controle:** é realizado paralelamente as fases de planejamento e a execução. Acompanha o andamento do projeto com o objetivo de propor ações corretivas e preventivas.
5. **Encerramento:** a fase de execução é avaliada através de uma auditoria interna ou externa. Ocorre a atividade de catalogação das lições aprendidas, isto é, as falhas ocorridas durante o desenvolvimento do projeto são discutidas e analisadas com o intuito de preveni-los em outros projetos.

A FIG. 2 ilustra um ciclo de vida genérico de um projeto. Como pode-se observar, no início os custos, quantidade de pessoas envolvidas e nível de atividades são baixos. Esses padrões aumentam gradativamente com o desenvolvimento do projeto, e no final caem rapidamente novamente.

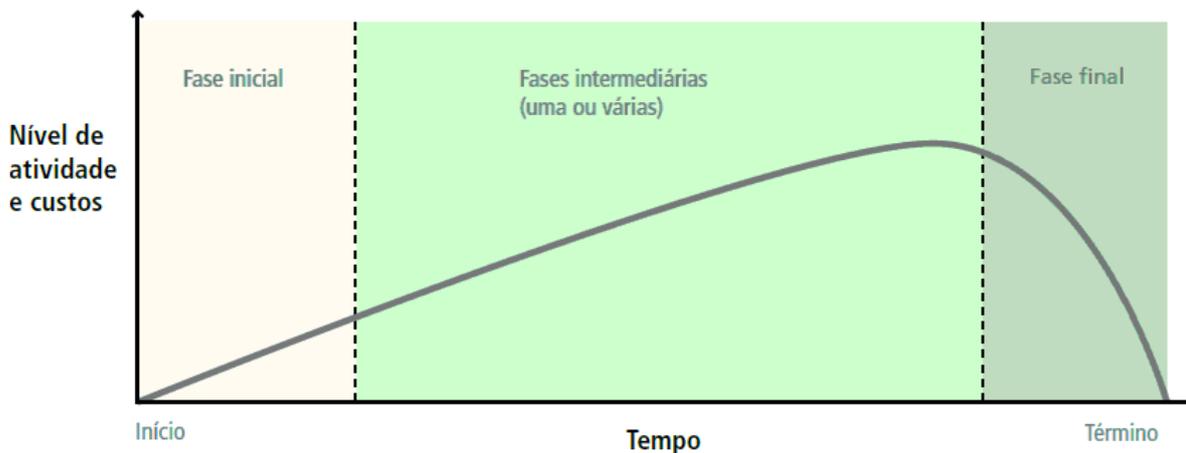


Figura 2 - Ciclo de vida genérico

Fonte: PMI(2004) *apud* Promon(2007)

### 1.3 O papel do gerente de projetos

O gerente de projetos é a pessoa responsável pelo gerenciamento de projetos e é também quem responde pelo seu sucesso. Ele deve ter habilidades como liderança, boa comunicação, negociação e resolução de problemas, influência na organização e experiência no ciclo de desenvolvimento do software (se essa for a área de gerenciamento). (PMI, 2009)

O gerente de projetos assume várias responsabilidades no projeto: controle e definição de riscos, avaliação e definição de fatores críticos de sucesso, avaliação e definição de pontos fortes e fracos do projeto, controle de cronograma, verificação de esforço, avaliação de desempenho da equipe, definição de prioridades, coordenação da interação entre os envolvidos, cumprimento de prazos, custos e qualidade dentro dos padrões estabelecidos, formalização de aceitação de artefatos, elaboração de relatórios de avaliação e acompanhamento. Além disso, ele deve estar aberto a receber sugestões, a dar e receber feedbacks, estabelecer um relacionamento aberto e franco divulgando amplamente o plano do projeto para a equipe. O gerente tem ganhado muito destaque nas organizações devido à evolução que o gerenciamento de projetos tem sofrido. (PMI, 2004)

### 1.4 PMBOK por área de conhecimento

O *PMBOK* é “a aplicação de conhecimentos, habilidades, ferramentas

e técnicas das atividades de um projeto a fim de alcançar seus objetivos”. (PMBOK, 2008). É um conjunto de melhores práticas em gerência de projetos, pesquisado, definido pelo *PMI* que são a base dessa metodologia.

Publicado pela primeira vez em forma de um guia em 1987, como um *White paper*, documentando e padronizando informações e práticas da gerência de projetos. Em 1996, foi publicada sua primeira edição, a segunda em 2000 e atualmente encontra-se na quarta edição, publicada em 2008. Nessa edição são abordados quarenta e quatro processos, divididos em nove áreas de conhecimento ou cinco grupos de processos. A FIG. 3 apresenta como é a interação desses grupos com as fases do ciclo de vida de um projeto. Pode ser observado que no início do projeto, os processos de iniciação consomem a maioria dos recursos. Com o seu desenvolvimento os processos de planejamento são os responsáveis pelo maior consumo, seguidos por execução e encerramento. Os processos de monitoramento e controle atuam uniformemente durante o ciclo de vida do projeto.

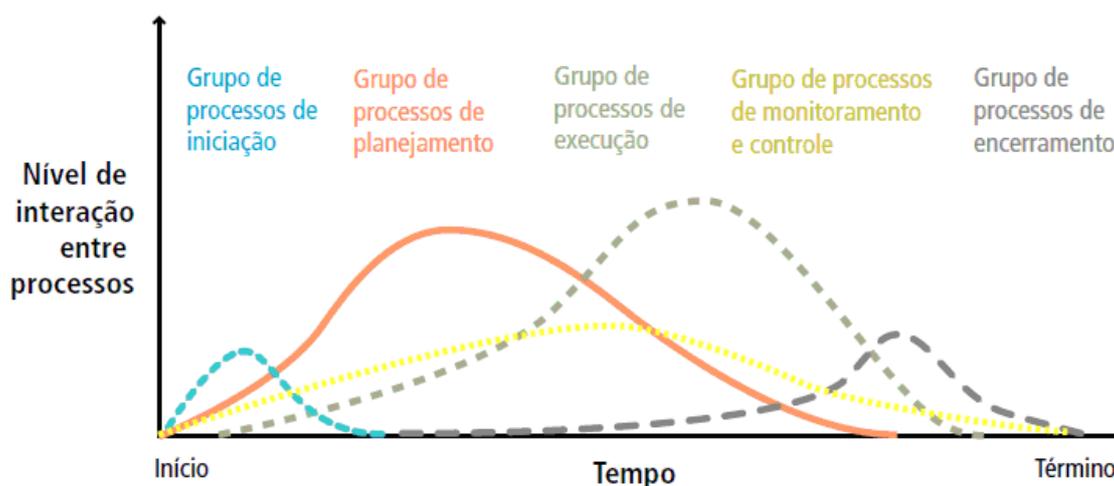


Figura 3 - Nível de interação dos processos ao longo do tempo

Fonte: PMI(2004) *apud* Promon(2007)

As áreas de conhecimento são: gerência de escopo, gerência de tempo, gerência de custo, gerência de riscos, gerência de recursos humanos, gerência de qualidade, gerência de comunicação, gerência de contratação e gerência de integração. Escopo, tempo, custo e qualidade são fundamentais para obter sucesso no projeto, recursos humanos e aquisições são insumos que devem mover o projeto. A comunicação e o risco são fatores de atenção e a integração é a responsável por combinar todas as outras áreas.

### 1.4.1 Gerenciamento da Integração

O gerenciamento da integração tem o objetivo de garantir que elementos estejam devidamente coordenados e integrados, ele é o responsável por combinar todos os outros gerenciamentos.

Os processos do gerenciamento de integração são (*PMBOK*, 2008):

1. **Desenvolver termo de abertura do projeto:** desenvolvimento do termo de abertura de um projeto, que é o aceite formal de início do projeto ou de uma fase.
2. **Desenvolver a declaração de escopo preliminar do projeto:** desenvolvimento da declaração de escopo que fornecerá uma descrição em um nível mais alto do escopo.
3. **Desenvolver o plano de gerenciamento do projeto:** documentação necessária para definir, preparar, integrar e coordenar todos os planos auxiliares de gerenciamento.
4. **Orientar e gerenciar a execução do projeto:** execução do trabalho previsto no plano de gerenciamento.
5. **Monitorar e controlar o trabalho do projeto:** acompanhamento dos processos usados durante o projeto para atender aos objetivos de desempenho desse.
6. **Controle integrado de mudanças:** revisão, aprovação e controle das solicitações de mudanças do projeto.
7. **Encerrar o projeto:** finalização de todas as atividades do projeto para encerrar formalmente o projeto ou uma de suas fases.

### 1.4.2 Gerenciamento de Escopo

O gerenciamento de escopo é responsável pelas características ou funcionalidades que caracterizam o produto ou serviço. (VARGAS, 2007) O escopo possui todo o trabalho que terá que ser realizado e seu gerenciamento garante que o projeto realizará todo e somente o trabalho necessário para seu sucesso.

Uma verificação constante é realizada durante o gerenciamento para se ter certeza do alinhamento entre o que está sendo feito e o necessário para obter

sucesso no projeto.

Os processos do gerenciamento de escopo são (*PMBOK*, 2008):

1. **Planejamento do escopo:** plano de gerenciamento que documenta como o escopo será definido, verificado e controlado.
2. **Definição do escopo:** declaração detalhada do escopo.
3. **Criar EAP:** subdivisão dos principais produtos em componentes menores e gerenciados mais facilmente.
4. **Verificação do escopo:** formalização da aceitação.
5. **Controle do escopo:** controle de mudanças.

#### 1.4.3 Gerenciamento de Tempo

O gerenciamento do tempo tornou-se necessário devido a projetos que foram entregues fora do prazo. Isso acarretou em insatisfação do cliente, aumento de custo do projeto e tensão da equipe de desenvolvimento. Ele surge para garantir que o projeto finalize no prazo desejado.

Os processos desse gerenciamento são (*PMBOK*, 2008):

1. **Definição da atividade:** identificação das atividades específicas do cronograma.
2. **Sequenciamento das atividades:** identificação e documentação de pendências entre atividades.
3. **Estimativa de recurso da atividade:** tipo e quantidade de recursos necessários.
4. **Estimativa de duração da atividade:** número de períodos de trabalhos necessários para finalizar seu desenvolvimento.
5. **Desenvolvimento do cronograma:** análise de recursos, restrições, duração e sequência das atividades.
6. **Controle dos cronogramas:** controle das mudanças.

#### 1.4.4 Gerenciamento de Custos

O gerenciamento de custos acontece para garantir que o projeto seja

executado dentro do orçamento aprovado pelo projeto. (VARGAS, 2007). Essa estimativa deve ser feita preferencialmente por quem realizará o trabalho. Informações históricas são importantes para garantir que erros nos custos não sejam repetidos.

Os processos do gerenciamento de custos são (*PMBOK*, 2008):

1. **Estimativa de custo:** estimativa de custo para terminar as atividades de um projeto.
2. **Orçamentação:** agregação dos custos estimados das atividades.
3. **Controle de custos:** controle do fator e de mudanças no orçamento.

#### 1.4.5 Gerenciamento de Qualidade

O gerenciamento da qualidade tem o objetivo de garantir que o projeto será concluído dentro da qualidade e padrões desejados e também que a satisfação dos envolvidos seja concretizada. O gerente de projetos é o maior responsável por essa área e deve ter a mesma prioridade que tem com custos e tempo.

Os processos do gerenciamento de projetos são (*PMBOK*, 2008):

1. **Planejamento da qualidade:** identificação dos padrões de qualidade do projeto e planejamento estratégico de como serão atingidos.
2. **Realizar a garantia da qualidade:** aplicação das atividades planejadas para alcançar os padrões determinados.
3. **Realizar o controle da qualidade:** acompanhar os resultados do projeto a fim de determinar se estão de acordo com os padrões de qualidade estabelecidos.

#### 1.4.6 Gerenciamento de Recursos Humanos

O gerenciamento de recursos humanos é o responsável por gerenciar os indivíduos envolvidos no projeto (*stakeholders*) e fazer o melhor uso desses

durante o desenvolvimento.

Os processos de gerenciamento de recursos humanos são (*PMBOK*, 2008):

1. **Planejamento de recursos humanos:** identificação e documentação das funções do projeto e criação do plano de gerenciamento pessoal.
2. **Contratar ou mobilizar a equipe do projeto:** obtenção de recursos para o projeto.
3. **Desenvolver a equipe do projeto:** melhoria de competências e integração entre os membros da equipe.
4. **Gerenciar a equipe do projeto:** acompanhamento de desempenho, *feedback* e resolução de problemas da equipe.

#### 1.4.7 Gerenciamento de Comunicações

O gerenciamento de comunicações é o responsável para que as informações desejadas cheguem às pessoas certas em tempo hábil e de maneira economicamente viável.

Os processos do gerenciamento de comunicações são (*PMBOK*, 2008):

1. **Planejamento das comunicações:** determinação de necessidades e comunicação dos envolvidos no projeto.
2. **Distribuição das informações:** disponibilizar informações necessárias aos interessados no projeto.
3. **Relatório de desempenho:** coleta e distribuição de informações de desempenho.
4. **Gerenciar as partes interessadas:** satisfação dos requisitos das partes interessadas.

#### 1.4.8 Gerenciamento de Riscos

O gerenciamento de riscos é a área responsável por identificar,

analisar e responder aos riscos de um projeto. Seu objetivo é maximizar os eventos positivos e minimizar os negativos. (VARGAS, 2007)

Os riscos tecnológicos são aqueles que envolvem a utilização de novas tecnologias ou de tecnologias não testadas completamente ou corretamente.

Os processos dessa área de conhecimento (*PMBOK*, 2008):

1. **Planejamento do gerenciamento de riscos:** como será feito o gerenciamento.
2. **Identificação dos riscos:** levanta hipóteses acidentais.
3. **Análise qualitativa dos riscos:** estimativa de riscos.
4. **Análise quantitativa dos riscos:** impacto desses riscos para o projeto.
5. **Planejamento de respostas a riscos:** medidas para mitigá-los.
6. **Monitoramento e controle de riscos:** acompanhamento dos riscos do projeto.

#### 1.4.9 Gerenciamento de Aquisições

O gerenciamento de aquisições é o responsável pela compra ou adesão de produtos, serviços ou resultados. Nele inclui o gerenciamento de contratos e o gerenciamento de controle de mudanças. Inclui administração de qualquer contrato emitido por organização externa que tem o interesse de adquirir o produto. (DIAS, 2008).

Os processos do gerenciamento de aquisições são:

1. **Planejar compras e aquisições:** determinação de que comprar e quando adquirir.
2. **Planejar contratações:** documentação de requisitos e identificação de possíveis fornecedores.
3. **Solicitar respostas de fornecedores:** obtenção de informações, cotações, preços, ofertas ou propostas.
4. **Solicitar fornecedores:** análise de ofertas, escolha de fornecedor e negociação por contrato.
5. **Administração do contrato:** gerenciamento do contrato e relação comprador x consumidor.

6. **Encerramento de contrato:** terminar e liquidar contrato.

## 2. METODOLOGIAS ÁGEIS

Uma das principais características de um projeto de desenvolvimento de software é a grande instabilidade de requisitos. As solicitações de mudanças por parte do cliente são constantes e com as metodologias tradicionais de desenvolvimento de software é muito complicado que um projeto possa ser entregue no prazo firmado, no custo previsto e, principalmente, com a qualidade desejada. (SANTOS, 2007)

As metodologias ágeis surgem com o princípio de permitir que o código de um software seja alterado constantemente sem prejudicar sua qualidade. Baseiam-se em uma menor burocracia de processos e menos documentação. Elas são uma constatação feita por diversos profissionais da área de engenharia de software, que apesar de terem adquirido muito conhecimento com a cartilha tradicional de desenvolvimento, alguns riscos só poderiam ser minimizados com o pensamento e atitudes diferentes. (ALMEIDA, 2007)

Esse capítulo apresenta uma introdução sobre as metodologias ágeis para desenvolvimento de software. O foco é o *Scrum*, “que é um processo ágil que pode ser utilizado para gerenciar e controlar desenvolvimentos complexos de softwares e produtos, usando práticas iterativas e incrementais.”(SCHNEIDER, 2008)

Na seção 2.1, é observado o surgimento das metodologias ágeis, um breve histórico de como essa prática se iniciou e também seu conceito. Uma visão geral sobre essas metodologias é apresentada. Seus princípios são estudados e o Manifesto Ágil, que é uma declaração de princípios que fundamentam o desenvolvimento ágil de software, é mostrado com seus doze princípios. Na seção 2.2, inicia-se a apresentação da metodologia ágil *Scrum*. Em que podem ser observados conceito, histórico e suas propriedades. Como subseções tem-se as atribuições de funções, onde são apresentados o *Product Owner*, *Scrum Master*, *Scrum Team* e *Stakeholders*. Os artefatos do *Scrum*, que incluem *Product Backlog*, *Sprint Backlog*, *Selected Backlog*, *Retrospective Items*, *Release Burndown*, *Scrum board* e *Impediment Backlog*. Os fluxos do *Scrum*, que são o Pré-game, o *Sprint* e todas as suas características e, por fim, o Pós-game. Na seção 2.3, uma breve introdução é feita sobre outras metodologias ágeis, o *XP (Extreme Programming)*,

*FDD (Feature Driven Development), MSF (Microsoft Solutions Framework) e DSDM (Dynamic Systems Development Method).*

## 2.1 Apresentação geral

O modelo de desenvolvimento de um software considerado tradicional é aquele baseado em processos e documentação. Um exemplo de modelo mais utilizado é o Cascata, cujos procedimentos seguem uma sequência de fases: levantamento de requisitos, análise, projeto, implementação, teste, implantação e manutenção. Essas metodologias tradicionais são consideradas pesadas, pois requerem muita burocracia e documentação, ao contrário das metodologias ágeis que se preocupam mais com a implementação. (ALMEIDA, 2008)

As metodologias ágeis surgiram em um contexto em que o número de casos de fracasso em projetos era muito grande e diversos profissionais da área de Engenharia de Software perceberam que, apesar de estudarem pela cartilha tradicional, na prática os riscos eram diminuídos usando-se métodos diferentes dos tradicionais. Em 2001, dezessete profissionais perceberam que seus métodos de trabalho iam contra o fluxo das metodologias tradicionais, porém eram além de eficazes, muito parecidos. Eles se reuniram e tentaram formar um modelo de desenvolvimento que substituísse o tradicional, mas chegaram à conclusão que é um processo muito complexo e resolveram propor alguns princípios determinantes para a obtenção de bons resultados. (BASSI, 2008) Tais princípios resultaram na publicação do Manifesto Ágil, documento baseado em doze princípios. (MANIFESTO ÁGIL, 2001):

1. Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
3. Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
4. Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto diariamente, durante todo o curso do projeto.
5. Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
6. O método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara-a-cara.
7. Software funcional é a medida primária de progresso.
8. Processos ágeis promovem um ambiente sustentável. Os

- patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
9. Contínua atenção à excelência técnica e bom design, aumenta agilidade.
  10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
  11. As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
  12. Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Em contrapartida com as metodologias tradicionais, as metodologias ágeis valorizam os indivíduos e as interações, o software executável, a colaboração com o cliente e as respostas rápidas as mudanças. Já as outras, dão valor aos processos e ferramentas, à documentação, à negociação de contrato e ao planejamento. (MANIFESTO ÁGIL, 2001)

O manifesto ágil vem aumentando seu número de adeptos. Em novembro de 2007, existiam sete mil trezentas e setenta e oito pessoas que o assinaram publicamente e a *Scrum Alliance* possui dezoito mil duzentos e vinte e cinco associados que pagaram para obter seu certificado em *Scrum*. (BASSI, 2008) As empresas adeptas, por exemplo, empresas renomadas como a *Google, Yahoo!, Microsoft, IBM, Cisco, Symantec, Siemens*. (ENGENHARIA DE SOFTWARE MAGAZINE, 2009)

Uma pesquisa realizada por Reifer em 32 organizações apontou que 14 delas usavam metodologias ágeis, motivadas por um histórico baixo no desempenho dos projetos quanto ao cumprimento de prazos e custos. A maioria das empresas analisadas era considerada nível superior na maturidade dos processos, contudo, buscavam algo novo que pudesse reverter o quadro de mau desempenho nos projetos. A TAB. 1, mostra as características dessas empresas estudadas nos diversos ramos de atuação, assim como detalhes quanto ao início da prática, quantidade de projetos realizados com o uso de agilidade e, o estágio de implementação. Após essa pesquisa, pode-se concluir que os principais resultados foram o incremento de produtividade, redução de custos, redução do tempo de entrega e melhoria na qualidade. (JAVA MAGAZINE, 2009)

<b>Indústria</b>	<b># Empresas que utilizam Métodos Ágeis</b>	<b># de projetos</b>	<b>Início da Utilização de Métodos Ágeis</b>	<b>Estágio da Implementação</b>
<b>Aeroespacial</b>	1	1	2001	Descoberta
<b>Computação</b>	2	3	2000	Piloto
<b>Consultoria</b>	1	2	2000	Piloto
<b>Comércio Eletrônico</b>	5	15	2000	Produção
<b>Pesquisa</b>	1	1	2000	Piloto
<b>Software</b>	2	4	2000	Produção
<b>Telecomunicações</b>	2	5	2000	Produção
<b>Total</b>	14	31	n/a	n/a

Tabela 1 – Características da empresas pesquisadas

Fonte: REIFER(2002) *apud* JAVA MAGAZINE(2009)

Algumas características dessas metodologias são comuns. Nas metodologias tradicionais, a implementação e os testes são etapas distintas, já nas ágeis, há um grande foco para os testes e dá-se preferência pela criação de testes de unidade antes da implementação do código. (ENGENHARIA DE SOFTWARE MAGAZINE, 2009)

Uma outra característica é o desenvolvimento em iterações, ou seja, divide-se em ciclos curtos que possuem os procedimentos que um projeto, desde o planejamento até a entrega de uma versão para o cliente com mais funcionalidades implementadas. São cooperativas, pois estimulam a proximidade com o cliente e a interação entre os programadores, diretas devido à simplicidade de aprendizado e documentação e, são extremamente adaptativas, possuem facilidade em acomodação com as mudanças ao longo do projeto. (ENGENHARIA DE SOFTWARE MAGAZINE, 2009)

As equipes ágeis baseiam-se na comunicação e na transparência, admitem a possibilidade de incertezas em suas estimativas e as expõem para o cliente. Quanto mais a longo prazo, maior a hesitação associada que diminui com o aumento do conhecimento sobre o assunto. (BASSI, 2008)

As metodologias ágeis baseiam-se em um gerenciamento com liderança e colaboração. Portanto, para uma implementação bem sucedida, é

necessário uma compatibilidade entre conceitos e valores da organização. A existência de uma equipe motivada é um fator crítico de sucesso, uma vez que as equipes normalmente são auto-organizáveis.

Alguns autores discutem algumas limitações encontradas nessas metodologias, como a inexistência de suporte a equipes distribuídas, subcontratação, dificuldade de apoio a grandes projetos e grandes equipes, sistemas de missão crítica e nem sempre é possível o cliente estar disponível. Alegam também que as metodologias não tratam da manutenção do software desenvolvido. (JAVA MAGAZINE, 2009)

## 2.2 Scrum

O *Scrum* é “um *framework* de processos leves para gestão de projetos seguindo os valores e princípios ágeis.”(KNIBERG, 2007) Criado em 1996 por Jeff Sutherland e Ken Schwaber, é uma metodologia que possui um foco maior nos aspectos gerenciais de desenvolvimento. A equipe trabalha unida com o objetivo de entregar o software funcional de alta qualidade, ela se compromete com uma meta e tem autonomia para definir como alcançá-la.

A não determinação de como a equipe executará suas atividades, favorece a auto-organização dessa e também a integração com outras metodologias ágeis, como o XP. Já existem relatos de sucesso com essas integrações. (KNIBERG, 2007)

O *Scrum* enfatiza a comunicação, trabalho em equipe, flexibilidade e fornecimento incremental de software funcionando. Possui uma abordagem empírica baseada em exploração e adaptação e assume que os clientes mudarão de ideia sobre os requisitos, concentrando assim, em responder rapidamente as funcionalidades mais prioritárias. (KNIBERG,2007)

### 2.2.1 Atribuição de funções

No *Scrum*, existem vários papéis para as pessoas envolvidas no projeto, pode-se observar o *Product Owner*, *Scrum Master*, *Scrum Team* e *Stakeholders*.

O *Product Owner* é um especialista do negócio, representa todos os *stakeholders*. Ele administra fundos para o projeto, estabelece e comunica a visão do produto, decide quando serão liberadas as versões do produto, responde e apóia o *Scrum Team* para sanear dúvidas sobre os requisitos, atualiza e prioriza o *Product Backlog*, aceita ou rejeita o resultado dos trabalhos e define as funcionalidades do produto. (KNIBERG, 2007)

O *Scrum Master* é o responsável por liderar o time ao sucesso. Ele remove os obstáculos, mais conhecidos como impedimentos, assegura que as práticas do *Scrum* estão sendo aplicadas corretamente, certifica que a cultura da organização está ajustada para que as metas sejam alcançadas. Ele também é uma espécie de gerente de projetos, pois responde pelo *Scrum Team* perante o *Product Owner*, acompanha o progresso do trabalho diariamente e atua ao lado dos membros do *Scrum Team* para treiná-los em situações de adaptação constante. É o *Scrum Master* também que organiza as reuniões de *Sprint*. (KNIBERG, 2007)

O *Scrum Team* é a equipe que desenvolverá o projeto. Deve ser composta entre 5 e 9 pessoas e multifuncional, ou seja, reunir todas as especializações necessárias para implementar segmentos completos do software a cada *Sprint*. Ele desenvolve os itens do *Product Backlog* do mais prioritário ao menos prioritário, estima o esforço necessário para implementá-los e então os quebra em atividades menores que não devem ultrapassar o limite diário de horas. Ainda identificam impedimentos e reportam ao *Scrum Master* para resolver. (KNIBERG, 2007)

Os *Stakeholders* são os interessados no projeto, a começar por clientes, usuários finais, equipe de *marketing*, *Scrum Team*, *Scrum Master* e outros. Podem consultar o *Product Backlog* a qualquer momento. (KNIBERG, 2007)

### 2.2.2 Artefatos

No Scrum existem diversos artefatos que auxiliam o funcionamento de um projeto, que são o *Product Backlog*, *Selected Backlog* ou *Sprint Backlog*, *Impediment Backlog*, *Retrospective Itens*, *Sprint Burndown Chart*, *Release Burndown* e *Scrum Board*.

O *Product Backlog* é uma lista incompleta e dinâmica que contem as

demandas de todos os *Stakeholders*, ordenada pela prioridade de desenvolvimento dos itens. Contém funcionalidades, correções de *bugs*, tecnologias e melhorias. Ela nunca está completa, evolui de acordo com a evolução do produto e do ambiente e quem faz a sua atualização e priorização é o *Product Owner*. Seus itens possuem descrição e estimativa de esforço, aqueles que estão no topo da lista possuem menor granularidade e maior detalhamento. A TAB.2 ilustra um *Product Backlog*. (BROD TECNOLOGIA, 2009) Nele existem duas colunas, uma com os nomes dos itens do *Backlog* e a outra com as estimativas de tamanho desses itens que já foram estimados pelo *Scrum Team*. O primeiro item é a permissão que um usuário faça uma reserva, sua estimativa foi três.

Item do Backlog	Estimativa
Permitir que o usuário faça uma reserva	3
Permitir que o usuário cancele uma reserva	5
Permitir a troca de datas da reserva	3
Permitir que empregados do hotel gerem relatórios de lucratividade	8
Melhorar manipulação de erros	8
...	30
...	50

Tabela 2: Exemplo de um *Product Backlog*.

Fonte: BROD TECNOLOGIA, 2009.

O *Sprint Backlog* é uma lista dos itens que serão desenvolvidos durante o *Sprint*. Esses itens são selecionados do *Product Backlog* na reunião de planejamento do *Sprint* e depois divididos em tarefas menores. Deve estar disponível para todos os membros do *Scrum Team* mas pode ser atualizado apenas pelo *Scrum Master*. (BROD TECNOLOGIA, 2009)

*Impediment Backlog* é uma pilha de impedimentos que impedem o *Scrum Team* a progredir no desenvolvimento. Ela é mantida pelo *Scrum Master* que é o responsável por resolver seus itens. (BROD TECNOLOGIA, 2009)

*Retrospective Itens* é uma lista com os itens citados pelo *Scrum Team* na reunião de retrospectiva. A lista organizacional é mantida pelo *Product Owner* e a dos itens do time pelo *Scrum Master*. (BROD TECNOLOGIA, 2009)

O *Sprint Burndown Chart* é um gráfico que mostra a soma dos tempos estimados restantes das tarefas de um *Sprint* por tempo. Ele é atualizado diariamente pelo *Scrum Master* após a reunião diária do *Sprint* para visualizar o

progresso conquistado pelo *Scrum Team* naquele dia. (BROD TECNOLOGIA, 2009) Na FIG. 4 é apresentado um *Sprint Burndown Chart*. Como pode ser observado, ele representa o número de horas que ainda faltam ser desenvolvidas pelo time ao longo dos dias úteis do *Sprint*. No início, o gráfico está alto pois poucas horas foram realizadas mas , a medida que os dias passam, essas horas vão diminuindo e o gráfico vai caindo.



Figura 4 - Exemplo de um *Sprint Burndown Chart*.

Fonte: BROD TECNOLOGIA, 2009

O *Release Burndown* é o gráfico que mostra o desempenho e progresso da equipe no projeto inteiro. É atualizado a cada *Sprint* pelo Scrum Master. (SYDLE<sup>3</sup>, 2009)

Na FIG. 5 é apresentado o *Release Burndown*, um gráfico que compara a produtividade do time no *Sprint* atual de acordo com a produtividade geral. A linha reta no gráfico é a produtividade esperada do time.

<sup>3</sup> Treinamento ministrado por Fábio Barone, colaborador da empresa Sydle Sistemas S.A. aos colaboradores da empresa. A Sydle é uma empresa especializada no desenvolvimento de sistemas de missão crítica, utiliza o Scrum para o desenvolvimento de alguns projetos. (Nota da autora)

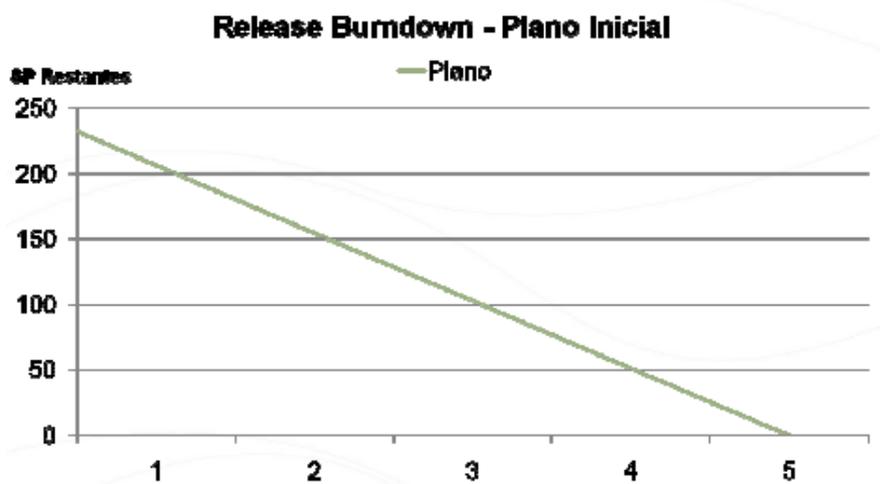


Figura 5 - Exemplo de um *Release Burndown*.

Fonte: SYDLE, 2009

O *Scrum Board* é um quadro que exhibe o progresso de cada atividade do *Sprint Backlog*. Ele é dividido em três partes e as atividades são representadas como estão no momento: aguardando para ser atendida, iniciada e concluída. É atualizado diariamente pelos membros da equipe e deve ficar visível a todos. (SYDLE, 2009)

Na FIG. 6 é apresentado um *Scrum Board*. Nele podem ser observadas quatro colunas: (i) atividades que ainda não foram iniciadas, (ii) as atividades em andamento, (iii) atividades concluídas; (iv) o *Burndown* e atividades não planejadas.

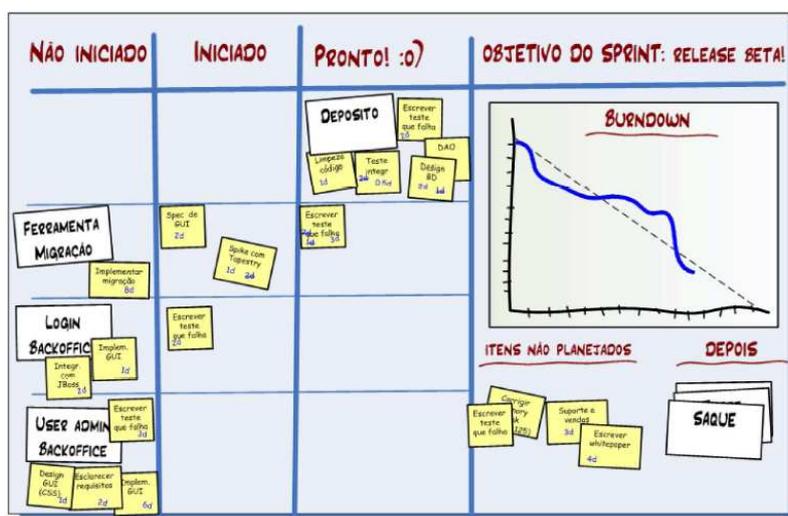


Figura 6 - Exemplo de um *Scrum Board*.

Fonte: BROD TECNOLOGIA, 2009

### 2.2.3 Fluxo do *Scrum*

Um projeto que é baseado em *Scrum* possui três fases em seu ciclo: pré-game, game ou *Sprint* e pós-game.

A fase de pré-game do *Scrum* é caracterizada pela organização inicial de tudo. Nela é feita uma análise de viabilidade e custos para o projeto, o *Scrum Team* é montado e já iniciam-se os treinamentos. O *Product Owner* começa a criação do *Product Backlog*.

No *Scrum*, os ciclos de iterações que existem nas metodologias ágeis são chamados de *Sprints*. Eles acontecem em um período de duas a quatro semanas e tem um objetivo específico, representado pelo desejo do cliente pelo incremento do software. O *Scrum Team* é o responsável por estimar os itens que compõem esse desejo, mas somente o *Product Owner* pode cancelar um *Sprint* se esses itens perderem o sentido. (SYDLE, 2009)

Durante o *Sprint*, são realizadas algumas reuniões. A primeira delas é a de planejamento, mais conhecida como *Sprint Planning*. É nela que o time define as estimativas para as histórias que serão desenvolvidas do *Sprint Backlog* e, em seguida, faz a divisão em atividades menores. Para acompanhamento do *Scrum Master* e interação entre os membros da equipe, diariamente é feita uma reunião, mais conhecida como *Daily Scrum*. A duração máxima é de quinze minutos e todos devem ficar em pé. O objetivo de todos permanecerem em pé durante essa reunião é de diminuir a sua duração, que seja discutido apenas o que foi feito durante o dia. Durante o *Daily Scrum* cada membro responde a três perguntas: “O que você fez hoje?”, “O que o atrapalhou?” e “O que você fará amanhã?”. (SYDLE, 2009)

Ao final de cada *Sprint* todos os *Stakeholders* se reúnem para duas reuniões. Primeiro para a chamada *Sprint Review*, que objetiva demonstrar o trabalho realizado durante o *Sprint* e obter aprovação do *Scrum Master*. Em seguida, é realizado o *Retrospective Meeting* e o time faz um alinhamento em grupo a respeito dos acontecimentos em relação às pessoas, relacionamentos, processos e ferramentas ao longo do *Sprint*. O objetivo dessa reunião é a identificação de possíveis melhorias.

Após o desenvolvimento de todo *Product Backlog* de todos os *Sprints*, acontece o pós-game, que é o fechamento oficial do projeto. Nesta fase, são feitas as etapas de integração, testes finais e documentação.

A FIG. 07 ilustra o ciclo do *Scrum*, com todas as suas etapas, desde a preparação do *Product Backlog* até a entrega do produto. O ciclo inicia no *Product Backlog* com seu desenvolvimento. Após inicializado, alguns itens são escolhidos pelo *Product Owner* e entram no *Sprint Backlog*. Um *Sprint* acontece no período de duas a quatro semanas (2 -4 weeks). Diariamente (24 hours) acontecem as *Daily Scrum Meeting* para acompanhamento do desenvolvimento e, por fim o que foi desenvolvido é entregue ao cliente.

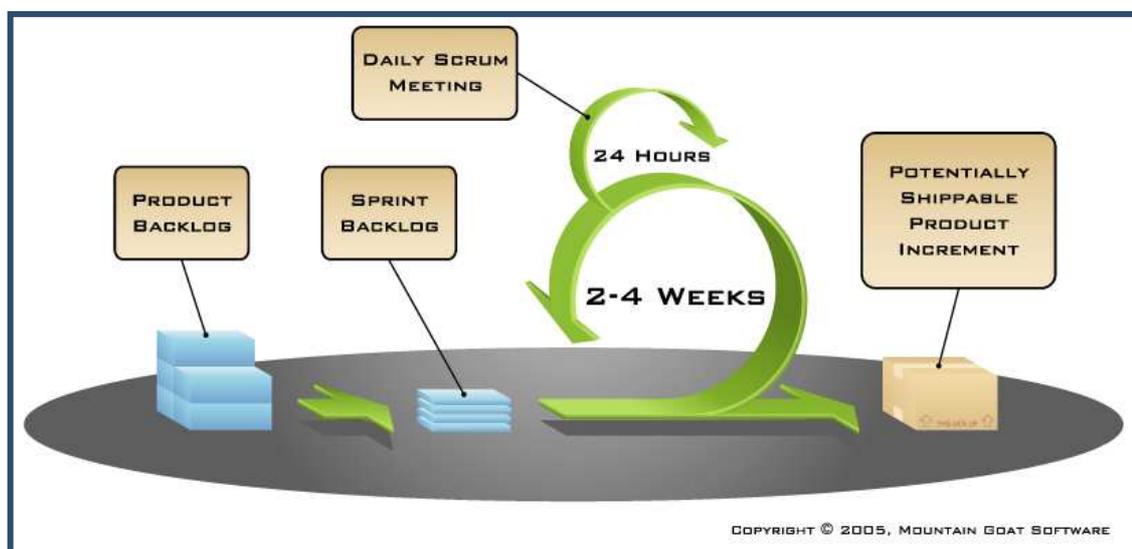


Figura 7 - Ciclo do *Scrum*

Fonte: SABBAGH(2006) *apud* MOUNTAIN GOAT SOFTWARE(2005)

### 2.3 Outras metodologias ágeis

O *Extreme Programming (XP)* é uma metodologia ágil criada em 1998 por Kent Beck, Ron Jeffries e Lunninghan. Representa aspectos técnicos de desenvolvimento e baseia-se em quatro atividades básicas: ouvir, desenhar, codificar e testar. Assim como as outras metodologias, ele é dividido em ciclos e, em cada ciclo, é feita uma reunião para decidir quais funcionalidades serão desenvolvidas. Algumas características do *XP* são o desenvolvimento do código em pares, todo código deve possuir testes unitários e prezar por um design simples e objetivo. (ENGENHARIA DE SOFTWARE MAGAZINE, 2009)

Outra metodologia muito usada é a *Feature Driven Development (FDD)*, que foi criada no final dos anos 90 em um projeto Java de grande porte de

um banco em Cingapura por Jeff de Luca. Ela é baseada em processos bem definidos, equipes auto-organizáveis e o papel do cliente é crítico. Cada iteração possui as fases de *kick-off meeting*, projeto, desenvolvimento, revisão do código e fechamento do release. (ENGENHARIA DE SOFTWARE MAGAZINE, 2009)

O *Dynamic Systems Development Method (DSDM)* foi projetado em meados dos anos 90 e é controlado por um consórcio de empresas. É a única metodologia compatível com a ISO9000, pois foi criada a partir do RAD. Seu ciclo consiste em pré-projeto, análise de aderência, estudo de negócio, modelagem funcional, projeto e desenvolvimento, implementação e pós implementação. Sua política é de primeiro fixar o prazo e o custo e depois ajustar as funcionalidades. (ENGENHARIA DE SOFTWARE MAGAZINE, 2009)

Por fim, a metodologia *Microsoft Solutions Framework (MSF)* foi criada pela Microsoft em 1994 a partir da análise de times de projetos e grupos. É um guia de procedimentos e boas práticas para projetos de desenvolvimento de software. Foca nos princípios de foco no negócio, comunicação, visão do projeto compartilhada, esclarecimento das responsabilidades compartilhadas, agilidade, investimento em qualidade, aprendizado com experiências anteriores e mais poderes aos membros do time. (ENGENHARIA DE SOFTWARE MAGAZINE, 2009)

### 3. GESTÃO DE PROJETOS EM SCRUM

A gerência de projetos tradicional é incapaz de se adaptar a mudanças e, as equipes têm cada vez mais dificuldade para reagir rapidamente a mudanças. Como citado no capítulo anterior, as metodologias ágeis surgiram para projetos que requerem inovação e criatividade ou que estejam sujeitos a constantes alterações. Um questionamento feito por vários estudiosos é se é possível integrar essas duas áreas e, com isso, aumentar a possibilidade de um projeto obter sucesso. (BASSI, 2008)

Nesse capítulo é apresentada a pesquisa feita sobre a gerência de projetos e as metodologias ágeis e, também, sobre o gerenciamento ágil de projetos. Na seção 3.1 é apresentada uma comparação entre essas duas formas de gerenciamento. Na seção 3.2, são descritas em detalhes todas as fases do ciclo de vida de um projeto ágil detalhadamente. E, por fim, na seção 3.3, é realizado um estudo de caso com o intuito de apresentar a aplicabilidade dessa combinação em um cenário real.

#### 3.1 Gerência Ágil X Tradicional

O gerenciamento clássico de projetos foca sua importância para o planejamento detalhado do projeto e aos processos formais de monitoramento e controle. Nele, é fundamental que toda a documentação seja feita e que um projeto possua escopo e custo fixo. Com esse foco, a pressão é cada vez maior em cima dos gestores para obterem resultados que impulsionem a melhoria no produto final. (WILLI, 2008)

O gerenciamento ágil tem o objetivo de maximizar a produtividade de um trabalho com a promessa de entrega rápida, flexibilidade e qualidade. Logo, ele valoriza as respostas a mudanças mais que o surgimento de um plano, a entrega do produto acima da documentação e prioriza o contato com o cliente. Elas favorecem o envolvimento direto dos desenvolvedores com os *stakeholders*, facilitando o entendimento dos requisitos. (WILLI, 2008)

A má interpretação de que o *PMBOK* é um processo e que tem que ser seguido para dar certo tem feito com que muitas pessoas questionem sobre o uso

de metodologias ágeis com o gerenciamento de projetos. Segundo Willi (2008) “o *PMBOK* não define um processo, ele é um conjunto de conhecimentos e as ditas melhores práticas.”

A TAB. 3 mostra uma comparação entre as características do gerenciamento ágil e o clássico. Nela pode ser observado que, na gestão clássica, as funções do gerenciamento são o planejamento, controle e tomada de decisão, os fundamentos teóricos são baseados no mecanicismo, o ambiente é estável, o acesso às informações é restrito, existe uma baixa capacidade de adaptação e a responsabilidade é centralizada no gerente. Por outro lado na gestão ágil, o ambiente apesar de ser turbulento, é centralizado, apresenta alta capacidade de adaptação, o acesso às informações é aberto e a gerência é descentralizada.

	<b>Gestão clássica</b>	<b>Gestão Ágil</b>
<b>Objetivos</b>	Prazo, custo e qualidade	Prazo, Custo, Qualidade e Capacidade de Transformação
<b>Funções do Gerenciamento</b>	Planejamento, Controle e Tomada de Decisão	Suporte
<b>Fundamento Teórico</b>	Mecanicismo: Divisão do Trabalho, Especialização e Controle	Projeto com um Sistema Complexo Adaptativo
<b>Meio Ambiente</b>	Relativamente Estável e Previsível	Turbulento e Imprevisível
<b>Estrutura Formal</b>	Centralizada e Hierárquica	Descentralizada: Times Pequenos e Semiautônomos
<b>Acesso às Informações</b>	Restrito: Gestores possuem as Informações	Aberto: Informação é Domínio Comum
<b>Controle</b>	Restrito e Centralizado	Descentralizada nos times semiautônomos
<b>Aprendizado</b>	Restrito	Contínuo e Adaptativo
<b>Capacitação de Adaptação</b>	Baixa	Elevada
<b>Responsabilidade</b>	Centralizada no Gerente	Descentralizada nos times semiautônomos.

Tabela 3: Comparação da gestão clássica com a gestão ágil

Fonte: Neuman, Baureis, Stock, 2009, p. 65.

De acordo com o *PMBOK* (2004), a gerência de projetos com:

todos os conhecimentos, as habilidades e os processos descritos não precisam ser sempre aplicados uniformemente em todos os projetos. O gerente de projetos, em colaboração com a equipe do projeto, é sempre responsável pela determinação dos processos adequados e do grau adequado de rigor de cada processo, para qualquer projeto específico.

Para DIAS (2005), o gerenciamento clássico dá uma grande importância ao planejamento detalhado do projeto e aos processos formais de monitoramento e controle. Já o gerenciamento ágil transfere o planejamento para a

execução, visando a entrega rápida de valor ao cliente e a apresentação de resultados ao longo de todo o projeto. É uma transferência do controle para a adaptação, permitindo alterações substanciais de escopo a cada iteração.

Para HIGHSMITH (2006), existem alguns princípios para os gerentes de projetos que atuam na agilidade:

1. entregar algo que seja útil para o cliente
2. cultivar patrocinadores compromissados com o projeto
3. empregar estilo de liderança colaborativa
4. criar times competentes e colaborativos
5. facilitar as tomadas de decisão pelo próprio time
6. utilizar funcionalidades rapidamente
7. encorajar adaptabilidade
8. premiar excelência técnica
9. focar mais em atividades que produzem artefatos utilizáveis pelo cliente do que atividades de apoio

O gerenciamento ágil busca desenvolver uma visão do futuro, baseada na exploração e desfazendo do planejamento prévio de ações e atividades que a gestão clássica possui. (WILLI, 2008)

### 3.2 Ciclo de vida de um gerenciamento de projeto ágil

O gerenciamento de projetos ágil possui cinco fases em seu desenvolvimento: Visão, Especulação, Exploração, Adaptação e encerramento. (DIAS, 2005)

Na fase de Visão é feito um acompanhamento geral do produto e do escopo, a identificação dos *stakeholders* e a definição de como será feito o trabalho da equipe. (DIAS, 2005)

Na segunda fase, a Especulação, uma identificação de requisitos iniciais para o produto é feita, definição das atividades, um plano de projeto é desenvolvido com entregas, marcos, iterações, cronograma de trabalho e alocação de recursos. Uma incorporação de estratégias é apresentada com o intuito de mitigar riscos, estimativas de custos e outras informações financeiras que podem ser necessárias. (DIAS, 2005)

A fase de Exploração é a fase em que os produtos planejados são entregues sob a forma de incrementos de funcionalidades e são divididos entre os ciclos do projeto. (DIAS, 2005)

Na adaptação é feito um estudo para se descobrir as melhores

práticas. Uma revisão do resultado entregue é feita inicialmente e, por consequência, uma análise da situação e avaliação do desempenho dos membros da equipe nessa iteração do projeto. Ela compara o resultado realizado com o esperado. (DIAS, 2005)

O Encerramento faz uma finalização das atividades do projeto e uma transferência dos aprendizados. (DIAS, 2005)

A fase de Visão é a primeira, enquanto que as fases de Especulação, Exploração e Adaptação alternam-se a cada iteração para o refinamento do projeto. Algumas vezes pode ser necessário voltar à fase de Visão quando as modificações são de extrema importância e podem alterar significativamente o produto. Uma vez obtido o produto final, o Encerramento do projeto é feito. (DIAS, 2005)

A FIG. 8 ilustra o fluxo geral de um projeto ágil. É um gráfico que compara o nível de atividade com os fluxos de um projeto ágil. Ele inicia na fase de planejamento onde o número de atividades ainda é alto pois nada foi desenvolvido ainda. Esse planejamento vai diminuindo com o passar do tempo e o gráfico demonstra isso. Após o início são apresentadas as iterações com os incrementos de funcionalidade, as mudanças no escopo e os aceites do cliente pelos itens desenvolvidos na iteração. Ao final, esse controle diminui, pois chega o final do projeto. A FIG. 9 apresenta as fases de um projeto baseado na agilidade. A fase inicial é a visão, onde a equipe do projeto é escolhida e uma visão geral sobre o que será desenvolvido. A segunda fase é a especulação, nela os itens da lista de funcionalidades que serão desenvolvidos são escolhidas e a seguir entra na fase de exploração. As funcionalidades são implementadas e, após isso, chega na fase de adaptação. Nessa fase, as mudanças solicitadas pelo cliente são estudadas e vão para a lista de funcionalidades ou o projeto vai para a fase final, que é a de encerramento.

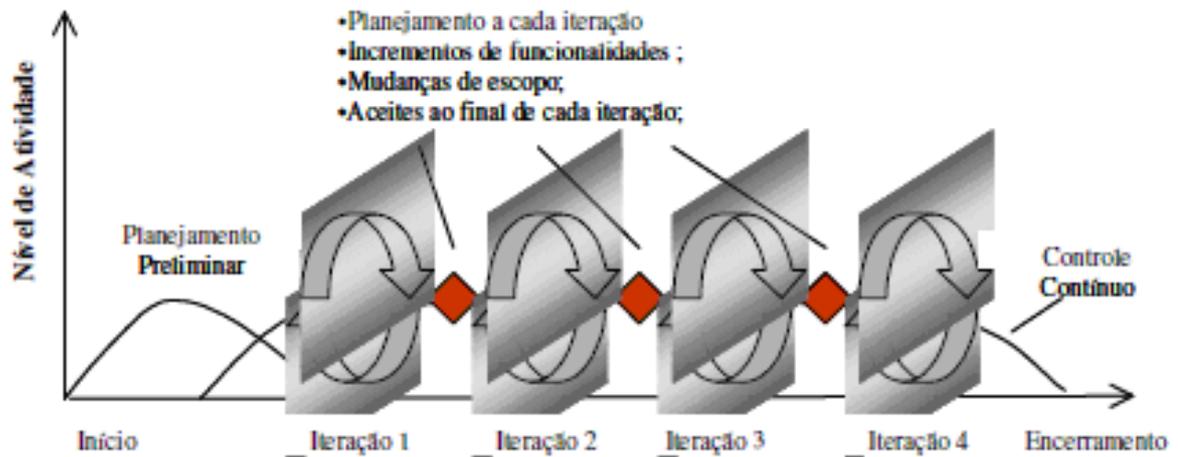


Figura 8 - Fluxo geral de um projeto ágil

Fonte: DIAS (2005) *apud* KOPPENSTEINER (2003)

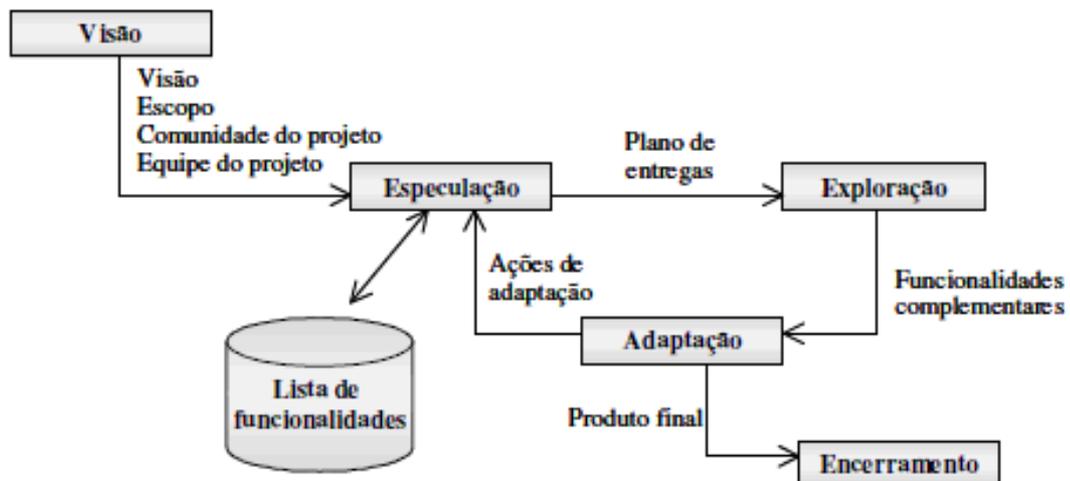


Figura 9 - Fases do Agile Project Management

Fonte: DIAS (2005) *apud* HIGHSMITH (2004)

### 3.3 Estudo de caso

A empresa A é uma fábrica de desenvolvimento de software e, trabalha com o gerenciamento de projetos clássicos. Após uma pesquisa da área de qualidade sobre metodologias ágeis, a empresa decidiu implantar o *Scrum* em seus projetos.

O primeiro projeto a se basear nessa metodologia foi um software para

modelagem, execução e otimização de processos orientado em *BPM*(*Business Process Management*). A empresa determinou que o *Sprint* teria uma duração de dez dias úteis e as demais características seriam baseadas no padrão.

A princípio, o dono da empresa resolveu investir nesse software a princípio, apenas para os processos internos da empresa, isto é não seria vendido. Ele próprio seria o *Product Owner*, o gerente de projetos dos membros da equipe foi nomeado *Scrum Master* e o time era composto por cinco desenvolvedores.

O *Product Owner* elaborou uma lista com itens de *Backlog*, e agendou-se a primeira reunião com o time. A reunião chamada de *Sprint Planning 1*, aconteceria entre os membros da equipe, *Scrum Master* e *Product Owner* com o intuito de o time entender o que deve ser feito e decidir a estimativa de cada história proposta no *Sprint Backlog*. Após essa reunião, somente a equipe se reúne para a chamada *Sprint Planning 2*. Nessa reunião, são discutidos aspectos técnicos de como as histórias serão desenvolvidas e também são quebradas em atividades menores.

Durante o *Sprint*, diariamente são feitas as *Daily Scrum*. Nela os membros da equipe mantêm-se informados sobre o andamento geral das histórias. Ao término dessa reunião, o *Scrum Master* atualiza o *Burndown Chart*.

No início da implantação, os membros da equipe ainda estavam se adaptando a essa nova forma de desenvolvimento e aperfeiçoando a sua produtividade (número de histórias no *Sprint*), sempre sobravam ou faltavam histórias. Aos poucos foram se ajustando até conseguir manter um número mais ou menos certo de histórias.

Nessa fase de adaptação o time ainda não era auto-gerenciável, sendo necessário o papel de um gerente de projetos para orientá-los. Na medida em que o time conseguiu alcançar essa auto-gestão, o papel do gerente de projetos tornou-se apenas um acompanhamento. Após alguns *Sprints*, um dos desenvolvedores tornou-se o *Scrum Master*, o gerente de projetos passou a ser apenas um gerente dos membros da equipe, ou seja, desempenhando um papel mais organizacional.

O desenvolvimento do produto durou vinte e oito *Sprints*, sendo desses dezoito com entregas no tempo correto de todas as funcionalidades.

O produto foi usado internamente durante um tempo e com isso, houve a identificação e correção de alguns *bugs*. Após esse período, o software apresentava mais estabilidade e o setor comercial começou a vendê-lo no mercado.

A primeira empresa a se interessar pelo software foi uma multinacional B de perfuração de petróleo. Os donos da empresa ainda não conheciam o *Scrum* e aceitaram segui-lo, mas com algumas ressalvas, seria uma adaptação nas metodologias ágeis e também na gestão de projetos. A forma de desenvolvimento é baseada no *Scrum*, ou seja, em pequenas iterações com até dez dias úteis. Existe uma lista de *Backlog* dos requisitos dos clientes, e as reuniões todas do *Sprint*. A princípio o custo seria fixo, eles pagariam por um número determinado de *Sprints* e teriam mais informações financeiras que pudessem ser necessárias. Foram escolhidos dois *Product Owners*, um no cliente e o outro interno, o *Scrum Team* e o *Scrum Master* foram os mesmos.

Em um momento inicial foi passado o custo de seis *Sprints*. Esses foram realizados mas a empresa resolveu implantar mais algumas funcionalidades que eram necessárias para o desenvolvimento do processo e, por fim acabaram contratando vinte *Sprints*.

O projeto foi realizado com sucesso, as dificuldades encontradas durante o desenvolvimento foram relativas aos problemas que a empresa teve para entender o como funciona a agilidade. Queriam mudar os requisitos quase todos os dias e esperavam receber pronta uma solicitação de um requisito no mesmo dia que a faziam. Dos vinte *Sprints* comprados, seis não foram entregues conforme o planejado, sendo três cancelados antes do término. Isso ocorreu devido a uma mudança nos requisitos e os outros três por impedimentos da empresa fizeram com que atrasasse a entrega. Nesse projeto, o papel do gerente de projetos foi fundamental para as negociações de prazo e liberação de recursos com o cliente, uma vez que não foi possível o cliente estar presente na empresa durante o projeto, pois estava em São Paulo e a empresa em Belo Horizonte.

## CONCLUSÃO

As duas abordagens de gerenciamento de projetos descritas neste estudo apresentam aspectos positivos e aspectos negativos.

O gerenciamento de projetos tradicional requer forte planejamento e muita disciplina, é baseado em grande montante de documentações. Portanto, deve ser usado para projetos mais estáveis, que estejam bem definidos e não tenham a possibilidade de haver muitas mudanças. Para início do projeto, um documento deve ser formalizado como um aceite para iniciar o desenvolvimento.

O gerenciamento de projetos ágil é mais adequado para projetos em que as mudanças de requisitos são constantes. é necessário muita criatividade e o cliente deve estar sempre presente. Esse é um grande dificultador da implantação dessas metodologias, pois com a globalização e o avanço das tecnologias, é muito comum o cliente estar distante da empresa contratada. Essa forma de gerenciamento apresenta documentos informais do projeto que devem estar visíveis a todos os interessados no projeto.

Este estudo proporcionou um comparativo entre essas metodologias de gerenciamento. Pode-se concluir que a metodologia correta vai variar de acordo com o perfil da empresa e com o trabalho que deve ser feito. Para o sucesso das metodologias ágeis, é necessário o comprometimento do time, enquanto que para o tradicional é necessário o comprometimento do gerente de projetos. A agilidade apresenta vantagens, pois possibilita maximizar produtividade, as entregas são rápidas, existe flexibilidade, simplicidade, favorecem o envolvimento direto entre os interessados e o cliente sempre recebe novas versões do software diminuindo sua ansiedade. As desvantagens ocorrem pois nem sempre é possível manter um contato constante com o cliente, o time deve estar totalmente comprometido e procurar se auto-gerenciar, o cliente pode não entender muito bem como funcionam, no estudo de caso mesmo aconteceu isso.

Por outro lado, a documentação da metodologia tradicional ajuda as pessoas a entender melhor o funcionamento do software, e ao contrário da agilidade, são programados para projetos grandes e com grandes equipes. Como desvantagem, o gerenciamento tradicional apresenta o alto índice de fracasso, os clientes só veem o resultado no final do projeto, o que os deixa muito ansiosos e a

falta de entendimento do que o cliente realmente quer.

O gerente de projetos no início do projeto é importante para a evolução do time, mas a medida que os membros vão se tornando auto-gerenciáveis, seu papel é mais como um acompanhamento do que é executado, auxílio aos membros, um papel mais organizacional. Com o estudo realizado nesse trabalho, foi possível comprovar que o gerenciamento de projetos pode ser usado em projetos baseados em agilidade, contanto que sejam feitas adaptações em ambas metodologias. Essas adaptações e implantação vão variar de empresa para empresa de acordo com a cultura de cada uma. A implementação de uma metodologia ágil em um ambiente tradicional é bastante complicada e requer muito comprometimento de todos os envolvidos para que seja efetuada com sucesso.

## REFERÊNCIAS

AC, Vinícius. **Introdução ao Desenvolvimento Ágil**. Disponível em: <<http://www.devmedia.com.br/post-5916-1-Introducao-ao-Desenvolvimento-Agil.html>>. Acesso em 08 set. 2010.

AGILE ALLIANCE. **Manifesto Ágil**. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 08 set. 2010.

BANKI, André Luis. **Metodologias ágeis: uma visão prática**. Disponível em: <<http://www.devmedia.com.br/post-9883-Artigo-Engenharia-de-Software-4-Metodologias-Ageis.html>>. Acesso em 08 set. 2010.

BASSI, Dairton. **Experiências com Desenvolvimento Ágil**. Disponível em: <[www.teses.usp.br/teses/.../Dissertacao\\_Metodos\\_Ageis\\_Dairton\\_Bassi.pdf](http://www.teses.usp.br/teses/.../Dissertacao_Metodos_Ageis_Dairton_Bassi.pdf)>. Acesso em: 08 set. 2010.

BENASSI, João Luis Guilherme. AMARAL, Daniel Capaldo. **Gerenciamento ágil de Projetos aplicado ao desenvolvimento de produto físico**. São Paulo, 2008, v. 1. Disponível em: <<http://www.scribd.com/doc/19545669/aleckssandrotavares>>. Acesso em: 08 set. 2010.

COHN, Mike. **Uma Introdução ao Scrum**. Disponível em: <<http://www.centrosoftware.com.br/wpcontent/uploads/2009/09/IntroducaoScrum.pdf>>. Acesso em: 08 out. 2010.

DIAS, Marisa Villas Boas. **Metodologias ágeis: Métodos ágeis de desenvolvimento de software**. Disponível em: <<http://www.devmedia.com.br/post-15459-Artigo-Engenharia-de-Software-20-Metodos-Ageis-de-Desenvolvimento-de-Software.html>>. Acesso em 15 set. 2010.

DIAS, Marisa V. B.; SOLER, Alonso M. **Agile Project Management: Um Novo Enfoque para o Gerenciamento de Projetos de Desenvolvimento de Sistemas de Tecnologia de Informação**. Disponível em: <<http://j2da.tempsite.ws/arquivos/Agile%20Project%20Management.pdf>>. Acesso em 08 out. 2010.

\_\_\_\_\_. **Gerenciamento de projetos**. Disponível em: <http://www.pmi.org.br>. Acesso em: 15 set, 2010.

GOMES, André Farias. **Desenvolvendo com agilidade:** Explorando as metodologias e práticas ágeis. Disponível em: <<http://www.devmedia.com.br/post-12849-Artigo-Java-Magazine-68-Desenvolvendo-com-Agilidade.html>>. Acesso em 15 set. 2010.

OLIVEIRA, Luiz Cláudio Vieira de; CORRÊA, Osvaldo Manoel. **Normas para redações de trabalhos acadêmicos, dissertações e teses.** Belo Horizonte: i-neti, 2008. 135 p.

PROJECT MANAGEMENT INSTITUTE. **A Guide to the Project Management Body of Knowledge: (Pmbok Guide).** 2008. USA: ANSI. 459 p.

PROMON. **Gerenciamento de projetos.** Disponível em: <[http://www.promon.com.br/portugues/noticias/download/PBTR%20GE\\_para%20web.pdf](http://www.promon.com.br/portugues/noticias/download/PBTR%20GE_para%20web.pdf)>. Acesso em: 15 set, 2010.

SCHWABER, K. **Agile Project Management with Scrum.** Cidade de edição:Microsoft Press, 2004.

SCHNEIDER, Guilherme. **Gerência de projetos:** o que é o processo Scrum. Disponível em: <<http://webinsider.uol.com.br/2008/08/22/gerencia-de-projetos-o-que-e-o-processo-scrum>>. Acesso em 06 ago. 2010.

SIVSimped. **Gerenciamento ágil de projetos aplicado ao desenvolvimento de produto físico.** São Paulo: 2007. Disponível em: <<http://www.scribd.com/doc/32088851/Gerenciamento-Agil-de-Projetos-aplicado-ao-desenvolvimento-de-produto-fisico>>. Acesso em 15 set, 2010.

SLIGER, Michele; BRODERICK, Stacia. **The Software Project Manager`s Bridge to Agility.** Cidade de edição: Addison Wesley, 2008.

TAVARES, Alecsandro. **Gerência de Projetos e Scrum:** um estudo de caso. Gravataí, 2008, v. 1. Disponível em: <<http://www.scribd.com/doc/19545669/aleckssandrotavares>>. Acesso em: 08 set. 2010.

TERTO, Alexsandro. **Relatório do Caos.** Disponível em: <<http://blogs.abril.com.br/alexsandro-terto>>. Acesso em: 09 set. 2010.

VARGAS, Ricardo. Manual **Prático do plano de projeto:** utilizando PMBOK GUIDE.

3. ed. Rio de Janeiro: Brasport, 2007. 230p.

VIEIRA, Marconi Fábio. **Gerenciamento de projetos de Tecnologia da Informação**. São Paulo: Campus, 2003. 400p.

WARDEN, Shane; SHORE, James. **The Art of Agile Development**. Cidade de edição:O'Reilly, 2007.

WILLI, Renato. **PMBOK e Agile... pode?** Disponível em:  
<<http://visaoagil.wordpress.com/2008/08/01/pmbok-e-agile-pode/>>. Acesso em: 08 nov. 2010.