

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE EDUCAÇÃO CONTINUADA
MBA EM ADMINISTRAÇÃO DE TI**

RODRIGO LEAL DA CUNHA

**GESTÃO ÁGIL DE PROJETOS: TRANSIÇÃO DO MÉTODO TRADICIONAL PARA
MÉTODOS ÁGEIS.**

Porto Alegre

2015

RODRIGO LEAL DA CUNHA

GESTÃO ÁGIL DE PROJETOS: TRANSIÇÃO DO MÉTODO TRADICIONAL PARA
MÉTODOS ÁGEIS.

Trabalho de Conclusão de Curso
apresentado como requisito parcial para
obtenção do título de Especialista em
Administração de TI, pelo MBA em
Administração de TI da Universidade do
Vale do Rio dos Sinos – UNISINOS

Orientador: Prof. Me Edson Wobeto

Porto Alegre

2015

Dedico este trabalho ao meu pai, Paulo Renato Dias da Cunha, que não mediu esforços para que eu chegasse nesta etapa de minha vida. Foi uma longa jornada, mas de uma forma especial sempre me deu força e incentivo para continuar a minha trajetória. Hoje você não está mais ao meu lado, mas tenho certeza que, de alguma forma, compartilha comigo a felicidade que sinto.

AGRADECIMENTOS

Para que fosse possível concretizar este trabalho, contei com o apoio, incentivo e carinho da minha esposa, Diane dos Santos Pacheco da Cunha, que de uma forma especial esteve ao meu lado.

Agradeço ao professor Edson Wobeto pelas orientações e apoio durante esta pesquisa, seu auxílio foi fundamental para a conclusão deste trabalho.

Aos meus colegas de trabalho que me acompanharam e contribuíram para a realização deste estudo de caso.

RESUMO

Este trabalho tem como objetivo avaliar a viabilidade de se implantar gerenciamento ágil para a gestão de projetos. Num contexto atual onde cada vez mais as organizações são competitivas, os clientes mais exigentes com relação a qualidade dos produtos disponibilizados no mercado, é fundamental que as organizações adéqüem seus processos para atender estas necessidades. Esta pesquisa insere-se dentro da área de TI de uma organização que possui enraizada o método tradicional para gerenciamento de projetos, porém ao longo do tempo não vem obtendo bons resultados. Ao longo deste trabalho serão realizados estudos, através das pesquisas bibliográficas, a cerca das principais metodologias ágeis e boas práticas sugeridas pelos autores, bem como conceituação do modelo tradicional. Com a realização das entrevistas, foi possível entender os processos do modelo atual e identificar os pontos de melhoria que a metodologia ágil pode vir a suprir. Ainda, considerando os aspectos da empresa estudada, o autor deste trabalho sugere a adoção de um modelo híbrido, com embasamento nos autores bibliográficos, onde há uma combinação do método ágil com o tradicional, reunindo assim os aspectos positivos de cada metodologia.

Palavras-chave: Metodologias Ágeis. *Scrum*. Tradicional. Projetos.

LISTA DE FIGURAS

Figura 1 – Modelo em Cascata	19
Figura 2 – Ciclo de vida do projeto.....	22
Figura 3 – Princípios do manifesto ágil	32
Figura 4 – O modelo Cynefin.....	37
Figura 5 – Ciclo do <i>Scrum</i>	41
Figura 6 – Estágios de Tuckman.....	42
Figura 7 – Artefatos do <i>Scrum</i>	44
Figura 8 – Itens do Backlog.....	45
Figura 9 – Fluxo do item.....	48
Figura 10 – Eventos do <i>Scrum</i>	49
Figura 11 – <i>Sprint</i>	51
Figura 12 – Quadro de Tarefas	54
Figura 13 – Quadro de Tarefas do <i>Kanban</i>	56
Figura 14 - Organograma da Empresa.....	73

LISTA DE TABELAS

Tabela 1 – Contextos do modelo de <i>Cynefin</i>	38
--	----

LISTA DE SIGLAS

EAP	Estrutura Analítica do Projeto
PMBOK	<i>Project Management Body of Knowledge</i>
PMI	<i>Project Management Institute</i>
PO	<i>Product Owner</i>
SGD	Sistema de Gestão de Demandas
TI	Tecnologia da Informação
WIP	<i>Work In Progress</i>
XP	<i>Extreme Programming</i>

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Situação Problemática e Pergunta de Pesquisa	11
1.2 Objetivos	16
1.2.1 Objetivo Geral	16
1.2.2 Objetivos Específicos	16
1.3 Justificativa	16
2 FUNDAMENTAÇÃO TEÓRICA.....	18
2.1 Método Tradicional de Gestão de Projetos	18
2.1.1 Ciclo de vida.....	21
2.1.2 Gerenciamento da Integração do Projeto.....	22
2.1.3 Gerenciamento do Escopo do Projeto.....	24
2.1.4 Gerenciamento do Tempo do Projeto.....	25
2.1.5 Gerenciamento dos Custos do Projeto.....	26
2.1.6 Gerenciamento dos Recursos Humanos do Projeto	27
2.1.7 Gerenciamento das Comunicações do Projeto	27
2.1.8 Gerenciamento dos Riscos do Projeto	28
2.1.9 Gerenciamento das Aquisições do Projeto.....	29
2.1.10 Gerenciamento das Partes Interessadas do Projeto	29
2.2 Métodos Ágeis.....	30
2.2.1 O Manifesto Ágil	31
2.2.2 Benefícios dos Métodos Ágeis	34
2.3 <i>Scrum</i>	36
2.3.1 Introdução	36
2.3.2 <i>Time Scrum</i>	41
2.3.3 Artefatos do <i>Scrum</i>	43
2.3.3.1 <i>Product Backlog</i>	44
2.3.3.2 <i>Sprint Backlog</i>	46
2.3.3.3 <i>Definição de Pronto</i>	46
2.3.3.4 Incremento do Produto	47
2.3.3.5 Metas de Negócio, Gráficos de Acompanhamento e Definição de Preparado.....	47
2.3.4 Eventos do <i>Scrum</i>	49
2.3.4.1 <i>Sprint</i>	50

2.3.4.2 <i>Sprint Planning</i>	51
2.3.4.3 <i>Daily Scrum</i>	52
2.3.4.4 <i>Sprint Review</i>	53
2.3.4.5 <i>Sprint Retrospective</i>	54
2.4 <i>Kanban</i>	55
2.5 <i>Extreme Programming</i>	58
2.5.1 Comunicação.....	59
2.5.2 Coragem.....	60
2.5.3 <i>Feedback</i>	60
2.5.4 Simplicidade.....	61
2.5.5 Práticas.....	61
2.5.5.1 Cliente Presente.....	61
2.5.5.2 Jogo do Planejamento.....	62
2.5.5.3 <i>Stand up Meeting</i>	62
2.5.5.4 Programação em par.....	63
2.5.5.5 Desenvolvimento guiado pelos testes.....	63
2.5.5.6 Refatoração.....	63
2.5.5.7 Código coletivo.....	64
2.5.5.8 Código padronizado.....	64
2.5.5.9 <i>Design</i> simples.....	64
2.5.5.10 Metáfora.....	65
2.5.5.11 Ritmo sustentável.....	65
2.5.5.12 Integração contínua.....	66
2.5.5.13 <i>Releases</i> curtos.....	66
2.6 Indicadores de desempenho para projetos ágeis.....	67
2.6.1 Tipos de indicadores.....	67
3 MÉTODOS E PROCEDIMENTOS.....	70
3.1 Delineamento da Pesquisa.....	70
3.2 Definição da Unidade-Caso e dos Sujeitos da Pesquisa.....	72
3.3 Técnicas de Coleta de Dados.....	73
3.4 Técnica de Análise de Dados.....	75
3.5 Limitações do Método.....	75
4 APRESENTAÇÃO E ANÁLISE DOS DADOS.....	77
4.1 Situação da empresa e entrevistas.....	78

4.2 Principais metodologias para gerenciamento ágil de projetos.....	81
4.3 Indicadores de desempenho para gerenciamento ágil de projetos	82
4.4 Métodos ágeis para gerenciamento de <i>softwares</i>	83
5 CONSIDERAÇÕES FINAIS	88
REFERÊNCIAS.....	90
APÊNDICE A – QUESTIONÁRIOS DE ENTREVISTA	93

1 INTRODUÇÃO

É fato que cada vez mais as organizações tem se preocupado em adotar novas metodologias para gerenciamento de projetos, a fim de obter maiores ganhos para o negócio (VERSIONE, 2013). Com base neste contexto, para Gomes (2013) métodos ágeis podem ser aplicados para diferentes tipos de projetos e necessidades, os quais auxiliam as organizações a obterem uma resposta mais rapidamente frente as necessidades que o mercado as impõe.

No capítulo seguinte, será apresentada a situação problemática e pergunta de pesquisa para uma organização que utiliza método tradicional para gerenciamento de projetos. Ainda neste capítulo são apresentados os objetivos que se pretende atingir, os quais se subdividem em objetivos gerais e específicos, e a justificativa da escolha deste assunto como objeto de pesquisa.

1.1 Situação Problemática e Pergunta de Pesquisa

Na atual conjuntura do ambiente de desenvolvimento de *softwares* (programas), cada vez mais as empresas estão buscando novas alternativas para alavancar seus negócios e mitigação de riscos. Gerentes de projetos são fortemente cobrados para atingir as metas traçadas dentro do orçamento estipulado, no prazo e qualidade acordados com o cliente final (GOMES, 2013).

A metodologia tradicional atende as necessidades em uma determinada característica de projeto, para Pressman (2006) este método é uma abordagem linear e seqüencial de todas as atividades envolvidas no processo de desenvolvimento do produto. Na medida em que o mercado exige entregas com mais freqüência e qualidade, novas opções frente a este modelo têm surgido.

Como alternativa ao gerenciamento de projetos tradicional, as empresas têm focado no gerenciamento ágil de projetos. Amaral (et al..2011, p 21) definem gerenciamento ágil de projetos da seguinte forma:

O gerenciamento ágil de projetos é uma abordagem fundamentada em um conjunto de princípios, cujo objetivo é tornar o processo de gerenciamento de projetos mais simples, flexível e iterativo, de forma a obter melhores resultados em desempenho (tempo, custo e qualidade), menor esforço em gerenciamento e maiores níveis de inovação e agregação de valor ao cliente.

No contexto relatado acima está situado um banco privado caracterizado por ter uma grande variedade de produtos e serviços com foco no agronegócio. Para manter a privacidade da empresa analisada, será utilizado o nome fictício empresa X.

Tendo como base mais de 2 milhões de clientes, possui um portfólio de serviços voltado para todos os tipos de pessoas que desejam ter um relacionamento bancário, porém tem na zona rural a sua maior rede de clientes e estratégia de negócio. Ao longo dos anos a empresa X tem percebido seus concorrentes também investirem cada vez mais seus recursos no meio rural, porém graças ao conhecimento adquirido ao longo das décadas, tem sua marca consolidada a nível nacional e atualmente ainda é considerada uma das principais referências no ramo em que atua.

Todavia, no decorrer dos últimos anos, a margem de lucro do banco tem reduzido significativamente. O volume anual de novos clientes está em uma curva descendente, resultando em baixa de 10% a cada período. Segundo os principais meios de comunicação, inclusive em amostragens elaboradas pela própria empresa, a marca da empresa X tem perdido visibilidade e espaço no mercado.

Com o objetivo de identificar as causas do baixo rendimento dos negócios, a empresa X constatou que seus concorrentes têm desenvolvido novos serviços com qualidade e velocidade maior que a sua. Entendendo estes serem os principais motivadores para o baixo desempenho, a área de metodologia da Tecnologia da Informação (TI) do banco irá reestruturar sua fábrica de projetos, onde pretende analisar a substituição do modelo de gerenciamento de projetos tradicional pelo ágil.

Referente ao modelo atual, o tradicional, a entrega do serviço somente é percebido pelo seu cliente após a finalização do projeto, resultando assim em prejuízo no lançamento de novos produtos, uma vez que a concorrência está a sua frente. Desta forma, em um ambiente competitivo e diversificado como o ramo bancário, a empresa avaliada neste trabalho entende que estas entregas de serviços que geram valor ao cliente devem ocorrer numa velocidade maior, onde seja possível perceber rapidamente novas funcionalidades desenvolvidas e com acréscimo de qualidade.

Na organização existem duas áreas: a de negócios e a de sistemas. A necessidade da criação de um novo projeto parte da área de negócios, a qual tem a responsabilidade de definir as necessidades da organização e escalonar quais

demandas devem ser entregues em uma fila de prioridades. Neste primeiro momento, as informações da demanda são elaboradas em uma visão macro, onde se entende qual é o produto fim que o desenvolvimento deve entregar. Na área de sistemas está alocada a equipe de desenvolvimento, os quais elaboram um detalhamento técnico do que deverá ser criado com base no entendimento da necessidade.

O ponto de partida para iniciar o ciclo de vida do projeto, que é elaborado pelas áreas de sistemas, é chamado de documento de visão, o qual irá detalhar tecnicamente o que se deseja desenvolver. Neste relacionamento, entende-se que a área de negócios é cliente interno da área de sistemas.

Na metodologia de desenvolvimento atual está previsto que o ciclo de vida dos projetos segue as mesmas fases do modelo tradicional. Neste direcionamento, Gido e Clements (2007, p. 1) afirmam que,

A primeira fase do ciclo de vida do projeto envolve a identificação de uma necessidade, problema ou oportunidade, que pode resultar na solicitação de propostas pelo cliente a pessoas, uma equipe de projeto ou organização (fornecedores) para atender à necessidade identificada ou resolver o problema. A segunda fase do ciclo de vida do projeto é o desenvolvimento de uma solução proposta para a necessidade ou problema. Essa fase resulta na entrega de uma proposta ao cliente por uma ou mais pessoas ou organizações. A terceira fase desse ciclo de vida é a implementação ou solução proposta. Essa fase, conhecida como execução, resulta na consecução dos objetivos, deixando o cliente satisfeito com a conclusão do escopo total do trabalho com qualidade, sem superar o orçamento e dentro do prazo. A fase final do ciclo de vida do projeto é sua conclusão.

No contexto da empresa desta pesquisa, é de responsabilidade da área de sistemas cadastrar no Sistema de Gestão de Demandas (SGD) a demanda que será desenvolvida, a qual será classificada em um dos seguintes tipos:

- 1. Projeto:** refere-se a novas demandas que visam atender uma necessidade da área de Negócios.
- 2. Melhoria:** aprimoramento de um projeto já existente em que será acrescido um novo escopo de desenvolvimento.

3. **Legal:** demanda obrigatória que visa atender determinada legislação, possui prioridade máxima frente os demais projetos.

Na medida em que o projeto é desenvolvido, as etapas no SGD são avançadas seqüencialmente até que ocorra o seu encerramento e entrega das funcionalidades criadas.

A utilização da metodologia Cascata¹, nos projetos de desenvolvimento de sistemas, inicialmente atendia as necessidades da organização, onde pequenas demandas eram criadas. Com o passar do tempo e aumento do porte da empresa, este modelo de desenvolvimento têm apresentado diversos problemas, tais como:

1. Os projetos tendem a sofrer mudanças ao longo de sua construção, embora que este modelo seqüencial preveja iterações, isto ocorra poucas vezes. Com o afastamento da equipe de projeto da área de negócios, a entrega final pode não ser a esperada.
2. Muito dificilmente o cliente sabe informar exatamente as suas necessidades. O modelo cascata tem dificuldade para lidar com as incertezas no início do projeto.
3. As entregas realizadas neste modelo são feitas próximo ao final do projeto, a percepção do cliente acaba sendo negativa em virtude da demora.
4. Uma versão do sistema entregue pelo time do projeto para o cliente somente ocorrerá quando o projeto estiver próximo de encerrar. Em caso de identificação de um erro grave nas etapas iniciais, tal como especificação mal entendida, demandará todo um retrabalho para o projeto corrigir o problema.
5. Como o modelo de desenvolvimento é seqüencial, por vezes ocorre de um membro da equipe permanecer ocioso pelo fato de estar aguardando

¹ Cascata é uma abordagem linear e seqüencial de todas as atividades envolvidas no desenvolvimento de *software* tradicional.

que outro complete suas tarefas para que ele possa dar prosseguimento no trabalho.

Dado este cenário, a empresa x passou a avaliar o desenvolvimento ágil de *software* como uma alternativa frente aos problemas mapeados. Gomes (2013) afirma que:

Uma das maiores motivações para a transição para métodos são os benefícios que são trazidos para a organização devido ao valor que é agregado ao cliente com qualidade e velocidade. Métodos ágeis ajudam organizações a responder mais rapidamente às necessidades do mercado, muitas vezes, resultando em grande vantagem competitiva (Gomes, 2013, p. 10).

Contudo, frente aos diversos métodos disponíveis na literatura, a empresa tem encontrado dificuldade para reestruturar seus processos de desenvolvimento de *software*, visto que ainda não identificou qual método ágil melhor se adapta para o seu contexto. A empresa vem de uma cultura de desenvolvimento tradicional, onde seus processos são engessados e burocráticos.

Na atual conjuntura, a empresa x não lança novos produtos na velocidade que o mercado exige e o padrão de qualidade dos seus produtos não tem atendido as expectativas da área de negócios. Através das análises realizadas em relatórios de desempenho dos projetos, é possível identificar que uma grande parte dos projetos abrem itens de garantia pós implantação. Caracteriza-se como item de garantia toda falha que é gerada quando o produto está em produção, o qual resulta em impacto para o negócio, seja prejuízo financeiro ou de imagem.

Pelo fato de o método atual não suprir todas as expectativas, a área de metodologia da TI entende que a mudança para métodos ágeis é necessária e, se bem implantada, será benéfica para todo o sistema, alavancando assim novos clientes. Os motivadores para esta mudança são claros e bem definidos, porém dada as diversas opções disponíveis, a identificação de qual a melhor metodologia para gerenciamento de projetos ainda é uma incógnita.

A dificuldade que a empresa está passando afeta diretamente o departamento de TI, o qual é responsável por desenvolver os novos serviços que serão entregues para os clientes. Diante do exposto, a questão de pesquisa para este trabalho é: **é possível implantar metodologia ágil no gerenciamento de projetos de *software* na empresa estudada?**

1.2 Objetivos

Para atender a questão da pesquisa formulada, serão apresentados o objetivo geral e objetivos específicos deste trabalho.

1.2.1 Objetivo Geral

Identificar qual é a metodologia de gerenciamento projetos de *software* que melhor atende as necessidades da empresa x.

1.2.2 Objetivos Específicos

Como objetivos específicos, este trabalho visa:

- a) Identificar o processo atual e os principais problemas apresentados no gerenciamento de projetos.
- b) Descrever quais os principais modelos de metodologias ágeis para gerenciamento de projetos.
- c) Identificar os indicadores de desempenho mais utilizados para gerenciamento de projetos ágeis.
- d) Estudar e propor uma metodologia para gerenciamento ágil de projetos ou sua combinação com a metodologia tradicional para obter agilidade no desenvolvimento de projetos.

1.3 Justificativa

Este trabalho justifica-se pelo atual momento vivenciado na organização, onde é oportuno para que se possa trabalhar uma forma diferente e melhor de se fazer gestão de projetos, onde seja possível contribuir de forma considerável para o

seu desenvolvimento. Quando concluída, esta pesquisa dará um importante suporte para auxiliar a empresa a entregar produtos com mais agilidade e qualidade, atendendo assim as expectativas dos clientes.

Tendo em vista que muitas empresas utilizam metodologia tradicional para gerenciamento de projetos, é muito provável que as dificuldades e incertezas que a empresa x tem passado sejam as mesmas vivenciadas por outras organizações.

Além disto, o trabalho justifica-se pelo fato do pesquisador trabalhar na área de TI da empresa avaliada e perceber que nesta área existem muitas dúvidas a respeito de qual prática ágil irá melhor solucionar os problemas identificados.

Com base na reestruturação da engenharia de *software* que está sendo conduzida na empresa x e também pelo escasso material literário disponível sobre o assunto, é oportuno o momento também para abordar como se trata a Gestão de Indicadores de desempenho para o novo modelo que será escolhido, bem como avaliar quais informações poderão ser mensuráveis. Quando esta pesquisa estiver concluída, será possível auxiliar a empresa a medir o que está sendo executado e avaliar se as metas estipuladas pela organização estão sendo atingidas dentro dos limites acordados.

Pelo fato de a empresa x ser uma das partes interessadas deste trabalho, para que fosse possível viabilizá-lo, a empresa permitiu acesso parcial em sua base de dados, tendo em vista que muitas das informações são classificadas como confidenciais por possuírem acordos sigilosos entre ela e seus fornecedores de TI.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo abordar as metodologias ágeis para gerenciamento de projetos e detalhar as diferentes práticas de cada método.

Esta fundamentação se faz necessária para que seja possível embasar o estudo de caso desta pesquisa, afim de que seja possível nortear a empresa analisada a respeito de qual a melhor prática a ser adotada para o seu contexto.

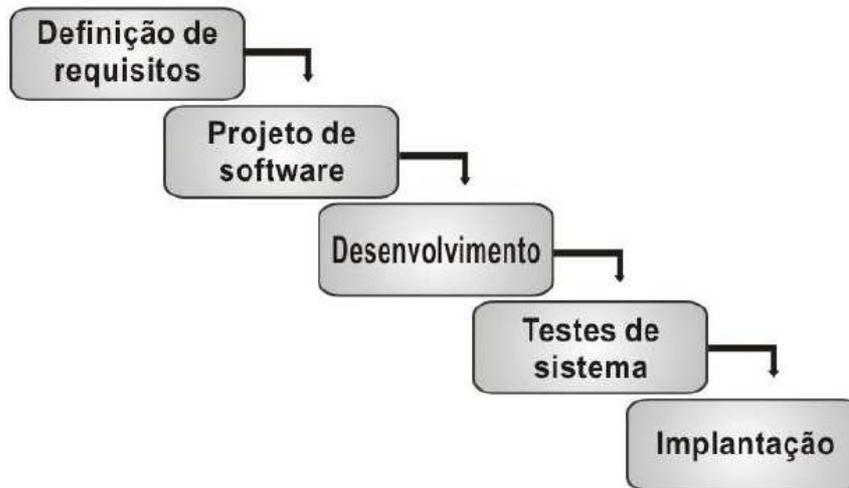
2.1 Método Tradicional de Gestão de Projetos

Um projeto é um esforço temporário, um objetivo específico através de um único conjunto de tarefas inter-relacionadas e com a aplicação eficaz dos recursos. Possui objetivo bem definido relacionado a escopo, cronograma e custo. O projeto nasce quando uma necessidade é identificada pela organização ou pessoas dispostas a fornecer recurso financeiro (GIDO; CLEMENTS, 2007).

A maioria dos projetos segue um padrão de desenvolvimento onde são consumidos menos recursos no início, mais recursos nas fases intermediárias e menos recursos na fase final. O modelo cascata, ou também conhecido como tradicional, é visto como um fluir constante para frente através de fases seqüenciais e distintas. Os objetivos são definidos no início do projeto e somente podem ser alterados mediante a um controle formal (MOURA, 2013).

Segundo Pressman (2006), o modelo em cascata foi o primeiro paradigma de desenvolvimento criado pela Engenharia de Software, que teve sua essência retirada de outras áreas da Engenharia. Ainda para Pressman (2006) o modelo em cascata sugere uma abordagem linear e seqüencial de todas as atividades envolvidas no desenvolvimento de *software*, e por se tratar de uma seqüência de eventos, esse modelo também ficou conhecido como “ciclo de vida” do *software*.

Figura 1 – Modelo em Cascata



Fonte: Adaptado de Pressman (2006).

- **Definição dos requisitos:** de acordo com Pfleeger (2004), os requisitos são características de algo que o sistema será capaz de realizar para atingir determinado objetivo. A equipe de desenvolvimento deve questionar o cliente sobre a forma que o sistema deve ser para satisfazer as suas necessidades, documentando esses requisitos.
- **Projeto de Software:** os requisitos são um conjunto de problemas que o sistema deve resolver para o cliente e com base nesses requisitos, a equipe de desenvolvimento deve, antes de codificar, projetar uma solução que satisfaça as necessidades do cliente (PFLEEGER, 2004).
- **Desenvolvimento:** de acordo com Pressman (2006), a atividade de desenvolvimento consiste em transformar, através de técnicas de programação, o projeto de *software* em um produto executável e operacional.
- **Testes de sistema:** ainda segundo Pressman (2006), os testes são um processo de execução de um programa que têm como objetivo revisar a codificação implementada e encontrar erros na estrutura do *software*.

- **Implantação:** implantação significa a entrega do *software* para o cliente e a utilização do usuário final, é neste momento que o cliente tem a oportunidade avaliar se o *software* atende às suas necessidades. Caso sejam identificadas não conformidades, a equipe deve registrar as modificações necessárias para dar início à manutenção do *software* (PRESSMAN, 2006).

De forma a consolidar as melhores técnicas e ferramentas para gestão de projetos, o *Project Management Institute* (PMI) criou um guia chamado *Project Management Body of Knowledge* (PMBOK) orientado para profissionais de gerenciamento de projetos onde são reunidas as melhores práticas nesta abordagem.

De acordo com Gido e Clements (2007), os seguintes atributos ajudam na definição de projeto:

- Um projeto tem um objetivo bem definido – um resultado ou produto esperado. O objetivo de um projeto costuma ser definido em termos de escopo, cronograma e custo [...].
- Um projeto é conduzido por meio de uma série de tarefas independentes, isto é, várias tarefas não repetitivas que precisam ser cumpridas em determinada seqüência, a fim de se atingir o objetivo do projeto [...].
- Um projeto utiliza vários recursos para realizar as tarefas. Tais recursos podem incluir diferentes pessoas, organizações, equipamentos materiais e instalações [...].
- Um projeto apresenta um esquema de tempo específico, ou uma vida finita. Ele tem uma data de início e uma data na qual o objetivo deve ser atingido [...].
- Um projeto pode ser um esforço único ou de uma única vez. Alguns projetos, como conceber e construir uma estação espacial, são únicos porque nunca foram tentados [...].
- Um projeto tem um cliente. Este é a entidade que fornece os recursos financeiros necessários para realizar o projeto. Pode ser uma pessoa, organização ou um grupo de duas ou mais pessoas ou organizações [...].
- Por fim, o projeto envolve certo grau de incerteza. Antes do início de um projeto, elabora-se um plano com base em certas suposições e estimativas. (GIDO; CLEMENTS, 2007, p. 4-5).

Já o gerenciamento de projetos é a gestão que se faz sob o mesmo através de conhecimento, habilidades, ferramentas e técnicas para atingir os seus objetivos (CRUZ, 2013).

2.1.1 Ciclo de vida

O ciclo de vida de um projeto são as etapas relacionadas a esforço e tempo dedicado em cada fase. Na medida em que as etapas avançam em seu ciclo de vida, diferentes papéis predominantes ou não são utilizados.

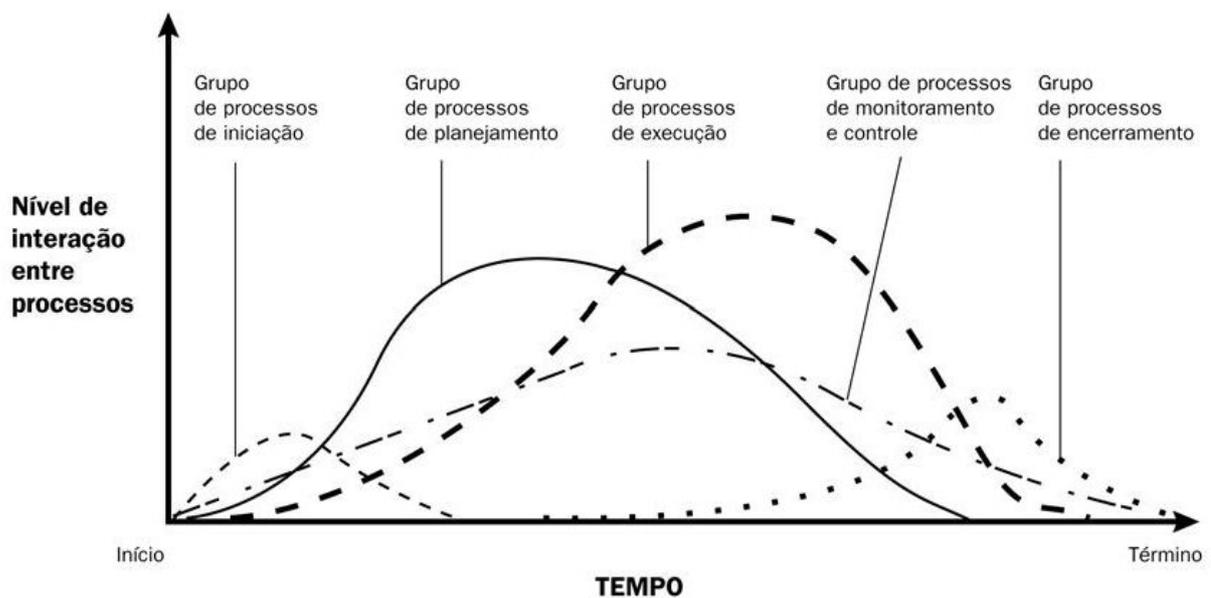
Quando pensamos em um projeto e em como vamos executá-lo, rapidamente pensamos em etapas que precisamos cumprir, uma a uma, de forma seqüencial, iterativa ou sobreposta, até terminarmos tudo que precisamos fazer. Essas etapas podem ser definidas ou moldadas de várias formas, de acordo com as necessidades de gerenciamento da organização, tendo nomes e números diferentes conforme os aspectos exclusivos de cada organização (CRUZ, 2013, p. 12-13).

De acordo com o guia PMBOK quinta edição, as seguintes fases fazem parte do ciclo de vida do projeto:

- **Iniciação:** nesta primeira fase do ciclo envolve a identificação da necessidade, problema ou oportunidade, onde um termo de abertura do projeto é elaborado para autorizar formalmente a criação do mesmo.
- **Planejamento:** esta fase reúne os processos relacionados ao planejamento do projeto. Neste documento constará o escopo, tempo e custo do projeto. Informações relevantes como quantidade de recursos e perfis envolvidos também são levados em consideração.
- **Execução:** esta fase agrupa os processos relacionados para executar o trabalho definido no plano de gerenciamento do projeto, ou seja, executar o que foi planejado na fase anterior. É neste processo que a maior parte do orçamento do projeto é gasto. Também é nesta fase que pode, se necessário, efetuar uma atualização do planejamento e mudar alguns planos previamente elaborados.

- **Monitoramento e Controle:** reúne os processos necessários para acompanhar, revisar e regular o progresso e o desempenho do projeto. Periodicamente deve mensurar o projeto para avaliar se o plano está sendo seguido conforme o planejado.
- **Encerramento:** nesta fase estão os processos de encerramento que incluem a aceitação do cliente ou patrocinador do projeto, revisão, documentação das lições aprendidas, atualização dos ativos e arquivamento de documentos relevantes.

Figura 2 – Ciclo de vida do projeto



Fonte: PMI (2013, p. 51).

2.1.2 Gerenciamento da Integração do Projeto

O gerenciamento da integração do projeto inclui os processos e atividades necessários para identificar, definir, combinar, unificar e coordenar os processos e atividades do grupo de processos do gerenciamento do projeto (PMI, 2013).

O gerenciamento da integração do projeto inclui fazer escolhas sobre alocação de recursos, concessões entre objetivos e alternativas conflitantes

e gerenciamento das dependências mútuas entre áreas de conhecimento de gerenciamento de projetos (PMI, 2013, p. 63).

O guia PMI (2013) cita que os seguintes processos fazem parte do gerenciamento da integração de projetos:

- **Desenvolver o termo de abertura do projeto:** desenvolve o documento que formalmente autoriza a existência de um projeto e dá ao gerente do projeto a autoridade necessária para aplicar recursos organizacionais às atividades do projeto.
- **Desenvolver o plano de gerenciamento do projeto:** processo de definir, preparar e coordenar todos os planos auxiliares e integrá-los a um plano de gerenciamento de projeto abrangente. As linhas de base e os planos auxiliares integrados do projeto podem ser incluídos no plano de gerenciamento do projeto.
- **Orientar e gerenciar o trabalho do projeto:** O processo de liderar e realizar o trabalho definido no plano de gerenciamento do projeto e a implementação das mudanças aprovadas para atingir os objetivos do projeto.
- **Monitorar e controlar o trabalho do projeto:** O processo de acompanhar, revisar e registrar o progresso do projeto para atender os objetivos de desempenho definidos no plano de gerenciamento do projeto.
- **Realizar o controle integrado de mudanças:** revisa todas as solicitações de mudança, aprova as mudanças e gerencia as mudanças nas entregas, ativos de processos organizacionais, documentos do projeto e no plano de gerenciamento do projeto, e comunica a decisão sobre os mesmos.

- **Encerrar o projeto ou fase:** finaliza todas as atividades de todos os grupos de processos de gerenciamento do projeto para encerrar formalmente o projeto ou a fase.

Para Gido e Clements (2007), o processo de gestão de projetos tem como significado planejar o trabalho a ser realizado e depois executar o plano. Conforme abordado no guia PMI (2013), não há uma única maneira de gerenciar projetos. São aplicados conhecimentos em gerenciamento de projetos, habilidades e processos necessários, em ordem e rigor, de forma a atingir o sucesso do projeto.

2.1.3 Gerenciamento do Escopo do Projeto

No gerenciamento do escopo do projeto são reunidos os processos necessários para garantir que todo o trabalho exigido, somente o exigido, seja completado com sucesso (PMI, 2013). Preocupa-se em definir e controlar o que está dentro do escopo e o que não deve estar.

O guia PMI (2013) define 6 processos para gerenciamento do escopo:

- **Planejar o gerenciamento do escopo:** define e documenta como o escopo será definido, validado e controlado.
- **Coletar os requisitos:** define e documenta as necessidades e requisitos das partes interessadas.
- **Definir o escopo:** desenvolve a descrição detalhada do projeto e do produto.
- **Criar a EAP (Estrutura Analítica do Projeto):** subdivide as entregas em componentes menores e fáceis de gerenciar.
- **Validar o escopo:** formaliza a aceitação das entregas concluídas do projeto.

- **Controlar o escopo:** monitora o andamento do escopo do projeto e gerencia as alterações na linha de base do escopo.

Os processos usados para gerenciar o escopo podem variar de um projeto para outro. A linha de base do escopo para o projeto é a versão já aprovada da especificação do escopo do projeto, da EAP e o dicionário EAP (PMI, 2013).

2.1.4 Gerenciamento do Tempo do Projeto

O gerenciamento do tempo do projeto inclui os processos requeridos para assegurar que o projeto termine dentro do prazo estipulado. Segundo o guia PMI (2013), os seguintes processos fazem parte do gerenciamento do tempo:

- **Planejar o gerenciamento do cronograma:** estabelece políticas, procedimentos e documentação para planejar, desenvolver, gerenciar, executar e controlar o cronograma do projeto.
- **Definir as atividades:** identifica e documenta as ações específicas para realizar a entrega do projeto.
- **Seqüenciar as atividades:** identifica a relação entre as atividades do projeto.
- **Estimar os recursos das atividades:** estima os tipos e quantidades de recursos para executar as atividades.
- **Estimar as durações das atividades:** estima a quantidade de tempo que será demandado para completar as atividades.
- **Desenvolver o cronograma:** analisar a seqüência de atividades, durações, recursos e restrições para criar o cronograma.

- **Controlar o cronograma:** controla as alterações do cronograma e efetua a atualização do seu progresso.

O guia PMI (2013) salienta ainda que algumas metodologias de desenvolvimento de cronograma ainda utilizam o mapa do caminho crítico e o método da corrente crítica.

2.1.5 Gerenciamento dos Custos do Projeto

O gerenciamento dos custos do projeto inclui os processos de planejamento, estimativas, orçamentos, financiamentos, gerenciamentos e controle dos custos, conforme abaixo:

- **Planejar o gerenciamento dos custos:** estabelece políticas, procedimentos e documentação para planejar, gerenciar, executar e controlar os custos dos projetos.
- **Estimar os custos:** estima os custos dos recursos necessários para concluir as atividades do projeto.
- **Determinar o orçamento:** agrega os custos estimados das atividades para estabelecer uma linha de base.
- **Controlar os custos:** monitora o projeto, atualiza seu orçamento e gerencia alterações na linha de base de custos.

É importante salientar que em muitas organizações a análise do desempenho financeiro é realizada fora do projeto. Em outras, o gerenciamento dos custos do projeto pode incluir esse trabalho (PMI, 2013).

2.1.6 Gerenciamento dos Recursos Humanos do Projeto

Este gerenciamento inclui os processos responsáveis por organizar e gerenciar a equipe do projeto, onde os membros podem também ser referidos como pessoal do projeto. Embora que os papéis e responsabilidades estejam bem definidos e designados, é importante que todos os membros da equipe se envolvam no planejamento. A participação de toda a equipe agrega conhecimento ao processo e fortalece o compromisso junto ao projeto (PMI, 2013). Abaixo constam os processos que norteiam este gerenciamento:

- **Planejar o gerenciamento dos recursos humanos:** identifica e documenta as funções, responsabilidades, habilidades, relações e cria um plano de gerenciamento do pessoal.
- **Mobilizar a equipe do projeto:** confirma a disponibilidade dos recursos e obtém a equipe necessária para concluir as atividades do projeto.
- **Desenvolver a equipe do projeto:** melhora as competências e interação dos membros da equipe para aprimorar o desempenho do projeto.
- **Gerenciar a equipe do projeto:** acompanha o desempenho dos membros da equipe, fornece *feedback* (realimentar), resolve os problemas decorrente do projeto e gerencia mudanças.

2.1.7 Gerenciamento das Comunicações do Projeto

De acordo com o guia PMI (2013), os gerentes de projetos passam a maior parte do seu tempo se comunicando com o time do projeto e as demais partes interessadas. Além disto, grande parte dos problemas na execução dos projetos é decorrente de uma má comunicação entre os seus envolvidos. Fazem parte do gerenciamento das comunicações do projeto os seguintes processos:

- **Planejar o gerenciamento das comunicações:** identifica as necessidades e requisitos das partes interessadas.
- **Gerenciar as comunicações:** disponibiliza as informações necessárias para as partes interessadas. É composto por criar, coletar, distribuir, armazenar, recuperar e descartar a informação.
- **Controlar as comunicações:** controla e monitora as informações no decorrer de todo o ciclo de vida do projeto de forma que todas as partes interessadas sejam atendidas.

2.1.8 Gerenciamento dos Riscos do Projeto

Os objetivos do gerenciamento de riscos são aumentar a probabilidade e impacto dos eventos positivos, bem como minimizar a exposição de eventos negativos no projeto (PMI, 2013). Os processos deste gerenciamento são:

- **Planejar o gerenciamento dos riscos:** define como conduzir as atividades de gerenciamento de riscos do projeto.
- **Identificar os riscos:** determina os riscos que podem afetar o projeto e documenta suas características.
- **Realizar a análise qualitativa dos riscos:** analisa as probabilidades dos riscos e os seus impactos no projeto.
- **Realizar a análise quantitativa dos riscos:** analisa numericamente o efeito dos riscos identificados para o projeto.
- **Planejar as respostas aos riscos:** desenvolve opções e ações para aumentar as oportunidades e reduzir as ameaças para os objetivos do projeto.

- **Controlar os riscos:** implanta plano de resposta aos riscos, monitora e controla os riscos durante o ciclo de vida do projeto.

2.1.9 Gerenciamento das Aquisições do Projeto

Segundo o guia PMI (2013), o gerenciamento das aquisições do projeto inclui os processos necessários para comprar ou adquirir produtos, serviços ou administração de todos os contratos emitidos por uma organização. Fazem parte deste gerenciamento os seguintes processos:

- **Planejar o gerenciamento das aquisições:** documenta as decisões de compras do projeto.
- **Conduzir as aquisições:** obtém as respostas dos fornecedores, seleciona um fornecedor e elabora o contrato.
- **Controlar as aquisições:** gerencia as relações de aquisição via monitoramento do contrato e realiza mudanças conforme necessário.
- **Encerrar as aquisições:** finaliza as aquisições do projeto.

2.1.10 Gerenciamento das Partes Interessadas do Projeto

O guia PMI (2013) sugere que o gerenciamento das partes interessadas está concentrado na comunicação contínua de todas as partes interessadas no projeto, de forma a atender todas as necessidades. Os seguintes processos fazem parte deste gerenciamento:

- **Identificar as partes interessadas:** identifica pessoas, grupos ou organizações que podem impactar o projeto.

- **Planejar o gerenciamento das partes interessadas:** desenvolve estratégias para engajar as partes necessárias e garantir seu engajamento no projeto.
- **Gerenciar o engajamento das partes interessadas:** comunica e trabalha as partes interessadas para atender suas necessidades e solucionar os problemas que ocorrem.
- **Controlar o engajamento das partes interessadas:** monitora o relacionamento das partes interessadas no projeto e ajusta as estratégias para que todos estejam engajados.

2.2 Métodos Ágeis

Cada vez mais as metodologias ágeis têm se tornado uma alternativa para o modelo de desenvolvimento de *software* tradicional. Embora tal modelo apresente diversas restrições quanto a sua aplicação, tais como percepção de entrega ao cliente somente no final do projeto e ociosidade de recursos quando as atividades são executadas em paralelo, o modelo cascata pode ainda ser bem aplicado quando os requisitos do sistema são bem claros e precisos e as entregas futuras estão bem definidas e documentadas.

Entretanto, na medida em que os projetos exigem constantes mudanças devido às incertezas dos requisitos dos sistemas e, por muitas vezes, antecipação das entregas para atender a legislação bancária, utilizar o modelo tradicional pode se tornar um alto risco para o negócio. Desta forma, se faz necessária a aplicação de métodos ágeis para desenvolvimento de *software*, se aplicado em um ambiente dinâmico como o ramo bancário, pode resultar em ganhos rápidos e de rápida percepção para os clientes.

Gomes (2013) salienta que, nos últimos anos, cada vez mais os métodos ágeis vêm ganhando popularidade, inclusive em grandes empresas renomadas no mundo da TI. Com a aplicação dos conceitos tidos como leves na programação, os métodos ágeis começaram a emergir, mostrando-se mais bem sucedido do que o modelo tradicional.

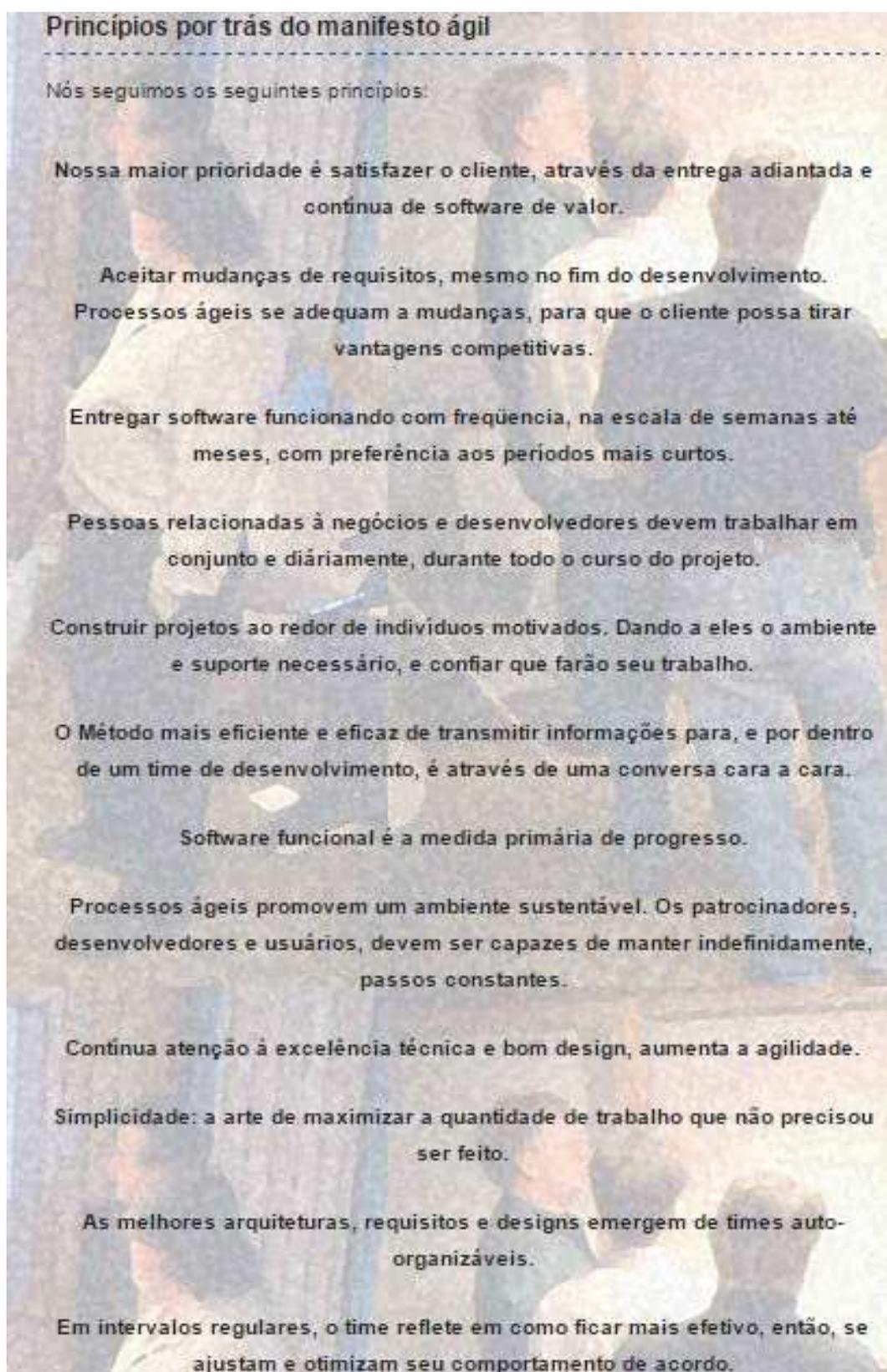
2.2.1 O Manifesto Ágil

O Termo “Métodos Ágeis” tornou-se popular em 2001 quando um grupo de especialistas em desenvolvimento de *software* se reuniu para discutir a respeito das melhores práticas para otimizar o desempenho dos seus projetos. Embora cada um possuísse suas próprias práticas e conceitos sobre execução de projetos, foi percebido que um pequeno conjunto de princípios era respeitado quando os projetos eram bem sucedidos. De acordo com Gomes (2013, p. 2),

Devido ao grande número de referências a esses processos leves, que emergiam como respostas aos constantes fracassos de projetos utilizando abordagens tradicionais, em fevereiro de 2001 um grupo de profissionais extraordinários do desenvolvimento de *software* reuniu-se em um *Resort* de *Ski* em *Wasatch Range* para discutir melhores maneiras de desenvolver *software*. Esse encontro deu origem ao manifesto ágil, uma declaração com os princípios que regem o desenvolvimento ágil.

O manifesto ágil vem como forma alternativa para as metodologias tradicionais que não conseguiam atender as necessidades do mercado de TI e até mesmo da própria organização. A figura 3 abaixo relaciona os 12 princípios:

Figura 3 – Princípios do manifesto ágil



Fonte: <http://manifestoagil.com.br/>

Segundo os profissionais que participaram do Manifesto Ágil, estes princípios são tidos como fundamentais para que um projeto tenha sucesso em sua execução. Cabe ressaltar que com o surgimento das metodologias ágeis não significa a rejeição dos princípios das metodologias tradicionais, mas sim se apresentam como alternativa para o desenvolvimento de *software*. As metodologias ágeis não relacionam conceitos novos, apenas alteram a perspectiva e os valores com que esses conceitos são encarados (HIGHSMITH; COCKBURN, 2001).

De acordo com Cruz (2015, p. 12), a principal justificativa para a criação do Manifesto Ágil veio da seguinte afirmação feita pelas pessoas que estavam presente no encontro “Estamos descobrindo maneiras melhores de desenvolver *software* fazendo-o nós mesmos e ajudando outros a fazê-lo”.

Cruz (2015, p. 12) afirma ainda que,

A partir desta simples afirmação, que também demonstra um estilo de trabalho e uma filosofia de organização e estrutura colaborativa, o Manifesto Ágil traz seus quatro valores que o sustentam e que também formam a base das principais práticas ágeis desde 2001, que são:

- Indivíduos e interações entre eles mais que processos e ferramentas.
- Software em funcionamento mais que documentação abrangente.
- Colaboração com o cliente mais que negociação de contratos.
- Responder a mudanças mais que seguir um plano. (Cruz, 2015, p. 12)

Portanto, o Manifesto Ágil é o embasamento filosófico de todos os métodos ágeis e os mais diversos tipos de métodos de desenvolvimento de *software* estão alinhados a ele. A maioria deles se utiliza de ciclos curtos e entregas rápidas, que são chamados de iterações e normalmente tem duração de poucas semanas, dessa forma garantindo *feedback* freqüente do cliente e respostas rápidas às mudanças.(GOMES, 2013).

Para um melhor entendimento, abaixo consta o detalhamento do significado de cada valor do Manifesto Ágil, segundo a visão de Prikladnicki, Willi e Milani (2014):

- Indivíduos e interação mais que processos e ferramentas. Passamos a crer tão cegamente nos processos e nas ferramentas que deixamos de nos comunicar. Esquecemos que são as pessoas que fazem *software*. Em vez de conversas e discussões, os desenvolvedores passaram a receber especificações escritas. Elas são importantes, sim, mas não comunicam tão bem como uma boa discussão presencial, ou esboços, rabiscos e modelos. Obviamente, ferramentas são importantes. É muito mais difícil fazer as coisas sem elas.

- Software em funcionamento mais que documentação abrangente. O Manifesto Ágil não nega a importância da documentação. No entanto, é preferível a entrega de *software* funcionando do que uma documentação abrangente, exagerada e cheia de desperdícios. Quando somos contratados, o resultado esperado é *software* funcionando, com qualidade. Documentação e manutenibilidade fazem parte dessa qualidade.
- Colaboração com o cliente mais que negociação de contratos. O Manifesto admite que é muito difícil se endereçar todas as complexas questões do desenvolvimento em contratos. Tentar criar muros de proteção não vai resolver nada se não houver colaboração entre a equipe e o cliente. Então, em vez de tentar resolver as coisas incluindo novas cláusulas, redigindo contratos super complexos, é preferível trabalhar em outro nível com o cliente, criando um clima de confiança e colaboração.
- Responder as mudanças mais que seguir um plano. Os processos de planejamento ágil normalmente incluem ciclos PDCA em diversos níveis (diário, semanal, mensal, trimestral, etc.), em que há a oportunidade de reflexão e readaptação dos rumos tomados pelo projeto. (PRIKLADNICKI; WILLI; MILANI, 2014, p. 6-7).

2.2.2 Benefícios dos Métodos Ágeis

Com base nas informações acima, entende-se que a transição para métodos ágeis pode resultar em diversos benefícios para toda a organização, fazendo com que a mesma possa realizar entregas com maior velocidade, qualidade e até mesmo sendo um diferencial competitivo perante o mercado.

Neste aspecto, Gomes (2013) informa que em 2011 uma pesquisa foi realizada com mais de 6.000 mil pessoas e organizações dos mais variados perfis no ramo de desenvolvimento de *software*. Como resultados foram extraídos os principais benefícios obtidos por essas organizações, que são:

- a) **Melhor *time-to-market* e maior retorno sobre o investimento:** quanto antes os clientes puderem começar a utilizar os produtos desenvolvidos, mais rápido será o retorno do valor investido, seja por meio de lucros diretos ou benefícios gerados pela utilização do produto em sua organização.
- b) **Maior satisfação do cliente e melhor gestão de mudanças de prioridades:** o planejamento iterativo permite que facilmente o cliente possa mudar suas prioridades, gerando pouco impacto na produtividade do time de desenvolvimento. A proximidade da

comunicação também resulta em um melhor alinhamento dos requisitos.

- c) **Melhor visibilidade dos projetos:** faz parte da cultura ágil manter as informações do projeto visíveis e disponíveis em ferramentas de gráficos, com as quais o time de desenvolvimento e a gestão podem acompanhar a evolução do projeto em relação as metas traçadas.
- d) **Maior produtividade:** não há uma aceitação universal a respeito da forma de se medir a produtividade de equipes de desenvolvimento de *software*. Para muitos este tema é impossível de se alcançar ou ainda muito subjetivo para se chegar a um consenso. No entanto, na pesquisa realizada, 75% dos participantes afirmaram ter alcançado uma produtividade maior depois da transição para métodos ágeis.
- e) **Equipes mais motivadas:** métodos ágeis tornam o ambiente mais agradável e em ritmo de trabalho sustentável, onde muitas organizações reportaram uma redução significativa na realização de horas extras e trabalhos de madrugada.
- f) **Melhor disciplina na engenharia e qualidade interna:** a utilização de práticas ágeis como forma de mitigação de falhas técnicas contribui para a entrega de produtos com melhor manutenibilidade, extensibilidade e com menos defeitos.
- g) **Processo de desenvolvimento simplificado:** os métodos ágeis são, de forma geral, menos prescritivos do que os métodos tradicionais, pois priorizam a iteração entre as pessoas e menos papéis e artefatos. Desta forma, tornam-se mais facilmente entendidos pela equipe.
- h) **Redução de risco:** o planejamento iterativo e as freqüentes entregas possibilitam que o cliente possa reajustar constantemente as prioridades do projeto. Métodos ágeis permitem que as incertezas do projeto sejam balanceadas de forma inteligente entre as entregas,

possibilitando assim uma melhor distribuição dos riscos. Além deste ponto, realizar entregas em um curto período de tempo permite também que haja uma maior visibilidade da velocidade do time de desenvolvimento, oferecendo assim maior previsibilidade do prazo necessário para concluir o projeto no tempo acordado junto ao seu cliente.

- i) **Redução de custos:** abordagens do modelo tradicional de desenvolvimento tendem a criar funcionalidades de baixa prioridade ou que se tornam desnecessárias ao longo do projeto. Equipes ágeis são menos propensas a isto, visto que o tempo entre o levantamento de requisitos e a entrega do produto ocorre em um curto prazo.

2.3 Scrum

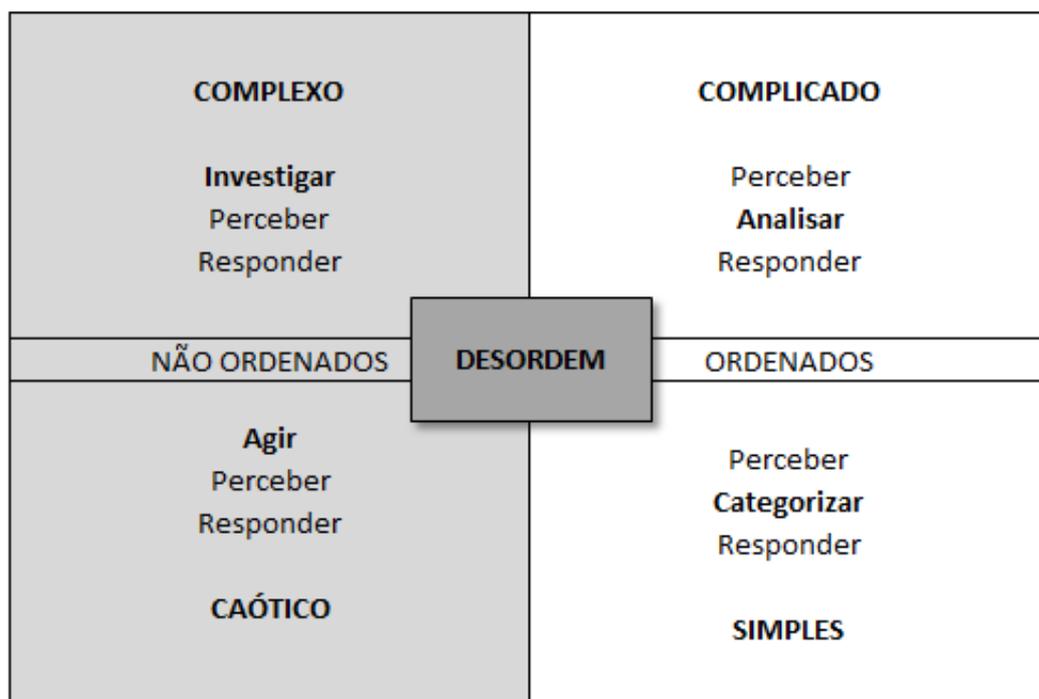
2.3.1 Introdução

O método ágil *Scrum* iniciou na década de 90, porém a prática tornou-se popular somente na década seguinte. *Scrum* foi apontado como o método de trabalho utilizado por dois em cada três participantes de uma pesquisa realizada em 2011 (versionOne, 2011). De acordo com Sabbagh (2013, p. 17) “*Scrum* é um *framework* ágil, simples e leve, utilizado para gestão do desenvolvimento de produtos complexos imersos em ambientes complexos”. Por trabalhar com uma abordagem iterativa e incremental, entrega valor ao cliente com uma maior frequência, reduzindo assim os riscos atrelados ao projeto.

Sabbagh (2013) salienta ainda que o *Scrum* é um conjunto de processos que não define práticas específicas e que deverão ser seguidas pelo time de desenvolvimento. Por não ser um método prescritivo, o *Scrum* não define como deverão ser levantadas as necessidades da área de negócio e nem como se deve lidar com o cliente. Com a adoção de diversas práticas de sucesso, o *Scrum* também serviu de influência para o desenvolvimento de outra metodologia ágil, o *Extreme Programmin*, método este que será mais detalhado ao longo deste trabalho (PRATES, 2013).

Muitas organizações ainda possuem dúvidas sobre como aplicar as práticas do *Scrum* no seu ambiente de desenvolvimento. Para resolver este impasse, Dave Snowden criou um modelo chamado *Cynefin* para auxiliar na tomada de decisões. Este modelo analisa o contexto de operação em que os sistemas da organização estão operando e identifica qual é a abordagem mais apropriada a ser utilizada. Na figura 4 é possível identificar em que sistemas a prática *Scrum* melhor se aplica (SNOWDEN; BOONE, 2007).

Figura 4 – O modelo Cynefin



Fonte: Adaptada de Snowden e Boone (2007).

Na tabela abaixo consta o descritivo de cada contexto do modelo analisado.

Tabela 1 – Contextos do modelo de *Cynefin*

(Continua)

Contextos	Descrição
Simples	Esse é o domínio das melhores práticas, onde há estabilidade e as relações entre causa e efeito são evidentes para todos, previsíveis e repetíveis. Assim, para uma dada situação, devem-se avaliar seus fatos (perceber) para definir de que tipo de situação se trata (categorizar) e então aplicar a melhor prática conhecida (responder).
Complicados	Esse é o método das boas práticas, onde existe a relação entre causa e efeito, mas nem todos podem vê-la. Diferentemente dos contextos simples, pode haver várias respostas viáveis para um mesmo problema e especialistas podem ser necessários para investigar e escolher a que melhor se aplica ao problema em questão. Assim, para uma dada situação, devem-se avaliar seus fatos (perceber) para definir qual das possíveis soluções é a mais adequada (analisar) e então aplicá-la (responder).
Complexo	Esse é o domínio da emergência, onde a relação entre causa e efeito somente se torna evidente em retrospecto, e não reproduzível. Não é possível prever o que irá acontecer e, assim, em vez de se impor por um caminho, uma abordagem empírica e adaptativa deve ser utilizada, a partir da qual as práticas irão emergir. A tolerância a falhas é um aspecto essencial da aprendizagem empírica. Assim, para uma dada situação, deve-se investigar por meio de experimentos para entender seus impactos (perceber) e então promover o que funciona e abandonar o que não funciona (responder).

Tabela 1 – Contextos do modelo de *Cynefin*

(Conclusão)

Caóticos	Esse é o domínio das práticas novas, onde as relações entre causa e efeito são impossíveis de serem determinadas, pois mudam constantemente. Não há padrões, apenas turbulências, de modo que qualquer prática utilizada será completamente nova. Quando possível, deve-se agir rapidamente de modo a restabelecer a ordem (agir), para se entender onde existe a estabilidade (perceber) e então trabalhar para trazer a situação do caos para a complexidade (responder), que é um contexto mais gerenciável.
Desordem	Neste domínio, múltiplas perspectivas se apresentam e não é possível determinar qual dos quatro contextos é o predominante. Assim, nada se sabe sobre a relação entre causa e efeito e não pode se determinar qual a melhor forma de se trabalhar. Na desordem, as pessoas competem de forma destrutiva por soluções que estejam em sua zona de conforto, ou seja, em algum dos outros quatro contextos. A saída para essa situação é quebrar o problema em partes e definir o contexto de cada uma delas – simples, complicado, complexo ou caótico – para então agir de acordo.

Fonte: Adaptado de Sabbagh (2013, p. 32-34).

Sabbagh (2013) conclui que os métodos tradicionais de gerência de projetos categorizam todos os tipos de projetos como pertencentes a um único domínio, como se fossem todos ordenados, ou seja, todos no contexto simples ou complicados. Este conceito, quando não verdadeiro, poderá resultar em fracasso na execução do projeto. A grande maioria dos projetos está contida no contexto complexo, gerenciá-los como se fossem simples e adaptativos não é uma boa solução. O *Scrum* foi criado justamente para lidar com projetos complexos.

Tudo começa com a contratação de um desenvolvimento para um projeto ou apenas a necessidade de suprir uma demanda do negócio. O objetivo que se quer alcançar é chamado de Visão do Produto. A responsabilidade por passar a necessidade da área de negócios para o time de desenvolvimento, bem como manter a Visão do Produto constante e atualizada, é do *Product Owner* (PO). Sabbagh (2013, p. 39) define *Product Owner* como sendo a pessoa que:

[...] trabalha com clientes do projeto e com quaisquer outras partes interessadas que possam contribuir para o entendimento e definição da Visão do Produto. O grupo de partes interessadas do projeto também inclui os próprios usuários do produto, que receberão ao longo do desenvolvimento partes prontas do produto para serem utilizadas (SABBAGH, 2013, p. 39).

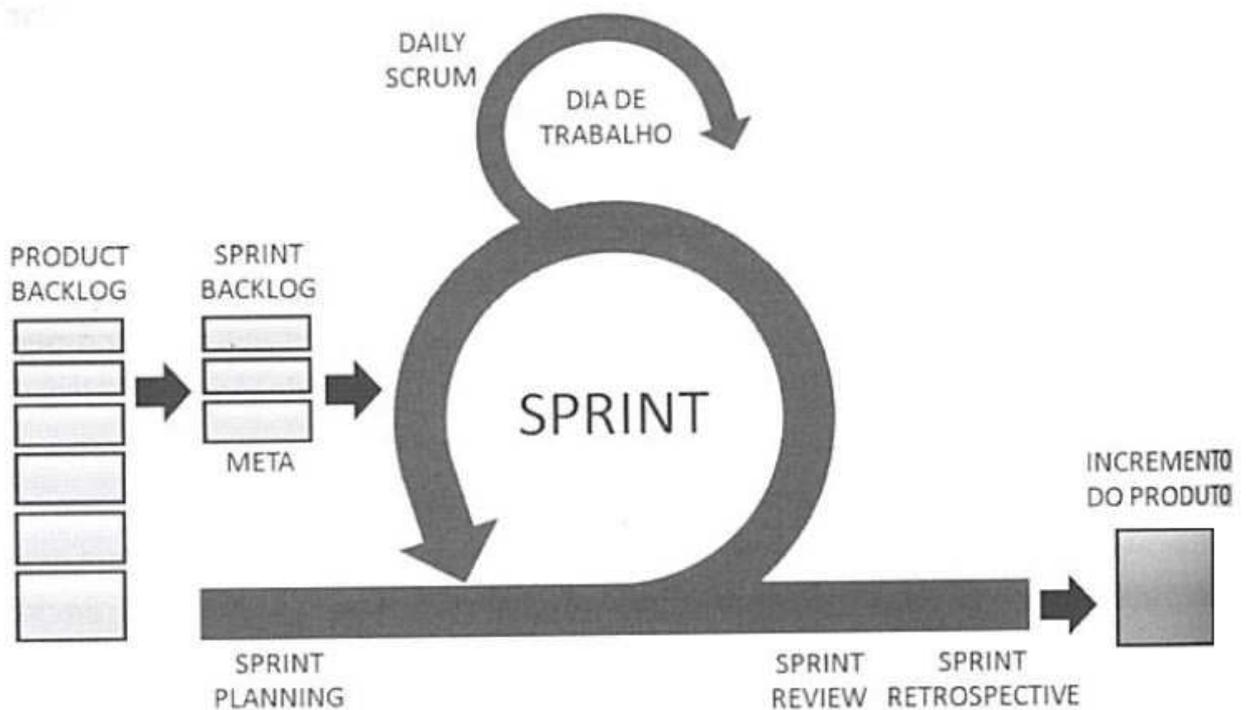
O *ScrumMaster* tem a responsabilidade de atuar como um facilitador para o time de desenvolvimento, removendo impedimentos que o grupo encontre para a execução do trabalho. O *ScrumMaster* deve estar presente em todas as reuniões diárias da equipe, assim como deve facilitar as interações do *Product Owner* e o time de desenvolvimento. Também é a pessoa que tem o domínio do *Scrum* e responsável por ensinar a metodologia para todo o time (CRUZ, 2013).

Antes de iniciar o desenvolvimento do projeto, o *Product Owner* deve iniciar, a partir da Visão do Produto, a criação de uma lista ordenada, incompleta e dinâmica de itens que deverão ser produzidos ao longo de todo o projeto e que futuramente se tornarão funcionalidades. Esta lista é chamada de *Product Backlog* (SABBAGH, 2013).

Cada um dos itens do *Product Backlog* é chamado de *User Story*, o qual descreve sob o ponto de vista do usuário qual é a necessidade ou objetivo da área de negócios. A *User Story* não possui uma definição de como se deve criá-la, cabe ao time de desenvolvimento identificar a melhor forma para representar os itens do *Product Backlog* (CRUZ, 2013).

Dado os contextos acima, o time já está pronto para iniciar o primeiro de vários ciclos de desenvolvimento do projeto. Estes ciclos são chamados de *Sprints* (arrancada), onde uma vez terminado um, imediatamente inicia-se o seguinte (SABBAGH, 2013). O ciclo completo do *Scrum* pode ser na figura 5:

Figura 5 – Ciclo do *Scrum*



Fonte: Sabbagh (2013, p. 42)

2.3.2 Time *Scrum*

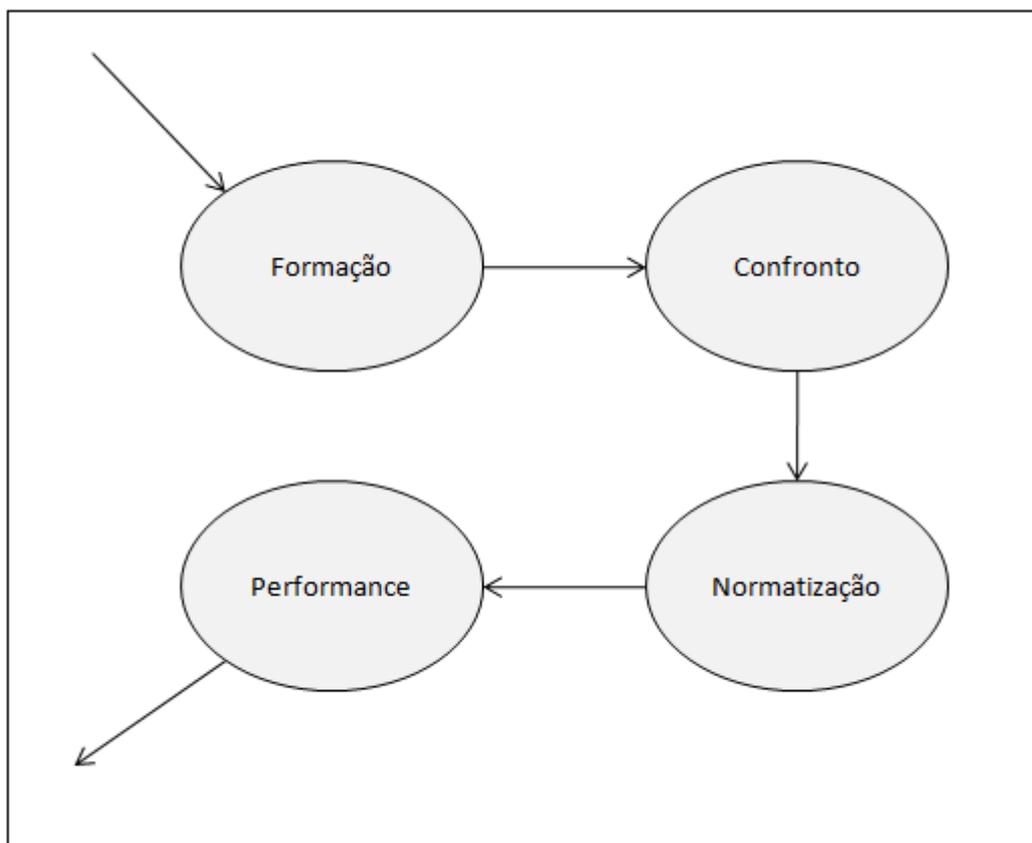
Sabbagh (2013) diz que o conceito de time de desenvolvimento é utilizado para designar um grupo multidisciplinar e multifuncional de pessoas que têm foco no desenvolvimento de projetos. Com base nas prioridades definidas pelo *Product Owner*, o time gera, em casa *sprint*, um Incremento do produto pronto, de acordo com a definição de Pronto, e que significa uma entrega de valor visível para os clientes do projeto.

O time é responsável por executar o desenvolvimento do projeto e transformar o *backlog* do produto em incrementos de funcionalidades, de forma que se crie um sistema que possa ser entregue ao cliente (CRUZ, 2015). No grupo devem existir perfis de profissionais com conhecimentos especializados no trabalho que será desenvolvido, tais como programação, arquitetura, testes e outros. No entanto, no time de desenvolvimento não há títulos e nem devem existir cargos ou áreas específicas de atuação. No *Scrum* todos os membros são designados como desenvolvedores.

Cruz (2015) também afirma que o time não deve ser subdividido para executar outras tarefas específicas, visto que os mesmos devem ser auto-organizáveis e devem dizer como transformar o *backlog* do produto em funcionalidades.

Em 1977, Bruce Tuckman, criou uma teoria referente as dinâmicas de grupos onde descreve o processo de maturidade dos grupos, conforme figura abaixo.

Figura 6 – Estágios de Tuckman



Fonte: Adaptado de Gomes (2013, p. 26).

Gomes (2013, p. 26) afirma que:

Cada um dos estágios citados anteriormente possui algumas características próprias:

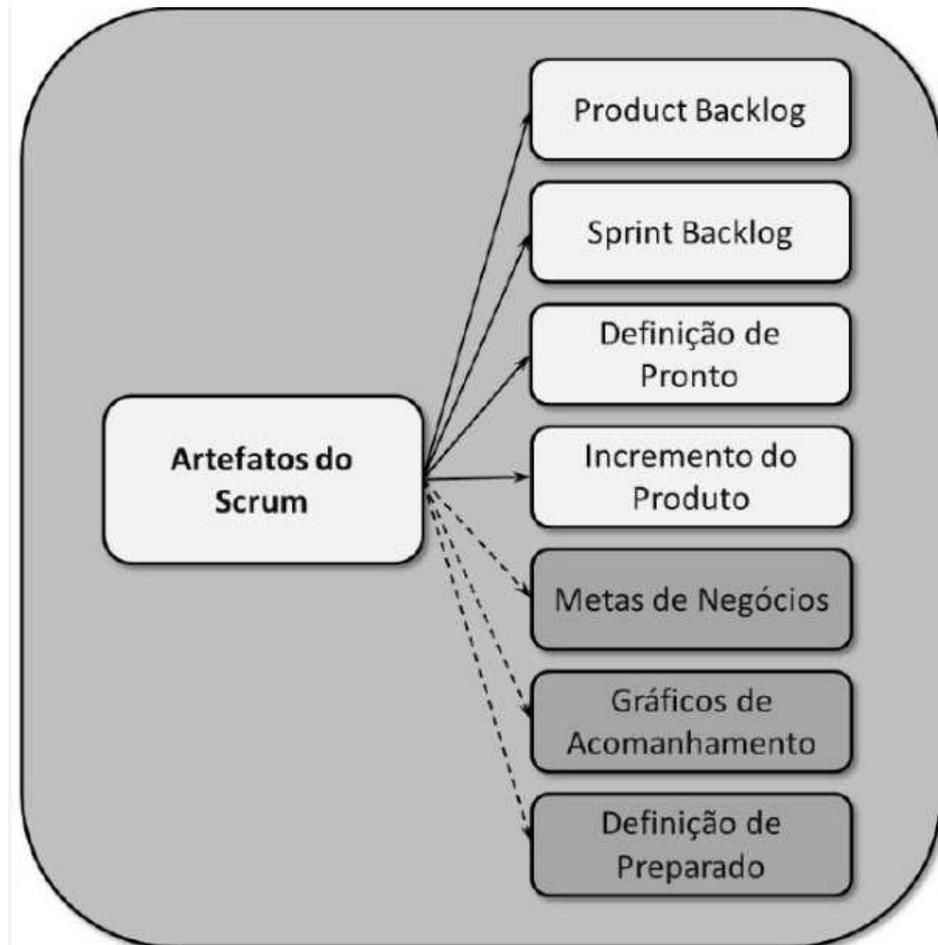
1. **Formação:** este é o primeiro estágio pelo qual uma equipe recém formada passará, é o momento em que os membros da equipe começa a conceber uns aos outros, procuram entender quais são as habilidades de cada um, e as tendências de comportamentos de cada pessoa. Nesse estágio também se estabelecem os primeiros objetivos coletivos e completam-se algumas tarefas, mas a ênfase das pessoas ainda é mais individual do que coletiva.
2. **Confronto:** conflitos aparecem à medida que diferentes idéias para atingir objetivos vão surgindo e as metas coletivas identificadas no estágio de formação são questionadas. Alguns membros podem se abalar por essa situação. Uma liderança forte é extremamente importante para que a equipe passe por este estágio.
3. **Normatização:** é neste estágio que as pessoas realmente passam a priorizar os objetivos coletivos da equipe sobre os objetivos individuais de cada membro, por isso, a equipe consegue entrar em consenso mais facilmente.
4. **Performance:** a equipe já possui todas as competências necessárias par atingir seus objetivos, todos estão comprometidos e motivados, e atuam com verdadeiro espírito de equipe, pensando em termos de “nós” em vez de “eu” ou “eles. Conflitos são facilmente solucionados.

2.3.3 Artefatos do *Scrum*

Um conjunto de artefatos deverá ser produzido pelo time de desenvolvimento ao longo da execução do projeto AMARAL (2011).

Sabbagh (2013) salienta que o *Scrum* define o uso de 4 artefatos: O *Product Backlog*, o *Sprint Backlog*, a Definição de Pronto e o Incremento no Produto. Conforme a figura 7, Sabbagh (2013) sugere ainda o complemento de três novos artefatos.

Figura 7 – Artefatos do Scrum



Fonte: Sabbagh (2013, p. 109)

2.3.3.1 *Product Backlog*

O *Product Backlog* é uma lista de itens contendo todas as funcionalidades que um produto deverá ter, a qual é definida pelo *Product Owner* e desenvolvida pelo time de desenvolvimento. De acordo com Sabbagh (2013, p 111),

O *Product Backlog* contém as necessidades ou objetivos de negócio dos clientes do projeto e demais partes interessadas e pode também conter melhorias a serem realizadas no produto, correções de problemas, questões técnicas, pesquisas que forem necessárias etc. Assim, tudo o que pode vir a ser desenvolvido para se alcançar a Visão do Produto é adicionado como um item do *Product Backlog*.

Para que o time de desenvolvimento possa trabalhar em cima do *Product Backlog* e transformar os itens em funcionalidades que estejam aptas para serem

entregues ao cliente, é necessário que se tenham detalhes suficientes sobre os itens do *Backlog* (CRUZ, 2015). As prioridades de execução dos itens são definidas pelo *Product Owner*, as quais serão desenvolvidas pelo time. No topo da listagem devem constar os itens mais sensíveis para o projeto de forma os clientes possam já receber um retorno sobre o investimento aplicado. Por serem os itens mais prioritários, são descritos com um maior detalhamento para que a entrega seja a mais assertiva possível. Por outro lado, os itens da parte final da listagem possuem menor importância para o projeto e, por conseguinte, menor detalhamento.

A metodologia *Scrum* não define nenhum leiaute ou qualquer padronização para o *Product Backlog*. Conforme Sabbagh (2013), o importante é que sejam exibidos itens em seqüência e que nenhum item tenha a mesma prioridade que outro, além de ser facilmente reorganizado, se necessário. A figura abaixo ilustra três tipos de exemplo para o *Product Backlog*:

Figura 8 – Itens do Backlog

<input type="checkbox"/> Lorem ipsum dolor sit amet consectetur	3
<input type="checkbox"/> Adipiscing elit duis molestie nibh quis ultricies	5
<input type="checkbox"/> Adipiscing sem odio eleifend lectus ut luctus	2

<input type="checkbox"/> Augue sapien mattis	>Lorem ipsum dolor sit amet consectetur	3
<input type="checkbox"/> Mollis elementum al	Adipiscing elit duis molestie nibh quis ultricies	5
<input type="checkbox"/> Erat vestibulum cons	Adipiscing sem odio eleifend lectus ut luctus	2

PRODUCT BACKLOG

<input type="checkbox"/> Eleifend id pretium t	Augue sapien mattis ante proin massa est
<input type="checkbox"/> laculis lacinia neque	Mollis elementum aliquet eget fringilla tempo
<input type="checkbox"/> Quam imperdiet a pr	Erat vestibulum consequat purus in ante
<input type="checkbox"/> Fringilla varius ante r	Eleifend id pretium tellus sagittis vestibulum
<input type="checkbox"/> Euismod tempor erat	laculis lacinia neque condimentum semper
<input type="checkbox"/> A lectus arcu non feu	Quam imperdiet a proin pulvinar odio eu
	Fringilla varius ante nunc faucibus odio
	Euismod tempor erat purus at purus aliquam
	A lectus arcu non feugiat ipsum cras dapibus

Fonte: Sabbagh (2013, p. 113)

2.3.3.2 *Sprint Backlog*

Sprint Backlog é uma lista de itens que o time de desenvolvimento precisa completar durante o *sprint* para que o item selecionado torne-se uma funcionalidade (LACEY, 2014). A seleção da lista é definida pelo time durante as reuniões de planejamento do *sprint* e negociada junto ao *Product Owner*, considerando-se que os itens mais acima do *Product backlog* possuem maior prioridade do que estão mais abaixo. O *sprint backlog* é uma visão de quais itens o time entende como possível de ser entregue ao final do *sprint*. Caso seja necessário, durante esta fase, o grupo pode obter mais detalhes junto ao *Product Owner* para identificar quais itens devem ser selecionados na *sprint*. Sabbagh (2013) diz que para representar o *sprint Backlog* deve-se usar um quadro contendo as tarefas do time de desenvolvimento para que seja visível em que situação estão os itens e suas tarefas correspondentes.

O time de desenvolvimento utiliza o melhor de seu conhecimento no momento do planejamento para criar o *Sprint Backlog*. Espera-se, no entanto, que esse plano evolua e seja atualizado no decorrer do *Sprint*, de forma a refletir o trabalho restante. Ou seja, alterações no plano como, por exemplo, a adição, reestimativa ou remoção de tarefas são aceitáveis. No entanto, uma quantidade de mudanças que dificulte ou até mesmo torne impossível de se atingir a meta do *Sprint* indicam, em geral, um planejamento para o *Sprint* mal executado. (SABBAGH, 2013, p. 150).

2.3.3.3 *Definição de Pronto*

A Definição de Pronto é, basicamente, uma avaliação do que deve ser feito pelo time de desenvolvimento para que a entrega possa ser considerada como entregável. Quando o item do *Product Backlog* ou incremento assume o estado de Pronto, é importante que todos os membros da equipe conheçam este significado, pois neste momento o trabalho que estava sendo realizado pelo grupo foi concluído. Para Gomes (2013),

Não há um *checklist* oficial que deva ser usado por todas as equipes ágeis do mundo, porque, considerando a complexidade do desenvolvimento de *software*, cada equipe é única e está resolvendo um problema dentro de um contexto único. Por isso, cada equipe deverá descobrir o que deve fazer parte de sua definição de pronto de acordo com o seu contexto, ou seja, levando em consideração a natureza do produto, a tecnologia que está sendo utilizada, as necessidades dos usuários etc.

Alguns itens são:

- Código refatorado.
- Código dentro dos padrões de codificação.

- Código revisado ou feito em par.
- Código integrado no sistema de controle de versão.
- Documentação de arquitetura atualizada.
- Testes de unidade realizados.
- Testes de aceitação realizados.
- Testes exploratórios realizados.
- Nenhum defeito conhecido pendente.
- Product Owner aceitou na história.
- Manual do Usuário atualizado.

É importante que o *Product Owner* participe da construção desta definição, uma vez que ele conhece quais são as expectativas que os clientes possuem com relação ao que o projeto deve entregar. Não é um problema que a definição mude conforme o desenvolvimento avança, visto que novas definições vão surgindo para agregar maior qualidade ao projeto.

2.3.3.4 Incremento do Produto

O Incremento do Produto reúne todos os itens do *Product Backlog* que foram finalizados durante um *sprint*. Em cada *sprint* do projeto, o time de desenvolvimento trabalha nos itens selecionados do *Product Backlog*, considerando suas prioridades do mais importante (topo da lista) para o menos importante (final da lista). O Incremento do Produto é a conclusão deste trabalho, ou seja, é a soma de todos os itens que foram completados no *sprint*. De acordo com Sabbagh (2013, p. 157),

Espera-se que o Incremento do Produto seja entregável, de forma que o *Product Owner* possa decidir por fazer uma *Release* para os clientes do projeto ao final do *Sprint*. O *Product Owner* pode, no entanto, optar por acumular alguns Incrementos do Produto para só então fazer uma *Release*. Um Incremento nem sempre representa valor suficiente para ser utilizado por seus usuários. Ele pode ser incompleto, portanto, e nesse caso não há sentido em entregá-lo ao final do *Sprint* em que foi produzido. Ainda que represente valor suficiente, diferentes razões podem justificar que não seja entregue ao final de cada *Sprint*: os clientes podem não conseguir absorver mudanças tão frequentes, ou mesmo questões políticas, burocráticas ou técnicas podem impedir o *Product Owner* de realizar a *Release* imediatamente.

2.3.3.5 Metas de Negócio, Gráficos de Acompanhamento e Definição de Preparado

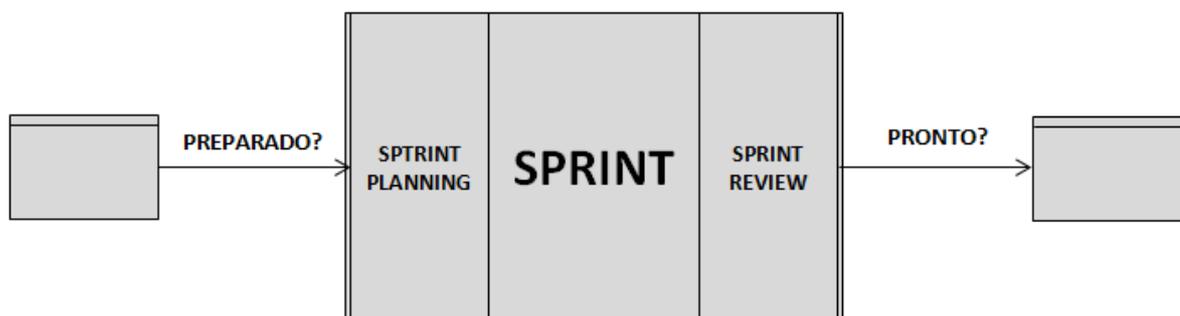
Nas Metas de Negócios, seja no contexto de uma *sprint*, *release* ou do projeto como um todo, o time de desenvolvimento trabalha no item de maior importância

para o item de menor importância, com o objetivo de realizar as entregas mais prioritárias para o negócio. Como os itens mais importantes são desenvolvidos primeiro, mesmo que com todo o esforço do time não seja possível entregar todos os itens previstos para a *sprint*, o time pode conseguir alcançar a meta estabelecida, visto que somente restarão os itens menos prioritários na listagem (SABBAGH, 2013). Cabe salientar que a meta de negócio não é um valor financeiro ou um escopo de serviço que o time de desenvolvimento deve realizar, são metas ou objetivos de negócios que tem como objetivo guiar a equipe para atender as necessidades do cliente.

Os gráficos de acompanhamento são ferramentas que servem para dar visibilidade quanto ao progresso do time de desenvolvimento para com as metas estabelecidas pelo *Product Owner*. Como este item está relacionado aos indicadores de desempenho, este assunto será tratado separadamente no objetivo específico relacionado a indicadores.

A Definição de Preparado é importante pelo fato de diminuir os riscos de ocorrer um *sprint* mal planejado, visto que quando há muitos detalhes é comum que alguns sejam deixados para serem discutidos na reunião de *Sprint Planning*. Sabbagh (2013, p. 187-188) diz que a Definição de Preparo “[...] é um artefato utilizado para garantir que os itens a serem considerados na reunião de *Sprint Planning* estejam preparados segundo um critério bem definido.”. Este trabalho de refinamento é realizado ao longo do *sprint* em sessões agendadas ou por um trabalho contínuo. Na figura abaixo pode-se observar a diferença entre Definição de Preparado e Definição de Pronto.

Figura 9 – Fluxo do item



Fonte: Adaptado de Sabbagh (2013, p. 188)

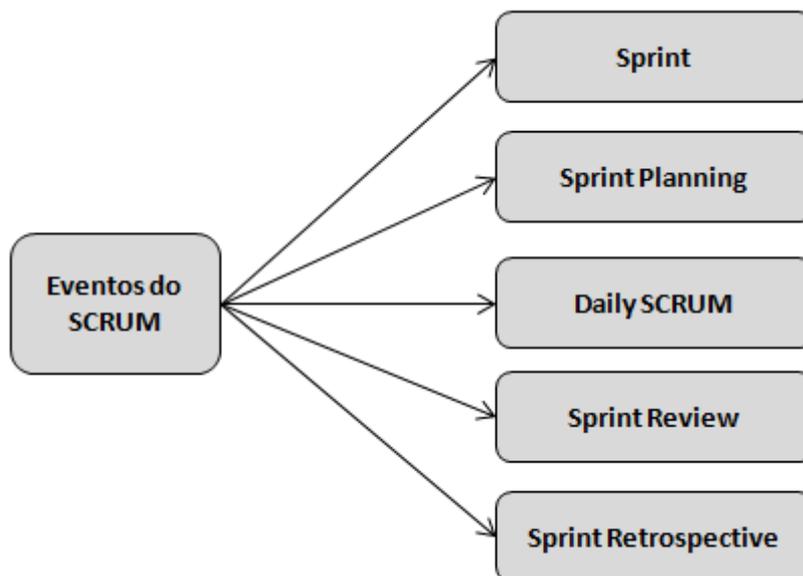
2.3.4 Eventos do Scrum

Em métodos ágeis, um evento é qualquer atividade realizada por uma ou mais pessoas, com um pequeno intervalo de tempo e com a definição de um objetivo específico (PRIKLADNICKI; WILLI; MILANI, 2014).

No *Scrum*, os eventos são criados para que uma rotina seja seguida, de forma que não se tenha a necessidade de se fazer reuniões não definidas na metodologia do *Scrum*. Possuem duração definida, a qual é chamada de *timebox* (caixa de tempo). Os eventos são o próprio ciclo de desenvolvimento, os quais são denominados de *Sprint*, *Sprint Planning*, *Daily Scrum*, *Sprint Review* e *Sprint Retrospective*. Para Sabbagh (2013, p. 194),

Os eventos do *Scrum* possuem uma duração definida, chamada de *timebox*. O *timebox* pode definir o tempo máximo em que um evento deve ocorrer, como nas reuniões, ou tempo exato, como no *Sprint*. O objetivo dos *timeboxes* é limitar o tempo em que um objetivo deve ser alcançado, de forma que não se gere desperdício com trabalhos intermináveis. Os *timeboxes* ajudam a criar um ritmo ou uma regularidade no trabalho do time de desenvolvimento.

Figura 10 – Eventos do Scrum



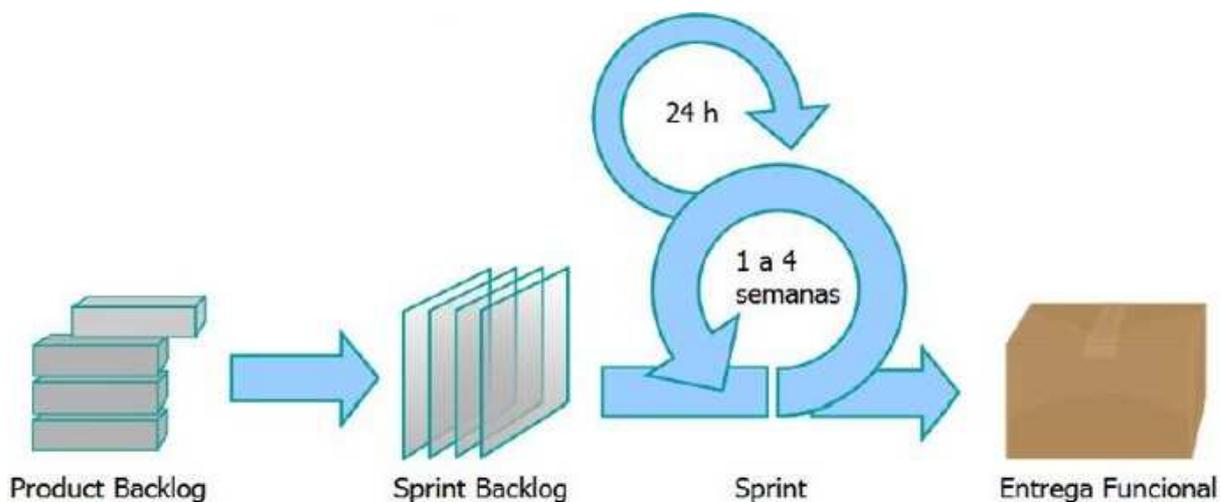
Fonte: Elaborado pelo autor.

2.3.4.1 *Sprint*

O *Sprint* é uma unidade básica do ciclo de desenvolvimento, em que o time entrega um incremento do produto, ou seja, uma funcionalidade desenvolvida a partir dos itens do *Product Backlog*. Possui um objetivo ou necessidade bem definida que é acordada entre o time de desenvolvimento e o *Product Owner*. Sabbagh (2013) resume da seguinte forma:

- a) **Objetivo:** atingir a meta do *sprint* que foi definida pelo *Product Owner*.
- b) **Quando:** ocorre durante todo o desenvolvimento do produto, um após o outro.
- c) **Duração:** de no mínimo 1 semana e no máximo 4 semanas.
- d) **Participantes obrigatórios:** time de desenvolvimento, *Product Owner* e *ScrumMaster*.
- e) **Saídas esperadas:** um novo incremento do produto no estado Pronto, o qual foi definido através da Meta do *Sprint*

A figura abaixo descreve em que momento ocorre o *sprint*.

Figura 11 – *Sprint*

Fonte: Adaptada WIKIPEDIA (2014)

2.3.4.2 *Sprint Planning*

O *Sprint Planning* é uma reunião que conta com a participação obrigatória de todos os membros do *Scrum*, podendo também contar com a presença do cliente, embora esta seja facultativa. Nesta reunião é negociado o que será desenvolvido e quais são as metas deste *sprint*.

Sabbagh (2013, p. 204) afirma que a reunião de *sprint* não deve durar mais que um dia de trabalho “Recomenda-se que ela seja um *timebox* de não mais que 5% da duração do *sprint*, ou no máximo oito horas para *sprint* de um mês.”.

Ao final do *sprint planning*, o time de desenvolvimento e o *Product Owner* negociam e definem uma meta de negócio que será o objetivo a ser perseguido pelo time para que a necessidade seja atendida.

Convém utilizar como meta de *sprint* uma analogia ao que o item se refere, como por exemplo, “permitir que o *internet banking* possibilite pagamento de boleto bancário” em vez de “desenvolver os itens X, Y, Z” (SABBAGH, 2013). Uma vez a meta tendo sido estabelecida, não é permitido que ela mude durante o *sprint*. Apenas o *Product Owner* pode mudá-la caso perceba que o *sprint* não faz mais sentido para o negócio.

2.3.4.3 *Daily Scrum*

O *Daily Scrum* é uma reunião rápida com duração máxima de 15 minutos que tem como objetivo planejar o próximo dia de desenvolvimento. O time de desenvolvimento se reúne, física ou remotamente, para avaliar a progressão do que foi estipulado no *sprint*. Ao término da reunião, três questões precisam ser respondidas:

1. que eu fui capaz de realizar desde a reunião de ontem?
2. que eu planejo realizar antes da reunião de amanhã?
3. que está me atrapalhando? (PHAM; PHAM, 2011, p. 269).

É facultativa a participação do *ScrumMaster*, porém se for necessário ele poderá participar como um facilitador na reunião do *Daily Scrum*, ajudando o time de desenvolvimento a manter o foco e não ultrapassar os 15 minutos de duração. É de responsabilidade do *ScrumMaster*, após a reunião, solucionar os impedimentos que o grupo identificou como sendo pontos de atenção para a execução das atividades previstos no item.

Como a execução das atividades é de responsabilidade do time de desenvolvimento, o *Product Owner* não participa do *Daily Scrum* para que não haja interferência. De acordo com Sabbagh (2013, p. 216),

A reunião de *Daily Scrum* é também conhecida como *Daily Meeting* (reunião diária) ou *Stand-up Meeting* (reunião em pé). Essa última designação é utilizada no *Extreme Programming* e indica uma prática potencialmente útil: a realização da reunião com todos os participantes de pé, com o objetivo de não permitir que eles sintam-se confortáveis o suficiente para quebrar o *timebox*. Cabe destacar, no entanto, que não há obrigatoriedade no *Scrum* de se realizar esta reunião em pé.

É importante salientar que não é objetivo do *Daily Scrum* informar os impedimentos ao *ScrumMaster*. Isto ocorre na reunião para que todo o time que atua no desenvolvimento do projeto tenha visibilidade dos problemas que foram identificados. Quando o time percebe um impedimento, o *ScrumMaster* deve ser notificado o mais breve possível e não somente na reunião do dia seguinte. Desta forma, a solução é encontrada mais rapidamente sem impactar nos prazos acordados junto ao *Product Owner*.

Por ter uma duração curta, o *Daily Scrum* não deve ser uma reunião de trabalho. Quando o grupo discute os pontos que estão sendo tratados, é comum na

equipe surgirem discussões ou questões relacionadas ao negócio. Nesta hipótese, geralmente o grupo cria uma reunião de trabalho específica para tratar o tema abordado (SABBAGH, 2013).

2.3.4.4 *Sprint Review*

O objetivo do *Sprint Review* é avaliar se o que está sendo entregue pelo time de desenvolvimento é o que foi previsto para ser entregue. Cruz (2015, p. 283) afirma que:

O objetivo maior desta reunião é a revisão, pelo PO ou pelo cliente, dos itens concluídos pelo Time. A melhor maneira de se fazer isso é justamente realizar uma apresentação ao PO, de todos os itens completados na *Sprint*. A sugestão mais indicada é que o Time realize uma demonstração do funcionamento do produto, apresentando o que está pronto e como funciona. Com isso, o PO poderá conferir e avaliar o que está sendo considerado como pronto, levando em conta o que está sendo entregue *versus* o que deveria ser entregue.

Para esta reunião não é preciso que ocorra grande preparação, o intuito é demonstrar para o *Product Owner* e partes interessadas o produto funcionando. Contudo, é interessante que haja uma pequena sessão no final do trabalho do *sprint* para que o time de desenvolvimento e o *Product Owner* possam alinhar o entendimento das ocorrências durante o *sprint* e definir o que deverá ser apresentado na reunião na revisão do *sprint* (SABBAGH, 2013).

Em geral, o *Product Owner* analisa os itens entregues e verifica se estão realmente com o estado de Pronto e não necessariamente todos os itens da *sprint* devam ser entregues. Na figura abaixo pode-se observar um modelo de apresentação de tarefas, onde quase todos os itens previstos foram planejados.

Figura 12 – Quadro de Tarefas



Fonte: Adaptado de Sabbagh (2013, p. 223).

2.3.4.5 *Sprint Retrospective*

Finalizar o *sprint* atual e entregar ao cliente o produto que lhe era esperado é importante, no entanto, sem revisar o que já passou e tomar como base as lições aprendidas para aplicação nos próximos *sprints* não haverá evolução do time e nem a melhoria contínua constante nos times de alto desempenho (CRUZ, 2015).

Times que utilizam *Scrum* estão sempre buscando melhores formas de se fazer seu trabalho, aprendendo com seus acertos e erros ao longo de todo o projeto. O objetivo da reunião de *Sprint Retrospective* (ou retrospectiva do *Sprint*) é estimular o Time de *Scrum* a realizar essa prática, que é chamada de melhoria incremental contínua. (SABBAGH, 2013, p. 226).

Nesta reunião, o time de desenvolvimento analisa o encerramento dos seus processos de trabalho, dinâmicas, comportamentos, práticas, ferramentas utilizadas e ambientes e planeja quais melhorias deverão ser desenvolvidas para os próximos ciclos de *sprint* (SABBAGH, 2013).

A metodologia aborda que o indicado é que a reunião de *Sprint Retrospective* ocorra no último dia de cada *sprint*, após terminar o *sprint review*. Esta prática assegurará que as melhorias identificadas sejam previstas nos *sprints* futuros. Conforme citado inicialmente, cabe ao *ScrumMaster*, conhecedor das práticas do *Scrum*, garantir que a equipe de desenvolvimento siga estas orientações com regularidade.

2.4 Kanban

O *Kanban* é um termo que significa literalmente “cartão” ou “sinalização”, tendo seus conceitos na literatura japonesa. Para GOMES (2013):

Kanban é um método ágil que, diferentemente da maioria dos métodos ágeis, não possui iterações. Ao invés disso, desacopla o planejamento, priorização, desenvolvimento e entrega, de forma que cada uma dessas atividades possa ter sua própria cadência para melhor se ajustar à realidade e necessidade que o processo demanda (GOMES, 2013, p. 18).

Neste método ágil as cadências, repetições que ocorrem de forma regular, é um conceito que determina o ritmo de um determinado tipo de evento. Desta forma, qualquer tipo de evento pode ter a sua própria cadência (GOMES, 2013). Por ter um processo contínuo, não é necessário que todo o planejamento seja concluído para que seja possível realizar uma entrega para o cliente.

Um dos benefícios do *Kanban* é que as tarefas classificadas como prontas podem ser entregues, enquanto que as demais que estão em progresso serão também entregues na medida em que forem sendo finalizadas. No *Scrum*, por exemplo, cada iteração dura em média quatro semanas e todos os períodos são pré-definidos.

De acordo com Massari (2014, p. 23), o *Kanban* possui cinco princípios básicos:

1. Fluxo de trabalho deve ser visível. É importante que o fluxo de trabalho esteja visível para todos, de forma que ele possa ser organizado, otimizado e rastreado.

2. Limitar WIP (*Work In Progress*). *Work In Progress*, ou trabalho em andamento, é a quantidade de tarefas existentes nas colunas do *Kanban*. É importantíssimo limitar essa quantidade, pois há riscos da criação de “gargalos” no processo [...].
3. Gerenciar o fluxo, para identificar problemas e melhorias. O fluxo não se autogerencia, então é importante sempre ficar atento para identificar possíveis problemas, mudanças e chances de melhoria.
4. Garantir clareza nas políticas do processo. Isso faz com que toda a equipe conheça as regras do jogo, evitando qualquer tipo de mal-entendido.
5. Colaboração na melhoria do processo. Trabalhar em equipe, sempre buscando identificar possíveis pontos de melhoria.

O *Kanban* trata-se de um quadro branco onde as tarefas são classificadas conforme ilustrado na figura 12. As tarefas são escritas em blocos de anotações e passam pelos diferentes estágios que foram definidos no quadro. O instrumento visual que é gerado, o quadro *Kanban*, auxiliará para que seja possível haver uma visão geral e organizada de todo o trabalho (PRIKLADNICKI; WILLI; MILANI, 2014).

“valor” e “Fluxo” passam a ser, então, elementos concretos e visíveis na rotina de trabalho da equipe, o que explicita os problemas. *Kanban* dará à equipe a capacidade de “enxergar” os gargalos que interferem no fluxo de valor. Ao se reorganizarem para resolvê-los, o processo de trabalho muda, uma nova forma de ver o trabalho se evidencia, e novas oportunidades de melhoria emergem. Um ciclo evolucionário de mudanças se estabelece (PRIKLADNICKI; WILLI; MILANI, 2014, p 120).

Figura 13 – Quadro de Tarefas do *Kanban*



Fonte: Massari (2014, p. 22).

Um exemplo de aplicação da metodologia *Kanban* foi a sua utilização no sistema Toyota de Produção, promovido pela *Toyota Motor Corporation*, cujo objetivo é eliminar os desperdícios de produção que estavam afetando a companhia. De acordo com Monden (2012, p. 3),

Neste sistema, o tipo e a quantidade de unidades necessárias são escritos num cartão similar a uma etiqueta, chamando *Kanban*, que é enviado pelos trabalhadores em um dos processos para os trabalhadores no processo precedente.

Estes cartões circulam por toda a empresa e, desta forma, o *Kanban* é capaz de repassar as informações necessárias para que a equipe possa adequar a produção com base no que foi proposto. Ainda no sistema Toyota de Produção, Monden (2012) sustenta o *Kanban* pelos seguintes itens:

- Sincronização da produção
- Padronização das operações
- Redução do tempo de preparação
- Atividades de melhoria
- Projeto de leiaute das máquinas
- Automação (MONDEN, 2012, p. 9).

Para Fusco e Sacomano (2007, p. 91) “[...] o sistema Toyota de Produção é um método de produção e que o *Kanban* é apenas o método através do qual este sistema é gerenciado”. É possível enumerar as seguintes funções do *Kanban*:

- Prover a coleta ou transporte de informação;
- Prover informação de produção;
- Impedir a produção e o transporte em excesso;
- Servir como pedido de fabricação, que é anexado ao produto semi-acabado;
- Prevenir a saída de produtos defeituosos, identificando os processos que levam à sua produção;
- Revelar problemas existentes e manter o controle dos estoques;
- Instruções para utilização (FUSCO; SACOMANO, 2007, p. 91).

Para que seja possível atender os objetivos citados acima, o autor cita que o uso do *Kanban* deve basear-se nos seguintes aspectos:

- Um processo anterior considera o número de itens indicado pelo *Kanban* em um processo mais recente;
- Um processo mais recente produz itens na quantidade e seqüência indicados pelo *Kanban*;
- Nenhum item é produzido ou transportado sem o *Kanban*;
- Anexar sempre um *Kanban* aos bens produzidos;
- Produtos defeituosos não são enviados para os processos subseqüentes.
- A redução do número de *Kanbans* (cartões) aumenta sua sensibilidade. (FUSCO; SACOMANO, 2007, p. 91).

De acordo com Moura (1992), um dos significados que o *Kanban* possui é o fato de ser um sistema que visa a melhora da produtividade, mudando-se o equipamento, os métodos de trabalho e as práticas de movimentação de material, o qual usa os cartões para que seja possível identificar com problemas e avaliar as mudanças necessárias.

2.5 Extreme Programming

Extreme Programming (XP) é uma metodologia ágil de desenvolvimento de *software* que surgiu no final da década de 90, o qual vem a ser uma alternativa aos modelos tradicionais. Para Teles (2014, p. 24), o XP é uma metodologia voltada para:

- Projetos cujos requisitos são vagos e mudam com freqüência.
- Desenvolvidos de sistemas orientados a objetos.
- Equipes pequenas, preferencialmente até 12 desenvolvedores.
- Desenvolvimento incremental (ou iterativo), em que o sistema começa a ser implementado logo no início do projeto e vai ganhando novas funcionalidades ao longo do tempo.

Proposto em 1996 por Kent Beck, ao pensar em quais seriam os melhores caminhos para se desenvolver um *software* com base nas suas experiências já

vivenciadas. A concepção da metodologia veio através dos fatores que Kent Beck identificou como sendo os responsáveis por tornar o *software* fácil de ser criado, bem como os fatores que tornavam essa criação difícil. (AUDY; PRIKLANDNICKI; 2007). Desta forma, baseado nestas quatro dimensões, surgiram os valores fundamentais desta metodologia: comunicação, coragem, simplicidade e *feedback* (realimentar).

Sendo assim, o *Extreme Programming* é uma proposta de desenvolvimento de *software* que visa entregar o que o cliente solicitou e no prazo acordado com ele. Neste modelo é utilizada uma equipe única de trabalho, ao contrário do *Scrum* que preconiza um time de desenvolvimento a parte dos demais envolvidos no projeto, aqui gerentes, clientes e desenvolvedores formam uma equipe única dedicada a entregar um *software* com qualidade. (AUDY; PRIKLANDNICKI; 2007).

2.5.1 Comunicação

Assim como ocorrem nas demais metodologias, um cliente possui um conjunto de problemas que deseja solucionar através do desenvolvimento de funcionalidades, as quais comporão um projeto. Por sua vez, os desenvolvedores detêm o conhecimento necessário para criar as funcionalidades que o cliente necessita. Para que os desenvolvedores possam ser assertivos na solução dos problemas, é de fundamental importância que exista uma boa comunicação entre as partes.

A comunicação ocorre de forma contínua entre o cliente e a equipe de desenvolvedores, pois dessa forma o cliente está acompanhando de perto o progresso do desenvolvimento. E o cliente decide o que deve ser construído e em que ordem isso deve ocorrer (AUDY; PRIKLANDNICKI, 2007, p 25).

Embora existam inúmeras formas de se comunicar idéias, é fato que algumas formas são mais eficientes que outras. Por exemplo, em diálogos presenciais, inúmeros elementos são levados em consideração, tais como gestos, expressões faciais, postura, tom de voz, etc. Em comunicações por vídeo conferência, alguns destes elementos não são percebidos, assim como o uso de telefone e e-mails. Portanto, o método XP prioriza o tipo de comunicação presencial para que a comunicação seja entendida da melhor forma possível (LEITE, 2006a).

2.5.2 Coragem

Assim como ocorre em todas as metodologias, é comum o cliente optar por mudar o escopo do projeto ao longo do seu desenvolvimento. Conforme as entregas vão ocorrendo, é natural que o cliente perceba que uma funcionalidade não faz mais sentido para o projeto ou que a mesma é mais prioritária que a outra. Embora isto faça parte quando se fala em desenvolvimento de *software*, gera na equipe uma preocupação por precisar alterar partes do sistema que já estavam prontas, correndo o risco de quebrar o que já estava funcionando (LEITE, 2006b).

Dado que o sistema é desenvolvido de forma incremental, a equipe está continuamente fazendo manutenção do *software* e criando novas funcionalidades. Em muitos casos, ela irá alterar algo que vinha funcionando corretamente, o que leva ao risco de gerar falhas no sistema. Por esta razão, a equipe precisa ser corajosa e acreditar que, utilizando as práticas e valores do XP, será capaz de fazer o *software* evoluir com segurança e qualidade. (TELES, 2014, p. 26).

2.5.3 Feedback

Tendo em vista que a construção de um projeto é um empreendimento que possui altos riscos, os desenvolvedores devem buscar *feedbacks* o quanto antes, para que eventuais falhas que venham a ocorrer sejam corrigidas o mais cedo possível, de forma que o esforço de retrabalho seja o menor possível a um custo sustentável pelo projeto. “Um princípio psicológico bem conhecido indica que para maximizar a taxa de aprendizado, a pessoa precisa receber *feedback* sobre quão bem ou mal ele está indo (WEINBERG, 1971, p.102, tradução nossa).”

Desta forma, o *Extreme Programming* é organizado para que se tenham *feedbacks* com maior frequência, possibilitando o cliente avaliar a funcionalidade entregue e com ela passar para o grupo seu retorno quanto a qualidade do produto. O time somente irá seguir adiante se a entrega for aprovada pelo cliente, pois caso seja identificada uma falha, ainda há tempo hábil para consertar o problema sem impactar o restante do projeto. “A razão básica para estratégias incrementais e iterativas é permitir que os inevitáveis erros das pessoas sejam descobertos relativamente cedo e reparados de forma metódica (COCKBURN, 2002, p.49, tradução nossa).”

2.5.4 Simplicidade

Diferentemente das metodologias tradicionais, o *Extreme Programming* preconiza que os desenvolvedores devem entregar primeiro aquilo que é essencial para o projeto e se evite desenvolver aquilo que ainda não está claramente definido como prioritário pelo cliente. Este receio é justificado pelo fato de que quando o cliente não tem certeza exata daquilo que ele quer, ele costuma adicionar no escopo do projeto tudo o que ele acredita que possa vir a precisar, inclusive itens desnecessários. No modelo de desenvolvimento tradicional, devido a falta de aproximação entre cliente e desenvolvedores, isto somente será percebido na etapa final do projeto. Já nos métodos ágeis, especialmente no *Extreme Programming*, com a adoção de um modelo de desenvolvimento iterativo, com *feedbacks* de curto prazo, no início do desenvolvimento o cliente já poderá passar suas primeiras impressões. (POPPENDIECK; POPPENDIECK, 2003).

Visto que os clientes normalmente não sabem direito o que desejam no início do projeto, eles tendem a pedir tudo o que acham que podem vir a precisar, especialmente se eles pensarem que só terão uma única chance de pedir. Esta é uma das melhores maneiras que conhecemos para aumentar o escopo do projeto muito além do necessário para alcançar a missão geral do projeto (POPPENDIECK; POPPENDIECK, 2003, p.32-33, tradução nossa).

2.5.5 Práticas

Para que seja possível aplicar os valores no desenvolvimento de *softwares*, de acordo com Teles (2014), o *Extreme Programming* baseia-se em 13 práticas, as quais são abordadas abaixo.

2.5.5.1 Cliente Presente

O *Extreme Programming* recomenda que o cliente esteja presente junto aos desenvolvedores para que as dúvidas de requisitos sejam sanadas rapidamente, evitando assim atrasos ou até mesmo falha nas funcionalidades desenvolvidas em virtude de uma má interpretação por parte dos desenvolvedores.

Ter especialistas de domínio à disposição o tempo todo significa que o tempo de *feedback* entre a solução ser imaginada e depois avaliada é o mais curto possível, freqüentemente de minutos ou algumas poucas horas. Tal rapidez no *feedback* significa que a equipe de desenvolvimento ganha uma compreensão mais profunda das necessidades e hábitos dos usuários e comete menos erros quando cria novas idéias. Eles tentam mais idéias diferentes, o que faz com que o produto final fique melhor. Havendo boa dose de colaboração, os programadores irão testar as idéias dos especialistas de domínio e oferecer contrapropostas. Isso irá aperfeiçoar as próprias idéias do cliente sobre como o sistema deve se parecer (COCKBURN, 2002, p.179, tradução nossa).

Embora que ter o cliente junto ao time o tempo inteiro durante o desenvolvimento seja o ideal, sua presença física não é o único ponto de atenção. O cliente precisa ser capaz de dar o devido direcionamento do valor do negócio quando houver dúvidas ao longo do projeto.

2.5.5.2 Jogo do Planejamento

Para Astels, Miller e Novak (2002), em se tratando de desenvolvimento de *software*, indiferente do tamanho do projeto, é necessário que seja elaborado um planejamento, contudo, não de forma detalhada que resulte uma grande demanda de tempo do time de desenvolvedores. Como o planejamento pode variar de acordo com as necessidades do negócio, é natural que em projetos grandes haja uma maior falta de precisão.

No XP, o planejamento é utilizado para que a equipe esteja trabalhando, a todo o momento, naquilo que é mais importante para o projeto. Desta forma, a exemplo do *Scrum*, o XP é dividido em *releases* e iterações, de modo que seja possível a equipe se reunir com o cliente e revisar o planejamento.

2.5.5.3 Stand up Meeting

Tendo como objetivo que as pessoas trabalhem em conjunto, o XP recomenda que se façam reuniões diárias chamadas de *stand up meeting*. Da mesma forma que utilizada no *Daily Scrum*, o objetivo é fazer com que o time esteja alinhado com o que foi feito no dia de ontem e o que se pretende fazer hoje. “Um dia de trabalho de uma equipe XP sempre começa com um stand up meeting. (...)

Primeiramente, ele serve para que todos os membros da equipe comentem rapidamente o trabalho que executaram no dia anterior (TELES, 2004, p.87)”.

É neste momento que a equipe tem oportunidade de trocar experiências e ter uma visão geral do projeto, forçando assim a aproximação e troca de informações de todos os membros.

2.5.5.4 Programação em par

Esta prática sugere que dois programadores trabalhem juntos em um mesmo computador. A dupla é formada com base em dois perfis: um profissional mais experiente e outro iniciante. A intenção é que o iniciante trabalhe na codificação do código enquanto que o mais experiente acompanha a programação e apóia no desenvolvimento das habilidades do colega (PIMENTEL;FUKS, 2011).

Com o uso da prática programação em pares, o código que está sendo desenvolvido sempre é revisado por duas pessoas e em tempo real, dirimindo as possibilidades de erros e potencializando a melhora do código fonte (PIMENTEL;FUKS, 2011).

2.5.5.5 Desenvolvimento guiado pelos testes

Ao longo do tempo, seja devido as alterações em funcionalidades mal desenvolvidas ou códigos mal projetados, os sistemas tendem a apresentar defeitos. A prática desenvolvimento guiado pelos testes é uma prática que consiste em escrever um roteiro de teste automatizado antes de codificar cada história e cada método do sistema (BECK, 2000).

2.5.5.6 Refatoração

Na medida em que um sistema é utilizado e novas funcionalidades são desenvolvidas, o sistema tende a se deteriorar ao longo do tempo, tornando-o lento e desorganizado. A refatoração é o ato de se modificar o código sem que as

funcionalidades sejam alteradas. Estas alterações visam aplicar pequenas melhorias no código do sistema, fazendo com que ele seja mais simples e de fácil compreensão.

Havendo uma cadência para esta prática, o resultado é um sistema compacto, ágil e quando houver a necessidade de se incluir uma nova funcionalidade, o desenvolvimento será realizado de forma mais fácil.

2.5.5.7 Código coletivo

Em continuidade à prática programação em par, no código coletivo o XP sugere que o programador tenha acesso a todo o código desenvolvido, inclusive aqueles que não foram criados por ele: “todas as classes e métodos pertencem à equipe e qualquer membro da equipe pode melhorar o que for necessário. (JEFFRIES; ANDERSON; HENDRICKSON, 2001, p.122, tradução nossa).”

Além disto, um dos benefícios na adoção desta prática é o fato de toda a equipe conhecer as mais variadas partes do sistema, tornando assim a equipe mais coesa.

2.5.5.8 Código padronizado

A adoção de um padrão para o desenvolvimento do código do sistema ajuda a simplificar a comunicação. “Não importa muito o formato. O que realmente importa é que todos os membros da equipe adotem o padrão e o utilizem sempre. (...) não são as especificidades (...) que contam; é a sua familiaridade com elas” (JEFFRIES; ANDERSON; HENDRICKSON, 2001, p.79, tradução nossa).

2.5.5.9 *Design* simples

No XP as equipes de desenvolvimento tentam manter o design o mais simples possível, não implementando alterações até que sejam realmente

necessárias. Através do *feedback* do usuário, identifica-se o correto ponto do sistema que se deve efetuar as implementações necessárias.

“Um design complicado é mais difícil de comunicar que um simples. Devemos, portanto, criar uma estratégia de design que gere o design mais simples possível, consistente com nossos demais objetivos (BECK, 2000, p.103, tradução nossa)”.

2.5.5.10 Metáfora

As metáforas são usadas freqüentemente no *Extreme Programming*, uma vez que os desenvolvedores criam situações no ambiente de programação e há a necessidade de, na comunicação com outras pessoas, fazer uma analogia destas situações para o mundo real. Podem-se citar como exemplo destes elementos os itens painel de controle, lixeira, arquivos, etc dos sistemas operacionais. No XP não é diferente, os desenvolvedores usam metáforas para facilitar a comunicação com o cliente, mitigando assim futuras falhas que poderiam vir a ocorrer devido a uma especificação de baixa qualidade.

Cockburn (2002) diz que:

1. O programador, tendo a teoria do programa, consegue explicar como a solução se relaciona com as questões do mundo real que ele procura tratar. (...)
2. O programador, tendo a teoria do programa, consegue explicar *porque* cada parte do programa é o que é (...).
3. O programador, tendo a teoria do programa, é capaz de responder construtivamente a qualquer requisição de modificação do programa, de modo a suportar as questões do mundo real de uma nova maneira (COCKBURN, 2002, p.231-232, tradução nossa).

Esta forma de utilização de metáforas é um potente mecanismo que auxiliará para tornar o sistema mais fácil e intuitivo de se utilizar.

2.5.5.11 Ritmo sustentável

Segundo Brooks (1995), cada vez mais o fator tempo tem influenciado para que os projetos de *softwares* não cumpram o que foi negociado com o cliente.

Quando isto ocorre, geralmente o time de desenvolvimento precisa prolongar sua jornada de trabalho para que o projeto seja entregue nos prazos acordados.

Na metodologia *Extreme Programming*, recomenda-se que a carga horária de trabalho seja limitada a oito horas diárias, evitando ao máximo a execução de horas extras. As atividades é que devem ser enquadradas na jornada de trabalho da equipe, conforme sua prioridade (BECK; FOWLER, 2001).

2.5.5.12 Integração contínua

No XP, as equipes de desenvolvimento trabalham em pares de acordo com o código coletivo, no entanto, esta prática associada com a integração contínua requer certo cuidado. “Sempre que diversos indivíduos estão trabalhando na mesma coisa, ocorre uma necessidade de sincronização (POPPENDIECK; POPPENDIECK, 2003, p.34-35, tradução nossa).”

Para resolver esta questão, a prática integração contínua sugere que os programadores trabalhem de forma isolada, porém em diversos momentos do dia eles entregam as versões mais recentes dos códigos produzidos, ou seja, cria-se uma sincronização para a atualização do código (BECK, 2000).

2.5.5.13 Releases curtos

No XP “um projeto de *software* é um investimento. O cliente investe uma certa quantidade de recursos na expectativa de obter um retorno dentro de certo prazo (TELES, 2004, p.185)”. A fim de garantir o maior retorno possível, o XP procura entregar valor de negócio em cada *release* e que cada uma tenha curta duração.

Desta forma, esta prática objetiva colocar o sistema em produção de forma recorrente, visto que “clientes gostam de entregas rápidas. (...) entrega rápida normalmente se traduz em aumento de flexibilidade no negócio (POPPENDIECK; POPPENDIECK, 2003, p.70-71, tradução nossa)”.

2.6 Indicadores de desempenho para projetos ágeis

Um indicador de desempenho é uma métrica, ou conjunto de métricas, que possibilita que haja uma visão do sistema. Através do indicador é possível que o gerente de projeto ou o time que o desenvolve possa avaliar a evolução do projeto ou produto e incluir melhorias (PRESSMAN, 2006).

Uma métrica é um número que representa um determinado estado do *software*, de um projeto como todo ou da equipe que o desenvolve. Cohn (2005) sugere que as métricas ágeis devam ter uma série de critérios, tais como: seguir tendências e não números, reforçar os princípios ágeis, ser de fácil coleta, fornecer *feedback*, etc.

Métricas podem ser úteis para ajudar o time a compreender onde estão, e ajudá-los a comparar com o estado em que querem estar. Assim como as metas, as métricas também devem estar sempre explícitas e visíveis para o time (GOMES, 2013, p 103).

2.6.1 Tipos de indicadores

Cohn (2005) e Gomes (2013) recomendam o uso de dois tipos de indicadores:

- a) **Indicadores indutores:** auxilia a detectar um aumento da probabilidade de ocorrência de um evento ou na sua gravidade, antes que o mesmo seja acusado nos indicadores de resultado. Quando há uma alteração no indicador indutor, significa que pode-se estar no caminho para atingir um objetivo ou meta.
- b) **Indicadores de resultado:** são métricas que verificam se o objetivo foi atingido depois de ter completado um determinado trabalho, contudo, não é possível que o resultado seja alterado.

Gomes (2013) sugere que se deva medir times e não indivíduos:

Quando as pessoas acreditam que serão afetadas pelos resultados de métricas sobre seus trabalhos individuais, elas rapidamente encontram formas de corromper tais métricas, além disso, medir indivíduos pode destruir a colaboração do time, uma vez que ajudar um outro membro do time pode significar melhorar a métrica de desempenho dele e prejudicar a sua própria. Já quando mede-se o time em vez do indivíduo, já que a colaboração tende a melhorar a performance do time como um todo, a colaboração entre as pessoas será mais incentivada (GOMES, 2013, p. 107).

Ainda de acordo com Gomes (2013), o autor sugere que os seguintes indicadores sejam aplicados para projetos ágeis:

- **Velocidade do time:** quantidade de pontos de história que um time consegue entregar na iteração.
- **Aceleração:** alteração na velocidade do time ao longo das iterações.
- **Prazo de execução:** tempo entre o pedido do cliente e a entrega da funcionalidade.
- **Horas trabalhadas:** somatório das horas dedicadas ao projeto por todos os membros da equipe.
- **Lucro:** influência das entregas no lucro obtido através dos incrementos do produto que está em desenvolvimento.

Existem diversos outros tipos de indicadores, os quais podem variar de projeto para projeto, fazendo sentido em determinada situação e em outra não. Para Cruz (2013), o gráfico de *Burndown* no *Scrum* é amplamente utilizado para mostrar para o time do projeto como está o avanço em relação ao que foi planejado. O gráfico deverá mostrar duas linhas, das quais a primeira refere-se ao avanço esperado após o planejamento da *sprint* e a segunda com o real avanço diário do projeto. A sugestão é de que o *Burndown* seja atualizado pelo time todos os dias para que se possa ter uma visão real da produção da equipe do projeto.

Sempre que um trabalho é concluído o gráfico desce, ao acrescentar um novo trabalho o gráfico sobe e quando um trabalho é reestimado também sobe. De acordo com Dimes (2014), os gráficos de *Burndown* permitem que o time veja se o produto

sendo desenvolvido será liberado dentro do prazo estipulado ou se sofrerá atraso. Havendo risco de ser liberado em atraso, o time deverá fazer os ajustes necessários, seja trabalhando de forma mais rápida ou removendo determinadas tarefas do *sprint*.

Uma alternativa ao gráfico de *Burndown* é o *Burnup*, onde diferentemente do primeiro citado, na medida em que o trabalho vai sendo concluído, o gráfico sobe (Gomes, 2013). Alguns times optam por utilizar o *Burnup* pelo fato de ficar mais evidente as alterações no escopo.

Ambos os gráficos representam o progresso do trabalho, a diferença se dá na sua forma de apresentação, os quais podem fornecer informações importantes, tais como:

- Quanto de trabalho a equipe do projeto é capaz de entregar?
- As entregas planejadas serão realizadas dentro do esperado?
- Qual será a data de conclusão aproximada do projeto?

3 MÉTODOS E PROCEDIMENTOS

Conforme abordado nas literaturas, pesquisa é um “conjunto de procedimentos sistemáticos, baseado no raciocínio lógico, que tem por objetivo encontrar soluções para os problemas propostos, mediante a utilização de métodos científicos” (ANDRADE, 2010, p. 110).

No que tange a metodologia, é definida como “conjunto de métodos ou caminhos que são percorridos na busca do conhecimento” (ANDRADE, 2010, p. 117). Já o método é “um caminho, uma forma, uma lógica de pensamento” (VERGARA, 2010, p. 3).

Este capítulo tem como objetivo delinear o método de pesquisa a ser utilizado neste trabalho de conclusão de curso e indicar as técnicas de coleta e de análise de dados utilizadas. Por fim, descreve as limitações que o método apresentou na pesquisa.

3.1 Delineamento da Pesquisa

Para a realização deste trabalho foi utilizado o tipo de pesquisa qualitativa baseada nos referenciais teóricos. “Uma pesquisa bibliográfica pode ser desenvolvida como um trabalho em si mesma ou constitui-se numa etapa da elaboração de monografias, dissertações, etc” (ANDRADE, 2010, p. 45). Além disto, utilizou-se o método de estudo de caso, o qual pode envolver tanto situações de estudo de um único caso quanto situações de estudo de múltiplos casos (MILES; HUBERMAN, 1994).

A pesquisa com caráter qualitativo tem sido amplamente utilizada em estudos voltados para a compreensão da vida humana em grupos, em campos como sociologia, antropologia, psicologia, dentre outros das ciências sociais. Esta abordagem tem tido diferentes significados ao longo da evolução do pensamento científico, mas pode-se dizer, enquanto definição genérica, que abrange estudos nos quais se localiza o observador no mundo, constituindo-se num enfoque naturalístico e interpretativo da realidade (DENZIN; LINCOLN, 2000). O “método qualitativo é subjetivo e envolve examinar e refletir as percepções para obter um entendimento de atividades sociais e humanas” (COLLIS; HUSSEY, 2005, p.26).

Para a elaboração desta pesquisa foi utilizado o método de estudo de caso, o qual se enquadra como uma abordagem qualitativa e é freqüentemente utilizado para coleta de dados na área de estudos organizacionais. Segundo Yin (2010), o método de estudo de caso é:

Uma investigação empírica que

- investiga um fenômeno contemporâneo em profundidade e em seu contexto real, especialmente quando
- os limites entre o fenômeno e o contexto não são claramente evidentes (YIN, 2010, p. 39).

Entende-se como caso uma unidade de análise, que pode ser um indivíduo, o papel que o mesmo desempenha ou uma organização, um pequeno grupo, uma comunidade ou até mesmo uma nação. No entanto, a definição de casos também pode ser definida temporariamente, eventos que ocorreram num dado período, ou espacialmente, estudo de um fenômeno que ocorre num dado local. Desta forma, um caso pode ser um fenômeno simples ou complexo, mas para ser considerado caso ele precisa ser específico (LINCOLN, 2001).

Yin (2010) afirma que,

A investigação do estudo de caso

- enfrenta a situação tecnicamente diferenciada em que existirão muito mais variáveis de interesse do que pontos de dados, e, como resultado
- conta com múltiplas fontes de evidência, com os dados precisando convergir de maneira triangular, e como outro resultado
- beneficia-se do desenvolvimento anterior das proposições teóricas para orientar a coleta e análise de dados (YIN, 2010, p. 40).

Esta pesquisa classifica-se como descritiva e exige do investigador uma série de informações sobre o que deseja pesquisar. Esse tipo de estudo pretende descrever os fatos e fenômenos de determinada realidade (TRIVIÑOS, 1987). Ainda segundo Triviños (1987), o ambiente, o contexto onde os indivíduos realizam suas ações e desenvolvem seu modo de vida, têm importância essencial na compreensão mais clara de suas atividades. O tipo de pesquisa descritiva tem por premissa buscar a resolução de problemas melhorando as práticas por meio da observação, análise e descrições objetivas, através de entrevistas com peritos para a padronização de técnicas e validação de conteúdo (THOMAS; NELSON; SILVERMAN, 2007).

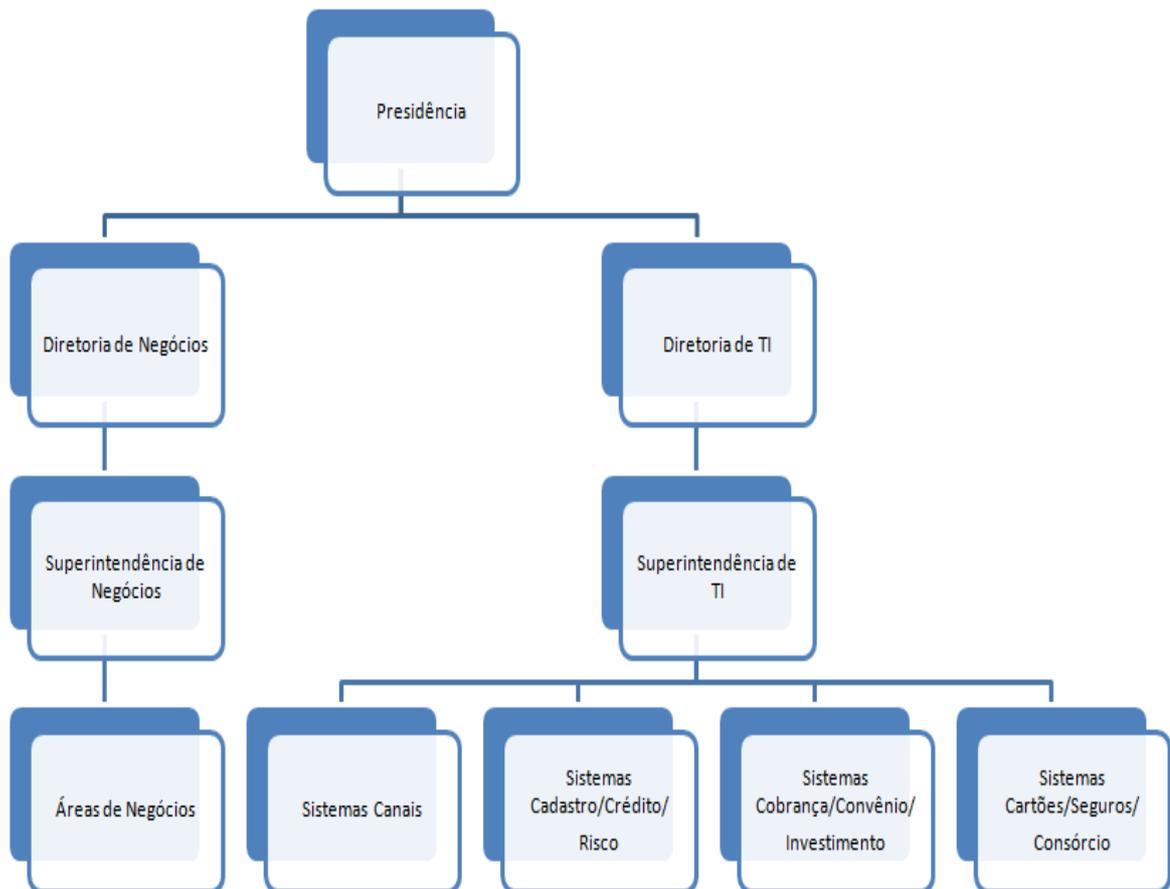
3.2 Definição da Unidade-Caso e dos Sujeitos da Pesquisa

A organização na qual se realizou o estudo de caso foi a empresa x. Atuante no ramo bancário, está no mercado há mais de 20 anos com o desenvolvimento de produtos serviços voltados para o agronegócio. O motivo pelo qual esta empresa foi escolhida é pelo fato de o pesquisador ser funcionário da mesma.

Na figura ilustrada abaixo consta o organograma da empresa, a qual detalho a seguir:

- a) Presidente e diretores executivos, responsáveis pelo direcionamento estratégico;
- b) Superintendentes, possuem a visão macro das linhas de serviços e garantem que o alinhamento estratégico esteja sendo seguido;
- c) Áreas de Negócios, responsáveis por identificar as tendências tecnológicas do mercado. Na empresa objeto do estudo de caso existem diversas subáreas, cada uma responsável por uma linha de produto do Negócio, porém para esta pesquisa não considero necessário abordá-las.
- d) Áreas de Sistemas, responsáveis por traduzir a necessidade do negócio em TI. São subdivididas por produtos, tais como Canais, Convênios, Investimentos, Cobrança, dentre outras. Nestas subáreas estão lotados os Gerentes de Projetos, Analistas, Testadores, Desenvolvedores, Arquitetos, bem como todos os demais perfis necessários para a execução completa de um ciclo de desenvolvimento de *software*.

Figura 14 - Organograma da Empresa



Fonte: Elaborado pelo autor.

3.3 Técnicas de Coleta de Dados

Para obter os dados utilizaram-se as seguintes técnicas de coleta de dados: pesquisa bibliográfica, observação participante, entrevistas em profundidade e documentos.

- a) **Pesquisa bibliográfica:** para que fosse possível identificar métodos ágeis para gerenciamento de projetos, foram reunidas diversas bibliografias relacionadas ao tema, sendo a principal fonte os livros que abordam este tipo de assunto. Andrade (2010) define a pesquisa da seguinte forma: “Tendo em mãos uma lista de obras identificadas como fontes prováveis para determinado assunto, procura-se localizar as informações úteis, por meio das leituras” (ANDRADE, 2010, p. 46).

Desta forma, através das leituras realizadas foram selecionadas as principais idéias abordadas pelos autores, as quais estão referenciadas neste trabalho, para auxiliar na solução da problemática.

- b) **Observação participante:** de acordo com Gil (1994), a técnica observação participante ou observação ativa é utilizada quando o observador está inserido dentro do contexto que está sendo analisado, onde o mesmo assume um papel de membro do grupo. Conforme citado em tópicos anteriores, o pesquisador deste trabalho atua na área de TI da empresa estudada e participa da gestão dos projetos.
- c) **Entrevistas em profundidade:** a técnica de entrevista é uma técnica de coleta de dados na qual o entrevistador tem um contato próximo com a pessoa. Segundo Yin (2010), as entrevistas, dentre os diversos outros métodos, é uma das mais ricas para análise de dados. As entrevistas “[...] também são fontes essenciais de informações para os estudos de caso. As entrevistas são conversas guiadas, não investigações estruturadas” (YIN, 2010, p.133). Para a realização deste trabalho, as pesquisas foram realizadas entre os meses de maio e junho de 2015. No apêndice A deste trabalho está o questionário que foi aplicado aos entrevistados.
- d) **Documentos:** Para Gil (1994) as fontes escritas são, na maioria das vezes, de grande valia e ajudam o pesquisador a não perder tanto tempo na hora da busca de material, sabendo que em determinadas circunstâncias só é possível a investigação através de documentos. Para a avaliação e interpretação dos dados coletados, é de extrema importância que o pesquisador seja cuidadoso, pois há risco de ocorrer uma má interpretação de algum determinado assunto ou uma análise tendenciosa, acarretando assim riscos ao trabalho realizado (YIN, 2010). Para este trabalho o pesquisador teve acesso a documentos relacionados a execução dos projetos, definições de escopos, relação de funcionários atuantes nos projetos, cronograma e andamento das fases do projeto.

3.4 Técnica de Análise de Dados

A análise de dados deste trabalho procurou relacionar o gerenciamento ágil em projetos e as proposições do objetivo geral, bem como os objetivos específicos. Para este estudo de caso, utilizou-se também a triangulação de dados, que segundo Vergara (2010), a técnica sugere a utilização de dados para viabilizar o estudo que se está propondo, de forma que seja possível obter novas visões e conhecimentos do caso. Seguindo nesta mesma linha, Yin (2010), afirma que com esta técnica pode mais de um ponto, as quais proporcionam várias avaliações do mesmo assunto.

Para esta pesquisa utilizou-se a análise qualitativa dos dados, onde foi possível compilar as informações coletadas e relacionar a gestão atual dos projetos para com as expectativas dos seus usuários. Através desta análise foi possível o pesquisador identificar os aspectos que apresentam problemas no gerenciamento de projetos atual da empresa estudada e encontrar soluções frente as bibliografias estudadas.

A triangulação de dados entre os conteúdos pesquisados e fundamentados apresentados no referencial teórico, os métodos de gestão de projetos da empresa estudada e as pesquisas com os funcionários da empresa.

3.5 Limitações do Método

A análise qualitativa é tratada por muitos autores como uma metodologia subjetiva que pode levar o pesquisador ao equívoco. Contudo, segundo Yin (2010), esta mesma consideração também se aplica para outros métodos de pesquisa científica onde o pesquisador pode não ter as habilidades necessárias para utilização do método. Desta forma, indiferentemente de qual método seja escolhido, é importante que os resultados estejam de acordo com o que o trabalho se propôs.

Esta pesquisa se limitou a estudar o contexto atual da empresa x e sua metodologia para gerenciamento de *softwares*. Pelo fato de existirem projetos de diferentes naturezas, o resultado deste trabalho não pode ser aplicado a todos projetos. Também não faz parte do objetivo implantar o gerenciamento ágil de

projetos, mas sim avaliar a sua viabilidade frente ao contexto atual da empresa estudada.

4 APRESENTAÇÃO E ANÁLISE DOS DADOS

Conforme citado na seção introdutória e situação problemática deste trabalho, a empresa x atua no ramo bancário, contudo, é na zona rural a sua maior rede de clientes e foco das suas estratégias de negócio. No que tange a área de TI, a estrutura da empresa é composta por uma diretoria de negócios e outra de TI.

As áreas de negócios são as responsáveis por identificar as necessidades de produtos para a organização com base na visão de mercado. É por meio destas áreas internas que as necessidades são endereçadas para as áreas de TI desenvolverem os produtos solicitados.

Por sua vez, são nas áreas de TI que ficam alocados os times de desenvolvimento, dentre eles o gerente de projeto e os demais membros. Atualmente todos os projetos, seja o mais complexo ou o mais simples de ser executado tem como modelo de gerenciamento o modelo tradicional. No entanto, a empresa estudada entende que não tem conseguido entregar valor para os seus clientes na velocidade desejada. Por conta disto, no ao longo dos últimos anos, tem-se percebido que a margem de lucro da organização reduz significativamente, da mesma forma que sua visibilidade no mercado bancário.

Com o objetivo de identificar uma solução como alternativa ao contexto atual, a empresa x tem avaliado a possibilidade de utilizar na sua gestão de projetos os métodos ágeis, de forma que a entrega de novas funcionalidades e produtos cheguem aos seus clientes com mais velocidade e eficiência.

Tendo em vista as necessidades da empresa x identificadas neste trabalho, a análise de dados das informações foi realizada com base na triangulação de dados entre as pesquisas realizadas e apresentadas no referencial teórico, a observação participante e entrevistas com os funcionários que atuam na área de projetos da organização.

Através da observação participante, pode-se identificar os diferentes processos da organização para o gerenciamento de projetos, seus pontos fortes e oportunidades de melhoria. Além disto, com o acesso aos documentos relacionados ao gerenciamento dos projetos foi possível haver um melhor entendimento do contexto geral da empresa x.

4.1 Situação da empresa e entrevistas

As entrevistas realizadas com os membros dos times de gerenciamento de projetos, os quais utilizam atualmente a metodologia tradicional para gestão de projetos, seguiram o questionário de entrevistas que se encontra no Apêndice A deste trabalho. Pode-se entender como membro dos times os profissionais que estão lotados nas áreas de negócios e sistemas, os quais atuam de forma parcial ou tempo integral no gerenciamento de projetos. Foram entrevistados profissionais que atendiam o seguinte perfil:

- a) **Analista:** com mais de 2 anos de experiência no negócio da empresa.
- b) **Desenvolvedor:** com mais de 2 anos de experiência no negócio da empresa.
- c) **Líder de projeto:** com mais de 3 anos de experiência no negócio da empresa.

Foram realizadas 10 entrevistas semi-estruturadas, das quais três com líderes de projetos, dois desenvolvedores e cinco analistas de diferentes papéis, tais como de sistemas, testes e requisitos. Quando os entrevistados foram questionados a respeito dos pontos negativos do modelo atual para gestão de projetos, grande parte entende que a velocidade das entregas não atende as necessidades da organização. O método atual é muito burocrático, exige que seja criada uma vasta documentação a respeito do escopo que se está desenvolvendo, onerando assim um tempo precioso da equipe que poderia estar focada em outras atividades mais essenciais para o projeto. Para todos os entrevistados que apontaram a velocidade das entregas como sendo o principal ponto fraco da metodologia tradicional, é consenso que se fosse aplicado métodos ágeis para o gerenciamento destes projetos, as entregas seriam realizadas com uma frequência maior. Da mesma forma, a área de negócio teria uma percepção de retorno do projeto mais rapidamente, uma vez que poderiam ser entregues etapas intermediárias antes da conclusão do projeto. Para Cruz (2013), é fundamental que o time de desenvolvimento do produto possa entregar as funcionalidades de forma rápida e

eficiente, de modo que a velocidade deste time possa ser medida para que melhores resultados possam ser obtidos.

Dentre outros apontamentos relatados pelos entrevistados, é possível relacionar os seguintes:

- **Definição do escopo:** muitos entendem que há um distanciamento considerável entre as áreas de negócios e as áreas de sistemas. Como consequência disto, as áreas de sistemas por vezes possuem uma má interpretação do escopo do projeto, as quais são responsáveis por fazer o detalhamento do que se deve desenvolver e então atuar junto as demais áreas de sistemas participantes do projeto. Quando uma entrega é realizada, geralmente já se passou bastante tempo desde a concepção do projeto. Não é incomum ocorrerem desalinhamentos com o que a área de negócios solicitou e o que foi desenvolvido, acarretando assim em retrabalho para o time de desenvolvimento. De acordo com Cruz (2013) e Sabbagh (2013), é essencial que haja uma aproximação do time do projeto com o cliente para que seja possível entender os requisitos que o projeto deve atender.
- **Processos de TI:** os processos das áreas de TI também foram apontados por diversos entrevistadores como sendo um dos fatores de insucesso no método tradicional. Atualmente a TI estabelece que para cada aprovação de execução de um projeto determinadas áreas chave da organização devem validar o escopo, no entanto, o prazo para esta validação é muito oneroso e não atende aos prazos do negócio. Para cada etapa do ciclo de desenvolvimento do método tradicional uma quantidade significativa de documentação é necessária, tais como excessivos formulários de aprovação e execução das entregas realizadas. Uma oportunidade de melhoria sugerida por uma parte dos entrevistados é evitar que alguns destes documentos fossem desenvolvidos ou que talvez fossem automatizados.
- **Garantia dos serviços:** garantia se refere ao período de 90 dias, a contar da data de entrega do projeto, que o mesmo não pode gerar

incidente. Considerando os aspectos desfavoráveis citados acima, alguns entrevistados consideram que grande parte das falhas identificadas em produção ocorre pelo fato de o responsável pelo projeto na área de negócios não atuar diretamente com a equipe que o desenvolve. Caso fossem adotadas práticas ágeis para gerenciamento de projetos, uma aproximação deste responsável seria necessário, bem como outras ações sugeridas na metodologia ágil, o que resultaria num melhor entendimento do escopo e redução das falhas quando o sistema for testado ou colocado em produção (CRUZ, 2013).

Quando questionados a respeito da liberdade que possuem para traçar as estratégias necessárias a fim de atingir os objetivos propostos, a maioria dos entrevistados respondeu que a equipe não possui a autonomia necessária para isto. Esta responsabilidade cabe, em parte, ao líder do projeto ou gerente da área de sistemas.

Quando questionados referente ao que percebem de positivo no modelo atual, muitos responderam que o método tradicional é eficaz e necessário para a organização em determinados tipos de projetos, tais como aqueles que possuem curto prazo de vigência e com um escopo muito bem definido e detalhado junto a área requisitante. Contudo, para projetos mais extensos ou complexos, o modelo tradicional não atende ao negócio da empresa, conforme exemplos citados anteriormente, havendo assim oportunidade para se avaliar o uso dos métodos ágeis para o gerenciamento dos projetos.

Por fim, questionados sobre a possibilidade de não eliminar o modelo tradicional, mas sim combiná-lo com as práticas ágeis, os entrevistados convergiram para esta idéia, entendendo como viável e benéfico para a organização. Contudo, é importante atentar para a forte questão cultural da empresa. Os entrevistados entendem que a adoção por si só da metodologia ágil não será a solução dos problemas. Acreditam que seja preciso haver uma mudança nas pessoas, onde uma relação mais próxima se estabeleça entre as “pessoas chave” do projeto, onde todos caminhem na mesma direção, com foco naquilo que realmente é essencial para o projeto.

4.2 Principais metodologias para gerenciamento ágil de projetos

Conforme abordado no referencial teórico, foi objetivo deste trabalho identificar as principais metodologias ágeis para gerenciamento de projetos e detalhá-las afim de que as melhores praticas possam ser identificadas e sugeridas para implementação na empresa x.

No método *Scrum*, dentre as suas diferentes aplicabilidades, os autores sugerem esta metodologia quando a organização possui a necessidade de realizar entregas freqüentes para seus clientes. São desenvolvidas preferencialmente as funcionalidades de maior valor para o negócio, afim de que antes mesmo da entrega do produto final já seja possível os clientes terem um retorno sobre o investimento. No contexto da empresa estudada, esta prática tende a suprir uma necessidade que o método tradicional não atende. Atualmente as áreas de negócios somente possuem visibilidade do produto quando o mesmo é concluído.

Outro benefício deste método ágil é com relação a redução de riscos, com a aproximação de todas as pessoas envolvidas no projeto, o escopo é melhor detalhado e as entregas são realizadas em curtos prazos, a fim de que seja possível coletar um *feedback* do cliente. Conforme citado pelos autores, em caso de mudança no planejamento, o esforço para isto será menor do que se a entrega fosse única e na conclusão do projeto.

Já no método ágil *Extreme Programming*, através das referências bibliográficas coletadas, pode-se citar como um de seus benefícios a utilização das histórias de usuários para definir quais funções o sistema de negócio deve fornecer. Como as mesmas não são estáticas, pode-se incrementar novas histórias ou removê-las de acordo com a necessidade levantada pela área de negócio. O planejamento, em oposição a metodologia tradicional, é realizado ao longo do projeto e o cliente pode definir o que é prioridade para ele.

Um outro conceito importante do *XP* é que sugere-se que o desenvolvimento das funcionalidades seja feito de forma incremental, desta forma, no caso de o cliente não estar satisfeito com uma entrega, a equipe do projeto não precisará avaliar todo o trabalho realizado até então. Para a empresa x, a adoção desta prática resolveria o problema atual no que tange a qualidade do produto que a TI está entregando com o uso do método tradicional.

A última metodologia ágil estudada foi o *Kanban*, onde de acordo com os autores pesquisados, o uso deste método tende a aumentar a produtividade da equipe, uma vez que os processos são mapeados, permitindo assim um panorama geral do trabalho. Outra grande vantagem na utilização do *Kanban* é com relação a redução de custos. Esta redução se deve ao fato de não se gerar estoques, evitando assim gastos com armazenagem. Somente é produzido aquilo que é capaz de ser suportado pela linha de montagem e absorvido pelo negócio.

4.3 Indicadores de desempenho para gerenciamento ágil de projetos

Conforme abordado no referencial teórico, indicador de desempenho é um instrumento utilizado para a avaliação de um trabalho, no contexto da empresa x, a qualidade dos seus projetos. Conforme bibliografias estudadas, pode-se aferir que o uso de indicadores pode ser utilizado para medir resultados, avaliar tendências e responder perguntas específicas para uma parte interessada no projeto.

Para avaliar a velocidade em que ocorrem as entregas do time do projeto ou medir o quanto das funcionalidades planejadas para a iteração ainda faltam ser implementadas, este trabalho sugere a adoção dos gráficos *Burndown* e *Burnup*.

Com a adoção do indicador *Burndown*, em uma determinada situação, pode-se chegar a conclusão que a equipe do projeto não conseguiu entregar o escopo que foi planejado. Dentre as possíveis causas para isto ocorrer, pode-se avaliar que o planejamento foi mal realizado, seja por falha da equipe do projeto ou mau detalhamento por parte do *Product Owner*. Outra é a identificação de algum impeditivo que a equipe encontrou durante a fase de desenvolvimento, o qual o *Scrum Master* não conseguiu sanar.

Já com o indicador *Burnup*, pode ser utilizado para demonstrar o quanto a equipe já evoluiu no projeto e o quanto ainda falta para atingir o planejado. Com a adoção deste indicador na metodologia ágil, a área de negócio poderá ter uma visão antecipada da evolução do trabalho realizado pela área de sistemas e até mesmo se planejar em caso de haver riscos de o objetivo não ser atingido.

Ambos os indicadores representam o progresso na entrega do trabalho, contudo, a diferença está na maneira como as informações são apresentadas.

Diferentemente do modelo tradicional, neste cenário é possível discutir as ações que devem ser realizadas para corrigir o rumo do projeto antes que ocorra a entrega.

Outros tipos de indicadores foram abordados no referencial teórico, porém cabe ressaltar a sugestão de Gomes (2013) em avaliar o lucro que cada funcionalidade gera para o negócio. Na metodologia tradicional esta análise somente é possível quando o projeto está em produção, podendo o retorno não ser satisfatório para a organização. No gerenciamento ágil, como as funcionalidades são entregues de forma segmentada, desenvolvendo primeiro aquelas que terão um maior retorno ao cliente, o lucro obtido tende a ser maior. Na atual conjuntura da empresa x, onde há uma grande espera pelas entregas dos projetos da área de TI, um indicador relacionado ao lucro tende a ser benéfico para o negócio. Considerando o lucro tangível, ou seja, aquele que é possível de ser mensurado, o cliente poderá avaliar o seu lucro em cada *release* do projeto.

4.4 Métodos ágeis para gerenciamento de softwares

Tendo em vista que a área de negócios possui como característica atender a diferentes tipos de projetos, sendo do mais simples ao mais complexo, considerando ainda a questão cultural da empresa, é possível que para projetos simples a metodologia ágil seja aplicada de forma a substituir o modelo tradicional. Pelo fato dos projetos considerados como simples pela organização terem um curto prazo de vigência (geralmente são dois meses) possuem escopo bem definido e o seu detalhamento é rico em informações, as práticas ágeis, seja *Scrum*, *Extreme Programming* ou *Kanban* podem ser aplicados para gerar valor ao negócio da empresa.

Contudo, para projetos mais complexos, onde pode haver incertezas na execução do trabalho, prazo de duração mais extenso ou escopo factível de modificação, recomenda-se uma combinação entre os métodos *Scrum* e o tradicional, utilizando os aspectos positivos de cada metodologia. De acordo com Cruz (2013),

[...] o Scrum sozinho não pode resolver todos os problemas de todos os projetos [...] muitos projetos não podem ser gerenciados 100% de forma ágil do seu início ao fim (como o Scrum propõe), o guia PMBOK é sugerido nesta obra como a principal ferramenta de complementação e apoio ao Scrum (CRUZ, 2013, p. 49).

Para que a implementação de métodos ágeis possa se tornar uma realidade na organização, deve-se considerar também que uma qualificação das diferentes áreas envolvidas em projetos será necessária. As lideranças e os diversos times, incluindo o negócio, deverão se apropriar dos conceitos da metodologia ágil, suas práticas e restrições, disseminando assim o conhecimento por toda a TI.

Não faz parte da proposta deste trabalho detalhar ou aprofundar os meios necessários para realizar esta combinação juntamente com todos os processos do guia do PMBOK. O ideal é permitir que as equipes optem por utilizar esta união das metodologias para identificar de forma natural os pontos de ligação entre as abordagens de gerenciamento, definindo assim os processos que oferecem apoio ao *Scrum* e em quais momentos dentro do ciclo do *Scrum* estes serão aplicados, levando em consideração cada projeto específico (GOMES, 2013). Ainda segundo o autor, mesmo em projetos ágeis é preciso que sejam realizadas algumas atividades formais para documentar a passagem no ciclo. Como proposição para a união das metodologias, Gomes (2013) sugere:

- **Iniciação do projeto:** mesmo os projetos que serão gerenciados via métodos ágeis requerem que determinadas documentações sejam criadas para formalizar o termo de abertura ou passagem das fases. Independente do projeto, tamanho ou características, é altamente recomendado que o termo de abertura contenha o conteúdo proposto pelo guia do PMBOK.
- **Aplicação do *Scrum*:** após as atividades formais terem sido executadas, o time pode iniciar a execução do projeto segundo a ótica do *Scrum*. Neste momento os conceitos do *Scrum* e o tradicional combinam bastante, pois assim como no guia PMBOK é sugerido que o projeto seja dividido em várias fases de acordo com a entrega que será realizada, no *Scrum* o encerramento da fase é a entrega de uma versão planejada (CRUZ, 2015). Em cada fase haverá um novo ciclo do *Scrum*, sempre começando pela visão do produto a ser entregue no final da fase e finalizando com a entrega de uma funcionalidade para o cliente. Como no método tradicional há o papel de gerente de projetos,

recomenda-se que no método ágil esta pessoa seja o *Product Owner* do *Scrum*.

- **Planejando a entrega:** a reunião de planejamento da entrega visa planejar a funcionalidade do produto que será entregue ao final do ciclo ou fase. É neste momento que o plano e a meta são criados e transmitidos para o time *Scrum* e restante da organização. Por conseguinte, os produtos são construídos iterativamente, começando por aqueles de maior valor e risco, de modo que cada *sprint* tenha um incremento do produto. O objetivo deste planejamento é para que o mesmo seja feito de forma breve, possibilitando que planejamentos complementares possam ser realizados conforme a necessidade.
- **Executando a *sprint*:** o time de desenvolvimento do projeto começa a construir o produto na fase de execução, a qual é marcada por ser o momento em que se executa o planejamento que foi realizado nas fases anteriores. Neste processo o time é autogerenciável, proporcionando assim uma agilidade que facilitará a execução dos trabalhos. O papel de facilitador é de responsabilidade do *ScrumMaster*, o poderá precisar da atuação do gerente de projetos para a remoção de impedimentos.
- **Monitoramento:** é a partir deste momento que o ciclo de desenvolvimento começa a tomar uma forma parecida com a do método tradicional, porém mantendo os aspectos dos métodos ágeis. Ambas as metodologias consideram importantes as práticas de reuniões com as pessoas envolvidas nos projetos, porém na forma tradicional as reuniões tendem a ser longas e cansativas. Como alternativa, o *Scrum* sugere uma proposta de reunião diária, onde a duração é de no máximo 15 minutos e são discutidos os pontos da última reunião, o que será realizado hoje e se há algum impedimento. Neste momento pode-se avaliar a qualidade do projeto se os padrões de qualidade estão sendo seguidos, além de prever e antecipar riscos iminentes para o projeto. É nesta fase de monitoramento do projeto

que os indicadores sugeridos no tópico anterior serão úteis para medir a qualidade geral do mesmo.

- **Revisão:** esta é uma das últimas fases do ciclo de desenvolvimento dentro da *sprint*, a qual pode ser resumida como sendo uma validação das funcionalidades construídas e para posterior liberação ao cliente. No modelo tradicional é sugerido que ao final das entregas uma revisão seja feita pelo responsável do escopo. Da mesma forma, o *Scrum*, atuando como apoio ao modelo tradicional, possibilita esta revisão através da revisão da *sprint*. Seu maior objetivo é a revisão do gerente de projeto, atuando como *Product Owner*, nos itens desenvolvidos.
- **Encerramento da fase:** quando uma *sprint* chega ao fim, outra deve iniciar imediatamente após a que terminou, não deixando assim o time parado. Contudo, quando a equipe está iniciando as fases iniciais da próxima *sprint*, existem outras atividades que devem ser realizadas, tais como novos incrementos que possam ser adicionados no produto e atividades de formalização da fase. Neste contexto de combinação entre as metodologias, a fase é relacionada a versão de entrega. Sendo assim, o encerramento da fase é a entrega da versão planejada.
- **Encerramento do projeto:** esta fase do projeto é caracterizada por sua formalidade, a qual garante que todo o trabalho foi realizado e validado pelo cliente. Para isto, atividades como termo de encerramento do projeto, atualização da documentação, coleta de lições aprendidas são muito importantes de se realizar. Por último, após findadas as formalizações necessárias, o gerente de projeto pode realizar a liberação do time do projeto para que possam ser deslocados para outros projetos, incluindo o próprio gerente.

Com a adoção destas práticas é possível conciliar ambas as metodologias sem que a organização tenha que escolher uma única abordagem para gerenciamento de *software*.

5 CONSIDERAÇÕES FINAIS

Estudar e avaliar o processo de gerenciamento de projetos da empresa x proporcionou um melhor entendimento a cerca da metodologia tradicional. Foi possível identificar que há uma forte cultura na organização, porém o método atual, por si só, não atende mais as necessidades do negócio. Com os conceitos dos métodos ágeis foi possível identificar oportunidades de melhoria na gestão de projetos, onde a combinação dos dois modelos obteve ótimos resultados.

Pelo fato de o autor deste trabalho atuar diretamente na área de TI, esta pesquisa contribui muito para o entendimento de novos conceitos para gerenciamento de projetos. Através das literaturas estudadas, foi possível identificar alternativas para a situação atual da organização, tema este que despertou ainda mais o interesse do pesquisador para colocar as práticas em ação na empresa x. Um grande desafio neste contexto será mudar a cultura das pessoas, tendo em vista que o método tradicional já é utilizado há muitos anos na organização. Contudo, através do compartilhamento das informações, bibliografias sobre métodos ágeis, adoção de boas práticas, pode-se chegar ao cenário ideal que é a combinação dos métodos.

Todos os objetivos específicos deste trabalho foram alcançados. O primeiro era identificar o processo atual da empresa e os seus principais problemas relacionados ao gerenciamento de projetos, onde através das pesquisas em documentos e a observação participante do autor diretamente na área de TI, foi possível identificar as dificuldades que foram amplamente discutidas neste trabalho. Com base nas entrevistas realizadas com os membros das equipes dos projetos, pode-se entender o processo atual da gestão de projetos, problemas identificados e verificou-se a oportunidade para adoção de métodos ágeis como uma prática positiva para a organização.

O segundo objetivo específico era descrever os principais modelos de métodos ágeis, o qual foi alcançado através dos estudos realizados na fundamentação teórica deste trabalho. Dentre as diversas bibliografias estudadas, cabe destacar o autor Fábio Cruz, o qual abordou a combinação do método tradicional e ágil como sendo algo benéfico para o negócio. Em sua abordagem ele cita que não é necessário que a empresa opte por um método para gerenciamento de projetos, mas sim obter os pontos fortes de cada metodologia para combiná-las.

Partindo deste entendimento, diversas oportunidades de melhorias no processo atual foram citadas ao longo deste trabalho, as quais o autor desta pesquisa irá levar para conhecimento da organização.

O terceiro objetivo específico era identificar os tipos de indicadores de desempenho mais utilizados em métodos ágeis. Com base no referencial teórico, este objetivo foi alcançado através da indicação de diversos modelos, cabendo um destaque para os indicadores de *Burndown* e *Burnup*.

O quarto objetivo específico era estudar e propor uma metodologia ágil ou sua combinação com o modelo tradicional, o qual obteve êxito através da abordagem de que ambas podem coexistir na organização. Com base nas leituras realizadas, pode-se entender que não há uma metodologia ideal, mas sim a combinação daquelas que mais se adéquam ao sistema da organização.

Por fim, o objetivo geral que é avaliar se é possível implantar métodos ágeis na empresa x foi alcançado. É possível sim sua implementação através das sugestões de práticas relacionando os conceitos do *Scrum*, ou outros modelos, e o modelo tradicional. Após haver um entendimento dos dois modelos, foi possível mapear onde a combinação se encaixa. Em linhas gerais, pode-se aferir que o modelo do *Scrum* tende a obter ótimos resultados quando aplicado para a parte de execução do projeto, por outro lado, a preocupação nas formalizações do método tradicional irá suprir esta lacuna no método ágil.

Para esta pesquisa não foi realizado um estudo sobre o retorno do investimento ou custo x benefícios com a adoção dos métodos ágeis na organização. Também não foi objeto de pesquisa avaliar o desempenho das equipes em caso de adoção das práticas ágeis. Estes estudos ficam como sugestões para o desenvolvimento de novas pesquisas, as quais poderão contribuir positivamente para o crescimento da organização.

REFERÊNCIAS

AGILE MANIFESTO. **Manifesto for Agile Software Development**. Agile Alliance, 2001. Disponível em: <<http://www.agilemanifesto.com.br/>>. Acesso em: 03 ago. 2015.

AMARAL, Daniel C. et al. **Gerenciamento Ágil de Projetos - Aplicação em produtos Inovadores**. São Paulo: Saraiva, 2011.

ANDRADE, Maria M. de. **Introdução à metodologia do trabalho científico: elaboração de trabalhos na graduação**. 10ª ed. São Paulo: Atlas, 2010.

ASTELS, D.; MILLER, G.; NOVAK, M. **eXtreme Programming – Guia Prático**. Rio de Janeiro, RJ: Campus, 2002.

AUDY, Jorge L. N.; PRIKLANDNICKI, Rafael. **Desenvolvimento Distribuído de Software**. Rio de Janeiro: Elsevier, 2007.

BECK, Kent. **Extreme Programming explained: embrace change**. 1. ed., MA: Addison-Wesley, 2000.

BECK, Kent; FOWLER, Martin. **Planning Extreme Programming**. 1. ed. Boston: Addison-Wesley, 2001.

BROOKS, Frederick P. **The mythical man-month: essays on software engineering**. 2. ed., MA: Addison-Wesley, 1995.

COCKBURN, Alistair. **Agile software development**. 1. ed. Boston: Addison-Wesley, 2002.

COHN, M. **Agile Estimating and Planning**. Massachusetts: Pearson Education, 2005.

CRUZ, Fábio. **Scrum e Agile em projetos: guia completo**. Rio de Janeiro: Brasport, 2015

CRUZ, Fábio. **Scrum e PMBOK unidos no gerenciamento de projetos**. Rio de Janeiro: Brasport Livros, 2013.

DENZIN, Norman K. ; LINCOLN, Yvonna S. **Handbook of qualitative research**. (2 Ed.). Thousand Oaks, Califórnia: Sage Publications. 2000.

DIMES, Troy. **Scrum essencial**. Canadá: Babelcube, 2014.

FUSCO, José P. A.; SACOMANO, José B. **Operações e gestão estratégica da produção**. São paulo: arte & ciência, 2007.

GIDO, Jack; CLEMENTS, James P. **Gestão de Projetos**. 3ed. Tradução Vertice Translate. São Paulo: Thomson Learning, 2007.

GIL, Antonio C. **Métodos e técnicas de pesquisa social**. 8ª ed. São Paulo: Atlas, 1994.

GOMES, Andre F. **Desenvolvimento de software com entregas frequentes e foco no valor de negócio**. São Paulo: Casa do Código, 2013.

Guia do *Scrum*. Disponível em: <<http://www.Scrumguides.org/docs/Scrumguide/v1/Scrum-Guide-PortugueseBR.pdf>>. Acesso em: 08 jul. 2015.

- HIGHSMITH, J.; COCKBURN, A. **Agile software development: The business of innovation**. IEEE Computer, v. 34, n. 9, p.120-122, 2001.
- JEFFRIES, Ron; ANDERSON, Ann; HENDRICKSON, Chet. **Extreme Programming installed**. 1. ed. Boston: Addison-Wesley, 2001.
- LACEY, Mitch. **The Scrum Field Guide: Practical Advice for Your First Year**. 1. ed. Indiana, 2014.
- LEITE, Vinícius Manhães. **Extreme Programmin – Valores - Comunicação**, 2006a. Disponível em: <<http://www.desenvolvimentoagil.com.br/xp/valores/comunicacao>> Acesso em: 09 Ago. 2015.
- LEITE, Vinícius Manhães. **Extreme Programmin – Valores - Coragem**, 2006b. Disponível em: < <http://www.desenvolvimentoagil.com.br/xp/valores/coragem>> Acesso em: 09 Ago. 2015.
- LINCOLN, Yvonna S. **The American tradition in qualitative research**. Vol. II. Thousand Oaks, California: Sage Publications, 2001.
- MASSARI, Vitor, L. **Gerenciamento Ágil de Projetos: uma visão preparatória para a certificação ágil do PMI (PMI-ACP)**. Rio de Janeiro: Brasport, 2014.
- MILES, Matthew B.; HUBERMAN, A. Michael. **Qualitative data analysis**. Thousand Oaks: Sage Publications, Inc. 1994.
- MONDEN, Yasuhiro - **Sistema Toyota de Produção**. Porto Alegre: Bookman, 2012.
- MOURA, Henrique. **PMP sem Segredos**. Rio de Janeiro: Elsevier, 2013.
- MOURA, R. Kanban. **A Simplicidade do Controle de Produção**. São Paulo: IMAM,1992.
- PFLEEGER, S. L. **Engenharia de Software: Teoria e Prática**, 2ª Edição, São Paulo: Prentice Hall, 2004.
- PHAM, Andrew; PHAM, Phuong-van. **Scrum em ação**. 1 ed. São Paulo: Novatec, 2011.
- PIMENTEL, Mariano; FUKS, Hugo. **Sistemas Colaborativos**. Rio de Janeiro: Elsevier, 2011
- PMI - PROJECT MANAGEMENT INSTITUTE. **Guia PMBOK: Um guia do conjunto de conhecimento do gerenciamento de projetos**. 5th edition Pennsylvania: Project Management Institute, 2013.
- POPPENDIECK, Mary; POPPENDIECK, Tom. **Lean software development: an agiletoolkit**. 1. ed. Upper Saddle River, NJ: Addison-Wesley, 2003.
- PRATES, Rubens. **Scrum Guia Prático para Projetos Ágeis**. 1. ed. São Paulo: Novatec, 2013.
- PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. Porto Alegre: McGraw-Hill, 2006.
- PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre: Bookman, 2014
- SABBAGH, Rafael. **Scrum – Gestão Ágil para Projetos de Sucesso**. São Paulo: Casa do Código, 2013.

Scrum. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2015. Disponível em: <<https://pt.wikipedia.org/w/index.php?title=Scrum&oldid=42901341>>. Acesso em: 8 ago. 2015

TELES, Vinícius M. **Extreme Programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. 1. ed. São Paulo: Novatec, 2004.

TELES, Vinícius M. **Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade**. 2 ed. São Paulo: Nopvatec, 2014.

THOMAS, J. R.; NELSON, J. K.; SILVERMAN, S. J. **Métodos de pesquisa em atividade física**. 5. ed. Porto Alegre: Artmed, 2007.

TRIVIÑOS, A. N. S. **Introdução à pesquisa em ciências sociais: a pesquisa qualitativa em educação**. São Paulo: Atlas, 1987.

VERGARA, Sylvia C. **Projetos e relatórios de pesquisa em administração**. 12ª ed. São Paulo: Atlas, 2010.

VERSIONONE. **8th Annual State of Agile Survey**. VersionOne, Inc., 2014. Disponível em: <<http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>>. Acesso em: 10 out. 2015.

WEINBERG, Gerald M. **The psychology of computer programming**. 1. ed. New York: Van Nostrand Reinhold Company, 1971.

YIN, Robert K. **Estudo de caso – planejamento e métodos**. 2 ed. Porto Alegre: Bookman, 2010.

APÊNDICE A – QUESTIONÁRIOS DE ENTREVISTA

Questões formuladas aos membros das equipes dos projetos.

- 1) Qual sua função ou papel dentro da empresa em que trabalha?
- 2) Você considera que os *softwares* desenvolvidos pelas áreas de sistemas são com qualidade?
- 3) Como vocês lidam com as mudanças de prioridade da área de negócios?
- 4) Quais desvantagens você identifica no modelo atual de gestão de projetos?
- 5) A equipe que você atua tem liberdade para decidir qual a melhor forma de atingir seus objetivos?
- 6) A visão do produto que você está ajudando a construir é clara e objetiva para você?
- 7) Há um contato próximo entre você e a área demandante do projeto (área de negócios)?
- 8) Se há necessidade de se efetuar uma alteração no escopo do projeto, como a equipe lida com esta situação?
- 9) O que você percebe de positivo no modelo atual?
- 10) Você acredita que a combinação de métodos ágeis e tradicional para gerenciamento de projetos pode contribuir de forma positiva para a organização?