SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

**FACULDADE DE TECNOLOGIA SENAI/SC FLORIANÓPOLIS**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ALEXANDRE THIESEN MATOS**

**AVALIAÇÃO DO PROCESSO DE DESENVOLVIMENTO DO PROJETO POLOPOLY COM BASE NA METODOLOGIA SCRUM**

FLORIANÓPOLIS

**2016**

ALEXANDRE THIESEN MATOS

**AVALIAÇÃO DO PROCESSO DE DESENVOLVIMENTO DO PROJETO POLOPOLY COM BASE NA METODOLOGIA SCRUM**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia SENAI Florianópolis como requisito parcial para obtenção do Grau de tecnólogo em Análise e Desenvolvimento de Sistemas sob a orientação do Professor Esp. Jayson Nienkotter de Melo.

FLORIANÓPOLIS

2015

ALEXANDRE THIESEN MATOS

**AVALIAÇÃO DO PROCESSO DE DESENVOLVIMENTO DO PROJETO POLOPOLY COM BASE NA METODOLOGIA SCRUM**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia SENAI Florianópolis em cumprimento a requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

APROVADA PELA **COMISSÃO EXAMINADORA**

EM FLORIANÓPOLIS, ... DE ... DE 2015

Prof. Nome Completo do(a) Coordenador(a), Título abreviado, (SENAI/SC) - Coordenador(a) do Curso

Prof. Nome Completo do(a) Coordenador(a), Título abreviado, (SENAI/SC) - Coordenador(a) de TCC

Prof. Jayson Nienkotter de Melo, Esp., (SENAI/SC) - Orientador(a)

Prof. Nome Completo do(a) Professor(a), Título abreviado, (SENAI/SC) – Examinador

Dedico este trabalho (escreva seu texto, caso queira dedicar o trabalho a uma pessoa, instituição. **Item opcional**.).

**AGRADECIMENTOS**

Agradeço a Deus pela vida!

Agradeço aos meus familiares que me apoiaram na realização deste trabalho.

Agradeço aos professores....

Faça alteração no texto de acordo com o que pretende agradecer!

(**Item opcional**)

“Toda a nossa ciência comparada com a realidade é primária e infantil, e, no entanto é a coisa mais preciosa que temos”

(ALBERT EINSTEIN)

(*Substituía esta frase por outra, caso queira manter um texto de epigrafe. Item opcional.)*

MATOS, Alexandre Thiesen. **Avaliação do Processo de Desenvolvimento do Projeto Polopoly com Base na Metodologia SCRUM.** Florianópolis, 2015. f. Trabalho de Conclusão de Curso (Graduação Tecnológica) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Faculdade de Tecnologia do SENAI, Florianópolis, 2015.

RESUMO

O resumo deve apresentar os pontos principais do documento como, o tema, objetivo principal, o método e conclusões. Deve ser composto por uma seqüência de frases concisas, afirmativas e não de uma enumeração de tópicos. Usa-se o verbo na voz ativa e na terceira pessoa. Deve apresentar no máximo 150 palavras. O espaçamento entrelinhas é simples. Deve ser apresentado em parágrafo único. Dúvidas consulte a NBR6028.

**Palavras-chave**: apresente no máximo três palavras chaves, que representam o assunto abordado no artigo. As palavras-chave são separadas entre si por ponto e vírgula e finalizadas por ponto.

MATOS, Alexandre Thiesen. **Avaliação do Processo de Desenvolvimento do Projeto Polopoly com Base na Metodologia SCRUM.** Florianópolis, 2015. f. Trabalho de Conclusão de Curso (Graduação Tecnológica) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Faculdade de Tecnologia do SENAI, Florianópolis, 2015.

abstract

Apresente aqui a tradução do resumo em língua inglesa.

**Key words**: apresente aqui a tradução das palavras-chaves para a língua inglesa.

LISTA DE FIGURAS

Insira aqui a lista de figuras. Caso o trabalho não incluir figuras, favor deletar esta página.

LISTA DE TABELAS

Insira aqui a lista de tabelas. Caso o trabalho não incluir tabelas, favor deletar esta página.

LISTA DE ABREVIATURAS E SIGLAS

Insira aqui a lista de abreviaturas e siglas. Caso o trabalho não incluir abreviaturas e siglas favor deletar esta página.

Lembre-se: no TCC as abreviaturas e siglas obedecem a seguinte regra. Na primeira ocorrência o termo deve ser escrito por extenso seguido da sigla ou abreviatura entre parênteses. Por exemplo: Serviço Nacional de Aprendizagem Industrial (SENAI). Nas demais ocorrências usar somente a sigla ou abreviatura. Na lista de abreviaturas e siglas deve-se seguir o seguinte padrão:

SENAI – Serviço Nacional de Aprendizagem Industrial

LISTA DE SÍMBOLOS

Insira aqui a lista de símbolos. Caso o trabalho não incluir símbolos, favor deletar esta página.

SUMÁRIO

[1 INTRODUÇÃO 15](#__RefHeading___Toc501_420151846)

[1.1 JUSTIFICATIVA 17](#__RefHeading___Toc503_420151846)

[1.2 OBJETIVOS 17](#__RefHeading___Toc505_420151846)

[1.2.2 OBJETIVO GERAL 17](#__RefHeading___Toc507_420151846)

[1.2.3 OBJETIVOS ESPECÍFICOS 17](#__RefHeading___Toc509_420151846)

[2 REVISÃO DE LITERATURA 19](#__RefHeading___Toc511_420151846)

[2.1 QUALIDADE 19](#__RefHeading___Toc513_420151846)

[2.2 PROCESSO 20](#__RefHeading___Toc515_420151846)

[2.3 MODELOS DE PROCESSOS DE SOFTWAREG 21](#__RefHeading___Toc517_420151846)

[2.4 METODOLOGIA ÁGIL 22](#__RefHeading___Toc519_420151846)

[2.5 SCRUM 23](#__RefHeading___Toc521_420151846)

[2.5.1 ORIGEM E CARÁCTERISTICAS 24](#__RefHeading___Toc523_420151846)

[2.5.2 PAPÉIS 25](#__RefHeading___Toc525_420151846)

[2.5.2.1 SCRUM MASTER 25](#__RefHeading___Toc527_420151846)

[2.5.2.2 PRODUCT OWNER 26](#__RefHeading___Toc529_420151846)

[2.5.3.3 EQUIPE 26](#__RefHeading___Toc531_420151846)

[2.5.4 EVENTOS 26](#__RefHeading___Toc533_420151846)

[2.5.4.1 SPRINTS 27](#__RefHeading___Toc535_420151846)

[2.5.4.2 SPRINT PLANNING 27](#__RefHeading___Toc537_420151846)

[2.5.4.3 DAILY 28](#__RefHeading___Toc539_420151846)

[2.5.4.4 REVIEW 28](#__RefHeading___Toc541_420151846)

[2.5.4.5 RETROSPECTIVE 28](#__RefHeading___Toc543_420151846)

[2.5.5 ARTEFATOS E FERRAMENTAS 29](#__RefHeading___Toc545_420151846)

[2.5.5.2 SPRINT BACKLOG 30](#__RefHeading___Toc547_420151846)

[2.5.5.3 BURNDOW CHART 30](#__RefHeading___Toc549_420151846)

[2.5.5.4 SCRUM BOARD 30](#__RefHeading___Toc551_420151846)

[2.5.5.5 PLANNING POKER 30](#__RefHeading___Toc553_420151846)

[2.6 CENÁRIO DO PROJETO POLOPOLY 31](#__RefHeading___Toc555_420151846)

[3 PROCEDIMENTOS METODOLÓGICOS 33](#__RefHeading___Toc557_420151846)

[4 CRONOGRAMA 34](#__RefHeading___Toc559_420151846)

[5 RESULTADOS E DISCUSSÃO 35](#__RefHeading___Toc561_420151846)

[5 CONCLUSÃO 36](#__RefHeading___Toc563_420151846)

[REFERÊNCIAS 37](#__RefHeading___Toc565_420151846)

[APÊNDICES 39](#__RefHeading___Toc567_420151846)

[Anexos 40](#__RefHeading___Toc569_420151846)

# 1 INTRODUÇÃO

Com o aumento da complexidade e dimensão dos softwares, em meados dos anos 70, eclodiu a chamada, crise do software, termo criado por Edsger Dijkstra, em sua apresentação na Association for Computing Machinery, intitulada “The Humble Programmer”, segundo Edger, “aqueles que quiserem softwares confiáveis, descobrirão que devem encontrar meios de evitar a maioria dos bugs” (DJKSTRA, 1972).

Evitar a maioria dos bugs, não significava que os programadores não deveriam errar, e sim que deveria existir uma forma de evitar que esses erros fossem descobertos só após a codificação do programa. Nasce assim a engenharia de software, em uma tentativa de contornar problemas como o citado anteriormente, e criar técnicas, métodos, ferramentas e processos de desenvolvimento de software.

Segundo Sommerville (2011), não existe um processo ideal de desenvolvimento de software, com isso, muitas empresas começaram a criar ou adaptar seus próprios métodos, havendo assim uma grande diversificação nesse sentido. E com a constante mudança do ambiente mundial em relação aos softwares, fazendo com que as empresas tenham que responder rapidamente a novas oportunidades e mercados, mudanças nas condições economicas e serviços concorrentes, rápido desenvolvimento e entregas se tornaram o principal requerimento no desenvolvimento de software, em vista deste novo cenário, nasceu o conceito de abordagens agéis.

Desenvolvimento ágil siginifica produzir um software eficiente rapidamente, e este não é desenvolvido por inteiro em uma única vez, e sim usa-se uma série de incrementos incluindo de cada vez uma ou mais novas funcionalidades do sistema (SOMMERVILLE, 2011).

Em 2001, nos Estados Unidos, 17 prestigiados profissionais da área de Engenharia de Software se encontraram com o objetivo de trocar suas experiencias de trabalho e melhorar o desempenho de seus projetos, nesta reunião chegou-se a um consenso dos princípios básicos para um projeto ser bem sucedido (SBROCCO, 2012).

Segundo essa analise e as caracteristicas do projeto POLOPOLY, que é uma aplicação CMS, usada para o gerenciamento de conteúdo online, onde as entregas são constantes, os requisitos são criados periodicamente pela equipe de análise do cliente e as prioridades mudam conforme a necessidade do mercado, destaca-se o SCRUM, que segundo Sommerville (2011), tem foco na gestão do desenvolvimento iterativo, e objetiva-se sempre na sua continúa melhoria dos ciclos de desenvolvimento.

Por mais que diversos casos comprovam os beneficios da implantação do SCRUM, cada empresa e cada equipe possuem necessidades e caracteristicas diferentes. Desta forma o seguinte trabalho fará uma analise do processo atual do projeto POLOPOLY, e apresentará um plano com as estapas de implantação, prevendo os seus benefícios e principais dificuldades segundo a metodologia.

## 1.1 JUSTIFICATIVA

O seguinte estudo será desenvolvido em uma empresa de desenvolvimento de software, onde situa-se uma equipe de desenvolvedores que tem como principal atuação a manutenção do projeto POLOPOLY para uma grande editora de jornais Holandesa. Neste contexto, devido a imprevisibilidade da area de atuação do cliente, o desenvolvimento ou manutenção das funcionalidades do sistema vem sendo realizado conforme conveniente à equipe, ou ao cliente no momento, de forma não padronizada, e questões como definições e distribuições de tarefas são feitas com base na experiencia dos membros da equipe.

Em busca da redução da dependência das pessoas chaves, e os beneficios que uma padronização do processo possa trazer para toda a equipe, e a receptividade que a equipe demonstrou, foi escolhida a metodologia ágil SCRUM.

## 1.2 OBJETIVOS

## 1.2.2 OBJETIVO GERAL

Desenvolver um plano de implantação da metodologia SCRUM para o projeto POLOPOLY.

## 1.2.3 OBJETIVOS ESPECÍFICOS

1. Acompanhar as etapas do processo de desenvolvimento do projeto Polopoly;
2. Descrever as etapas do processo de desenvolvimento do projeto Polopoly;
3. Estudar os processos, artefatos e ferramentas utilizados na metodologia SCRUM;
4. Criar uma plano de implantação, com as etapas de aplicação do SCRUM, prevendo os impactos positivos e negativos de cada processo, artefato e ferramenta a ser utilizado.

# 2 REVISÃO DE LITERATURA

## 2.1 QUALIDADE

Segundo David Garvin (1984) apud Pressman (2011) a qualidade é um conceito complexo e pode ser definidos em cinco visões diferentes, são elas:

1. Visão transcendental – qualidade não é algo de fácil definição, mas reconhecido facilmente;
2. Visão do usuário – esta visão baseia-se nas metas impostas pelo usuário, sendo assim se elas são atendidas, o produto possui qualidade;
3. Visão do fabricante – baseia-se nas especificações impostas a criação do produto, se o produto condiz com as mesmas, ele possui qualidade;
4. Visão do produto – sugere que a qualidade está relacionada com as características essenciais do produto;
5. Visão baseada em valor – diz respeito ao valor que o usuário estaria disposto a pagar pelo produto.

O conceito de qualidade de projeto é referente às especificações que são definidas pelos projetistas para a elaboração do produto. É importante lembrar que também contribuem para a qualidade do projeto certos fatores, como: tolerâncias, especificações de desempenho e a qualidade dos materiais. Quanto mais o produto atender a esses fatores, maior será a qualidade final do mesmo (PRESSMAN, 2011).

Para o desenvolvimento de um software a qualidade é um aspecto extremamente considerável, para busca-la é necessário conceitua-la. Pressman traz o seguinte conceito, “uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o produzem e para aqueles que o utilizam” (PRESSMAN, 2011, p.360). Esse conceito enfatiza três pontos cruciais, são eles:

1. Gestão de qualidade efetiva – para a construção de um software de alta qualidade é necessário uma infraestrutura que forneça subsidio para a elaboração de processos que auxiliam no seu desenvolvimento;
2. Produto útil – atende as expectativas do usuário final, fornecendo confiança e estando livre de erros;
3. Fornecer valor mensurável para seu usuário e criador – Um produto com a qualidade garantida para o fabricante gera menos esforço no seu suporte após desenvolvimento, deste modo os projetistas poderão trabalhar na criação de novos produtos, por exemplo. Já para o usuário, um produto de qualidade fornece confiabilidade e agilidade de acordo com a sua proposta inicial (PRESSMAN, 2011).

## 2.2 PROCESSO

“Processo é um conjunto de atividades, ações e tarefas realizadas na criação de algum produto de trabalho” (PRESSMAN, 2011, p.40). No contexto de software, o processo de software pode ser definido como um conjunto de atividades, métodos, práticas e transformações, utilizados para a criação ou manutenção de um produto de software, esses processos podem ser definidos conforme os objetivos e necessidades de cada empresa desenvolvedoras deste produto. Estes processos podem ser definidos ou informais, para os processos informais não existem documentações escritas, eles são repassados muitas vezes oralmente ou por imitação, já os processos definidos, existem documentações detalhadas com o que é feito, por quem, quando, o que precisa ser usado e o objetivo destes processos (PAULA FILHO, 2001).

Ainda segundo Paula Filho (2000), para a maturação e capacitação de uma empresa é necessária à existência deste tipo de processo, pois eles que definem padrões, facilitando assim a qualificação dos colaboradores e tornando a empresa mais independente. Vale ressaltar que somente definir e seguir processos não garante a qualidade, é preciso que o objetivo final dos mesmos seja entregar os produtos no menor prazo possível, com menor custo e garantindo a melhor qualidade.

## 2.3 MODELOS DE PROCESSOS DE SOFTWARE

Cada atividade, ação e tarefa de um processo são definidas por um modelo que os objetiva, e os relaciona entre si, e entre outros processos. Pressman (2011) estabelece um modelo genérico que é dividido em cinco atividades, são elas:

1. Comunicação – Consiste em realizar as necessidades reais dos interessados no desenvolvimento do produto, alinhando assim suas características e funções;
2. Planejamento – Determina como o desenvolvimento será feito, identificando previamente seus riscos, as necessidades de recursos humanos e materiais, assim como os objetivos e o cronograma de trabalho;
3. Modelagem – Para melhor compreensão do negócio, e identificação de pontos de melhoria ou necessidades, anteriormente ao inicio do desenvolvimento, é preciso desenhar esse software;
4. Construção – Atividade de desenvolvimento do produto;
5. Emprego – Implantação do produto e entrega final para o cliente.

É necessária a escolha de um fluxo de processo, que define a ordem de execução de cada atividade, são elas:

1. Linear – Executa cada uma das atividades em sequencia;
2. Iterativo – Pode repetir uma ou mais atividades antes de ir para a próxima;
3. Evolucionário – Executa as atividades em forma circular, acrescentando uma versão a cada volta completa;
4. Paralela – Executa as atividades sem a necessidade de ser finalizar uma para começar outra (PRESSMAN, 2011).

Para a melhoria dos seus processos de desenvolvimento, diversas empresas no mundo inteiro criaram modelos, ou desenvolveram modelos já existentes, criando assim modelos de capacitação, que são usados como referencia para a avaliação dos processos de uma organização, ele faz a comparação dos métodos que a empresa utiliza, com aquelas que o modelo descreve. O resultado dessa conferição é que determina o nível de maturidade do processo, recomendando assim, formas de melhorias para o mesmo. (PAULA FILHO, 2001).

## 2.4 METODOLOGIA ÁGIL

Criada em uma época em que com o desenvolvimento dos computadores, e o abrangente aumento do mercado de novos sistemas, que diferentes dos antigos, que eram desenvolvidos para carissimos e imensos computadores, e por essa razão, obrigados a seguir toda uma metodologia que exigia uma boa documentação para seus usuários, e especialmente para outros desenvolvedores, os novos sistemas não necessitavam que eles fossem especificados por inteiro, e sim evoluissem conforme necessidade dos clientes (BROD, 2013).

A partir desta conclusão, novos conceitos de metodologias de desenvolvimento foram criadas, conhecidas inicialmente por “métodos leves”. Em 2001, no estado de Utah, nos Estados Unidos, dezessete profissionais da área de software se encontraram com o objetivo de trocar ideias sobre os seus métodos de desenvolvimento, e discutir formas de melhorar o seus desempenhos, eles concluiram que mesmo utilizando praticas adpatadas as necessidades de suas próprias companhias, todas elas seguiam principios básicos para um projeto ser bem sucedido (SBROCCO, 2012).

Cria-se assim um documento conhecido por Manifesto Ágil, que tem como foco o conjunto de quatro valores (PHAM, A. e PHAM, P., 2011):

1. Priorizar indivíduos e interações em vez de processos e ferramentas;

2. Ter um software funcional é mais importante que uma documentação extensa;

3. Envolvimento e colaboração com o cliente em vez de somente negociação de contratos;

4. Sempre estár preparado a mudanças, sabendo que essas podem ocorrer durante todo o processo de desenvolvimento.

Destaca-se também doze princípios (PHAM, A. e PHAM, P., 2011):

1. Prioridade máxima em satisfação do cliente, com entregas antecipadas e contínua de software com valor;

2. Mudanças dos requisitos são sempre bem vindas, visto que elas trazem vantagens competitivas aos clientes;

3. Entregas em curtos periodos do produto funcionando;

4. Trabalho em conjunto de desenvolvedores e gestores;

5. Construção dos projetos com pessoas motivadas, confie nelas, e forneça suporte e ambientes necessários para a realização do trabalho;

6. O jeito mais eficiente de transmitir informações para, e entre uma equipe é através da conversa cara a cara;

7. A principal medida de progresso é o produto funcionando;

8. Processos Ágeis promovem o desenvolvimento sustentável, os clientes, desenvolvedores e usuários devem ser capazes de menter um ritmo constante indefinidamente;

9. Atenção contínua à excelencia técnica, e um bom projeto melhoram a agilidade;

10. Simplicidade, máximizar a quantidade de trabalho não necessário é essencial;

11. O melhores requisitos, arquiteturas e projetos, surgem de equipes auto-organizáveis;

12. Em periodos regulares de tempo, a equipe deve repensar sobre como se tornar mais eficiente, e então reajustar-se de acordo com as necessidades percebidas.

Com base nos itens citados, cada método ágil criado posteriormente ao manifesto, oferecem um conjunto de atividades que podem ser adotadas durante o desenvolvimento de software, sua principal ideia é a redinifinição constante das prioridades do projeto, que apesar de parecer informal, as metodologias ágeis não deixam de utilizar processos, ferramentas e documentos, porém, de uma forma reduzida e objetiva (SBROCCO, 2012).

## 2.5 SCRUM

Scrum é um framework de desenvolvimento iterativo e incremental, onde os processos são divididos em ciclos, sendo qual representando uma parte de um conjunto de atividades a serem executadas (SUTHERLAND, 2011).

## 2.5.1 ORIGEM E CARÁCTERISTICAS

O termo Scrum relacionado a desenvolvimento de software, surgiu de um artigo publicado por Hirotaka Takeuchi e Ikujiro Nonaka na Havard Business Reviews de 1986, intitulado “The new product development game”, nele os autores descrevem uma abordagem na qual projetos utilizam equipes pequenas e multidisplinares trabalhando com sucesso rumo ao um objetivo comum, no qual eles utilizaram a comparação a formação Scrum do rugby (PHAM, A. e PHAM, P., 2011).

Na decada de noventa, Jeff Sutherland e sua equipe, incorporando também estilos de gerenciamento observados por Takeuchi e Nonaka, criaram, documentaram e implementaram o SCRUM, que diferente dos métodos tradicionais, onde o foco sempre foi a criação de documentação e no cumprimento rigido dos processos, o Scrum concentra sua atenção no produto final e nas interações dos indivíduos (SBROCCO, 2012).

Os principais temas do Scrum são, inspeção e adaptação, visto que o desenvolvimento de um produto envolve aprendizagem e surpresas, sendo no maior conhecimento do negócio, ou na constante evolução do mercado, o Scrum infatiza a constante evolução, inspecionando praticas utilizadas, e o produto, e depois adptar os objetivos do produto e os processos de seu desenvolvimento (SUTHERLAND, 2011).

Segundo Sbrocco, não existe uma solução definitiva que irá solucionar todos os problemas, mas existem algumas considerações que podemos usar no Scrum, são elas:

1. Resultados flexiveis, visto que requisitos complexos podem sempre mudar de acordo com as necessidades dos clientes;

2. Flexibilidade dos prazos, deve-se se gerenciar e controlar o desenvolvimento;

3. Equipes pequenas, onde elas devem gerenciar a si mesmas;

4. Implementar o conceito iterativo e incremental no desenvolvimento de software, sendo o produto construido em partes(iteração), e ao final dessas iterações, uma nova parte do sistema é criada(incrementando sempre uma nova funcionalidade) até termos o produto completo;

5. Constantes revisões, identificando problemas e suas causas, e removendo os impendimentos;

6. Valorização do indivíduo.

## 2.5.2 PAPÉIS

O Scrum estabelece um conjunto papéis, cada qual com suas práticas e regras, que a equipe deve cumprir, esses papéis se dividem em três principais: o Scrum Master, Product Owner e a Equipe (SBROCCO, 2012).

## 2.5.2.1 SCRUM MASTER

O scrum master é responsável por planejar, instruir e coordenar a equipe no Scrum, ele deve garantir que as regras e ferramentas utilizadas no Scrum sempre estejam relacionadas com a constante melhoria dos processos. (BROD, 2013). Segundo Mike Cohn (2011), o trabalho do scrum master é semelhante ao de um personal trainer, mas ao invés de auxiliar o time na correta execução dos exercicios fisicos, ele os guiarão de forma correta nos processos, removendo obstaculos, garantindo que a equipe trabalhe em conjunto, sempre os direcionando ao objetivo final.

## 2.5.2.2 PRODUCT OWNER

O product owner é o responsável pelo produto, ele que define os objetivos a serem alcançados, enquanto o scrum master direciona a equipe a objetivos pré definidos, o product owner os cria (COHN, 2011).

Ele tem a visão clara e todo conhecimento sobre o produto, é função dele coletar os requisitos para o backlog do produto, e definir as prioridades do projeto (PHAM, A. e PHAM, P., 2011).

## 2.5.3.3 EQUIPE

A equipe é formada por membros com variadas especialidades de acordo com as necessidades do projeto, não havendo niveis de hieraquia, fazendo assim que uma equipe scrum seja autogerenciavel (BROD, 2013), ela tem o poder de se auto organizarem e gerenciarem seu trabalho de acordo com os desafios encontrados na execução do projeto (COHN, 2011).

## 2.5.4 EVENTOS

Uma caracteristica importante no Scrum, são seus eventos, sejam eles, o ciclo de desenvolvimento, conhecido por Sprints, ou suas reuniões, que fazem parte desse ciclo, são elas, Sprint Planning, Daily Meeting, Sprint Review e Sprint Restrospective.

## 2.5.4.1 SPRINTS

O Scrum é uma abordagem de desenvolvimento iterativa e incremental, seguindo essa definição as sprints são as iterações realizadas durante um projeto, com a função de incrementar funcionalidades em busca do objetivo final (COHN, 2011). As sprints começam com a reunião de planajamento, onde é definido o sprint backlog que a equipe irá trabalhar, em busca do objetivo final, elas podem durar até quatro semanas(PHAM, A. e PHAM, P., 2011).

## 2.5.4.2 SPRINT PLANNING

É a reunião inicial de preparação da Sprint, onde a equipe define os requisitos a serem desevolvidos, suas prioridades e objetivos, é importante a presença de toda a equipe, porque nem sempre a equipe de desenvolvimento consegue avaliar a importância de um requisito a ser entregue, ou o product owner entender dependencias entre módulos do sistema (BROD, 2013).

Segundo Jeff Sutherland (2011) a planning é dividida em duas partes, sendo que a primeira deve durar até quatro horas, onde o Product owner dá a sua visão dos requisitos a serem desenvolvidos, e a equipe deve realizar as questões referentes ao conteúdo, suas finalidades e significados para que a equipe entenda o objetivo da Sprint. A segunda parte da reunião tem como foco, o detalhamento da implementação das atividades e quais a equipe irá se comprometer a entregar ao final da sprint. É desta reunião que é criado o Sprint backlog.

## 2.5.4.3 DAILY

A daily é uma reunião em que permite a equipe gerencie seu trabalho e descubra impedimentos diariamente (PICHLER, 2011), ou seja, a daily deve ser realizada todos os dias, é aconselhavél que ela seja feita em pé e com duração de no máximo 15 minutos, nela todos os membros da equipe reportam o que estão trabalhando, seus objetivos e impedimentos na realização de suas tarefas até a próxima reunião, cabe ao scrum master intermediar a solução dos obstaculos falados pelos membros da equipe (SUTHERLAND, 2011).

## 2.5.4.4 REVIEW

Ao término da sprint, é realizada a reunião chamada review, onde o product owner juntamente com a equipe revisam a Sprint, ela tem como objetivo a demonstração dos itens realizados durante a iteração, mas não com a intenção de inspeção, mas sim de entender quais os objetivos foram alcançados e saber a situação do produto no contexto atual (SUTHERLAND, 2011).

## 2.5.4.5 RETROSPECTIVE

A retrospective permite que a equipe identifique problemas, suas causas e suas soluções do processo da sprint (PICHLER, 2011). Segundo Jeff Sutherland (2011), na retrospective, todos os membros da equipe defem participar, e responder duas questões, o que foi feito de bom durante a Sprint, e o que se pode melhorar na próxima, e é papel do Scrum master anotar, e juntamente com o time definir ações e a priorização para o constante melhoramento das iterações, diferente do sprint review que serve para melhoramento do produto, a retrospective vida a relação da equipe no processo Scrum.

## 2.5.5 ARTEFATOS E FERRAMENTAS

O scrum possui alguns artefatos e ferramentas que auxiliam a equipe no desenvolvimento do produto e na melhoria continua dos processos.

2.5.5.1 PRODUCT BACKLOG

É a lista de funcionalidades que compoem o produto, essa lista é feita pelo product owner, sendo ele o responsável pela priorização, adição ou exclusão dos seus itens (SUTHERLAND, 2011). O gerenciamento do product backlog é um processo continuo, segundo Roman Pichler (2011) ele pode ser entendido nas seguintes etapas:

- Conforme a necessidade, novos items podem ser adicionados, e os existente removidos ou alterados;

- É necessário a priorização da lista, sendo os itens mais importantes encontrados no topo;

- Os itens com alta prioridade devem estar preparados para a próxima sprint planning, ou seja, o product owner deve sua clara visão;

- A equipe deve sempre participar no dimensionamento do backlog.

## 2.5.5.2 SPRINT BACKLOG

É o resultado da sprint planning, uma lista de tarefas selecionadas do product backlog, em que a equipe se propos a trabalhar durante a realização da sprint, onde cada item deve estar priorizado e entendidos por todos (SUTHERLAND, 2011).

## 2.5.5.3 BURNDOW CHART

É utilizado para visualizar o progresso da equipe, correlacionando a quantidade de trabalho e o tempo em qualquer ponto no tempo da sprint (SUTHERLAND, 2011). Segundo Brod (2013), é a relação das horas estimadas com as efetivamente realizadas de uma atividade, servindo assim como um registro futuro para novas astimativas.

## 2.5.5.4 SCRUM BOARD

Um quadro utilizado para o acompanhamento cronologico das atividades durante as Sprints, ele é dividido por etapas de desenvolvimento das tarefas, tornando-se possível a observação rapida do andamento das atividades (SBROCCO, 2012).

## 2.5.5.5 PLANNING POKER

Utilizado durante o planejamento das Sprints, é uma técnica que serve para estimar o tamanho das atividades a serem realizadas, ela é aplicada em uma dinamica de grupo, onde se dá a impressão que estão jogando Poker. Nesta atividade cada tarefa é discutida e cada integrante dá sua nota de complexidade baseada em uma escala definida anteriormente, havendo consenso, é atribuida à tarefa, mas se houver divergencias, abre-se o espaço para discussões, a após uma nova rodada de pontuações (SBROCCO, 2012).

# 3 PROCEDIMENTOS METODOLÓGICOS

O presente estudo será realizado em uma fábrica de software lozalizada na cidade de Florianópolis, a empresa, em atividade no mercado há 10 anos presta serviços para diversas empresas que procuram terceirizar sua equipe de desenvolvimento, entre elas, uma grande empresa Holandesa, que atua no ramo mídias fisicas e digitais. Nesta empresa situa-se o projeto em questão.

O projeto POLOPOLY é incumbido pela manutenção do sistema de gerenciamento de conteúdo de mesmo nome, este responsável pelo gerenciamento de diversos sites regionais e nacionais da Holanda, criado por uma empresa Inglesa, o sistema polopoly foi totalmente baseado na tecnologia J2EE, e é uma plataforma independente e totalmente escalável.

O estudo em questão será abordado de maneira qualitativa, buscando os significados e intenções do que é pesquisado, de forma não ter como intuito a enumeração de resultados, diferente da pesquisa quantitativa (MINAYO, 2010), e de carater explorátorio que tem como foco, desenvolver, esclarecer e modificar conceitos e ideias, visando a formulação de problemas específicos (GIL, 2010), e fazendo uso de procedimento técnico bibliogŕafico que tem como objetivo uma tomada de contas, sobre o que foi publicado acerca de um determinado tópico (TAYLOR; PROCTER, 2001).

Como primeiro passo do estudo o pesquisador irá observar durante quatro semanas o ciclo de desenvolvimento do projeto POLOPOLY, o que corresponde no projeto em questão, a dois ciclos de desenvolvimento, do seu inicio com à definição das tarefas até a finalização das mesmas.

Serão identificados e descritos os papéis dos integrantes da equipe e suas funções, os processos de desenvolvimento do projeto e suas etapas, assim como os artefatos e ferramentas utilizadas. Nesta estapa também serão levantados os problemas enfretados pela equipe, fundamentando a necessidade da aplicação de um processo de dsenvolvimento.

É necessário um estudo em literatura existente a cerca dos conceitos relacionados as metodologias ágeis, com enfoque na métodologia de desenvolvimento ágil SCRUM, seus objetivos, processos, papéis, artefatos e ferramentas utilizadas, assim como casos de sucesso para o embasamento de sua escolha.

Como etapa final da pesquisa, busca-se a elaboração de um plano de implantação do SCRUM no projeto POLOPOLY, onde nelas são descritas estratégias que preveêm, mudanças de papéis e responsabilidades dos integrantes da equipe, inclusões ou alterações nos processos e utilização de novos artefatos e ferramentas, objetivando os motivos de tal aplicação, assim como suas possíveis dificuldades.

# 4 CRONOGRAMA

# 5 RESULTADOS E DISCUSSÃO

## 5.1 CENÁRIO DO PROCESSO DE DESENVOLVIMENTO DO PROJETO POLOPOLY

Como citado anteriormente, o sistema polopoly é usado para gerenciamento do conteúdo de diversos sites, seus usuários são definidos em, editores, responsáveis por todo o conteúdo digital, gerenciamento dos sites e artigos, e leitores, que são os consumidores do conteúdo criado.

No projeto em questão existem duas frentes, a equipe do cliente localizada na Holanda, formada pelos analistas de negócio, equipe de teste e pelo gerenciador de atividades, e a frente de desenvolvimento, formada somente por programadores contratados pela empresa fábrica de software citada. Devido ao fato dos editores de conteúdo e os leitores não fazerem parte do projeto, e apenas utilizam o sistema, ambos serão tratados como usuários.

Os analistas de negócio tem como função o levantamento, a priorização e o aceite das atividades, essas consistem em bugs e novas funcionalidades, ambas criadas por solicitação dos usuários. A equipe de teste é resposável por testar as tarefas desenvolvidas, e verificar se elas estão em conforme com o que foi descrito pelos analistas. O gerenciador de atividades é responsável por, juntamente com a equipe de analistas, criar uma lista das atividades, prioriza-las e depois apresenta-las para equipe de desenvolvimento. O gerenciador de tarefas também faz o papel de desenvolvedor.

|  |  |
| --- | --- |
| Analistas de negócio | * Criação das atividades por solicitação dos usuários * Decisão de quais atividades serão desenvolvidas de acordo com as necessidades dos usuários * Priorização das atividades * Aceitação das atividades |
| Equipe de testes | * Teste do que foi desenvolvido * Verificação do que foi pedido o resultado |
| Gerenciador de atividades | * Priorização das atividades com os Analistas * Repasse das atividades para equipe de desenvolvimento * Desenvolvimento |

As atividades tem quatro tipos de prioridades, são elas: Sem prioridade, geralmente atividades de melhoria de alguma funcão não primordial do sistema; Normal, requisitos de novas funções, alterações de layout dos sites ou bugs que não impedem a utilização do sistema; Urgente, bugs no layout dos sites, ou nas funções do primordiais do sistema; Criticas, bugs que ocasionam o não acesso ao sistema, ou aos sites, ou o desenvolvimento de algo relacionado a ações de marketing em dias especificos.

Na frente de desenvolvimento, existem duas equipes, a equipe de Front-End, formada por dois desenvolvedores e é responsável pelo layout do site e pela criação dos códigos HTML, CSS e JavaScript, ela não trabalha diratamente com o sistema, e sim com um conjunto de sites de exemplo, com conteúdos estáticos apenas para demonstração de como ficarão as atividades que serão desenvolvidas para os sites.

A outra equipe é a de Back-End, composta por três desenvolvedores, que tem como função o desenvolvimento das atividades, que envolve a manutenção do código JAVA, linguagem em que o sistema é feito e a aplicação do HTML, CSS e JavaScript desenvolvido pela equipe Front-End e implantação do que foi desenvolvido no servidor de testes para posterior verificação pelo cliente. O gerenciador de atividades realiza também as funções realizadas pela equipe de Back-End dependendo da quantidade de atividades solicitadas, ou se as atividades estão atrasadas.

|  |  |
| --- | --- |
| Equipe Front-End | * Criação do layout do site * Desenvolvimento do código HTML, CSS e JavaScript |
| Equipe Back-End | * Aplicação do código HTML,CSS e JavaScript * Manutenção do código JAVA * Implantação das atividades no servidor de teste |

O processo de desenvolvimento de sofware se inicia-se com uma reunião onde os analistas de negócio reúnem as atividades criadas, e juntamente com o gerenciador de atividades, é discutida uma lista do que vai ser desenvolvido nas próximas semanas, criando assim uma lista menor com atividades priorizadas.

Após esse levantamento o gerenciador de atividades e a equipe de desenvolvimento se reúnem com o objetivo de apresentação das tarefas, estas discutidas suas complexidades, dúvidas e objetivos. Então é definido o tempo que a equipe de desenvolvimento levará para cada atividade, e quais a equipe se compromete a entregar.

É nesta etapa que verifica-se a necessidade de desenvolvimento Front-End de alguma atividade, caso positiva, a equipe Front-End define a complexidade e o tempo que levará.

A equipe geralmente se compromete entregar uma quantidade de atividades referente as duas semanas de desenvolvimento, mas dependendo da quantidade das tarefas que os analistas definem como prioridade, a quantidade pode variar pra menos ou mais semanas, um dos motivos para o não cumprimento do tempo definido, são as constantes inserções de atividades críticas na lista. Outro motivo é o aparecimento de problemas não previstos na criação das atividades, como mudanças dos requisitos, ou dúvidas, quando os imprevistos surgem, os desenvolvedores e analistas entram em contato diretamente.

Todos os dias as 10 horas da manhã é realizada uma reunião onde cada membro da equipe relata o que fez no dia anterior, e o que pretende fazer. Participam da reunião os desenvolvedores Front-End, Back-End e conforme disponibilidade, o gerenciador de atividades, devido ao fato do gerenciador de atividades estar situado na Holanda, a reunião algumas vezes é realizada utilizando a ferramenta de videochamadas Google Hangouts.

É utilizado o sistema de repositório e controle de versão git, e o serviço GitHub para compartilhamento de código.

O gerenciamento das atividades, é feito com a ferramenta de monitoramento de tarefas e acompanhamento de projetos, JIRA. As atividades são separadas em três estados:

|  |  |
| --- | --- |
| A fazer | Atividades definidas que não foram iniciadas o desenvolvimento |
| Em desenvolvimento | Atividades em que algum desenvolvedor está trabalhando no momento |
| Finalizada e pronta para testes | Atividade finalizada e pronta para ser entregue a equipe de testes |

Ao término de todas as atividades, os desenvolvedores fazem a implantação no servidor de testes, e as atividades são repassadas para equipe de testes, onde serão testadas contra erros de funcionamento e erros de desenvolvimento. Conforme a conclusão dos testes, os analistas verificam as tarefas e às aceitam, concluindo-as, ou havendo problemas, elas voltam para a equipe de desenvolvimento. Em razão dos testes serem realizados após conclusão de toda a lista, as tarefas rejeitadas são incluidas no próximo ciclo de desenvolvimento, este muitas vezes já iniciado, ocasionando atrasos.

## 5.2 IMPLANTAÇÃO DO SCRUM

Serão usados algumas iterações pra adequação ao SCRUM. Cada iteração adicionando alguma prática.

Como primeiro passo para implantação do SCRUM no projeto POLOPOLY foi realizada uma análise em cima dos papéis no processo de desenvolvimento atual, correlacionando-os aos utilizados no Scrum. Após foi possível definir um planejamento da implantação no que diz respeito as atividades, artefatos e ferramentas. O resultado dessa análise é apresentado através do mapeamento com base nos critérios descritos na tabela:

|  |  |
| --- | --- |
| Identificador | Significado |
| T | Totalmente atendido |
| P | Parcialmente atendido |
| N | Não atende |

A tabela apresenta o resultado da analise realizada sobre o processo de desenvolvimento atual. Ao lado do processo, artefato e ferramenta está o identificador do resultado encontrado, de acordo com os critérios estabelecidos na tabela 1.

|  |  |
| --- | --- |
| Processo/Ferramenta/Artefato | RESULTADO |
| Scrum master | P, o gerenciador de atividades faz o papel de facilitador, e o gerenciamento das atividades, mas como ele também é desenvolvedor, quando ele está ocupado, o programador busca diretamente informações sobre tarefas, com os analistas do cliente |
| Product owner | T, os analistas fazem o papel de dono do produto, pois é eles que detem todo o conhecimento do que vai ser, e o que foi implementado, mas existe uma equipe de analistas, e todos com a responsabilidade de criar tickets e definir prioridades conforme a gerencia do cliente necessita |
| Equipe | T, a equipe é auto gerenciavel, e busca soluções para os problemas encontrados durante a execução da sprint, é também papel da equipe definir o tempo de execução das atividades |
| Sprint | P, O projeto já realiza uma abordagem de desenvolvimento iterativa e incremental, mas as iterações não duram sempre o mesmo tempo, uma iteração pode começar com o plano de durar duas semanas, mas conforme a urgencia de algumas atividades elas podem durar mais, e nem sempre existe um objetivo final, ou se existe o objetivo pode mudar. |
| Sprint planning | P, A equipe realiza uma reunião de apresentação das próximas atividades a serem realizadas, onde são solucionadas as duvidas, e estimadas, mas nem sempre uma atividade passa por essa reunião, conforme urgencia de algumas atividades, elas podem ir direto para a lista de desenvolvimento, e o programador precisa conversar com os analistas sobre elas. |
| Daily | P, a equipe se reune todos os dias, e apresenta o que está desenvolvendo, e se existe algum problema, mas muitas vezes a reunião toma um bom tempo, porque problemas de natureza tecnica são discutidos, e com a não participação do gerenciador de atividades, a equipe inteira discute o que se precisa fazer para solucionar impedimentos. |
| Review | N, A equipe não realiza nenhuma reunião nesse sentido, as atividades são testadas conforme descrição dos analistas de sistema, e a equipe não fica sabendo quando, ou quais atividades vão para produção. |
| Retrospective | N, a equipe não realiza nenhuma reunião com o objetivo de rever os processos de desenvolvimento, muitas vezes, decisões nas mudanças de processos são tomadas no meio do ciclo de desenvolvimento, ou são decididas pelos analistas. |
| Product backlog | P Existe a lista de atividades criadas, onde os analistas recebem as solicitações dos usuários do sistema, e criam as tarefas. |
| Sprint backlog | P Após a reunião dos analistas, e do gerenciador de atividades, é criada uma lista priorizada com as atividades que os desenvolvedores trabalharão durante as próximas semanas, essa lista nem sempre mantém a priorização de quando foi criada, e conforme aparecimento de novas atividades com mais urgencia, ela pode aumentar, ou diminuir. |
| Burndow chart | N Não existe nenhum registro gráfico sendo usado para a visualização do progresso as atividades. |
| Scrum board | T A equipe utiliza a ferramenta JIRA, onde é disponibilizada um quadro com as atividades em andamento. |
| Planning poker | P A equipe estima as atividades, mas elas são estimadas conforme o tempo que demorarão as serem finalizadas, e essa extimativa é realizada conforme relato do desenvolvedor com maior experiencia no módulo afetado. |

“TODO:” Como o SCRUM está descrito na bibliografica separado por, papeis, eventos e artefatos e ferramentas, pensei em identificar em capitulos separados o uso deles já no projeto e mostrando o porque ele está adequado já, ou o porque não, mas não sei se devo falar já no mesmo capitulo o que precisa ser feito para essa adequação. Em vez de usar essa tabela, talvez só usar a tabela pra mostrar rapidinho o que tá acontecendo.

5.2.1 – PAPÉIS

Conforme citado anteriormente, o Scrum possui três papéis: o Scrum Master, o Product Owner e a Equipe. O papel de Scrum Master equivale conforme analise do processo atual, ao gerenciador de atividades, que é o principal meio de comunicação entre os analistas e a equipe de desenvolvimento, e é a pessoa que informa a equipe sobre mudanças de escopo do projeto, ou mudanças sobre o processo de desenvolvimento e também coordenando a maioria das reuniões.

Ao assumir a função de Scrum master, o gerenciador adiciona a suas responsabilidades a de garantir o uso do Scrum e trabalhar em pról dele, não sendo mais o lider da equipe, e sim trabalhando para resolver os impedimentos conforme apareçam, garantindo a continuidade do processo corretamente. Devido ao fato do gerenciador também participar do desenvolvimento das atividades, deverá ser incubido a ele somente atividades sem prioridades, não importando a interrupção delas pelas responsabilidades do Scrum master.

Segundo a analise do processo atual, o papel de Product Owner é totalmente representado pelos analistas, pois é deles a responsabilidade da criação dos objetivos do projeto, assim como a manutenção da lista de todas as atividades e sua priorização. Não havendo a necessidade de mudança de suas atribuições.

O Scrum diz que uma equipe deve ser auto-gerenciavel e ter o poder de organizar seu trabalho conforme os desafios aparecem, e é essa a principal caracteristica identificada na equipe do projeto, a razão disto, é o fato de a equipe de desenvolvimento estar situada em outro país, e na maior parte de sua jornada de trabalho não ter contato imediato com o gerenciador de atividades ou os analistas. A equipe fará parte de todas as mudanças realizadas no projeto, como novos eventos e a utilização de novos artefatos ou ferramentas.

5.2.2 – ARTEFATOS E FERRAMENTAS

Segundo dito anteriormente, o Scrum trabalha com a criação de alguns artefatos, e o uso de ferramentas para o auxilio do processo de desenvolvimento. Um dos maiores problemas encontrados no processo de desenvolvimento atual, é a falta de padrão na criação das atividades, dependendo do analista que cria a atividade, elas tem mais ou menos descrição, Cohn(2011) fala que historias de usuários são as melhores maneiras para descrever um problema ou uma nova funcionalidade, elas são escritas pela perpesctiva de um usuário do sistema. O projeto irá utilizar o seguinte modelo: Como<Usuario>, desejo <Objetivo da tarefa> para que <Razão(Alguma nova funcionalidade, ou correção de um problema)>.

A lista de atividades atual é muito semelhante ao que o Scrum define como backlog do produto, pois são os analistas, agora product owners que a gerenciam, e será mantido o sistema de priorização, definindo as atividades em estados, que são: sem prioridade, normal, urgente e criticas, com seus significados mantidos conforme descrito no capitulo 5.1. A única alteração a ser feita é nas atividades consideradas criticas, elas também irão para o backlog com prioridade máxima, e só serão desenvolvidas no inicio da próxima sprint.

A lista de atividades do próximo ciclo, agora irá se chamar sprint backlog, e nela terão somente atividades vindas do product backlog, e para acompanhamento posterior e utilização do gráfico burdown, todas as atividades serão estimadas conforme sua complexidade na reunião de planejamento da sprint e não mais pelo tempo. A inserção ou exclusão de alguma atividade nela ou dela, só serão aceitas conforme acordo entre o product owner, o Scrum master e a equipe de desenvolvimento. O sprint backlog será criado conforme a soma das complexidade das atividades chegue a um valor especifico, este chega-se devido a experiencia com sprints anteriores.

Para o gerenciamento das atividades, a equipe já utiliza a ferramenta JIRA, que disponibiliza um plugin para desenvolvimento agil, chamado de JIRA Agile, que permite o gerenciamento de projetos ageis, do Product Backlog até o acompanhemento das atividades e seus estados de desenvolvimento. Devido a este fato, não será necessário a utilização do tradicional quadro na parede, e nem um treinamento especifico ou aquisição de novas ferramentas.

FIGURIA DO SPRINT BOARD DO JIRA

O JIRA também disponibiliza relatórios sobre o andamento das sprints, o gráfico burndown é um deles, antes não utilizado devido ao fato das tarefas criadas no JIRA não possuirem definição das estimativas. Com a inserção dessa prática será possível o acompanhamento da sprint por esse gráfico, que utiliza a relação da soma das complexidades das atividades, com o tempo definido para a realização da Sprint.

EXEMPLO DO BURNDOW CHAT

Para as estimativas das tarefas, atualmente a equipe utiliza a experiencia dos desenvolvedores nos módulos afetados pela atividade para definir a quantidade de horas que serão tomadas para a conclusão da atividade.

EXPLICAR MAIS SOBRE ISSO

# 5 CONCLUSÃO

# REFERÊNCIAS

BROD, Cesar. Scrum – Guia prático para projetos ágeis. Primeira edição. Santos: Novatec, 2013.

COHN, Mike. Desenvolvimento de software com Scrum – Aplicando métodos ágeis com sucesso. Porto Alegre: Bookman, 2011.

DIJKSTRA, Edsger Wybe. The humble programmer. 1972, 17p. Disponível em: http://www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF>. Acesso em: 24 jun 2015.

GIL, Antonio Carlos. Métodos e técnicas de pesquisa social. 6. ed. São Paulo: Atlas, 2010.

MINAYO, Maria Cecília de Souza; DESLANDES, Suely Ferreira; GOMES, Romeu. Pesquisa social: teoria, método e criatividade. 29. ed. Petrópolis, Rio de Janeiro: Vozes, 2010.

PAULA FILHO, Wilson. Engenharia de Software: Fundamentos, Métodos e Padrões. Segunda Edição. Rio de Janeiro: LTC p.184 – 198, 2001.

PHAM, Andrew; PHAM, Phuong-Van. Scrum em ação – Gerenciamento e desenvolvimento ágil de Projetos de Software. Primeira edição. Santos: Novatec, 2011.

PICHLER, Roman. Gestão de produtos com Scrum – Implementando métodos ágeis na criação e desenvolvimento de produtos. Rio de Janeiro: Elsevier, 2011.

PRESSMAN, Roger S. Engenharia de Software. Sétima Edição. São Paulo: 2006.

REZENDE, Denis Alcides. Engenharia de Software e Sistemas de Informação. Terceira Edição. Rio de Janeiro: Brasport, 2001.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar. Metodologias ágeis: engenharia de software sob medida. Primeira edição. São Paulo: Érica, 2012.

SOFTEX – ASSOCIACAO PARA PROMOCAO DA EXCELENCIA DO SOFTWARE BRASILEIRO. MPS.BR – Melhoria de Processo do Software Brasileiro: Guia Geral. Versão 1.2. 2009. Disponível em: <http://www.softex.br/mpsBr>. Acesso em: 24 jun. 2015.

SUTHERLAND, Jeff, The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework. Paris, 2011. Disponível em: <<http://www.hbagency.com/scrum/tl_files/scrum_inc/documents/ScrumPapers.pdf>>. Acesso em: 16 maio 2016.

# APÊNDICES

# Anexos