



PUC-SP

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO COGEAE

ENGENHARIA DE SOFTWARE

**UM ESTUDO DE VIABILIDADE DE IMPLEMENTAÇÃO DO SCRUM EM UMA
EMPRESA DE DESENVOLVIMENTO DE SOFTWARE**

TÁSSIO HENRIQUE PEREIRA

São Paulo

2016

TÁSSIO HENRIQUE PEREIRA

**UM ESTUDO DE VIABILIDADE DE IMPLEMENTAÇÃO DO SCRUM EM UMA
EMPRESA DE DESENVOLVIMENTO DE SOFTWARE**

Monografia apresentada ao Curso de Especialização em Engenharia de Software, da Pontifícia Universidade Católica de São Paulo - COGEAE, como pré-requisito para a obtenção do título de Especialista em Engenharia de Software

Orientador: Prof. André Luiz Garcia Pereira.

São Paulo

2016

TÁSSIO HENRIQUE PEREIRA

**UM ESTUDO DE VIABILIDADE DE IMPLEMENTAÇÃO DO SCRUM EM UMA
EMPRESA DE DESENVOLVIMENTO DE SOFTWARE**

Resultado: _____

Orientador: Prof. André Luiz Garcia Pereira.

São Paulo, 7 de Maio de 2016.

DEDICATÓRIA

*Dedico este trabalho aos meus pais, a
minha família e a todos aqueles que
estiveram presentes em minha vida ao
longo desta etapa de aprendizagem.*

AGRADECIMENTOS

Agradeço a Deus pelas inúmeras bênçãos concedidas!

Agradeço ao Prof. André Luiz Garcia Pereira por me orientar e pela atenção durante a concretização deste trabalho.

Agradeço a minha noiva por permanecer ao meu lado enquanto eu realizava essa pesquisa.

Agradeço a Paula, minha amiga, que me incentivou a concluir essa pesquisa.

Agradeço a minha família e amigos, pessoas especiais, que me auxiliaram e me apoiaram na concretização deste trabalho.

Obrigado a todos

LISTA DE FIGURAS

Figura 1 - Modelo Clássico Cascata.....	18
Figura 2 - Modelo Incremental.....	20
Figura 3 - Modelo Espiral	21
Figura 4 - Visão Geral do Scrum	27
Figura 5 - História sobre o comprometimento do <i>Scrum Team</i>	30
Figura 6 - Sprint.....	33
Figura 7 - Gráfico Burndown - modelo	36
Figura 8 - Fluxograma de desenvolvimento de solicitações antes da implantação do Scrum.....	40
Figura 9 - Fluxograma de desenvolvimento de solicitações após a implantação do Scrum.....	43
Figura 10 - Ferramenta Web Asana	44
Figura 11 - Diagrama de Ishikawa para a empresa antes do <i>Scrum</i>	49
Figura 12 - Diagrama dos “5 Por quês” para definição da causa raiz	49
Figura 13 - Solicitações concluídas por mês.....	50
Figura 14 - Prazo médio de conclusão das solicitações.....	51

LISTA DE TABELAS

Tabela 1 - Solicitações antes da implantação do Scrum.....	41
Tabela 2 - Papéis do Gerente de Projetos e Testes.....	45
Tabela 3 - Solicitações depois da implantação do Scrum.	48

RESUMO

A globalização da economia e da tecnologia trouxe junto um mercado atual de desenvolvimento de software cada vez mais exigente, em busca de desenvolver projetos com eficiência e de forma rápida. Se faz primordial que as empresas utilizem de ferramentas para uma melhor saída para possíveis alterações inerentes a qualquer execução de projeto, estabelecendo assim a necessidade de utilizar métodos inovadores para garantir bom desenvolvimento dos trabalhos. Como uma excelente resposta à essa nova exigência do mercado, surgem então as metodologias ágeis, que possibilitam a criação de softwares com mais funcionalidades e que atendam às necessidades dos clientes de forma eficiente e em tempo menos, e também com valor agregado reduzido. Dentro dessa ótica, a metodologia ágil Scrum assiste às mudanças recorrentes, sem perder a qualidade na entrega do produto final, em comparação com os métodos clássicos que costumam mais, portanto, se faz necessária a compatibilização desta metodologia para dentro do escopo das empresas, sejam elas pequenas ou médias, para que estas possam apresentar competitividade em relação às grandes empresas. O presente trabalho busca examinar a viabilidade de implantação da metodologia ágil Scrum numa empresa de desenvolvimento de software, relatando as etapas da implementação de cada atividade e relatar as dificuldades encontradas neste processo, demonstrando as melhorias alcançadas após a conclusão de sua implantação.

Palavras-chave: Gerenciamento de Projetos, Scrum, Metodologia Ágil.

ABSTRACT

The globalization of the economy and technology brought together a current market software development increasingly demanding, seeking to develop projects efficiently and quickly. If it is essential that companies use tools for better output for possible changes inherent in any project execution, thus establishing the need to use innovative methods to ensure smooth execution of work. As an excellent response to this new market requirement, then come the agile methodologies, which enable the creation of software with more predicates that meet the impositions of customers efficiently and in less time, and with reduced added value. Within this perspective, the agile Scrum attend the recurring changes without losing quality in the delivery of the final product, compared to the classical methods that cost more, so it is necessary to reconcile this methodology into the scope of the companies, are they small or medium so that they can present competitiveness in relation to large companies. This paper seeks to examine the feasibility of implementing the Scrum agile methodology in a software development company, reporting stages of the implementation of each activity and report the difficulties encountered in this process, and demonstrate the improvements achieved after its implementation.

Keywords: Project Management, Scrum, Agile Methodology.

SUMÁRIO

INTRODUÇÃO	11
OBJETIVOS	12
JUSTIFICATIVA	13
METODOLOGIA	14
ORGANIZAÇÃO	14
1 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	16
O DESENVOLVIMENTO DE SOFTWARE NA HISTÓRIA	16
1.1 MODELOS DE DESENVOLVIMENTO DE SOFTWARE	18
1.1.1 Modelo Cascata	18
1.1.2 Modelo Incremental	19
1.1.3 Modelo Espiral	20
2 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE	23
2.1 METODOLOGIA ÁGIL	23
2.1.1 Manifesto Ágil	25
3 SCRUM	27
3.1 DEFINIÇÃO	27
3.1.1 Pilares do Scrum	28
3.1.2 Papéis do Scrum	30
3.2 TIME	31
3.2.1 Product Owner	31
3.2.2 Time de Desenvolvimento	31
3.2.3 Scrum Master	32
3.3 EVENTOS SCRUM	32
3.3.1 Sprint	32
3.3.2 Cancelamento de Sprint	33
3.3.3 Reunião de planejamento de Sprint	34
3.3.4 Reunião diária	34
3.3.5 Revisão de Sprint	35
3.3.6 Retrospectiva da Sprint	35
3.4 ARTEFATOS DO SCRUM	36
3.4.1 Backlog do Produto	37
3.5 TRANSPARÊNCIA DO ARTEFATO	37

4 ESTUDO DE VIABILIDADE DE IMPLEMENTAÇÃO DO SCRUM.....	38
4.1 EMPRESA.....	38
4.2 EQUIPE.....	38
4.3 SOFTWARE.....	38
4.4 PROCESSO ATUAL	38
4.5 ETAPAS PARA IMPLEMENTAÇÃO	42
4.5.1 Mudança de papéis e funções	45
4.6 ANÁLISE DE RESULTADOS.....	46
5 CONSIDERAÇÕES FINAIS	48
5.1 OBSERVAÇÕES.....	48
5.2 BENEFÍCIOS DA IMPLEMENTAÇÃO DO SCRUM	48
5.3 SUGESTÕES PARA TRABALHOS FUTUROS	53
REFERÊNCIAS BIBLIOGRÁFICAS.....	54

INTRODUÇÃO

A indústria de desenvolvimento de software atualmente é uma das que recebe maior destaque, diante da crescente demanda por novas tecnologias. Produto indispensável para a contemporaneidade, o software se faz presente em todos os campos de nossa vida e por isso se apresenta como uma das propriedades intelectuais de maior valor atualmente.

Transformações ocorridas nos hardwares, a partir da década de 70, geraram um grande aumento na eficiência, com fácil aquisição a todas as classes por conta da redução de tamanho e custo. A demanda por softwares que são responsáveis hoje por gerir boa parte de nossa vida, sendo usada como forma de entretenimento, comunicação e principalmente como ferramenta de trabalho, tem aumentado por conta da constante evolução do hardware.

A crise do software foi assim intitulada depois que desenvolvedores atravessaram períodos com muitos problemas inerentes às atividades de desenvolvimento de sistemas, evidenciados pela complexa atividade de criação de softwares, que demandava melhoria. Foi então identificada a carência de ferramentas e métodos que amparassem o processo de desenvolvimento de softwares. A partir disso é que foram criadas ferramentas para auxiliar as equipes, como as metodologias ágeis, guias com práticas, entre outros, que ampararam as empresas, e foi evidenciada uma diminuição nos problemas que ocorriam durante a crise do software.

Mas isso só se aplicava apenas às grandes empresas detentoras de poder econômico e social para bancar os custos de implantação dessas ferramentas. Então as empresas pequenas e médias, por possuírem recursos limitados, optaram por empregar uma maneira própria de desenvolvimento, privando-se das novas tecnologias, o que gerou inúmeros problemas futuros, mesmo após 40 anos do fim da crise do software.

A ferramenta de desenvolvimento ágil Scrum, se destaca no âmbito dessas pequenas empresas, podendo elas então aprimorar o desenvolvimento dos sistemas, trazendo melhorias aos processos.

As características que mais se destacam no Scrum são a busca pela comunicação ágil, redução de documentos burocráticos, e interações tratadas com

mais seriedade.

Ao criar o Guia Scrum, Schwaber (2011), instituiu que essa metodologia pode ser utilizada para desenvolver qualquer tipo de sistema, mesmo os mais complexos.

Ao comparar as características do Scrum com os ambientes de desenvolvimento das empresas, foi possível verificar que a metodologia se encaixa aos cenários da empresa, permitindo que os desenvolvedores efetuem entregas mais rápidas dos produtos, pois faz com que haja uma mudança das ferramentas utilizadas e da organização da equipe, bem como da mentalidade de cada membro, do processo, garantindo entregas com mais qualidade, o que resulta em mais confiança e valor por parte do cliente.

Este trabalho visa explorar a viabilidade da implantação do Scrum em uma empresa de desenvolvimento de software, que possui algumas equipes de desenvolvimento divididas por categoria. Esta empresa ainda não utilizava nenhuma metodologia de desenvolvimento de software, e as atividades dos projetos eram executadas sem se basear em nenhum critério, conforme era pertinente ao grupo, sem padrões, e as tarefas eram divididas entre o grupo conforme experiência de cada membro. Essa divisão de tarefas resultava muitas vezes em sobrecarga de trabalho para os mais experientes do time. Com a nova demanda de clientes e projetos, se fez necessária a implantação da metodologia, por conta da frequente alteração de escopo dos novos projetos.

Este trabalho busca então avaliar que tipo de resultado serão trazidos pela implantação do Scrum, bem como identificar as melhorias que ocorreram nos processos.

OBJETIVOS

Com base na necessidade das empresas em tornar o seu processo de desenvolvimento mais ágil, a proposta do presente trabalho é entender e averiguar as ferramentas do método de implantação do Scrum em uma empresa de desenvolvimento de software, mensurando a repercussão positiva e/ou negativa no gerenciamento dos procedimentos.

JUSTIFICATIVA

A metodologias ágeis têm sido muito difundidas no âmbito do desenvolvimento de softwares. (KNIBERG ,2007)

A metodologia ágil se fundamenta em seis características: flexibilidade dos resultados, flexibilidade dos prazos, times pequenos, revisões frequentes, colaboração e orientação de objetos (SCHWABER, 2004).

Empresas que buscam agilidade nos processos podem usar as metodologias ágeis, que trazem inúmeros benefícios ao desenvolvimento de softwares. Sem uma correta implantação de métodos de gerenciamento, o trabalho pode tomar um rumo errado, e causar grande perda de produtividade da equipe.

O Scrum busca resolver esses problemas, valorizando o recurso humano, visando à melhoria contínua dos processos e pode ser implantado de diversas formas, podendo se adequar à cultura de cada empresa atendendo suas necessidades.

Assim este trabalho pretende averiguar a viabilidade de implementação do Scrum em uma empresa de desenvolvimento de software, avaliar os benefícios e identificar as dificuldades encontradas nesse processo.

A empresa atua no desenvolvimento de software para o setor de Varejo e está há mais de 28 anos no mercado. Possui aproximadamente 60 colaboradores, sendo 30 colaboradores da área de desenvolvimento de software. Como a empresa atende diversos clientes, sempre existe a necessidade modificação do software. A empresa é especializada em produto de software de venda para comércio varejista para todo o território nacional, neste ramo as alterações de legislações são constantes pois cada unidade federativa pode adotar suas próprias obrigações tributárias.

Todas as novas necessidades dos clientes são avaliadas, algumas delas chegam a durar meses, pois como não existe um processo definido algumas prioridades acabam sobrepondo tais necessidades.

Outro ponto relevante, é a comunicação entre a equipe do suporte e a equipe de desenvolvimento, pois sempre são liberados *releases* sem o prévio conhecimento da equipe de suporte, o que acarreta na falta de conhecimento da equipe do suporte sobre o produto em questão.

METODOLOGIA

A pesquisa realizada no projeto é de Natureza Aplicada, objetivando gerar conhecimentos para aplicação prática, dirigida a ambientes de desenvolvimento de softwares. Sua abordagem será qualitativa, de modo a analisar a interação de cada indivíduo com a metodologia e com o novo ambiente de desenvolvimento a ser criado.

Quanto ao seu objetivo, a pesquisa é caracterizada como investigatória fazendo uso de procedimento técnico bibliográfico.

Desta forma, contempla as seguintes etapas:

- Estudo geral sobre metodologias ágeis: pesquisa sobre as metodologias ágeis, história e propósitos;
- Estudo aprofundado sobre a metodologia ágil *Scrum*;
- Definição da adaptação da metodologia *Scrum* para o ambiente a ser aplicado.
- Tabulação e avaliação dos resultados. Esta etapa busca avaliar os ganhos ou perdas encontradas com o uso da metodologia ágil *Scrum* para desenvolvimento e criação de sistemas.

ORGANIZAÇÃO

Este trabalho está organizado em quatro capítulos:

- Capítulo 1 – Processo de desenvolvimento de software: é feita uma apresentação sobre alguns processos de desenvolvimento de software;
- Capítulo 2 – Metodologias de desenvolvimento de Software: é apresentado o fundamento da metodologia ágil;
- Capítulo 3 – *Scrum*: é apresentado o estudo da metodologia escolhida para implementação na empresa em questão;
- Capítulo 4 – Estudo de viabilidade de implementação: é feito o estudo com base nas características da empresa e a base teórica apresentada nos

capítulos anteriores para identificar os passos necessários para implementação da metodologia *Scrum* na empresa em questão.

1 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

O DESENVOLVIMENTO DE SOFTWARE NA HISTÓRIA

Com a criação da informática nos anos 60, uma inovadora ferramenta também foi criada: o computador. Era ainda uma peça grande, que consumia muita energia, e não oferecia muitos recursos.

O acesso a esse tipo de tecnologia se restringia ao governo e instituições de ensino, com a finalidade de pesquisa. Equipes desenvolvedoras foram recrutadas para tornar isso possível, e atender á necessidades de cada organização. (KOSCIANSKI; SOARES, 2007).

No advento da década de 70, houve o lançamento do computador pessoal com tamanhos menores, incentivado pelas transformações de hardware, o que provocou a criação de uma gama de variantes desses novos equipamentos.

Edsger Dijkstra (1972), relata que:

A maior causa da crise do software é que as máquinas se tornaram várias ordens de magnitude mais potentes! Em termos diretos, enquanto não havia máquinas, programar não era um problema; quando tivemos computadores fracos, isso se tornou um problema pequeno e agora que temos computadores gigantescos, programar tornou-se um problema gigantesco. (DIJKSTRA, 1972, p. 5).

Koscianski e Soares (2007) relatam que a primeira crise do software se deu na primeira geração de desenvolvedores, relacionado às dificuldades enfrentadas por estes nos processos e na busca de soluções, dada no início da revolução digital. Os problemas mais comuns da época eram a complexidade da definição de requisitos e escopo, que geralmente gerava mau funcionamento, estouros no orçamento devido às mudanças ao longo do processo, e à instabilidade de funcionamento dos sistemas.

Alavancada pelo crescente interesse sobre metodologias e ferramentas de desenvolvimento de software, os investimentos na área têm aumentado, e divulgam o conceito de engenharia de software. O termo “engenharia de software” foi utilizado

pela primeira vez na Alemanha, em uma conferência, no final da década de 60, criado por Fritz Bauer (KOSCIANSKI E SOARES, 2007).

As instituições de ensino têm estado na vanguarda da disseminação do aprendizado referente a cursos e disciplinas da engenharia de software, que corre em paralelo á evolução desse novo cenário.

Ferramentas como *Unified Modeling Language* (UML), modelos de processos de desenvolvimento de software, modelos de controle de versão de sistemas, ferramentas de comunicação, metodologias de desenvolvimento ágil podem ser úteis para auxiliar o processo de desenvolvimento de softwares.

Todavia, Koscianski e Soares (2007) afirma que ainda hoje:

[...] mais de trinta anos depois, quais são os problemas enfrentados na construção e utilização de software? Ao lermos o relatório da conferência da NATO de 1968 e outros documentos produzidos na década de 1970, fazemos uma descoberta assustadora: os problemas são os mesmos que encontramos atualmente. (KOSCIANSKI E SOARES 2007)

Desenvolver um software é como seguir uma receita, ou seja, executar uma sequencia de atividades que deve ser realizada a fim de alcançar os propósitos do projeto (PAULA FILHO, 2009).

A Engenharia de software, segundo Wirth (2012) propõe a adoção da disciplina para lidar com o desenvolvimento do software, tentando reduzir a influência de dificuldades ocasionais, tomando por base ferramentas que auxiliam nesse processo.

Quatro atividades são fundamentais para o processo de desenvolvimento de software, apesar de cada um deles apresentar uma característica diferente (SOMMERVILLE, 2011):

- Definição das funcionalidades e limitações do software, especificação;
- Realização de um bom projeto e implantação, para garantir o alcance dos objetivos propostos;
- Avaliação do produto a fim de verificar se o mesmo atende aos requisitos inicialmente acordados com o cliente;
- A evolução do software.

1.1 MODELOS DE DESENVOLVIMENTO DE SOFTWARE

Os processos de software visam estabelecer, para a equipe de desenvolvimento, as atividades que devem ser realizadas para se obter um produto de software. Existem alguns modelos que são como representações desses processos, como, modelo cascata, modelo incremental e modelo espiral (SOMMERVILLE, 2011).

1.1.1 Modelo Cascata

O primeiro modelo de desenvolvimento de software que se tem conhecimento é o modelo clássico denominado modelo cascata. De fácil compreensão, o modelo cascata se fundamenta em etapas sequenciais. Cada fase gera um documento que deve ser aprovado para haver continuidade no processo (PRESSMAN, 2006). Este modelo é coordenado em esquemas de atividades que devem ser concebidas antes do início da sua execução (SOMMERVILLE, 2011). A figura 1 demonstra as fases do modelo clássico em cascata.

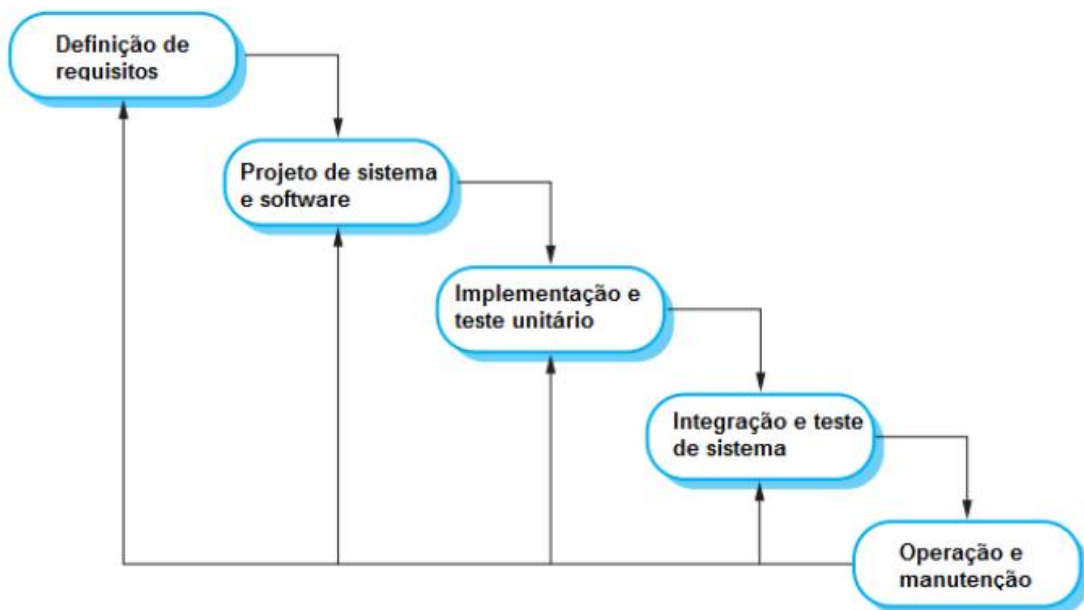


Figura 1 - Modelo Clássico Cascata
Fonte: Sommerville (2011)

O modelo cascata apresenta alguns percalços apesar de ser muito difundido e utilizado. Ele apresenta carência na definição do escopo e requisitos no início de seu desenvolvimento, e o produto geralmente demora a ser entregue ao cliente (Jubileu 2008).

Este modelo, que apresenta um processo rígido de desenvolvimento é mais utilizado na execução de sistemas de níveis altos de segurança, onde se faz necessária a demanda por conhecimento específico, portanto é um modelo que não certifica benefícios diante de outros modelos para execução de outros tipos de sistemas (SOMMERVILLE, 2011).

Pressman (2006) afirma que este modelo não se adapta à atualidade, num mercado que trabalha em um compasso veloz a fim de atender à crescente demanda de alterações que são inerentes à diversos projetos. Pode ser utilizado em processos que apresentem escopo fixo e linearidade na sua execução.

1.1.2 Modelo Incremental

O modelo incremental utiliza-se do modelo cascata, porém o aplica de forma interativa. Como mostra a figura 2, as sequencias do modelo em cascata são utilizadas de forma racional com o passar do tempo. Assim, cada fluxo de continuidade resulta num incremento do produto a ser entregue para o cliente. Como exemplo podemos citar o desenvolvimento de um software editor de texto, que pode ter um incremento como a edição básica de textos, e o próximo incremento pode ser a habilidade de editar texto de forma avançada e ainda integrar um corretor ortográfico numa próxima entrega (PRESSMAN, 2006).

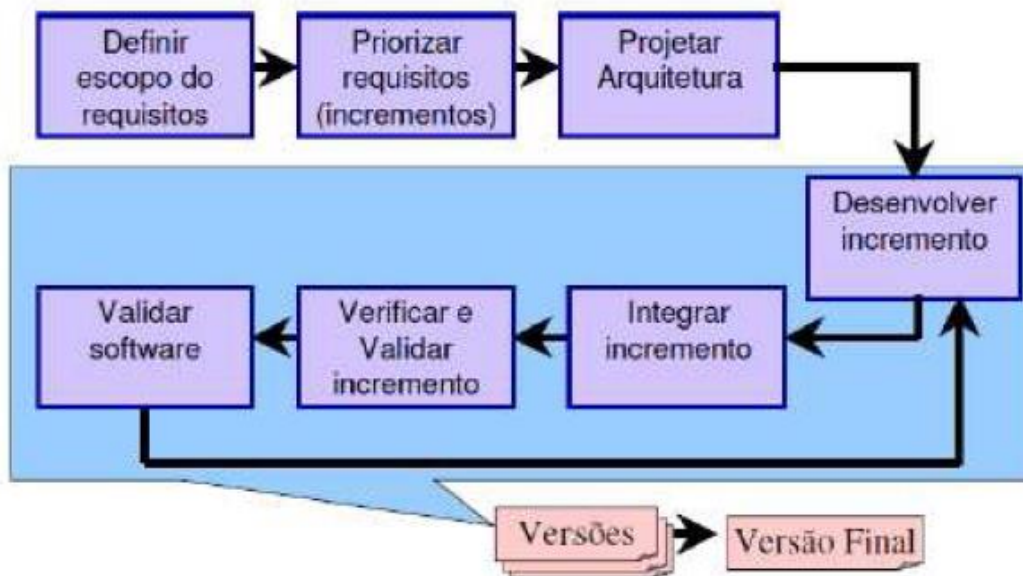


Figura 2 - Modelo Incremental
Fonte: Jubileu, 2008

O principal objetivo deste modelo é apresentar um produto operacional ao final de cada fase, podendo este produto operacional passar ou não pela análise do cliente. Os primeiros incrementos geralmente são versões simplificadas do produto, que poderão ser utilizadas pelos usuários, servindo de plataforma de avaliação (PRESSMAN, 2006).

Para Jubileu (2008), as principais vantagens deste modelo são:

- Não há necessidade de esperar uma versão final para iniciar o uso do software;
- As primeiras entregas podem ser usadas como um tipo de protótipo, visando definir melhor requisitos e incrementos futuros;
- É menor o risco de fracasso do software, uma vez que entregando inicialmente as funções mais prioritárias se torna desnecessário que as mesmas passem por um período de testes mais intensos.

1.1.3 Modelo Espiral

Paula Filho (2009), discorre que o ciclo de vida no formato de uma espiral permite adicionar ao modelo novos requisitos durante o fluxo. Usa-se então uma

abordagem alternada para desenvolver o sistema, resultando em entregas incrementais, combinando a natureza iterativa com os aspectos controlados, que requerem uma equipe de desenvolvimento experiente e disciplinada.

Este modelo apresenta bastante flexibilidade e maior visibilidade, dando a ao cliente a chance de avaliar as partes entregues do produto com possibilidade de alterar requisitos ou fornecer dados mais concretos a respeito do produto. Porém, toda essa flexibilidade requer uma gestão aprimorada do projeto para que sua estrutura continue sólida a cada iteração finalizada (PAULA FILHO, 2009).

A Figura 3 é um esboço do funcionamento do modelo:

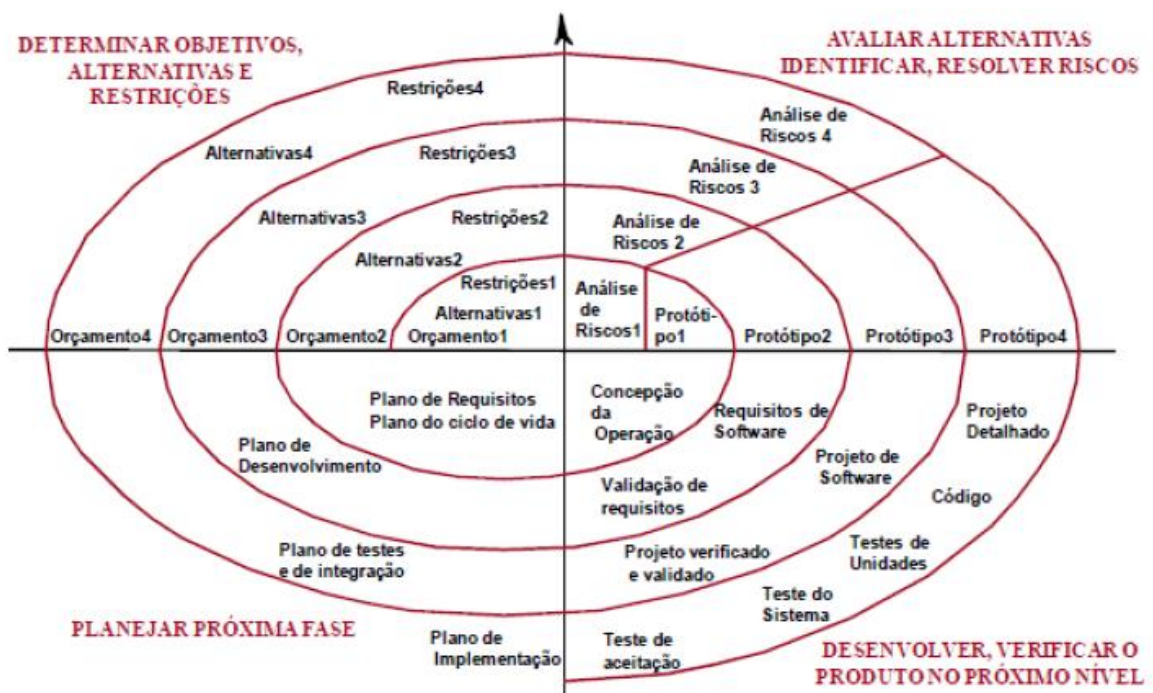


Figura 3 - Modelo Espiral
Fonte: Jubileu, 2008.

Como pode ser observada, na figura 2, cada volta do espiral é dividida em quatro partes. A primeira parte é a definição dos objetivos e também são identificadas as restrições ao processo de desenvolvimento do produto e os riscos do projeto, elaborando-se um plano de gerenciamento. A segunda parte é a avaliação e redução de riscos. Para cada risco identificado é realizada uma análise detalhada do mesmo e as medidas visando a sua redução. A terceira é o desenvolvimento e a validação. Após a avaliação dos riscos, deve ser escolhido o modelo mais adequado para o desenvolvimento da solução. A quarta e ultima parte é o planejamento, é a etapa em que o projeto é revisado, e é tomada a decisão de

continuar, ou seja, elaborar os planos para uma nova volta, ou se o mesmo será finalizado. (SOMMERVILLE, 2011).

2 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Metodologia é conceituada por Houaiss (2001) como sendo um apanhado de regras e princípios que estabelecem métodos para a realização de alguma atividade. Tais regras seriam as formas de agir a fim de chegar a um denominador comum. Alguns objetivos do uso das metodologias inerentes ao desenvolvimento de software, segundo Wirth (2012) podem ser definidos como a pessoa que realiza, o que, quando, onde e como essa pessoa executa o processo. Dessa forma todos os envolvidos saberão de suas funções e papéis dentro do desenvolvimento, sejam eles, equipe de desenvolvedores, usuários, clientes ou a administração da empresa.

Leite (2014) afirma que o uso de uma metodologia ágil proporciona ao gerente a possibilidade de controlar o desenvolvimento do produto, sem desviar o foco do planejamento e do intento do projeto a fim de se evitar o seu insucesso, caso haja negligência ou má gestão e ao desenvolvedor produzir de forma eficiente e a partir de um escopo bem definido, um software que atenda às necessidades do cliente de forma hábil.

Uma metodologia deve servir de instrumento de gerenciamento, permitindo facilidade ao entendimento de todos, comunicação e coordenação de toda a equipe envolvida, não podendo se limitar à originalidade profissional (LEITE, 2014).

De importância fundamental para o desenvolvimento dos produtos, uma metodologia deve ser escolhida com base nos requisitos a serem exigidos, e com base na natureza do projeto, além dos recursos que serão utilizados para seu desenvolvimento (WIRTH, 2012).

2.1 METODOLOGIA ÁGIL

Como já relatado previamente, o ambiente de desenvolvimento de software atual caracteriza-se por sua alta capacidade de mudança, adaptação e aperfeiçoamento durante o processo de desenvolvimento de soluções em software. Então como os modelos de desenvolvimento tradicionais são pesados, tornou-se necessária a criação de metodologias leves, que levassem em consideração a

entrega incremental de software e a adaptação às mudanças de requisitos durante o processo de desenvolvimento. (SOMMERVILLE, 2011)

Daí a origem do desenvolvimento ágil que se popularizou em 2001, quando alguns especialistas em desenvolvimento de software se reuniram, estudaram e analisaram medidas que possibilitariam o aumento dos índices de sucesso dos projetos. Contudo, o grupo de especialistas chegou à conclusão de que independentemente das práticas e ênfases de cada metodologia ágil, quatro princípios são primordiais a qualquer uma delas. Foi então publicado um documento chamado Manifesto Ágil, com o intuito de divulgar para a comunidade científica os quatro valores alcançados. Os valores são: pessoas e interações no lugar de ferramentas e processos, software executável em substituição a documentos, participação do cliente e não transações comerciais e respostas ágeis e inteligentes para mudanças no lugar de cumprir metas (KOSCIANSKI; SOARES, 2007).

Porém, observamos que o Manifesto Ágil não rejeita processos e ferramentas, documentação, negociação de contratos nem planejamento, mas simplesmente mostra que estes têm importância secundária quando comparados com os indivíduos, com o software executável, com a colaboração dos clientes e as respostas rápidas as mudanças. (KOSCIANSKI e SOARES, 2007, p. 194)

Segundo Pressman (2006), a agilidade não é apenas uma resposta às modificações que ocorrem no decorrer de um projeto. Ela engloba também a filosofia apresentada no manifesto ágil, o qual que estimula as atitudes da equipe, facilitando sua comunicação e torna o cliente uma parte da equipe de desenvolvimento com o objetivo de tornar o projeto viável e flexível.

Segundo Cohn (2011), tornar as equipes de desenvolvimento mais ágeis é o que as empresas hoje em dia têm buscado com mais frequência, com o intuito de desenvolver softwares que apresentem mais qualidade e que sejam mais fiéis aos requisitos inicialmente acordados com o cliente. Isso dá pelo fato de que equipes eficientes garantem a implantação dos produtos da empresa com mais efetividade, garantindo assim a satisfação do cliente.

Surgiram ao longo dos anos, diversas metodologias de desenvolvimento ágil, como por exemplo, *Scrum*, *Crystal Clear*, *Adaptive Software Development* (ASD), *Feature Driven Development* (FDD), *Dynamic Systems Development Method* (DSDM). Dentre elas, a principal e mais utilizada é a *Extreme Programming* (XP) e (PAULA FILHO, 2009). Porém, este trabalho se pautou na metodologia ágil Scrum,

que será explicitada no decorrer do desenvolvimento deste.

2.1.1 Manifesto Ágil

No ano de 2001, devido à insatisfação com os modelos de software clássicos, ocorreu uma reunião com um grupo de 17 notáveis desenvolvedores. Esse grupo era formado por produtores e desenvolvedores de software que trocaram experiências, teorias e práticas referentes ao desenvolvimento de software.

Neste encontro, reuniu-se um conjunto de princípios básicos necessários para o sucesso de um projeto denominado “Manifesto para o Desenvolvimento Ágil de Software” (SBROCCO, 2012).

Nesse manifesto foi declarado: Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

O manifesto ágil consiste em doze princípios. Estes foram criados pela *agile alliance*, instituição que promove o desenvolvimento ágil, e são listados a seguir (AGILLE MANIFESTO, 2013):

- Tem como prioridade satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado;
- Mudanças nos requisitos são bem-vindas, mesmo que o desenvolvimento esteja avançado;
- Garantir a entrega do software funcionando, com preferência numa escala de tempo menor;
- Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto em prol do projeto;
- Construir projetos em torno de indivíduos motivados. Dar a eles ambiente e suporte necessário e demonstrar confiança na realização de seu trabalho;

- O método mais eficiente e eficaz de transmitir informações para a equipe de desenvolvimento é através de conversa face a face;
- Software funcionando é a medida principal do progresso;
- Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante de trabalho;
- Contínua atenção a excelência técnica e design aumentam a agilidade;
- Simplicidade é essencial;
- As melhores arquiteturas, requisitos e designs provêm de equipes que se auto-organizam;
- Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e a fim de ajustar seu comportamento.

Com base nesses doze princípios, a constante alteração de escopo e inclusão requisitos passa a ser o conceito central de adaptação às mudanças do projeto. Cada um dos métodos ágeis busca oferecer um conjunto de boas práticas e atividades para serem utilizados durante o desenvolvimento de projetos.

É cada vez mais comum o uso das metodologias ágeis, no desenvolvimento de software. A aproximação do cliente do processo de desenvolvimento de software, juntamente com o uso de processo de desenvolvimento iterativo, resulta em um recurso eficiente, onde as equipes recebem o *feedback* logo após a apresentação do produto, o que acaba se tornando um excelente aliado para lidar com as constantes mudanças de funcionalidade.

3 SCRUM

3.1 DEFINIÇÃO

O Scrum é um framework de desenvolvimento de software ágil de abordagem interativa e incremental fundamentada em teorias empíricas que afirmam que o conhecimento é adquirido da tomada de decisões e da experiência baseada no que é conhecido. (SOMMERVILLE, 2011)

De acordo com Koscianski e Soares (2007, p. 200), “o foco da metodologia é encontrar uma forma de trabalho dos membros da equipe para produzir o software de forma flexível e em um ambiente em constante mudança”. Assim sendo, o Scrum prega um trabalho ágil, que pode ser executado de forma maleável e que se adapta a todos os tipos de ambientes, com a certeza de que quaisquer problemas podem ser resolvidos, desde a mudança de escopo até problemas de implantação.

A fim de suprir tais necessidades, o Scrum é composto por papéis, eventos, artefatos e regras sendo cada um deles de suma importância para seu desenvolvimento eficaz e consequente sucesso e, segundo seus criadores, Schwaber e Sutherland (2011), “as regras do Scrum integram os papéis, eventos e artefatos, administrando as relações e interações entre eles”.

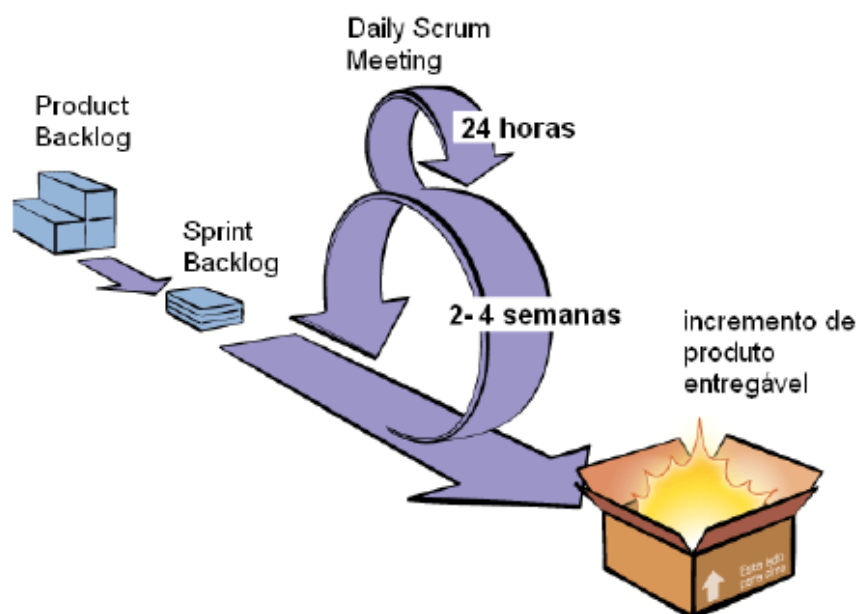


Figura 4 - Visão Geral do Scrum
Fonte: Universidade Federal de Pernambuco

A figura 4 acima demonstra uma visão abrangente do Scrum, onde estão evidenciados os eventos, artefatos, e papéis, que serão estudados a seguir.

3.1.1 Pilares do Scrum

3.3.1.1 Transparência

O *Scrum Guide* tem a seguinte definição para a transparência no Scrum:

Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados”. Esta transparência requer aspectos definidos por um padrão comum para que os observadores compartilhem um mesmo entendimento do que está sendo visto. Por exemplo:

- Uma linguagem comum referindo-se ao processo deve ser compartilhada por todos os participantes; e,
- Uma definição comum de “Pronto” 1 deve ser compartilhada por aqueles que realizam o trabalho e por aqueles que aceitam o resultado do trabalho. (Schwaber e Sutherland 2011).

A reunião diária, que será exemplificada logo mais, é uma forma de transparência que os membros da equipe podem ter entre si, e que ajuda a traçar estratégias para os próximos passos.

O Kanban, artefato do Scrum, é uma ferramenta bastante utilizada para ajudar no processo de transparência, e é bem-vindo para ajudar a transparecer o trabalho que vem sendo realizado. Pode se estender por todo o ciclo do Scrum, de tarefas que ainda não foram iniciadas até as que já foram implantadas e liberadas para uso.

3.3.1.2 Inspeção

Schwaber e Sutherland (2011) têm a seguinte definição sobre a Inspeção no *Scrum*:

Os usuários Scrum devem, frequentemente, inspecionar os artefatos Scrum e o progresso em direção a detectar variações. Esta inspeção não deve, no entanto, ser tão frequente que atrapalhe a própria execução das tarefas. As inspeções são mais benéficas quando realizadas de forma diligente por inspetores especializados no trabalho a se verificar. (SCHWABER E SUTHERLAND, 2011).

A inspeção do produto é uma prática muito importante e que deve ser levada em consideração para todo e qualquer processo que deseja ser bem-sucedido. Se feita com a devida frequência, ela dá a possibilidade de identificar rapidamente qualquer desvio sobre a meta que deve ser atingida e ainda garante que o valor agregado ao cliente possa ser medido e aprimorado junto com a adaptação, que é o terceiro pilar do Scrum.

No Scrum a reunião de revisão se apoia no pilar da inspeção visando garantir que a equipe faça a apresentação do incremento do produto, para que haja aprovação ou reprovação das atividades desenvolvidas pelo product owner. O processo deve ser inspecionado constantemente para evitar desvios. A dinâmica da equipe também deve ser inspecionada, pelo Scrum Master, na reunião de retrospectiva a fim de evitar problemas que possam ocorrer dentro da equipe. Mesmo que a solução não seja imediata, o problema precisa ser conhecido para que se encontre uma solução posteriormente a fim de evitar que o atrito da equipe atrapalhe o bom andamento do processo.

3.3.1.3 Adaptação

O Scrum Guide tem a seguinte definição para a transparência no Scrum:

Se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o produto resultado será inaceitável, o processo ou o material sendo produzido deve ser ajustado. O ajuste deve ser realizado o mais breve possível para minimizar mais desvios. “O Scrum prescreve quatro Eventos formais, contidos dentro dos limites da Sprint, para inspeção e adaptação, como descrito na seção Eventos do Scrum deste documento:

- Reunião de planejamento da Sprint;
- Reunião diária;
- Reunião de revisão da Sprint;
- Retrospectiva da Sprint” (Schwaber e Sutherland, 2011).

As características da adaptação estão fortemente presentes no processo Scrum através das cerimônias e na estipulação do *timebox* das *Sprints*.

A reunião de retrospectiva é um outro tópico importante sob o ponto de vista da adaptação do Scrum. Nessa reunião a adaptação é feita na equipe, com a correção de problemas que possam estar atrapalhando o bom andamento dos processos permitindo a reorganizando da dinâmica da equipe. Essa é uma das adaptações mais importantes, pois sem organização, o esforço da equipe para se finalizar uma Sprint será muito maior.

3.1.2 Papéis do Scrum

Podemos elencar três funções principais ao Scrum, sendo eles o Scrum Master, o Product Owner, e o time. Toda equipe envolvida no projeto do Scrum se denomina de “porco”, ficando o restante com a denominação de “galinha” (*chicken*). Essa forma de organização consiste em não permitir interferência da equipe externa, galinhas, no desenvolvimento das funções e atividades dos porcos. A figura 5 abaixo demonstra a abrangência e a ligação dos envolvidos no Scrum.



Figura 5 - História sobre o comprometimento do *Scrum Team*.
Fonte: Vizdos (2006).

3.2 TIME

3.2.1 Product Owner

O *Product Owner*, traduzido para o português como “dono do produto”, é responsável por aumentar o *Return on Investment* (ROI) pela identificação de funcionalidades do produto, que devem ser transformadas em prioridades, decidindo qual item será executado primeiro na próxima etapa. A cada início de novo ciclo, a lista de prioridades deverá ser repriorizada. Quanto ao ROI, este não se refere ao sentido comercial, e sim ao sentido de realizar a escolha certa para cada ciclo priorizados os itens a serem desenvolvidos.

Product Owner e o cliente podem ser a mesma pessoa. Enquanto que em outros casos o *Product Owner* pode representar milhões de pessoas que irão utilizar o produto final, com uma grande variedade de necessidades (SUTHERLAND, 2012).

Segundo Sbrocco (2012), o *Product Owner* é responsável por garantir que a equipe agregue valor ao negócio e faz a ponte entre a equipe e o cliente. Tem a responsabilidade de manter a equipe funcional e de definir as funcionalidades do produto, elaborar e manter o *Product Backlog*, definir o lançamento do produto, e aceitar ou rejeitar os resultados dos trabalhos.

3.2.2 Time de Desenvolvimento

É a equipe de desenvolvimento que não é denominada com os papéis tradicionais da Engenharia de Software, como programadores, analistas de qualidade, engenheiros de software, executores de testes, etc. Toda equipe é envolvida, de forma que todos possam participar e consigam gerenciar suas atividades de forma prática e útil. É composta por até 9 pessoas, e todos os membros são responsáveis por atingir os objetivos definidos em cada ciclo [SCHWABER, 2004]. Tem a responsabilidade de priorizar os itens que serão

executados em cada iteração, cumpre os objetivos com total liberdade demonstra o trabalho desenvolvido ao *Product Owner*.

3.2.3 Scrum Master

O *Scrum Master* é responsável pelo processo Scrum, por garantir que seja entendido e aplicado, por implementá-lo - fazendo dele uma cultura na organização - e ainda por garantir que toda a equipe siga as regras e as práticas do Scrum (SCHWABER; SUTHERLAND, 2013). É responsável também por proteger a equipe e assegurar que ela não se comprometa excessivamente.

Atua como intermediador na Reunião diária e torna-se responsável por remover quaisquer obstáculos que sejam levantados pela equipe durante essas reuniões. Pode ser exercido por qualquer membro da equipe, porém, normalmente é exercido por um gerente de projetos (SCHWABER; SUTHERLAND, 2013).

3.3 EVENTOS SCRUM

3.3.1 Sprint

Os projetos que fazem uso de Scrum são divididos em ciclos chamados de *Sprint*, os quais correspondem às iterações. Esses ciclos duram em torno de duas a quatro semanas, tendo o principal objetivo o de gerar um produto de valor ao cliente e dentro dos requisitos previamente estabelecidos. O cliente e a equipe devem ter ciência de que nenhuma mudança deve ocorrer durante um *Sprint* (SBROCCO, 2012).

Sbrocco (2012, pg. 163) define o *Sprint* como “[...] a unidade básica de desenvolvimento do Scrum, entendida como um esforço dentro de uma “caixa de tempo”; em outras palavras, um esforço restrito de duração específica”.

Já Sutherland (2012), define o *Sprint* como:

Uma iteração, ou um ciclo repetitivo de trabalho, que produz um incremento de produto ou sistema. Não mais que um mês e, geralmente, mais de uma semana de ciclo. A duração é fixada ao longo do trabalho total e todas as equipes trabalhando no mesmo sistema ou produto utilizam o mesmo tempo de ciclo (SUTHERLAND, 2012, pg. 30).

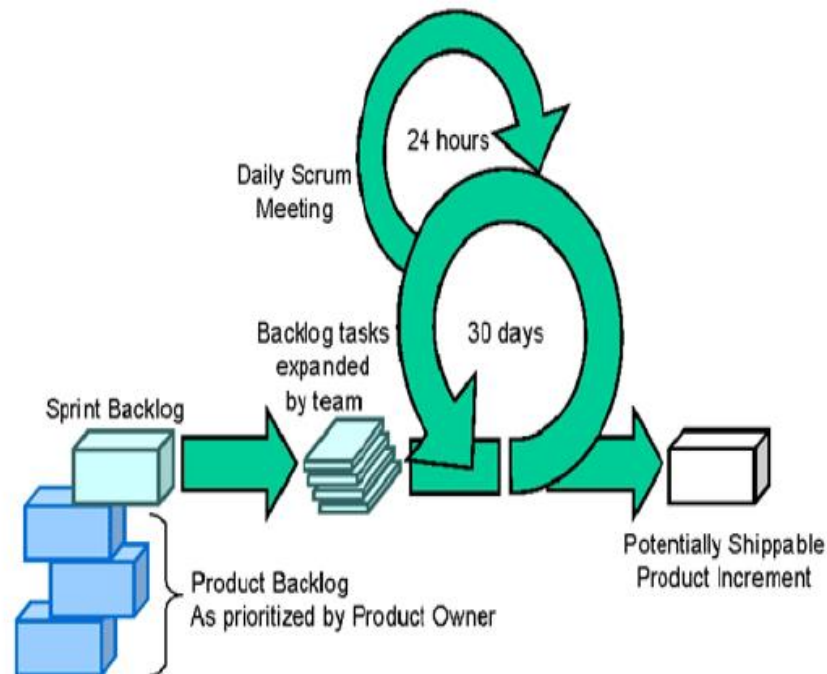


Figura 6 - Sprint
Fonte: Pereira et al 2007

3.3.2 Cancelamento de Sprint

Inerente a qualquer execução de projeto, cancelar uma Sprint pode vir a ocorrer a qualquer momento. Os usuários, o *Scrum master* e o time podem chegar a um consenso da necessidade de cancelamento da Sprint, porém, somente Product Owner é o responsável por autorizar seu cancelamento.

Mudanças de procedimento e tecnologia, negociação, alterações no mercado, legislação ou até mesmo estratégia empresarial podem ser impactantes nos objetivos do projeto e consequentemente serem responsáveis pelo cancelamento da Sprint.

Ao cancelar uma *sprint*, o *product owner* deve revisar os itens da *backlog* que já estão prontos, e planejar o momento em que estes entrarão novamente em desenvolvimento, para que se reduzam os impactos no rendimento e motivação da

equipe e no planejamento futuro das atividades, além de reduzir também o grande consumo de recursos para se iniciar uma nova Sprint. Como não pode ser comum o cancelamento de uma Sprint, seguem abaixo algumas precauções que devem ser tomadas a fim de reduzir as chances de isso acontecer:

- a) Projetar Sprint com duração menor, no máximo 2 semanas;
- b) Oferecer maleabilidade no ajuste do prazo final para entrega do produto, considerando o fim de cada meta da Sprint um pouco antes do plano total;
- c) Aperfeiçoar o nível de detalhe do *backlog*

3.3.3 Reunião de planejamento de Sprint

Durante esta reunião de planejamento, na qual todo o time Scrum deve estar presente, o trabalho a ser realizado durante a Sprint é planejado. Esta reunião deve ter, no máximo, oito horas de duração para uma Sprint de quatro semanas. Os resultados desta reunião resumem-se na definição do produto a ser entregue pela Sprint em questão e como o trabalho a ser desenvolvido durante a Sprint será realizado.

3.3.4 Reunião diária

A reunião diária de *Scrum (Daily Scrum)* é um encontro rápido com todos os membros time com a finalidade de definir quais serão as tarefas do dia e explicar sobre os resultados do dia anterior. Esta reunião é também chamada de *Stand Up Meeting* (reunião em pé), de forma a conseguir maior agilidade. Três perguntas são respondidas por cada membro sobre suas responsabilidades (RISING; JANOFF, 2000): O que foi feito ontem? O que será feito hoje? Há algum obstáculo à realização das atividades? Porém essas perguntas não são respondidas como meios de prestar contas sobre o serviço que vem sendo executado, e sim como uma ferramenta de comprometimento para com o restante do time. Se faz importante para que os demais membros do time conheçam as metas individuais de cada

integrante, seus impedimentos e logo cobrar por compromissos assumidos.

3.3.5 Revisão de Sprint

No final de cada *Sprint* é realizada uma revisão onde é necessário apresentar e verificar o produto gerado e caso seja necessário, adaptar o *Product Backlog*.

Como parte do princípio do Scrum, essa é uma reunião é rápida e informal, normalmente essa reunião dura até 4 horas pois é necessário definir o *Product Backlog* e a mesma é composta pelo time *Scrum* e as partes interessadas convidadas pelo *Product Owner*.

3.3.6 Retrospectiva da Sprint

Com o intuito de complementar o ciclo do *Scrum*, é realizado a Retrospectiva do *Sprint*. Essa reunião é realizada entre o *Scrum Master* e a equipe, após a revisão do *Sprint* e antes da reunião de planejamento da *Sprint*, onde o *Product owner* pode ser convidado a participar. Segundo Sbrocco (2012), o principal objetivo dessa reunião é verificar o que houve de bom durante o processo e deve ser mantido, e os pontos negativos que deverão ser melhorados. Duas perguntas devem ser respondidas pela equipe: quais os pontos positivos que podem ser apontados e que ocorreram durante a *Sprint*, e o que pode ser feito para aprimorara a próxima *Sprint*. Fica ao encargo do *Scrum Master* tomar nota de todos os itens apontados na reunião para que seja criada lista ordenada com as alterações. Quando comparados o *Sprint Review* e o *Sprint Retrospective*, podemos identificar que o *Sprint Review* envolve inspeção e adaptação com relação ao produto enquanto *Sprint Retrospective* tem o foco no processo *Scrum* (SBROCCO, 2012).

3.4 ARTEFATOS DO SCRUM

Sbrocco (2012) discorre sobre os artefatos do Scrum como sendo o *Product Backlog*, o *Sprint Backlog* e o *Burndown Chart*, que são gerados em fases diferentes de cada Sprint.

Sprint Backlog relaciona uma série de tarefas oriundas do *Product Backlog*, que a equipe deve executar numa *sprint*. Ou seja, a equipe fica responsável pela implementação dos itens e entregar seus sistemas funcionando (SCHWABER, 2004). A equipe é a única que pode alterar a *Sprint Backlog* e as tarefas devem durar em turnos de 4 a 16 horas.

O gráfico *Burndown* (*Burndown Chart*) tem como objetivo oferecer clareza do progresso do trabalho à equipe. Ele correlaciona as etapas de trabalho que ainda faltam ser executadas e o progresso das equipes na execução destes. O gráfico apresenta então duas retas: a tendência que é o ideal que deveria estar sendo executado e a outra reta é a real, que apresenta como o trabalho está efetivamente sendo desenvolvido. As retas sempre estão decrescendo à maneira que as atividades vão sendo realizadas e finalizadas (SCHWABER, 2004), com mostra a figura 7 abaixo.

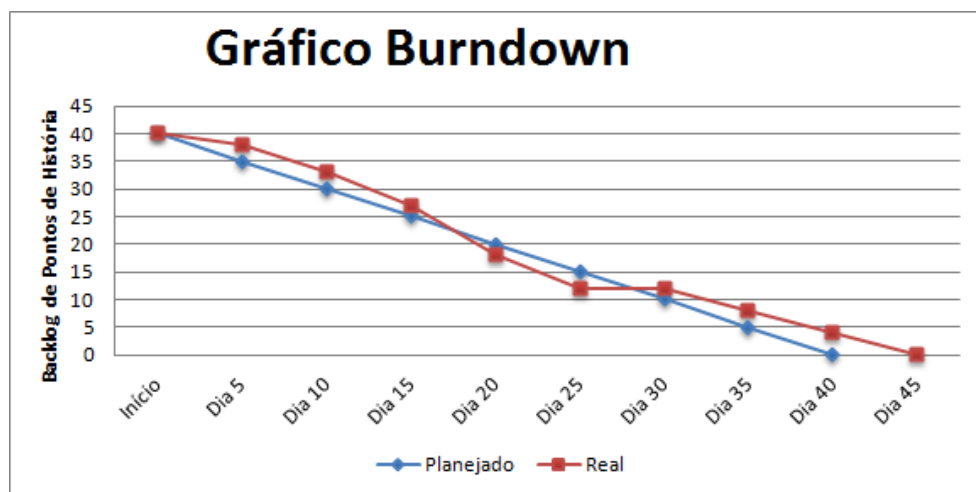


Figura 7 - Gráfico Burndown - modelo
Fonte: Próprio autor

3.4.1 Backlog do Produto

São tarefas que devem ser executadas durante a *sprint*, composta por preceitos funcionais. Ele permite que o processo seja controlado e gerenciado. O *product owner* é o responsável por manter o *backlog* do produto atualizado e disponibiliza-lo à equipe sempre que preciso.

3.5 TRANSPARÊNCIA DO ARTEFATO

O guia definitivo do Scrum relata que esta metodologia exige transparência nas decisões, de forma a aprimorar o valor e o controle. Caso não haja transparência, os artefatos podem apresentar falhas e aumentar os riscos, fazendo com que ele perca valor.

Schwaber (2011) afirma que o Scrum Master, juntamente com a equipe deve relatar os problemas apresentados na transparência. Várias atividades existentes ajudam a equipe a aprimorar a transparência do artefato. As falhas nas transparências podem aparecer por exemplo, nas diferenças entre o esperado e o resultado efetivo, e o Scrum Master é quem responde pela extinção de tais falhas. Trabalho este que exige experiência do Scrum Master e é um caminho que deve ser seguido sempre.

4 ESTUDO DE VIABILIDADE DE IMPLEMENTAÇÃO DO SCRUM

4.1 EMPRESA

A empresa objeto deste estudo atua no desenvolvimento de software para o setor de varejo e está há mais de 28 anos no mercado. Os donos optaram por não fornecerem dados societários e de cadastro, então chamaremos somente de “empresa”.

4.2 EQUIPE

A empresa conta com aproximadamente 60 colaboradores no total, sendo 30 colaboradores da área de desenvolvimento de software, divididos por assunto/módulo.

4.3 SOFTWARE

A empresa é especializada em produto de software de venda para comércio varejista para todo o território nacional. É importante ressaltar que neste ramo as alterações de legislações são constantes, pois cada unidade federativa pode adotar suas próprias obrigações tributárias.

4.4 PROCESSO ATUAL

Após vários estudos, decidiu-se pela implantação do Scrum na empresa pois ela não contava com nenhuma metodologia de desenvolvimento ágil. O fluxo de

desenvolvimento de um software antes da implantação pode ser definido como abaixo, e está demonstrado também na figura 8.

- Abre-se uma nova solicitação de criação ou correção de software;
- Analisa-se a urgência da solicitação, se urgente, vai para o desenvolvedor de imediato, se não, vai para uma fila de espera;
- O desenvolvedor realiza as correções e incrementos sem definição do prazo de entrega;
- Após o termino, a versão é testada e liberada para uso.

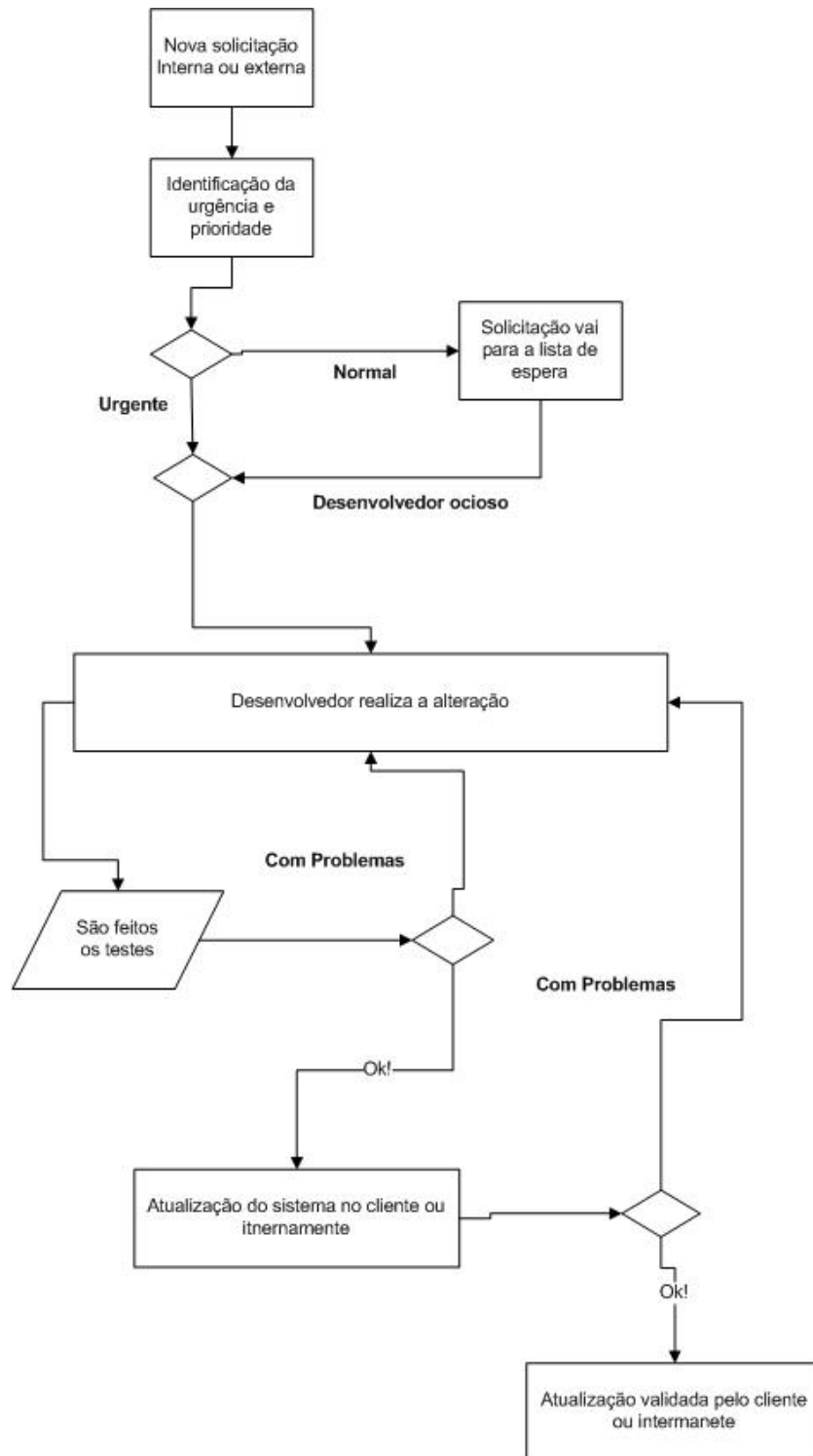


Figura 8 - Fluxograma de desenvolvimento de solicitações antes da implantação do Scrum
Fonte: Elaborado pelo autor com base em dados fornecidos pela empresa.

Um dos grandes problemas enfrentados pela falta de desenvolvimento com metodologia ágil foi a falta de controle sobre as solicitações, e falta de informação dos envolvidos nos processos e do gerente de desenvolvimento e do solicitante.

Sempre que uma tarefa urgente é solicitada, como não havia um

planejamento prévio, eram necessárias diversas horas extras dos colaboradores da equipe de desenvolvimento para conseguirem dar conta da demanda, o que afetava diretamente na qualidade do serviço prestado e desejado pela empresa.

O gerente de desenvolvimento detinha o controle das operações, e as mantinha em planilhas de Excel, acessível a todos da equipe. A equipe é dividida por módulos e/ou assunto, então sempre que as solicitações chegavam, o gerente as direcionava para a equipe correta, nomeando conforme produto ou cliente, porém, as solicitações não eram bem detalhadas o que acabava dificultando e atrasando o desenvolvimento do produto.

Alguns dos problemas enfrentados pela falta de controle eficaz era a mesma solicitação em mais de uma equipe, por falha do gestor da área em não saber direcionar corretamente as solicitações ou por sobrecarga de tarefas, e o baixo nível de detalhamento das solicitações que demandava ainda mais tempo para conclusão da correção.

A tabela 1 abaixo revela dados das solicitações abertas e concluídas num prazo de 8 meses antes da implantação do Scrum.

Tabela 1 - Solicitações antes da implantação do Scrum

Mês / Ano	Solicitações abertas	Solicitações Concluídas e encerradas	Prazo médio para conclusão
JUN/2015	5	1	26 dias
JUL/2015	23	5	62 dias
AGO/2015	7	1	32 dias
SET/2015	2	1	25 dias
OUT/2015	18	5	45 dias
NOV/2015	16	2	28 dias
DEZ/2015	20	4	38 dias
JAN/2016	31	13	98 dias
TOTAIS	122	32	-
MÉDIA MENSAL	15,25	4	-

Fonte: Elaborado pelo autor com base em dados fornecidos pela empresa.

Com base na tabela 1 acima, podemos verificar que os meses de Junho, Agosto e Setembro tiveram apenas uma solicitação concluída e com prazo muito longo de conclusão. Isso se deu porque toda a equipe estava envolvida na implantação de um novo sistema em dois novos clientes. Podemos observar também que para os demais meses, o prazo também está elevado para a conclusão das tarefas.

Com a implantação da nova metodologia, levando em conta o planejamento

estratégico da empresa que inclui a captação de novos clientes, uma reestruturação na equipe de desenvolvimento será necessária e urgente. Com base nos objetivos da empresa, o gerente de desenvolvimento ressaltou alguns pontos importantes para a implantação de uma metodologia de desenvolvimento ágil:

- Carência de transparência no planejamento e desenvolvimento, sendo então necessárias mais reuniões periódicas com todos os envolvidos a fim de melhorar o monitoramento do processo;
- Conseguir atender às mudanças frequentes nos requisitos do projeto;
- A falta de agilidade na correção dos problemas que surgem após a implantação do software;
- Otimização do trabalho, a fim de reduzir as horas extras que além de frequentes deixavam a equipe desmotivada.

4.5 ETAPAS PARA IMPLEMENTAÇÃO

Com base nos requisitos apresentados pelo gerente de desenvolvimento, e pelas características de cada colaborador das equipes de desenvolvimento, ficou decidido que ao Scrum seria a melhor metodologia ser implantada na empresa. Segundo Cohn (2011), é primordial observar que o processo atual não está gerando os resultados esperados que saber o momento de mudar o esquema de trabalho.

A metodologia ágil Scrum foi apresentada a toda equipe, com foco nos objetivos e processos de funcionamento e utilização.

Dessa forma, pode ser feito um planejamento para implantação da metodologia, como pode ser observado abaixo:

- Sistematizar as solicitações por ordem de urgência com uma nova ferramenta, que seja mais prática e de mais fácil acesso, produzindo o *backlog* do produto. Também será feita uma alteração para melhorar o nível de detalhamento das solicitações com a inclusão de dados mais consistentes como por exemplo um *print* das telas com os erros a serem corrigidos.

- Definir reuniões para definir os prazos e o *backlog* para o primeiro *Sprint* a ser executado;
- Realizar a primeira reunião de encerramento da *Sprint*;
- Definir a primeira entrega aos clientes após a implantação do Scrum.

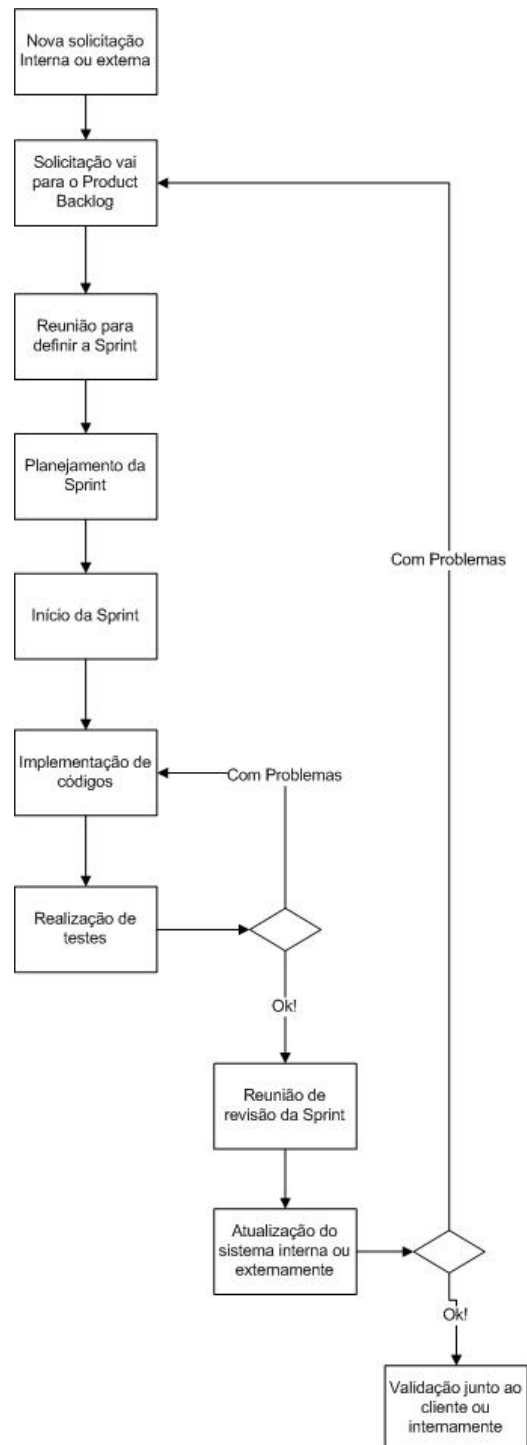


Figura 9 - Fluxograma de desenvolvimento de solicitações após a implantação do Scrum
Fonte: Elaborado pelo autor com base em dados fornecidos pela empresa.

A ferramenta web Asana (2015)¹ foi escolhida pela equipe para gerenciar o controle do *Backlog* do produto. Essa ferramenta é considerada segura para gerenciar esse tipo de projeto e possui um corpo de interação prático e de fácil assimilação, além de estar disponível para aparelhos móveis, para Android e iOS, sem haver então a necessidade de um computador para gerenciá-la.

Após implantada, a ferramenta demonstrou ser de grande valia para a equipe e para o processo de implantação do Scrum quando relacionada às *Sprints* e *Backlogs* pois facilitava à equipe ter uma visão ampliada de todos o processo em execução. Ela também permite que informações sejam adicionadas individualmente para cada etapa de desenvolvimento e solicitação, o que simplificou o processo de comunicação entre o desenvolvimento e a equipe de testes.

As solicitações foram então divididas por segmento. As solicitações de módulos e clientes fazem parte da *backlog* do produto e as outras fazem parte da *backlog* da *Sprint* que está em desenvolvimento, que é subdividida em três partes: a parte que está no departamento de desenvolvimento, a parte que está sendo testada e a parte que já está concluída.

A figura 10 abaixo exemplifica a ferramenta web Asana.

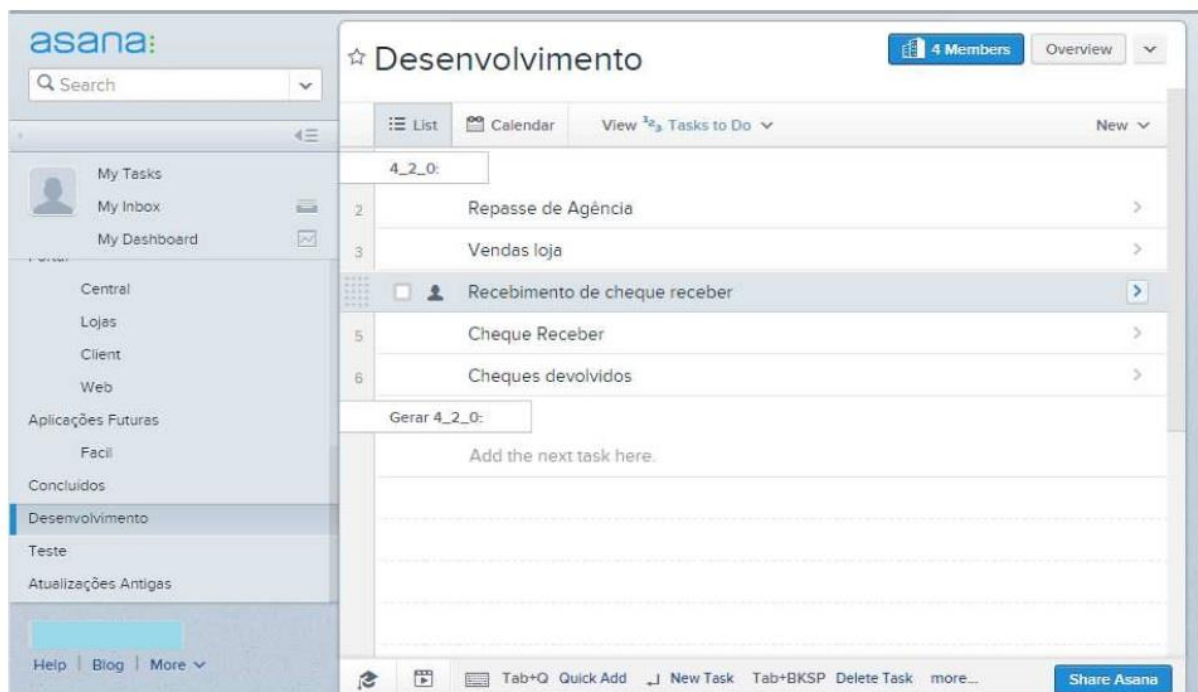


Figura 10 - Ferramenta Web Asana
Fonte: Souza, 2014.

¹ Disponível em :< <https://asana.com/>> Acessado em 03 de maio 2016

4.5.1 Mudança de papéis e funções

A mudança de papéis após a implantação do Scrum se torna inevitável podendo ser torturante para a equipe, seja na criação de novos ou extinção de papéis antigos (COHN, 2011).

Para a empresa objeto deste estudo, dois papéis foram criados: o Scrum Master e o dono do produto. Quem ocupa a função de analista de sistemas, adquiriu o papel de dono do produto, pois o analista de sistemas geralmente tem mais conhecimento sobre os produtos e suas funções. O papel de Scrum Master ficou com o gerente de desenvolvimento, que passa agora a obter o título de Gerente de Projetos. A tabela 2 abaixo exemplifica as novas funções do Scrum Master. Este papel foi o que mais sofreu alteração nas tarefas.

Tabela 2 - Papéis do Gerente de Projetos e Testes.

Gerente de Projetos e Testes	Scrum Master
Garantir o desenvolvimento de processos conforme o que solicitado pelo cliente.	É o líder da equipe. Ajuda a aprimorar o rendimento do time e garante a utilização do Scrum.
Gerencia todo o processo, riscos, recursos humanos, despesas, entre outros.	Possui alçada de controle sobre o processo, mas não possui autoridade sobre a equipe.

Fonte: Elaborado pelo autor conforme informações cedidas pela empresa.

Com a implantação da nova metodologia, algumas mudanças foram acontecendo gradativamente. A mudança nos papéis como já foi citado anteriormente é uma delas. Outra mudança é nas funções do desenvolvedor ou programador que agora se torna parte de todo o processo e no entendimento dos requisitos de todo o produto.

Os responsáveis pelos testes dos produtos também tiveram suas funções alteradas. Eles não recebem perfeitamente as especificações que garantem as funcionalidades do sistema (COHN, 2011).

A comunicação da equipe era ruim antes da implantação do Scrum. Agora essa comunicação melhorou bastante o que garante uma boa execução dos trabalhos garantindo assim a qualidade dos processos.

Ao final da Sprint, é feita a reunião para garantir que o produto a ser entregue está funcionando corretamente. Cohn, (2011) afirma que aprender a entregar um software que funciona perfeitamente é o principal desafio da implantação da metodologia Scrum. A fim de garantir sucesso na entrega do produto, a empresa passou a controlar as entregas por número de versões.

4.6 ANÁLISE DE RESULTADOS

Após a inserção do Scrum na metodologia de trabalho, foi possível observar as alterações tanto em processos como em resultados, partindo da premissa dos processos iniciais.

Num primeiro momento, havia a perda de tempo e aumento de custo num projeto devido o fato de haverem desencontros de informações dentro da organização e até mesmo dentro da equipe dos chamados, seja por falta de detalhes nas requisições ou pelo encaminhamento de chamados em duplicidade, gerando aumento do tempo de avaliação do chamado sem necessidade.

Com a implantação da metodologia, passou-se a dar maior importância à programação prévia da distribuição das solicitações, fato não observado na metodologia atual. Pelo fato de não haverem reuniões de definição de técnicas e distribuição mais clara das tarefas, observava-se certa falta de conhecimento sobre como seguir as etapas do processo, e até mesmo dúvidas sobre as próprias etapas, devido o baixo grau de detalhamento e falta de gerenciamento. Com a implementação do Scrum e a realização de reuniões prévias para definição de requisitos iniciais dos projetos foi possível observar um aumento no atendimento de chamados devido uma melhor gestão dos chamados e da equipe, uma vez que dividida em segmentos e com um maior detalhamento e repasse de informações, pôde aumentar o atendimento à demanda de chamados e novos projetos.

Com todos esses benefícios, a maior observação de viabilidade se dá pelo fato de ter aumentado o número de chamados e reduzido os custos finais de projeto, observado também pela redução no número de horas extras necessárias para que fossem concluídos os chamados pendentes, bem como o aumento qualitativo da produção, com redução de retrabalhos e problemas com prazos de entrega, muito observados antes da implantação do Scrum. Podemos observar isso por meio do número de chamados encerrados no mês de março após a implementação, onde foram encerrados 33 chamados no mês, valor igual ao de chamados encerrados no período de junho de 2015 a janeiro de 2016. Além disso, os prazos médios de espera foram reduzidos, chegando até à conclusão de chamados em até 19 dias, valor quase 27% quando comparado ao melhor mês do período pré-implantação do Scrum, onde foram gastos 26 dias para resolução de um único chamado, sendo

que no mesmo mês de março foram corrigidos 3300% de casos além do que o mês de junho de 2015. A melhor gestão dos prazos e das informações possibilitou melhora na tratativa dos projetos, onde houve maior tempo hábil para realização de testes e verificações nos comandos e códigos gerados, podendo assim disponibilizar maior parcela de tempo para possíveis retrabalhos e alterações, também reduzidas com a implementação do Scrum, que possibilitou ganhos quantitativos e qualitativos para toda a organização.

5 CONSIDERAÇÕES FINAIS

5.1 OBSERVAÇÕES

Pôde ser verificado que as metodologias tradicionais de não atendem às necessidades atuais de desenvolvimento de software, que exigem sempre inovações, respostas rápidas e eficientes além de, é claro, o aumento na competitividade das empresas. Com isso fica claro que as metodologias ágeis chegaram para suprir essa demanda e atender às exigências do mercado atual, assegurando formalização dos processos, alto desempenho na execução dos trabalhos, considerando sempre o atendimento aos requisitos do sistema, requisitos de segurança e de controle do processo como um todo.

5.2 BENEFÍCIOS DA IMPLEMENTAÇÃO DO SCRUM

Para que seja possível expor a diferença de entregas e resultados observados após a implementação da nova tecnologia, a tabela abaixo confronta dados apresentados na Tabela 1, que indica os dados antes da implementação do Scrum e as variações de quantidade de chamados atendidos e a redução dos prazos médios de entrega dos mesmos, reduzidos em proporções consideráveis. Para ilustrar a situação inicial, foi realizado o diagrama abaixo, a fim de filtrar as adversidades para chegar às causas e efeitos do problema, conforme mostrado:

Tabela 3 - Solicitações depois da implantação do Scrum.

Mês / Ano	Solicitações abertas	Solicitações Concluídas e encerradas	Prazo médio para conclusão
FEV/2016	25	23	52 dias
MAR/2016	32	33	19 dias
ABR/2016	42	39	22 dias
MAI/2016	12	11	16 dias
Totais	111	106	-
Média Mensal	27,75	26,5	-

Fonte: Elaborado pelo autor com base em dados fornecidos pela empresa.

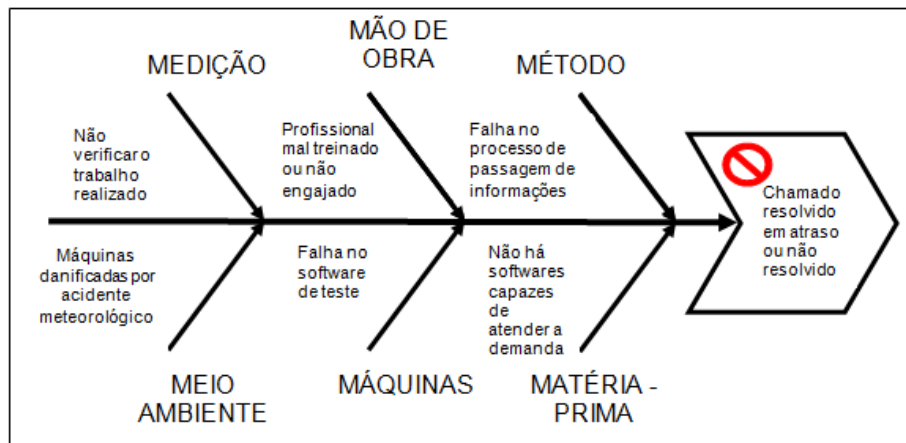


Figura 11 - Diagrama de Ishikawa para a empresa antes do Scrum
 Fonte: O Autor (2016)

Pelo diagrama acima, observamos que inúmeras podem ser as causadoras do problema conhecido na organização. Variando desde método até mão de obra, é possível que tenham ocorrido desencontros que dificultaram a realização das atividades. Para encontrar a causa-raíz do problema, optou-se pela idealização do método conhecido como “5 Por quês” ou “5 Why”, para que chegue-se a fundo na questão e possa identificar a causa real do problema, conforme feito no modelo abaixo:

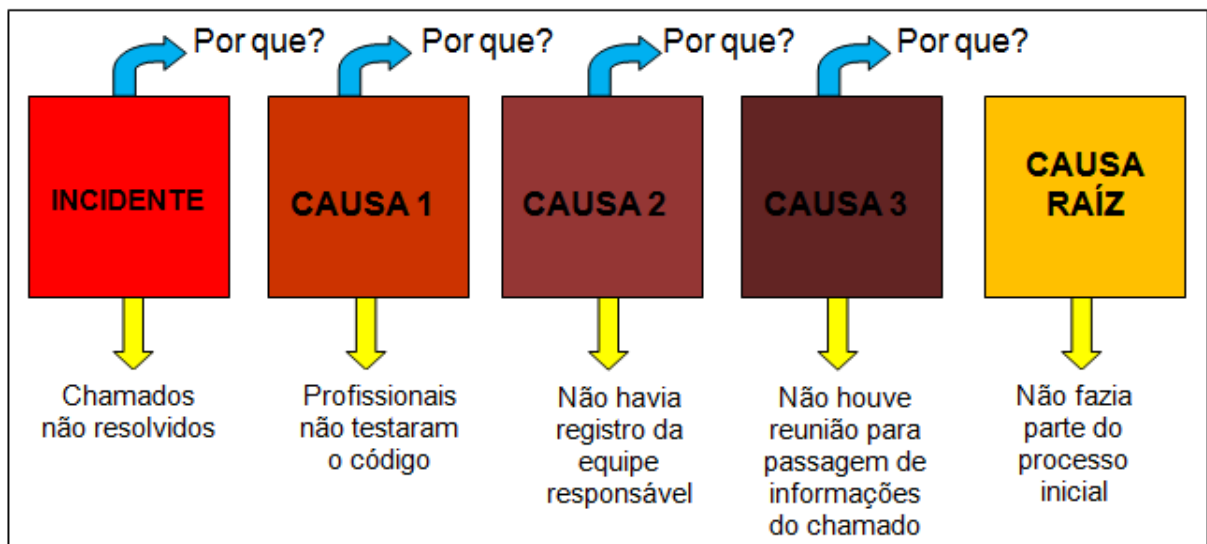


Figura 12 - Diagrama dos “5 Por quês” para definição da causa raiz
 Fonte: O Autor (2016)

Podemos então observar que em quatro níveis do diagrama podemos verificar que por conta a falta de comunicação e por um processo fragilizado de planejamento, houve desencontro de ideias, chegando à falha na etapa final, que

consiste na não resolução do chamado, ou até mesmo podendo ser inserido no contexto de atendimento fora de prazos.

Podemos observar que houve um aumento relevante no número de solicitações atendidas e concluídas e um decréscimo no prazo médio para conclusão das tarefas. A metodologia Scrum permitiu que as solicitações fossem atendidas dentro de prazos aceitáveis e com o máximo de eficiência. Esse aumento gradativo pode ser observado no gráfico abaixo, que relaciona as solicitações concluídas no prazo de junho de 2016 à maio de 2016, período previamente mencionado acima como sendo o período de implementação do Scrum, que passou a apresentar aumento no número de resoluções e queda no tempo médio de atendimento à cada novo chamado:

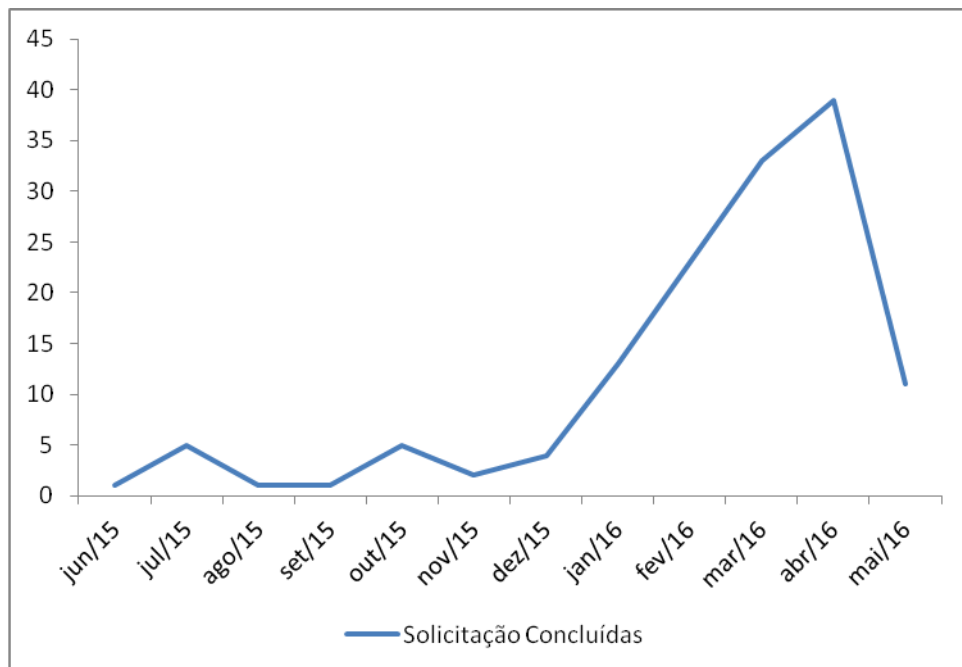


Figura 13 - Solicitações concluídas por mês
Fonte: Elaborado pelo autor

A figura 13 acima demonstra o fluxo de solicitações concluídas por mês. Pode ser observado que os meses de março e abril apresentaram o maior número de solicitações atendidas. Esses dados deixaram a equipe mais motivada ainda, e com a certeza de que a metodologia Scrum era a melhor forma de gerenciar os processos de desenvolvimento de software na empresa.



Figura 14 - Prazo médio de conclusão das solicitações
Fonte: Elaborado pelo autor

A figura 14 acima demonstra a média de dias que as tarefas levaram para ser concluídas. A queda no tempo de execução demonstra que equipe se tornou mais eficiente após a implantação da metodologia.

As Metodologias ágeis têm se tornado cada vez mais importantes no desenvolvimento de softwares, pois apresentam respostas rápidas e eficazes à nova demanda de clientes e à alta complexidade dos novos softwares de mercado. Elas surgiram para preencher um lócus existente nos departamentos de desenvolvimento, suprimindo as necessidades destes e garantindo bom desempenho na criação e qualidade do produto entregue, sem é claro, se esquecer das exigências de segurança e controle do processo.

A implantação do Scrum na empresa estudada gerou estímulo para a equipe, que pode aplicar todo seu potencial à nova metodologia. De início houve dificuldade de adaptação da equipe, mas com a ajuda e participação direta do gerente de projetos esse processo foi menos doloroso.

A repercussão causada pelas mudanças foi vista de forma positiva por toda a empresa, garantindo trabalho em conjunto e aumentando o rendimento da equipe de desenvolvedores.

Um ponto importante a ser destacada é a quantificação dos dados informados nas tabelas 1 e 3. Foi observado que a metodologia Scrum melhorou todo o

processo de desenvolvimento, desde o cadastro de novas solicitações até a entrega final do produto.

A mudança cultural na empresa é um importante ponto a se destacar, uma vez que o processo de desenvolvimento da empresa está ancorado na gestão bem definida de processos a serem executados por todos. Devemos preconizar que o envolvimento de toda a equipe e a decisão tomada por todos em síncrono foi crucial para o sucesso da implementação da ferramenta.

Podemos concluir ao final então que os objetivos propostos foram alcançados integralmente. Primeiramente realizamos a revisão da literatura sobre as metodologias ágeis de desenvolvimento de software, com foco no Scrum. Posteriormente realizamos o estudo da empresa objeto deste trabalho elaborando um planejamento estratégico de implantação da metodologia escolhida, preparando e adaptando a empresa às mudanças que viriam a ocorrer e realizando todas as mudanças que foram julgadas como necessárias para obter sucesso na implantação do Scrum. Finalizamos então com a análise final do processo após a implantação da metodologia, ratificando então o sucesso da utilização da metodologia no desenvolvimento de softwares.

Além disso, houve mudanças nas reuniões de levantamento de requisitos e de início de novos projetos, que antes não existiam e passaram a fazer parte da rotina após a implantação do Scrum.

No cenário anterior, o processo tinha início com o cliente que requisitava alguma melhoria ou uma nova função, e esse contato era realizado diretamente com o gerente de projetos. O gerente de projetos analisava as solicitações a fim de decidir se a alteração será realizada e as organizava em listas a fim de distribuir as tarefas. Em consequente, o gerente de projetos elencava os itens da lista de acordo com a experiência de cada membro da equipe e ordena o início do desenvolvimento. De posse do trabalho a ser realizado, cada membro da equipe de desenvolvimento estima o tempo que gastará para desenvolver tal ferramenta e repassava a informação para o gerente de projetos.

Após a implantação, as solicitações de alteração de escopo ou início de um novo projeto continuam sendo repassadas diretamente ao gerente de projetos, mas agora são realizadas as reuniões diárias, além das reuniões de planejamento, revisão e retrospectiva, a fim de gerar um para posicionamento e decisão em relação ao desenvolvimento das atividades a serem executadas ou concluídas, o

que gerou agilidade no processo, aumentou a qualidade do produto entregue e conseqüentemente uma melhor coesão no trabalho da equipe que agora trabalha em conjunto para a conclusão de cada projeto.

5.3 SUGESTÕES PARA TRABALHOS FUTUROS

Como sugestão para trabalhos futuros é sugerida implantar um sistema de gestão da qualidade, onde o controle da qualidade seja feito através de avaliação periódica do desempenho geral do projeto, através das informações obtidas nas reuniões diárias e no gráfico de Burndown.

Também como sugestão de trabalhos futuros, pode ser feita a elaboração de uma técnica de indicação de dependências entre as tarefas a serem realizadas pela equipe a fim de ajudar a equipe na execução de suas tarefas.

Alem disso, os trabalhos futuros também podem:

- Comparar abordagens ágeis e tradicionais de desenvolvimento de *software*;
- Analisar modelos de melhoria de processos de desenvolvimento de *software* e seus benefícios;
- Analisar se a implantação de outro método ágil é mais executável e útil que a metodologia *Scrum*. Neste trabalho, não pôde ser feita a análise das vantagens e desvantagens de outros métodos ágeis, mas esta avaliação já foi realizada em outros trabalhos, como em Abrahamsson et al. (2002).

REFERÊNCIAS BIBLIOGRÁFICAS

ABRAHAMSSON, P. et al. **Agile software development methods** – review and analysis. Espoo: VTT Publications, 2002.

AGILE MANIFESTO. **Manifesto for Agile Software Development**. Disponível em: <<http://www.agilemanifesto.org>>. Acesso em 26 abril de 2016.

BECK, K. **Embracing Change With Extreme Programming**. IEEE Computer. p. 70–77. 1999.

BECK, K. **Programação Extrema (XP) explicada: acolha as mudanças**. Bookman, Porto Alegre, 2004.

COHN, M. **Desenvolvimento de Software com Scrum: Aplicando métodos ágeis com sucesso**. Porto Alegre: Bookman Companhia Editora Ltda., 2011.

DIJKSTRA, E. W. **The humble programmer**. 1972, 17p. Disponível em: <<https://www.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD340.html>>. Acesso em: 22 Abril 2016.

HIRAMA, Ki. **Engenharia de Software: Qualidade e Produtividade com Tecnologia**. Rio de Janeiro: Elsevier Editora Ltda., 2012.

HOUAISS, A.; VILLAR, M. de S.; FRANCO, F. M. de M. Houaiss: **Dicionário da língua portuguesa**. Rio de Janeiro: Editora Objetiva Ltda., 2001.

JUBILEU, A. P. **Modelo de Gestão do Processo de Venda e Desenvolvimento de Software On-Demand para MPE's**. 2008. 330 f. Tese (Doutorado) - Curso de Engenharia de Produção, Universidade Federal de São Carlos (ufscar), São Carlos, 2008.

KNIBERG, H. **Scrum e XP direto das Trincheiras – Como Fazemos Scrum**. Publisher C4Media editor of InfoQ.com, 2007.

KOSCIANSKI, A.; SOARES, M. dos S. **Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. 2.ed. São Paulo: Novatec, 2007.

LANDIM, H. F. **Uma Abordagem de Monitoramento dos Fatores e Condições que influenciam nas Práticas Ágeis**. 2012. 185 f. Dissertação (Mestrado) - Curso de Mestrado em Informática Aplicada, Fundação Edson Queiroz Universidade de Fortaleza Centro de Ciências Tecnológicas, Fortaleza, 2012.

LEITÃO, M. de V. **Aplicação de Scrum em Ambiente de Desenvolvimento de Software Educativo**. 2010. 72 f. TCC (Graduação) - Curso de Engenharia da Computação, Escola Politécnica de Pernambuco – Universidade de Pernambuco, Recife, 2010.

LEITE, Alessandro Ferreira. **Metodologia de desenvolvimento de Software**. Disponível em: <<http://www.devmedia.com.br/metodologiade-desenvolvimento-de-software/1903>>. Acesso em: 08 abr. 2016.

PAULA FILHO, W. de P. **Engenharia de Software: Fundamentos, Métodos e Padrões**. 3. ed. Rio de Janeiro: Ltc – Livros Técnicos e Científicos Editorial S.a, 2009.

PRESSMAN, R. S. **Engenharia de Software**. 6ª Ed. São Paulo: McGraw-Hill, 2006.

PRESSMAN, Roger S.. **Engenharia de Software**. São Paulo: Pearson Education do Brasil, 2010.

RAMOS, Almir et al. **Egenharia de Software: Modelo Cascata**. Disponível em: <<http://modelocascata.blogspot.com.br/>>. Acesso em: 01 mai. 2016.

RISING, L.; JANOFF, N. S. **The Scrum software development process for small teams**. IEEE, 2000 Disponível em: <<http://ieeexplore.ieee.org/>>. Acessado em 28 abr 2016.

ROCHA, I. **Lean Software Development: Desenvolvimento de Software Leve**. 2013. Disponível em: <<http://igorrocha.com.br/leansoftware-development/>>. Acesso em: 05 mai 2014.

SAVOINE, M.; et al. **Análise de Gerenciamento de Projeto de Software Utilizando Metodologia Ágil XP e Scrum: Um Estudo de Caso Prático**. In: XI Encontro de Estudantes de Informática do Tocantins, 2009.p. 93-102. Disponível em: <<http://tinyurl.com/yfsvnwz>>. Acesso em 26 Março 2016.

SBROCCO, J. H. T. de C.; MACEDO, P. C. **Metodologias ágeis: engenharia de software sob medida**. 1ª ed. São Paulo: Érica, 2012.

SCHWABER, K. **Agile Project Management with Scrum**. Redmond, Washington: Microsoft Press, 2004.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum: Um guia definitivo para o Scrum: as regras do jogo**. 2011, 18p. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 16 abr. 2016.

VARASCHIM, J. D. **Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet**. 2009. Curso de Ciência da Computação, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

SOUZA, D. R. **Implantação da metodologia ágil Scrum em um ambiente de desenvolvimento**. Dissertação de Mestrado. Universidade Federal de Santa Catarina, Curso de Tecnologias da Informação e Comunicação. 2014.

WIRTH, Niklaus. **Uma breve História da Engenharia de Software**. 2012. Disponível em: <http://marathoncode.blogspot.com.br/2012/07/uma-breve-historiadaengenharia-de_10.html>. Acesso em: 08 de abr. 2016.