

Brunno dos Passos Alves

**Proposta de Processo Ágil de Desenvolvimento  
de *Software* em uma Instituição de Educação  
Pública**

São Paulo, SP - Brasil

Junho de 2016

Brunno dos Passos Alves

## **Proposta de Processo Ágil de Desenvolvimento de *Software* em uma Instituição de Educação Pública**

Monografia apresentada ao Programa de Pós-Graduação *Lato Sensu* do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – *campus* São Paulo, como requisito para obtenção do título de Especialista em Gestão da Tecnologia da Informação, Área Gestão do Desenvolvimento de Sistemas, Linha de Pesquisa Metodologia de Desenvolvimento de *Software*

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP  
Especialização em Gestão da Tecnologia da Informação

Orientador: Prof<sup>o</sup> Me. José Oscar Machado Alexandre

São Paulo, SP - Brasil

Junho de 2016

*Este trabalho é dedicado à minha mãe  
que me ensinou a aprender.*

# Agradecimentos

Para a realização deste trabalho agradeço:

- Aos meus pais, Elcio e Elenir, por me apoiarem e trilharem pelos caminhos corretos durante a vida.
- A minha esposa, Livia, pela paciência e amor a mim dedicados.
- A minha filha recém-nascida, Luna, fonte de alegrias e inspiração.
- Aos meus amigos pela compreensão e palavras de incentivo.
- Aos professores por guiarem e auxiliarem durante o curso.
- Ao Professor e orientador, José Oscar Machado Alexandre, pela confiança e por me auxiliar durante os trabalhos desenvolvidos.
- Também ao IFSP pelo ambiente acadêmico que viabilizou este trabalho.
- E mais uma vez à minha mãe que de alguma forma deve estar me acompanhando e ajudando.

*“Não há nada tão inútil quanto fazer eficientemente  
o que não deveria ser feito.”  
(Peter Drucker)*

# Resumo

A boa qualidade dos serviços públicos é um anseio da sociedade e foco de propostas de diversos estudos para sua melhoria. Com a Tecnologia da Informação (TI) assumindo um papel estratégico dentro das organizações, utilizar conceitos de Gestão de TI pode ser uma forma de alcançar esse objetivo. Atuar para tal, mais especificamente na Gestão do Desenvolvimento de *Software*, trazendo os benefícios desse novo paradigma, que vem fazendo com que cada vez mais as organizações privadas o adotem, é uma maneira de alcançar a satisfação do cliente, no caso de uma instituição pública, a sociedade. Porém alguns conceitos da agilidade vão de encontro às características das instituições públicas. Assim, visando integrar conceitos de Metodologias Ágeis e Qualidade de *Software* para atender as especificidades da administração pública brasileira, esse trabalho propõem um processo de desenvolvimento de ágil de *software* integrado a alguns padrões de qualidade e quatro formulários que propiciam uma burocracia mínima necessária a fim de melhorar a qualidade dos serviços públicos oferecidos a população, em especial uma instituição de educação pública devido à transversalidade de seus processos.

**Palavras-chaves:** desenvolvimento ágil. qualidade de *software*. setor público.

# Abstract

The good quality of public services is a yearning of society and focus proposals for various studies for improvement. With the Information Technology (IT) assuming a strategic role within organizations, use of IT Management concepts can be a way of achieving this goal. Acting for that, specifically in Development Management Software, bringing the benefits of this new paradigm, has increasingly private organizations to adopt it, it is a way to achieve customer's satisfaction in the case of a public institution, the society. But some agility concepts go against the characteristics of public institutions. Thus, to integrate concepts of Agile Methodologies and Software Quality to meet the specific characteristics of the Brazilian public administration, this paper proposes an agile development process software integrating some quality standards and four ways to provide the order to improve the quality of public services offered to the population, especially a public education institution due to the transversality of its processes.

**Key-words:** agile development. quality software. public sector.

# Lista de ilustrações

|   |    |
|---|----|
| Figura 1 – Exemplo de um <i>kanban</i> (KNIBERG; SKARIN, 2009). . . . .   | 18 |
| Figura 2 – Exemplo da estória “Cadastrar Cliente” com suas respectivas tarefas<br>(Elaborado pelo autor). . . . . | 19 |
| Figura 3 – Comparativo entre CMMI e MPS.BR. Adaptado pelo autor de (BAR-<br>BIERI, 2011). . . . .                 | 20 |
| Figura 4 – Processo de recepção de novas demandas para atualização do <i>software</i> . . . . .                   | 36 |
| Figura 5 – Processo da metodologia ágil de desenvolvimento para o setor público . . . . .                         | 38 |

# Lista de quadros

|  |    |
|--|----|
| Quadro 1 – Níveis de maturidade e seus processos. Adaptado pelo autor de (SOF-TEX, 2012). . . . .              | 21 |
| Quadro 2 – Comparativo entre organizações públicas e privadas. Adaptado pelo autor de (PARENTE, 2009). . . . . | 22 |
| Quadro 3 – Objetos de Fluxo BPMN. . . . .  | 26 |
| Quadro 4 – Objetos de Conexão BPMN. . . . .  | 27 |
| Quadro 5 – <i>Swimlanes</i> BPMN. . . . .  | 28 |
| Quadro 6 – Artefatos BPMN. . . . .   | 29 |
| Quadro 7 – Valores da pontuação para cálculo do Fator de Risco. . . . .  | 52 |
| Quadro 8 – Relação entre os formulários, nível de maturidade e processos. . . . .                              | 59 |

# Sumário

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>  | <b>11</b> |
| 1.1      | Motivações   | 12        |
| 1.2      | Objetivo   | 13        |
| 1.3      | Organização  | 14        |
| <b>2</b> | <b>Conceitos Fundamentais</b>  | <b>15</b> |
| 2.1      | Agilidade  | 15        |
| 2.1.1    | Scrum  | 16        |
| 2.2      | Qualidade  | 19        |
| 2.2.1    | MPS.BR   | 20        |
| 2.3      | Desenvolvimento de <i>Software</i> no Setor Público                              | 22        |
| 2.4      | Engenharia de Processos  | 25        |
| 2.4.1    | <i>Business Process Model and Notation</i>                                       | 25        |
| 2.4.1.1  | Objetos de Fluxo   | 25        |
| 2.4.1.2  | Objetos de Conexão   | 26        |
| 2.4.1.3  | <i>Swimlanes</i>   | 27        |
| 2.4.1.4  | Artefatos  | 28        |
| <b>3</b> | <b>Método de Pesquisa</b>  | <b>30</b> |
| 3.1      | Tipo de Pesquisa   | 30        |
| <b>4</b> | <b>Frentes de Pesquisa e Trabalhos Relacionados</b>                              | <b>32</b> |
| <b>5</b> | <b>Integração Agilidade e Qualidade</b>  | <b>35</b> |
| 5.1      | Modelo do Processo   | 36        |
| 5.1.1    | Processo de Demandas   | 36        |
| 5.1.2    | Processo da Metodologia Ágil de Desenvolvimento de Software para o Setor público | 37        |
| 5.1.3    | Papéis   | 39        |
| 5.1.4    | Fases  | 40        |
| 5.1.5    | Eventos  | 41        |
| 5.1.6    | Fluxos   | 42        |
| 5.1.7    | Atividades   | 42        |
| 5.1.8    | Artefatos  | 45        |
| 5.2      | Formulários  | 47        |
| 5.2.1    | Proposta para Formulário de Análise  | 48        |

|          |   |           |
|----------|---|-----------|
| 5.2.2    | Proposta para Formulário de Risco . . . . .                 | 51        |
| 5.2.3    | Proposta para Formulário de Configuração . . . . .          | 53        |
| 5.2.4    | Proposta para Formulário de Memória . . . . .               | 54        |
| <b>6</b> | <b>Conclusão . . . . .</b>                                  | <b>57</b> |
| 6.1      | Considerações Finais . . . . .                              | 58        |
| 6.2      | Trabalhos Futuros . . . . .                                 | 59        |
|          | <b>Referências . . . . .</b>                                | <b>62</b> |
|          | <b>Apêndices . . . . .</b>                                  | <b>66</b> |
|          | <b>APÊNDICE A Formulário de Análise . . . . .</b>           | <b>67</b> |
|          | <b>APÊNDICE B Formulário de Risco . . . . .</b>             | <b>68</b> |
|          | <b>APÊNDICE C Formulário de Configuração . . . . .</b>      | <b>69</b> |
|          | <b>APÊNDICE D Formulário de Memória . . . . .</b>           | <b>70</b> |
|          | <b>Anexos . . . . .</b>                                     | <b>71</b> |
|          | <b>ANEXO A Manifesto Ágil . . . . .</b>                     | <b>72</b> |
| A.1      | Manifesto para o Desenvolvimento Ágil de Software . . . . . | 72        |
| A.2      | Os Doze princípios do Software Ágil . . . . .               | 72        |

# 1 Introdução

Em (FERNANDES; ABREU, 2014) é exposta a importância que a área de Tecnologia de Informação (TI) cada vez mais tem alcançado nas organizações, passando a ser considerado como um setor estratégico. Não mais tem sido vista como um setor que presta serviços, tais como consertar e configurar computadores, ou ainda auxiliar na elaboração de planilhas para automatização de cálculos. Agora, a TI assume um papel fundamental no desenvolvimento e principalmente no crescimento das instituições, sendo suas atividades de grande importância, suas ações devendo estar consonantes com as metas organizacionais e seus gestores em contato direto com os altos níveis hierárquicos para seu total alinhamento com o negócio.

O Desenvolvimento de Sistemas de Informação é um dos campos que se destaca dentro da área de TI, evidência disso é essa ser uma grande área dentro da Ciência da Computação, e até mesmo no mercado de trabalho pode-se notar a separação de três tipos de profissionais de TI: gestores, analistas de sistemas e de infraestrutura e redes. A disciplina de Metodologias de Desenvolvimento de *Software* une dois perfis dos citados anteriormente, o gestor de TI e o analista de sistemas, o desenvolvedor de *software*.

A implantação de um processo de desenvolvimento de *software* em um setor de TI traz a padronização do fluxo de desenvolvimento de sistemas de informação. Essa padronização propicia o aumento da qualidade final do produto desenvolvido, proporciona um bom ambiente organizacional e também a melhoria na satisfação do cliente e do usuário final (ROCHA; MALDONADO; WEBER, 2001).

Metodologias Ágeis de Desenvolvimento de *Software* tem ganho notoriedade na última década. Elas priorizam aspectos por muitas vezes antes ignorados por outras metodologias. Esse novo paradigma promove resultados diferenciados e satisfatórios, motivos esses fazem com que as metodologias ágeis se destaquem (FERREIRA; LIMA, 2006).

Embora muito difundida em organizações privadas, a mudança de paradigmas dos métodos ágeis provoca controvérsias. Frequentemente são encontrados na literatura pareceres da impossibilidade ou dificuldades de implementá-la no setor público. Segundo (BALLARD, 2011) é impossível aplicar a agilidade na administração pública da Inglaterra, já em (BENITES; PAIVA; FERNANDES, 2011) e (KAMEI; QUEIROZ; FILHO, 2012) relatam as dificuldades de se implantar metodologias ágeis em órgãos públicos brasileiros. Com a leitura e análise desses trabalhos é possível identificar que algumas características dessa metodologia se tornam motivo para tal, como:

1. Não prezarem tanto por documentação, contrariando a característica burocrática

do setor público;

2. Os membros da equipe de desenvolvimento devem estar alinhados no que se refere ao conhecimento da tecnologia e da área de atuação do negócio, características essas difíceis de se obter por meio de concursos públicos;
3. Motivação e pró atividade da equipe, aspectos contrários ao estereótipo construído socialmente do servidor público;
4. Foco nos resultados, sejam eles produtos ou serviços, sendo que na administração pública o foco é no processo.

Analisando preliminarmente, essas características são uma barreira para a implantação da agilidade no setor público pois, devido a suas estruturas rígidas, acabam por validar tais argumentos.

Porém, se analisados com mais profundidade, parte desses argumentos podem ser derrubados empregando conceitos da Qualidade de *Software*. Essa área do conhecimento visa aumentar a qualidade do *software* através da definição e padronização de seu desenvolvimento. Assim, se durante o fluxo de desenvolvimento de *software* forem incorporadas fases como documentação de requisitos, análise de riscos, padrões na codificação, testes, formalização de início e fim das fases, dentre outras etapas, se terá qualidade no processo de *software* que refletirá diretamente na qualidade do produto final. Modelos de qualidade como CMMI, ISO e MPS.BR são reconhecidos e a certificação neles é buscada pelas empresas privadas a fim de se destacarem no mercado.

Com essa atenção à definição e normatização do fluxo de processo de desenvolvimento pode-se gerar artefatos que cubram as especificidades da Administração Pública, em especial a brasileira, aplicada a uma instituição de educação pública, tornando possível a implantação de uma metodologia ágil e trazer consigo seus benefícios ao órgão público e sociedade.

## 1.1 Motivações

O setor público, de uma maneira geral, é estereotipado socialmente com um viés de ineficiência e burocracia exacerbada. É grande o anseio da população por um serviço público de qualidade, que forneça serviços e produtos eficientes e a um baixo custo financeiro. Esse desejo da sociedade acontece nos mais diversos setores: educação, saúde, transporte, segurança, justiça, dentre outros.

Há muitos trabalhos que abordam diversos aspectos e ressaltam a importância de um serviço público de boa qualidade, propondo sua melhoria através de ações que envolvam a TI. Por exemplo, em (OLIVEIRA, 2013) há a proposta para a melhoria

da saúde pública através do gerenciamento dos prontuários, o trabalho de (ROCHA; OLIVEIRA; OLIVEIRA, 2013) se aplica à educação, atuando na melhoria do ensino público por meio da inclusão digital e em (LÜBECK; WITTMANN; GOMES, 2012) é abordada a gestão da informação para a melhoria do transporte público.

Com o setor de TI passando a ser visto como um departamento fundamental e estratégico para as organizações, sejam elas privadas ou públicas, entende-se então que uma forma de contribuir para a melhoria do serviço público seria aumentar a eficiência e qualidade de seu setor de TI atuando na gestão do desenvolvimento de sistemas de informação.

Já que pesquisadores buscam entender e construir modelos que melhorem a qualidade dos serviços públicos, e a TI ser estratégica nas organizações, atuar nessa intersecção de forma a melhorar a qualidade no processo de desenvolvimento de *software* em instituições públicas influencia de forma direta e indireta na qualidade do serviço fornecido para a população.

A aplicação de uma metodologia ágil no setor público para o desenvolvimento de *software* visa trazer os benefícios que esse novo paradigma proporciona e que vem incentivando diversas organizações privadas a adotarem essa ideia. E assim atender as características particulares da administração pública brasileira, integrando padrões reconhecidos para aumentar a qualidade do processo de desenvolvimento, para que a melhora reflita no serviço prestado para a sociedade.

## 1.2 Objetivo

Esse trabalho tem como objetivo a elaboração e modelagem de um processo que integre Metodologia Ágil de Desenvolvimento de *Software* com Qualidade de *Software*, de modo a atender as especificidades e outros instrumentos regulatórios da Administração Pública Brasileira, como legislação e normas, aplicados a uma instituição de educação pública. E assim melhorar o processo de desenvolvimento de *software* nas organizações públicas que adotarem esta metodologia e conseqüentemente os benefícios refletirem na comunidade atendida por esse instituto.

Embora existam trabalhos como (CATUNDA; NASCIMENTO; CERDEIRAL, 2011) e (TELES, 2004) abordando a junção da Agilidade com a Qualidade, não foi encontrado pelo autor algum que tratasse a questão da sua aplicação em uma instituição de ensino pública, quiçá a uma organização do setor público, e muito menos que propusesse uma maneira de o implantar. Com isso, este trabalho propõe duas frentes:

O primeiro desafio deste trabalho consiste em modelar um processo de desenvolvimento ágil de *software* levando em consideração um padrão de qualidade de *software*.

Tal modelagem permitirá visualizar o fluxo do processo e entender melhor como se dá a sequência de atividades necessárias no processo de desenvolvimento de sistemas de informação, compreendendo desde o momento inicial do projeto, passando pelo levantamento de requisitos, codificação, testes, entre outras etapas, até sua entrega e finalização.

Como segundo desafio tem-se a descrição detalhada dos fluxos e atividades do processo, bem como a definição e detalhamento de artefatos que atenderão as especificidades do setor público, ou seja, os documentos para registro da evolução do projeto. A partir desses detalhamentos será realizada uma comparação com a teoria e possíveis ajustes, para então ser realizada uma análise crítica do processo e avaliado sob seus aspectos fortes, fracos e pontos a melhorar.

### 1.3 Organização

Até então foi apresentada a parte introdutória a este trabalho, assim foram descritas as Motivações, Objetivo e Contribuições desse projeto respectivamente nas Seções 1.1 e 1.2.

Nas próximas páginas, primeiramente, serão contextualizadas as grandes áreas que fundamentam essa pesquisa. Assim sendo, no Capítulo 2 (Conceitos Fundamentais) serão expostas disciplinas como Agilidade (Seção 2.1) e Qualidade (Seção 2.2). Na Seção 2.3 são descritas características de como se dá o Desenvolvimento de *Software* no Setor Público, e por fim na Seção 2.4 (Engenharia de Processos) a técnica e notação utilizadas para a estruturação do modelo do processo.

Para o levantamento do estado da arte para a proposta dessa monografia foi seguida a metodologia de pesquisa exposta no Capítulo 3, Método de Pesquisa, sendo considerados alguns trabalhos como base para seu desenvolvimento, esses são descritos no Capítulo 4, Trabalhos Relacionados. No Capítulo 5 (Integração Agilidade e Qualidade) é exposto e detalhado o modelo do processo proposto para a metodologia ágil de desenvolvimento de *software* elaborado.

As Considerações Finais e Trabalhos Futuros são apresentadas no Capítulo 6 (Conclusões) nas Seções 6.1 e 6.2, respectivamente.

## 2 Conceitos Fundamentais

Segundo (BECK; BEEDLE; BENNEKUM, 2001), no desenvolvimento de sistemas de informação, a maior prioridade é satisfazer o cliente com entrega adiantada e contínua de *software*. Partindo dessa premissa alguns paradigmas foram quebrados no processo de desenvolvimento. Metodologias que antes primavam pela burocracia e eram extremamente formais e com etapas bem “engessadas” apresentavam desvantagens em projetos e equipes com características muito comuns em diversas organizações.

Já a qualidade é um fator importante em diversas áreas do conhecimento, como visto em (CARVALHO; PALADINI; BOUER, 2012), sendo aplicada em produtos, serviços, processos e indivíduos. Direcionar esforços em sua aplicação interfere diretamente com grau negativo ou positivo de excelência percebida. A qualidade também é aplicada no desenvolvimento de *software*, tanto no produto final quanto no serviço do processo de seu desenvolvimento, existindo padrões que a definem.

A seguir serão expostas de forma breve os conceitos teóricos das disciplinas fundamentais que esse trabalho utiliza. Assim sendo, essa pesquisa promove a junção Agilidade (Seção 2.1), Qualidade (Seção 2.2) aplicados no desenvolvimento de *software* em um órgão da Administração Pública (Seção 2.3). Na Seção 2.4 é descrita de forma sucinta a Engenharia de Processos, disciplina utilizada para a modelagem da metodologia proposta.

### 2.1 Agilidade

Metodologia de Desenvolvimento de *Software* é um campo integrante da Engenharia de *Software* e do Gerenciamento de Projetos. Essa define práticas, estruturadas em fluxos de atividades, de forma a se padronizar o processo para que possa ser seguido e repetido durante o desenvolvimento de *software* (TRECCANI; SOUZA, 2010).

Essas atividades podem compreender etapas como: análise, levantamento de requisitos, desenho da arquitetura do sistema, codificação, testes, implantação, suporte, dentre outras fases.

Ao longo dos anos, os processos de desenvolvimento de *software* têm evoluído de forma a definir regras, etapas, processos e documentos como forma de garantir a qualidade nos sistemas implementados. As metodologias mais conhecidas na engenharia de *software* são: *Waterfall* (conhecida como modelo Cascata), *Rational Unified Process* (RUP), Modelo em V, Incremental, Evolutivo, Prototipagem, Espiral (PRESSMAN, 2011) e Scrum. Esse último é o abordado neste projeto, uma metodologia ágil.

As metodologias ágeis são baseadas no Manifesto Ágil (BECK; BEEDLE; BEN-

NEKUM, 2001). Dentre seus doze princípios (vide Anexo A - Manifesto Ágil) podemos destacar: (i) a prioridade em satisfazer o cliente, (ii) as pessoas envolvidas trabalham em conjunto e estão motivadas, (iii) eficiência e eficácia ao transmitir informações, (iv) excelência técnica, (v) simplicidade e (vi) reflexão sobre como ser mais eficaz.

Esses são alguns dos princípios que empregados em conjunto com os demais trazem bons retornos à organização, e são essas vantagens que fazem com que cada vez mais um número maior de instituições adotem metodologias ágeis para o desenvolvimento de *software*.

### 2.1.1 Scrum

Existem diversas metodologias ágeis, cada uma com suas características (FAGUNDES; DETERS; SANTOS, 2008). Encontrado na literatura ora como metodologia, ora como *framework*, o Scrum vem sendo amplamente difundido para o processo de desenvolvimento de *software* (SCHWABER; BEEDLE, 2002). Este é totalmente aderente ao Manifesto Ágil prezando pela entrega contínua de algo funcional ao cliente sempre gerando ou agregando algum valor. A escolha do Scrum como metodologia ágil a ser adotada deve-se ao fato de sua ampla utilização, farta literatura e por possuir um certo formalismo por aplicar fases e atividades bem definidas, podendo incorporar características de outras metodologias conforme for conveniente.

O Scrum tem como base conceitos do processo de Manufatura Lean, utilizado nas linhas de montagens de automóveis, que prega a redução de desperdícios, tempo e custo e a melhoria da qualidade (RUBIN, 2012). Aplicando ferramentas, processos contínuos de análise, produção e elementos que reduzam as falhas, através de melhoria contínua, por exemplo. Em 1995, Ken Schwaber, adaptou esses conceitos para o desenvolvimento de *software*, formalizando o Scrum. Esse nome é devido uma formação de Rugby, onde o trabalho em equipe é essencial e há a constante mudança de estratégia durante uma partida para que se alcance a vitória (SALGADO; MELCOP; ACCHAR, 2010).

Em (CAELUM, 2011) observa-se que o Scrum, como um método ágil, possui etapas e papéis bem definidos. Também se utiliza de ferramentas e artefatos que auxiliam na execução de suas iterações e práticas, que devem ser seguidas por toda equipe de forma cooperativa e responsável. Por não definir de forma rígida um processo e possuir características mais flexíveis e dinâmicas, o Scrum se aproxima mais de um *framework* que de um método, assim podendo ser adaptado para outras áreas e não somente no desenvolvimento de *software*. Por exemplo em (MORAES, 2010), que trata da aplicação do Scrum para a rotina do dia a dia, como gerenciar a vida objetivando suas ações para o alcance de metas de projetos pessoais e profissionais.

No Scrum para desenvolvimento de *software* a equipe, chamada de time, que for-

mará o grupo para desenvolver o projeto deve conter de cinco a nove pessoas com papéis bem definidos. Esses são listados na sequência (CAELUM, 2011):

**Scrum Master** Um membro do time, pode ser um desenvolvedor, responsável por manter a equipe focada e também fazê-la seguir os princípios do Scrum, com o cumprimento dos horários e a objetividade nas reuniões. Esse também deve auxiliar nas dificuldades e impedimentos do dia a dia atuando para resolvê-los.

**Product Owner** Também conhecido como P.O., é o dono do produto. Esse deve estar ligado à área do negócio e assume o papel de representante do cliente. Deve definir os requisitos, estabelecer prioridade e sanar dúvidas pertinentes ao projeto. O P.O. participa de todo o processo do Scrum.

**Time** Equipe de desenvolvimento do projeto, composta por exemplo de desenvolvedores, testadores, levantadores de requisitos, administrador de dados, *designers*, dentre outros profissionais que atuem no desenvolvimento do projeto.

As etapas, também chamadas de *time-box*, são as reuniões que ocorrem com o time. Essas regras possuem tempo de duração máximo estabelecidos e definições claras de papéis e visam garantir a objetividade e foco dessas reuniões de maneira a minimizar o desperdício de tempo, maximizando a produtividade. Dentre outros benefícios, essas reuniões também promovem a integração e instigam a motivação do time, viabilizam o acompanhamento mais frequente do progresso do projeto e a construção e execução de estratégias de forma colaborativa com todos os envolvidos. Todas essas características beneficiam o gerenciamento e andamento do projeto, que reflete na qualidade do *software* e conseqüentemente na satisfação do cliente. Abaixo são apresentadas as etapas do Scrum (CAELUM, 2011):

**Sprint** É uma iteração do processo que segue um ciclo *Plan-Do-Check-Adjust* (PDCA). Esse deve ser longo o suficiente para desenvolver incrementos significantes para o cliente, e curto o suficiente para o cliente definir melhor os requisitos. Um Sprint tende a durar de duas semanas a um mês, e compreende o tempo para desenvolver os incrementos do produto (85%) e reuniões (15%) que serão descritas a seguir.

**Planning** Reunião que inicia um Sprint e ocupa 5% do seu tempo total. Nessa reunião é feito o planejamento, definindo os requisitos (estórias) e as tarefas com base nas necessidades, estimativa de esforço para executá-las e a meta que se deseja alcançar ao final do Sprint.

**Daily** Reunião que ocorre diariamente sempre na mesma hora e local, devendo durar no máximo 15 minutos. Nessa, todos os membros do time relatam o que fizeram desde o último Daily e o que vão fazer até o próximo.

**Game** Fase onde se dá a execução das tarefas definidas, ou seja, a implementação do incremento de *software* para o cumprimento da meta estabelecida. Nessa fase é onde ocorrem os Dailys.

**Review** Reunião onde é feita a entrega do incremento de *software* e ocupa 2,5% do tempo total do Sprint. É apresentada a nova versão do sistema produzida ao cliente e esse a aprova.

**Retrospectiva** Reunião que encerra o Sprint, ela visa a melhoria contínua do processo e da equipe. Dura 3,75% do tempo total do Sprint, onde se levantam os pontos negativos e positivos que ocorreram durante o Sprint, e se atua de forma a identificar ações a serem executadas para corrigi-los e melhorá-los para os próximos Sprints.

É comum encontrar no ambiente de equipes que seguem o Scrum um *kanban*. *Kanban* é uma palavra de origem japonesa que significa cartão ou sinalização. Consiste em uma ferramenta desenvolvida pela empresa automotiva japonesa Toyota, onde para melhorar a eficiência e eficácia em sua linha de produção, em um quadro dispunha cartões com descrições de tarefas para indicar o andamento do fluxo de trabalho, indicando em qual estágio do processo determinada tarefa está. Sua utilização permite o controle da produção com informações sobre quando, quanto e o que produzir. A representação de um *kanban* é exposta na Figura 1.

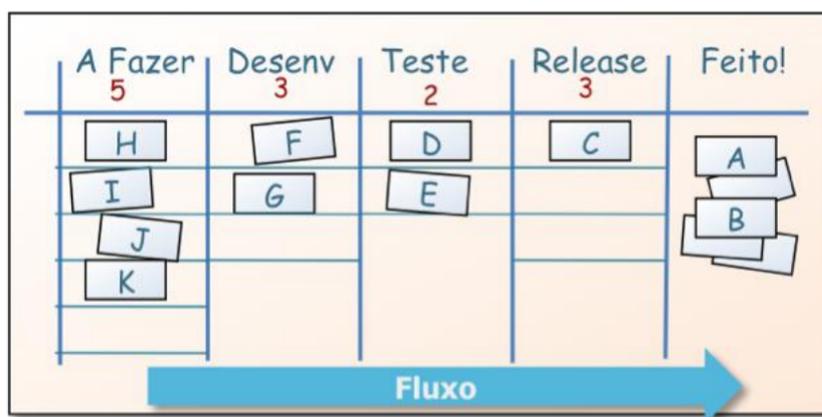


Figura 1: Exemplo de um *kanban* (KNIBERG; SKARIN, 2009).

Uma das ferramentas do Scrum é o *kanban*. Em um *kanban* do Scrum os estágios podem ser os mais diversos, e essa definição é feita conforme a necessidade identificada pelo Time. Podem ser, por exemplo, fases “aguardando execução”, “em desenvolvimento”, “em testes”, “finalizada”, dentre outras. A utilização do *kanban* permite visualizar o fluxo de trabalho, acompanhar o desempenho do Time, controlar o progresso da execução das tarefas, estimar prazos, identificar possíveis gargalos no processo e riscos para o cumprimento de metas, dentre outros benefícios.

A necessidade de *software* do requisitante é chamada de Estória, essa é dividida em partes, denominadas Tarefas. Essas tarefas são as atividades técnicas e operacionais para viabilizar as necessidades apontadas pelo P.O. e descrita na Estória. Um exemplo é apresentado na Figura 2 em um sistema de compras, a estória “Cadastrar Cliente” envolverá tarefas como “Criar tabela no banco de dados”, “Criar tela de cadastro” e “Validar o CPF”. Uma vez essas três tarefas concluídas, então o requisito da estória “Cadastrar Cliente” foi desenvolvido e cumprido.

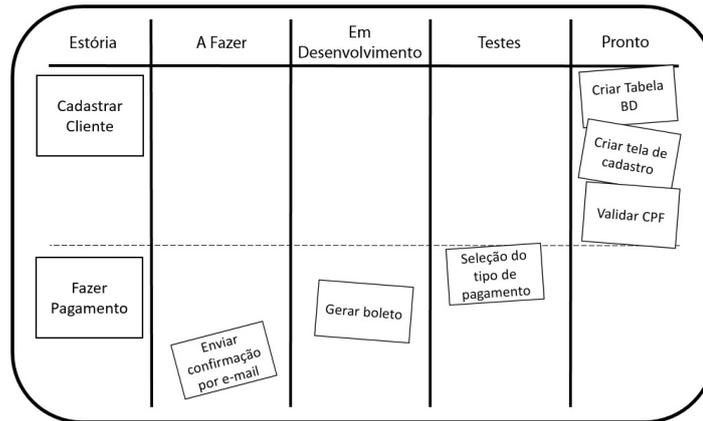


Figura 2: Exemplo da estória “Cadastrar Cliente” com suas respectivas tarefas (Elaborado pelo autor).

## 2.2 Qualidade

A Qualidade de *Software* também é um campo da Engenharia de *Software*. Seu foco é prover a qualidade do *software* através da normatização dos processos de desenvolvimento de sistemas de informação (ROCHA; MALDONADO; WEBER, 2001). Os padrões em geral são aplicados aos processos de desenvolvimento, mas o objetivo final é garantir o cumprimento dos requisitos e satisfazer às expectativas do cliente.

Dentre os padrões de qualidade pesquisado, os de maior recorrência são: o grupo da (i) ISO 9000 (ABNT, 2000), mais especificamente na qualidade de *software* a ISO 9126, que dispõe sobre a qualidade de produto de *software*, correspondente a norma brasileira NBR ISO/IEC 9126, ISO 15504 que trata do processo de desenvolvimento de *software* e ISO 12207 que foca no processo de ciclo de vida do *software*, (ii) CMMI - *Capability Maturity Model - Integration* (CMMI, 2010) - modelo de referência internacional que define a maturidade da qualidade de *software*, (iii) MPS.BR - Melhoria de Processos do *Software* Brasileiro (MARTINS; ANDRADE; GONÇALEZ, 2013) - é um modelo de qualidade adaptado para pequenas e médias empresas brasileiras baseado nas normas ISO/IEC 12207 e ISO/IEC 15504 e compatível com o CMMI e (iv) MoProSoft - *Modelo de Procesos para la Industria del Software* (GUARDATI; PONCE, 2010) - padrão mexicano baseado no CMMI e ISO/IEC 15504.

### 2.2.1 MPS.BR

A Melhoria de Processos do Software Brasileiro (MPS.BR) é uma iniciativa da Associação para Promoção da Excelência do Software Brasileiro (Softex) com o apoio do Ministério da Ciência e Tecnologia do Brasil, Financiadora de Estudos e Projetos (FINEP), Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e o Banco Interamericano de Desenvolvimento (BID) (SOFTEX, 2012). A escolha do MPS.BR como padrão de qualidade a ser adotado deve-se ao fato de ser uma iniciativa para a realidade brasileira no cenário de desenvolvimento de *software*.

Para o MPS.BR adaptou-se os conceitos e níveis do modelo de certificação internacional *Capability Maturity Model – Integration* (CMMI) para realidade das pequenas e médias indústrias brasileiras em um modelo mais enxuto. Dessa forma, o MPS.BR fornece certificações em níveis de maturidade, onde as empresas que se submetem obtêm a certificação mais rapidamente e também a um menor custo quando comparado ao CMMI.



Figura 3: Comparativo entre CMMI e MPS.BR. Adaptado pelo autor de (BARBIERI, 2011).

Na Figura 3 é possível observar a hierarquia dos níveis de maturidade do MPS.BR e um comparativo com os níveis do CMMI. Esses níveis são dependentes entre si, assim sendo para atingir um determinado nível de maturidade, a organização deve possuir todos os requisitos e exigências dos níveis anteriores àquele e realizar uma nova avaliação em seus processos. A certificação MPS.BR inicia-se com o nível de maturidade G (Parcialmente Gerenciado) e termina no nível A (Em Otimização). São sete os níveis maturidade (SOFTEX, 2012):

- A – Em Otimização** O último nível de maturidade, nesse a organização realiza a melhoria contínua, assim sendo seus processos estão em constante avaliação e atualização para seu melhor desempenho.
- B – Gerenciado Quantitativamente** Considera-se no nível B quando há avaliação e gerenciamento do desempenho de seus processos de forma quantitativa.

- C – Definido** Para o nível C de maturidade o gerenciamento do risco deve ser realizado e formalizado.
- D – Largamente Definido** Entende-se como Largamente Definido quando a atenção está voltada às atividades de validação e verificação.
- E – Parcialmente Definido** No nível E deve-se considerar processos como treinamento de seus profissionais e a melhoria e controle dos processos organizacionais.
- F – Gerenciado** Neste nível há a introdução ao controle de medições, a gerência de configuração, como versionamento, a gerência de aquisições e a garantia da qualidade.
- G – Parcialmente Gerenciado** Ponto de partida, primeiro nível de maturidade no MPS.BR. Neste deve-se gerenciar os projetos e seus requisitos, identificando se estão sendo seguidos padrões.

Para cada nível de Maturidade é necessário a implantação de Processos na organização, que por sua vez lista uma série de Atributos a serem atendidos. A seguir no Quadro 1 são apresentados as relações entre o nível de maturidade e seus respectivo processos:

Quadro 1: Níveis de maturidade e seus processos. Adaptado pelo autor de (SOFTEX, 2012).

| Nível de Maturidade                      | Processo   |
|--|--|
| <b>A</b><br>Em otimização                | Implantação de Inovações na Organização<br>Análise de Causas e Resolução   |
| <b>B</b><br>Gerenciado Quantitativamente | Desempenho do Processo Organizacional<br>Gerência Quantitativa do Projeto  |
| <b>C</b><br>Definido                     | Gerência de Riscos<br>Análise de Decisão e Resolução   |
| <b>D</b><br>Largamente Definido          | Desenvolvimento de Requisitos<br>Solução Técnica<br>Validação<br>Verificação<br>Integração do Produto  |
| <b>E</b><br>Parcialmente Definido        | Treinamento<br>Definição do Processo Organizacional<br>Avaliação e Melhoria do Processo Organizacional<br>Adaptação do Processo para Gerência de Projeto |
| <b>F</b><br>Gerenciado                   | Gerência de Configuração<br>Garantia da Qualidade<br>Medição<br>Aquisição  |
| <b>G</b><br>Parcialmente Gerenciado      | Gerência de Projeto<br>Gerência de Requisitos  |

## 2.3 Desenvolvimento de *Software* no Setor Público

A administração pública é um campo de estudo da Administração e ela deve organizar o Estado em todas suas instâncias. Esta é composta por órgãos, serviços, agentes do Estado, pessoas coletivas públicas a fim de garantir a satisfação das mais diversas necessidades coletivas, por exemplo setores como educação, saúde, segurança, justiça, mobilidade, cultura, dentre outros.

A separação do estudo da administração das organizações públicas e privadas se faz necessária, pois elas possuem características diferentes que influenciam no modo de como administrá-las (PARENTE, 2009). Algumas das características que diferenciam e influenciam diretamente no modo de como gerir esses tipos de organizações podem ser vistos no Quadro 2.

Quadro 2: Comparativo entre organizações públicas e privadas. Adaptado pelo autor de (PARENTE, 2009).

| Iniciativa Pública                | Iniciativa Privada                        |
|-----------------------------------|---|
| Maximizar custo-benefício         | Busca de lucro                            |
| Só realiza se estiver na lei      | Tudo é permitido, exceto o proibido       |
| Elevado grau de formalismo        | Grau de formalidade variável              |
| Propensão a inatividade           | Dinâmico, visa o alcance de metas         |
| Processos decisórios indefinidos  | Processos decisórios bem definidos        |
| Responsabilidades pouco definidas | Responsabilidades claramente distribuídas |
| Ênfase no processo                | Ênfase no resultado                       |
| Foco na ação punitiva             | Foco nas ações corretivas                 |

Em especial um instituto de ensino público objetiva construir práxis educativas que contribuam para a inserção social e a formação integradora. Sendo referência de excelência e atendendo às demandas da sociedade, de forma a acompanhar os processos de transformação do ensino e do trabalho com a perspectiva da diminuição das desigualdades sociais. Para tal, visa formar e qualificar profissionais nos diversos setores da economia. Em (CHAVES; TAKADA; MACIEIRA, 2014) mostra que dentre as particularidades de uma instituição de educação pública, está o fato de as diretrizes a serem seguidas estarem definidas pelo Ministério da Educação - MEC em consonância com o Plano Nacional de Educação - PNE, Lei nº 13.005 de 25 de Julho de 2014. Além das características de transversalidade de seus processos, onde as suas atividades são inter-relacionadas a fim do alcance das metas organizacionais, partindo da alta-gestão e integrando as atividades finais (educação) e de suporte (planejamento, gestão e transparência).

Um ponto recorrente e alvo de críticas da administração pública é a sua burocracia. Há a necessidade das organizações estabelecerem controles, regras, processos fim de diminuir perdas consequentes da desorganização. Porém a burocracia hoje possui uma

conotação pejorativa devido a sua extrema valorização e utilização, resultando em muitos papéis, assinaturas, carimbos, protocolos e outras formalidades, assim deturpando a intenção inicial que era de trazer mais eficiência à organização. A burocracia é importante, porém no atual cenário vale refletir e avaliarmos o que de fato é útil, seguindo a ideia da desburocratização (BELTRÃO, 1982).

Ainda que haja a burocracia, dentro da administração pública há regulamentações, normas e legislações específicas que regem o setor de TI e devem ser considerados em sua gestão. Como exposto em (ALVES, 2013) se comparado ao setor privado, na administração pública federal a aplicação da governança de TI é relativamente recente e emergente. Tais instrumentos regulatórios são aplicáveis a todos os órgãos dos três poderes do governo federal, não somente aos ligados à Educação.

Segundo (ALVES, 2013), o governo federal, juntamente com seus órgãos de controle cada dia mais publica decretos, leis e recomendações, bem como aplica auditorias para o bom andamento dos trabalhos da TI e para regulamentação da administração de seus recursos nos mais diferentes órgãos e esferas. Exemplo é o decreto federal no 7579/11 que dispõe sobre o Sistema de Administração dos Recursos de Tecnologia da Informação, visando estabelecer a governança de TI em órgãos federais. Outras iniciativas do governo federal são os mais diversos órgãos para fiscalizar a TI, como a Secretaria de Fiscalização de Tecnologia da Informação (Sefti), a Secretaria de Logística e Tecnologia da Informação (SLTI) e o Sistema de Administração de Recursos de Informação e Informática, antigo Sistema de Informática do Serviço Público (SISP).

Ainda em (ALVES, 2013), estudos realizados pelo Tribunal de Contas da União (TCU) e apresentados no Levantamento de Governança de TI ainda há muito a se fazer para que os pontos avaliados pela pesquisa apresentem resultados satisfatórios. Observa-se a existência de pouco planejamento da TI, pouco apoio da alta administração e baixa garantia da qualidade dos serviços prestados pela TI.

No que tange a área de *softwares*, em geral a Administração Pública orienta a adoção de sistemas livres e de código aberto. Conforme pode ser observado no acórdão do Tribunal de Contas da União (MONTEIRO, 2013) ainda é grande a quantidade de compra de sistemas informatizados em detrimento ao desenvolvimento próprio, sejam essas soluções prontas ou ainda a contratação de empresas que prestam serviços de desenvolvimento como fábricas de *software*, por exemplo. E para uma instituição de ensino leva-se em conta que os sistemas devem atender tanto as áreas fins, voltadas ao ensino, como por exemplo plataformas de ensino a distância. Também devem cobrir as áreas de suporte, como aplicações de gerenciamento de almoxarifado, por exemplo.

Porém a aquisição de soluções que não sejam de desenvolvimento próprio, ou ainda a adoção de sistemas *open-source* (de código livre e aberto), acabam acarretando na dependência do órgão que o implanta, isso com relação à manutenção e adaptação do

produto adquirido. E ainda assim, quando adquirido o serviço de uma empresa para o desenvolvimento da solução, é de baixa a aderência a utilização de metodologias ágeis, isso devido aos riscos que esse tipo de processo oferece (MONTEIRO, 2013):

**Riscos relativos a processos** contratação de desenvolvimento de *software* com adaptação de metodologia ágil que desvirtue sua essência; alteração da metodologia ágil adotada no instrumento convocatório no decorrer da execução contratual; ausência de definição dos artefatos ou alteração dos artefatos exigidos da contratada no instrumento convocatório durante a execução contratual; exigência de artefatos desnecessários ou que se tornam obsoletos rapidamente; utilização de contrato para desenvolvimento de *software* por metodologias tradicionais para desenvolvimento por métodos ágeis.

**Riscos relativos a pessoas** falta de comprometimento ou colaboração insatisfatória do responsável indicado pela área de negócios no desenvolvimento do *software*; falta do conhecimento necessário do indicado pela área de negócios para o desenvolvimento do *software*; excessiva dependência da visão do indicado pela área de negócios; equipe da empresa contratada não ter expertise em desenvolvimento de *software* com métodos ágeis; dificuldade de comunicação entre a equipe de desenvolvimento da contratada com o indicado pela área de negócios.

**Riscos relativos a produtos** alteração constante da lista de funcionalidades do produto; iniciação de novo ciclo sem que os produtos construídos na etapa anterior tenham sido validados; falta de planejamento adequado do *software* a ser construído; pagamento pelas mesmas funcionalidades do *software* mais de uma vez, em virtude de funcionalidades impossíveis de serem implementadas em um único ciclo, ou em virtude da alteração de funcionalidades ao longo do desenvolvimento do *software*; não disponibilização do *software* em ambiente de produção para utilização e avaliação dos reais usuários; forma de pagamento não baseada em resultados.

Assim sendo, devido à rigidez da estrutura e dos processos do setor público e sua burocracia exacerbada, e às características de uma instituição de ensino, pode-se enxergar uma dificuldade na implantação de metodologias ágeis. Porém, ao analisar os riscos listados anteriormente, apontados pelo TCU em “Levantamento sobre Aplicação de Metodologias Ágeis em Desenvolvimento de *Software*” no setor público (MONTEIRO, 2013) e analisá-los pontualmente com a intenção de saná-los, há maneiras de tratá-los. E na verdade o que é um empecilho pode ser interpretado como uma oportunidade ao se construir um processo que cubra tais riscos e garanta as especificidades do setor público, como será apresentado nesse trabalho.

## 2.4 Engenharia de Processos

Como visto em (CARVALHO, 2009), a engenharia de processos fornece conhecimentos e práticas capazes de observar, mapear, estabelecer, gerenciar e melhorar processos. Quando aplicada em organizações possuem o objetivo de gerenciar recursos, equipes, orçamentos, dentre outros componentes.

(JUNIOR; PINHEIRO, 2015b) expõe que sua finalidade, de maneira geral, é melhorar processos e métodos que são executados nas instituições. Para tal, otimizar o tempo da execução de atividades, analisar e propor melhor aproveitamento dos recursos (mão de obra, matéria prima, recursos financeiros, dentre outros), aumentar a qualidade, reduzir o desperdício, promover maior produtividade, inovar os processos, aumentar a margem de lucro, diminuir os custos, para que ofereça produto e/ou serviço de melhor qualidade, em menos tempo e a um menor custo.

Uma das ferramentas da Engenharia de Processos é o *Business Process Model and Notation* (BPMN) - Notação de Modelagem de Processos de Negócio. Essa é uma notação utilizada para a modelagem visual dos processos, composta por elementos gráficos padrões que facilitam a modelagem, entendimento, implantação e gerenciamento dos processos. Diversas ferramentas computacionais utilizam essa notação e oferecem suporte desde a simples modelagem até a construção colaborativa dos processos, navegação e validação dos fluxos, construção da arquitetura dos processos, automatização e construção de aplicações para o gerenciamento de indicadores, identificando gargalos ou sobras para que o processo seja otimizado (JUNIOR; PINHEIRO, 2015a).

### 2.4.1 *Business Process Model and Notation*

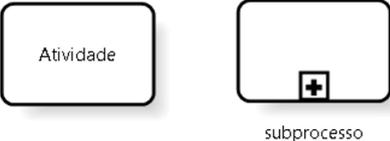
A notação BPMN é um padrão bastante adotado e ao utilizá-la há facilidade para implementar ferramentas conceituadas no mercado para gestão de processos, uma vez que essas utilizam o padrão BPMN para controlar os processos mapeados.

Segundo (JUNIOR; PINHEIRO, 2015b) a notação BPMN possui quatro categorias básicas de elementos para utilização nos mapas dos processos: Objetos de Fluxo, Objetos de Conexão, *Swimlanes* e Artefatos. Tais elementos são descritos nas próximas Seções:

#### 2.4.1.1 Objetos de Fluxo

São os principais elementos do BPMN, estes representam eventos que acionam ou finalizam a execução de atividades, atividades que representam tarefas e os *gateways* como controladores de fluxo. No Quadro 3 é possível observar a representação gráfica dos Objetos de Fluxo e na sequência esses são detalhados (WHITE, 2004).

Quadro 3: Objetos de Fluxo BPMN.

| Objetos de Fluxo | Representação Gráfica   |
|------------------|---|
| Evento           | <br>Início      Intermediário      Fim      Evento de Tempo |
| Atividade        | <br>Atividade      subprocesso                              |
| Gateway          |   |

**Evento** É algo que acontece durante um processo. Esses eventos afetam o fluxo do processo e têm geralmente uma causa ou um impacto. Há três categorias de eventos: Início, Intermediário e Fim . Eventos podem ser de vários tipos, envio e recebimento de mensagem, envio e recebimento de sinal, um evento condicional, por exemplo. Para esse trabalho será utilizado o evento de Tempo, que possui data e hora para ocorrer.

**Atividade** É um termo genérico para um trabalho executado. Pode representar tarefas de um processo ou sub processos dentro de um macroprocesso. O sub processo é distinguido por uma pequena cruz no centro inferior da representação da atividade.

**Gateway** É usado para controlar a divergência e a convergência da sequência de um fluxo. Assim, determinará decisões tradicionais, como juntar ou dividir trajetos do fluxo do processo.

#### 2.4.1.2 Objetos de Conexão

Os objetos de conexão, como o nome diz, conectam os elementos de um diagrama BPMN. No Quadro 4 são expostas as representações gráficas dos tipos de Objetos de Conexão e a seguir sua diferenciação conforme a utilização (WHITE, 2004).

Quadro 4: Objetos de Conexão BPMN.

| Objetos de Conexão | Representação Gráfica  |
|--------------------|--|
| Fluxo de Sequência |  |
| Fluxo de Mensagem  |  |
| Associação         |  |

**Fluxo de Sequência** É usado para mostrar a ordem (sequência) que as atividades serão executadas em um processo.

**Fluxo de Mensagem** É usado para mostrar o fluxo das mensagens entre dois participantes diferentes, que as emitem ou as recebem.

**Associação** É usada para associar dados, texto e outros artefatos com os objetos de fluxo. As associações são usadas para mostrar as entradas e saídas das atividades.

#### 2.4.1.3 *Swimlanes*

As *Swimlanes* funcionam como um mecanismo de organização das atividades em categorias visuais separadas. Sua organização se dá da seguinte forma (WHITE, 2004):

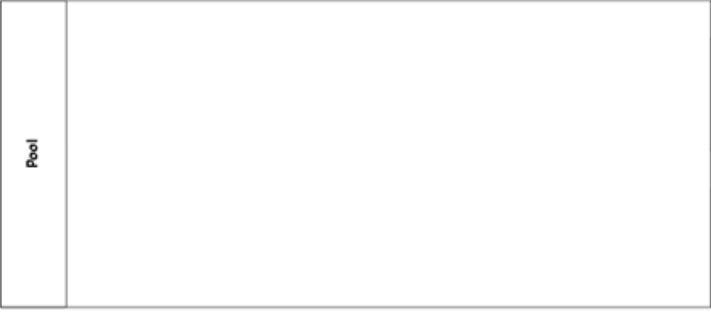
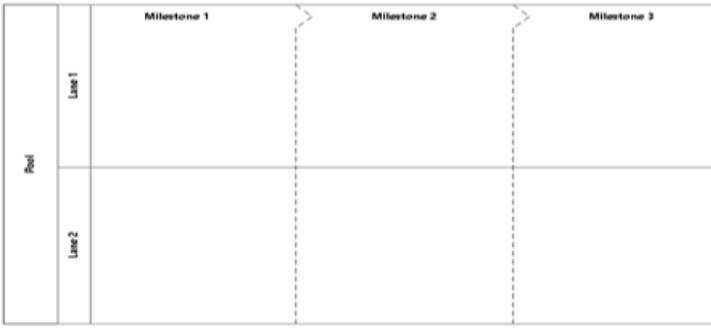
**Pool** Um *Pool* representa um processo. Atua como um separador gráfico para dividir um conjunto de atividades de outros *pools*. Uma *Pool* pode se relacionar com outra, isso é comum em organizações onde, em geral, os produtos de um processo são entradas de outros.

**Lane** Uma *Lane* é uma subdivisão dentro de um *Pool* usada para organizar e categorizar as atividades. São utilizadas para representar os papéis envolvidos no processo, identificando quem é o responsável por determinada atividade dentro de um processo, ou seja dentro de um *Pool*.

**Milestone** Um *Milestone* divide o processo em etapas, por exemplo uma mudança de fase. Auxilia na identificação do início e fim de determinados estágios que ocorrem durante o processo, também deixando mais claro quais são as atividades e envolvidos em cada uma dessas etapas.

A seguir, no Quadro 5, são exibidas sequencialmente a integração das representações gráficas do *Pool*, *Lane* e *Milestone*.

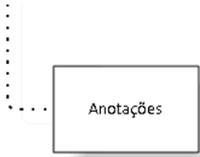
Quadro 5: *Swimlanes* BPMN.

| Swimlanes                       | Representação Gráfica  |
|---------------------------------|--|
| Pool                            |    |
| Pool com 2 lanes                |   |
| Pool com 2 lanes e 3 milestones |  |

#### 2.4.1.4 Artefatos

Os Artefatos são elementos que auxiliam no entendimento do processo. Podendo ser insumos ou produtos de atividades, agrupamentos ou observações. Os Artefatos estão representados no Quadro 6 (WHITE, 2004).

Quadro 6: Artefatos BPMN.

| Artefatos       | Representação Gráfica  |
|-----------------|--|
| Objeto de Dados |  |
| Grupo           |  |
| Anotações       |  |

Para facilitar o entendimento do diagrama deve-se utilizar cada tipo conforme o objetivo pretendido, sendo:

**Objeto de Dados** O objeto de dados é um mecanismo para mostrar como os dados são requeridos ou produzidos por atividades. Ou seja, esses são a entrada ou saída de uma atividade, assim sendo são conectados às atividades com as associações. Podem ser documentos como fichas e formulários, ou ainda bancos de dados.

**Grupo** Um Grupo é representado por um retângulo pontilhado e pode ser usado para finalidades de documentação, de organização ou de análise. Um grupo pode conter diversos elementos do processo.

**Anotações** As Anotações são elementos para fornecer informações adicionais ao leitor do diagrama BPMN. Observações que se julgue relevantes deve-se utilizar anotações para seu registro vinculadas ao elemento que se deve descrever a informação adicional.

## 3 Método de Pesquisa

De acordo com (GIL, 2002), uma pesquisa é um processo formal e sistemático de desenvolvimento do método científico, que objetiva descobrir respostas para problemas mediante o emprego de procedimentos científicos. Todo planejamento e execução de uma pesquisa tem como subsídio o uso do método, que em pesquisa significa a escolha de procedimentos sistemáticos para a descrição e explicação de fenômenos.

A seguir são apresentados os aspectos metodológicos do presente trabalho, onde são evidenciados o tipo de pesquisa e os métodos para coleta e análise de dados.

### 3.1 Tipo de Pesquisa

Em análise de (GIL, 2002) e (GERHARDT; SILVEIRA, 2009) o presente trabalho é uma pesquisa bibliográfica com justificativas. Foram procurados trabalhos que integrassem metodologias e processo de desenvolvimento ágil, qualidade de *software* e setor público. Dessa forma, seguindo a classificação:

A abordagem da presente pesquisa é Qualitativa, já que essa não se preocupa com representatividade numérica, e sim com o aprofundamento da compreensão de uma organização e cenário. Procurando explicar o porquê e apresentar o que convém ser feito sem evidenciar com valores quantitativos. Preocupando-se com aspectos da realidade difíceis de serem mensurados e quantificados. Em resumo, essa pretende descrever, compreender, explicar um cenário e situação e estabelecer as relações entre o global e o local. Por ter o intuito de gerar conhecimentos para aplicação prática, a fim de solucionar problemas específicos, a natureza dessa pesquisa é definida como Aplicada.

Quanto aos objetivos, essa é uma pesquisa Exploratória, isso pois tem como objetivo proporcionar maior familiaridade com o problema exposto, evidenciá-lo e propor uma solução. Para tal foi realizado o levantamento bibliográfico e análise de trabalhos que descrevem experiências práticas com o problema pesquisado e sua compreensão para proposta de solução.

Classificando quanto aos procedimentos é uma pesquisa Bibliográfica. Durante o desenvolvimento dessa pesquisa é realizado um levantamento e análise de referências teóricas como publicações de livros, artigos científicos, páginas de *web sites* dentre outras fontes para o conhecimento do estado da arte para o levantamento de informações e conhecimentos prévios sobre o problema exposto e então propor uma solução.

Conceitos e soluções apresentadas são baseadas na experiência do autor que possui pós graduação em gestão pública experiência de três anos no setor privado e seis no setor

público, desses atuando cinco anos como gestor de TI na área de sistemas de informação atuando principalmente na análise e desenvolvimento de *software*.

## 4 Frentes de Pesquisa e Trabalhos Relacionados

Neste capítulo tem-se a apresentação de trabalhos de pesquisa relacionados com os referenciais teóricos deste projeto, quais sejam: Métodos Ágeis e Scrum, Qualidade de Software e MPS.BR e desenvolvimento de *software* no setor público, mais especificamente em uma instituição de educação. Para tal feito foi levantada uma série de trabalhos destas áreas, procurando sempre que possível verificar o uso destes conceitos em conjunto.

Analisando a literatura, pôde-se observar que a grande maioria dos trabalhos que abordam métodos ágeis e qualidade de *software* são aplicadas a organizações da iniciativa privada, não especificando sua implantação no setor público, quiçá em um instituto de educação, embora possa ocorrer nesse segmento. Também foram procuradas publicações sobre Scrum, MPS.BR e setor público, sendo que não foi encontrado algum projeto que propusesse a junção dos três campos de forma detalhada. Sendo assim, tem-se o reforço que a proposta deste projeto é inovadora e inédita.

Um fator importante que justifica a abordagem destes três campos em conjunto (Scrum, MPS.BR e Organizações Públicas) é o benefício que a aplicação desses podem trazer para a comunidade. Por exemplo, um sistema implantado de forma a garantir a qualidade do produto, do processo de desenvolvimento e o alinhamento das necessidades da sociedade, contribuem para a melhor prestação de serviço à população e melhor ambiente organizacional da equipe, servidores públicos, que compõem a equipe do projeto.

Com a análise de publicações foi possível detectar algumas frentes de pesquisa que estão relacionadas e algumas são mutuamente dependentes, orientando e servindo de apoio para o desenvolvimento deste projeto.

Um trabalho relevante é o “SCRUMMPS 2.0 - Evolução de uma Ferramenta Iterativa para Suporte ao Scrum” (CARVALHO, 2013). Nesse é abordada a importância da gerência de projetos para que as atividades sejam executadas de acordo com planejamento. Ainda como exposto em (CARVALHO, 2013) metodologias ágeis tem como foco tornar os processos mais céleres, mas a qualidade não pode ser posta de lado. Isso pois é a qualidade de um produto que conquista clientes e aumenta a participação de uma organização no mercado.

O foco em (CARVALHO, 2013) é obter a qualidade através da melhoria do processo que é desenvolvido o *software*, no caso o MPS.BR. Para tal há a adoção de práticas documentadas em alguns processos dos níveis de maturidade F e G do MPS.BR, gerando assim a ferramenta computacional “ScrumMps 2.0” de modo a auxiliar a gerência de

projetos que seguem o Scrum. Nessa ferramenta é possível, por exemplo, acompanhar tarefas e atividades do projeto, gerenciar usuários e estórias de usuários, interagir com equipe de desenvolvimento e abordar conceitos de MPS.BR, tais como registro de reuniões, gerenciamento de risco, rastreamento de *Bugs* e portfólio de projetos.

Em (SZIMANSKI; ALBUQUERQUE; FURTADO, 2009) “Implementando Maturidade e Agilidade em uma Fábrica de Software Através de Scrum e MPS.BR nível G” é realizado um estudo de caso aplicado em uma pequena fábrica de *software* para introduzir conceitos ágeis, buscando equilíbrio entre agilidade e maturidade como alternativa para melhoria da qualidade dos produtos e, conseqüentemente, aumento da competitividade no mercado. Assim foi proposta uma extensão do Scrum para as áreas de processo do MPS.BR nível G a fim de manter a compatibilidade entre agilidade e qualidade.

Para (SZIMANSKI; ALBUQUERQUE; FURTADO, 2009) não há total compatibilidade entre o Scrum e o guia MPS.BR nível G. A integração dessas duas frentes foi uma alternativa capaz de proporcionar uma melhoria de de 42,50% (inovação, diminuição do tempo de entrega dos projetos, desenvolvimento de produtos de trabalho que agregam valor para o cliente e resultados para a empresa) se comparado ao método antigo. Um exemplo de uma área que proporcionou a melhoria foi a do gerenciamento de requisitos, que através de abordagens ágeis proporciona uma forma eficaz para a gerência desses, bem como solicitações de mudanças durante o desenvolvimento. Sendo assim, o processo proposto demonstrou um avanço inicial para o sucesso do projeto, ainda existindo a necessidade de monitoramento para a melhoria contínua e obtenção de melhores resultados.

(BENITES; CAGNIN; PAIVA, 2014) em “IAMPS: an process to support the MPS.BR implementation together with agile methods” - um processo para apoiar a implementação MPS.BR em conjunto com métodos ágeis, cita que independente de se adequar ou não na há uma avaliação formal para algum nível de maturidade é importante para a organização possuir processos com aderência ao MPS.BR, pois esses influenciam na qualidade e melhoria dos resultados a serem obtidos. Nesse trabalho é evidenciado que o MPS.BR não determina como os processos devem ser definidos para alcançar um certo grau de maturidade, dessa forma pode-se considerar que a metodologia ágil pode ser usada para permitir a implementação de processos “leves” para apoiar a implementação do MPS.BR.

Ao final (BENITES; CAGNIN; PAIVA, 2014) chega à conclusão que possuir e seguir um modelo de processo contribui com a melhoria da qualidade de processo, do *software*, da produtividade e a familiarização com os requisitos, facilitando a implantação, já que essa pode ser traumática quando a organização já possui padrões culturais estabelecidos e de difícil mudança. Tais características similares às das organizações públicas, em que ainda em um contexto muito geral seja estabelecido um conjunto de diretrizes que ofereçam suporte à condução de projetos de implantação de melhorias. Esse processo

genérico, quando aplicado em uma organização pública, evidenciou ser necessário outros estudos para conduzir e observar o comportamento do processo proposto em outros níveis do MPS.BR. Uma vez que não possui mecanismos de análise que permitam determinar quais atividades ou ferramentas devem ser adotadas ficando subjetivos e poucos específicos.

Aplicado ao setor público, (BALLARD, 2011) em “*Agile will fail GovIT*”, apresenta argumentos contrários à utilização de metodologias ágeis em suas instituições devido suas características serem contrárias aos princípios ágeis, e indicando que sua implantação não terá sucesso. Porém em (KAMEI; QUEIROZ; FILHO, 2012) “Scrum no Serviço Público: um Relato de Implantação nas Secretarias Estaduais da Fazenda e da Gestão Pública do Estado de Alagoas”, são citados os benefícios de sua utilização relacionados ao envolvimento do cliente e comprometimento da equipe perante os resultados, permitindo assim melhorias na qualidade de entrega dos projetos e viabilizando trabalhos mais colaborativos. Ressalta-se ainda que embora houvesse resistências dos membros de alguns projetos, foi possível o amadurecimento e a busca por melhorias no processo.

Em (BENITES; PAIVA; FERNANDES, 2011) com “Implantação de Resultados Esperados do Processo Gerência de Projetos com o Apoio do Scrum no Setor Público”, são levantados pontos positivos da utilização do Scrum, porém nas propostas de trabalhos futuros destaca-se a construção de uma base de conhecimento, métodos e ferramentas de apoio à implantação ágil e de modelos de maturidade no serviço público.

Necessidade essa também ressaltada no estudo de caso em que foi avaliado a aplicação do Scrum em uma instituição pública, em (ROCHA, 2015). Mostrou-se que esse trouxe uma nova perspectiva ao gerenciamento de desenvolvimento de *software*, apontando a necessidade de sua adaptação para um órgão público. O Scrum trouxe benefícios pois o projeto foi entregue aos seus usuários havendo aumento da produtividade da equipe, bem como melhoria na comunicação entre os envolvidos, também houve diminuição dos custos, prazos e riscos. Porém, alguns fatores como a burocracia, falta de motivação e outros empecilhos impactaram negativamente. Ao final, para trabalhos futuros sugere-se analisar a viabilidade de agregar as recomendações do MPS.BR nível G afim de contemplar o gerenciamento de requisitos e projetos.

E é nessa necessidade de especificação, aprofundamento e definição de atividades, ferramentas e documentos que esse trabalho se conduzirá no próximo Capítulo (5 “Integração Agilidade e Qualidade”).

## 5 Integração Agilidade e Qualidade

Em diversas organizações públicas existem atividades que não agregam valor para os resultados gerados, podendo até em certos casos atrapalhar o desempenho, consumindo tempo e recursos, enfim atuando como “gargalos”. Sendo que para eliminá-las, geralmente não há uma receptividade favorável, isso pois há baixa cobrança por resultados, falta de vontade política e acomodação de uma parcela dos servidores públicos, que acarreta na manutenção de atividades inúteis que reforçam a ineficiência dos serviços. Tais fatos desgastam a imagem do setor público e seus servidores perante a sociedade (JUNIOR; PINHEIRO, 2015a).

A análise e redesenho dos processos podem contribuir para eliminação dessas atividades que não apresentam vantagens à organização. Para tal deve-se identificar onde os esforços devem ser direcionados para melhor qualidade do trabalho, alinhando as suas tarefas às necessidades da comunidade, que é seu cliente.

E é buscando de forma contínua a satisfação do cliente que as organizações abandonam o modelo em que direcionam seus esforços para o ambiente interno, com pouca preocupação com aquele que irá receber o resultado de seus processos. Assim, com o foco no cliente, as melhorias e mudanças nos processos estão alinhadas ao aumento da satisfação desses.

Adiante são apresentadas as propostas de processo ágil de desenvolvimento de *software*, através do diagrama BPMN (Seção 5.1), e de formulários para sua documentação (Seção 5.2). Assim visando a obtenção de melhores resultados a fim de auxiliar as organizações públicas a alcançarem a satisfação esperada para seus clientes, unindo a Agilidade da metodologia Scrum com os conceitos da Qualidade do MPS.BR.

A escolha do Scrum se deve por essa ser uma metodologia ágil que possui um grau razoável de formalismo como pode ser visto em (FAGUNDES; DETERS; SANTOS, 2008), bem como ser amplamente difundida no mercado e com grande quantidade de literatura disponível, como artigos e livros que transcrevem a teoria, estudos de caso de sua aplicação, além de também fornecerem ferramentas e conceitos ideais para sua adaptação ao setor público.

Já a escolha do MPS.BR como padrão de qualidade a ser adotado é por esse ser uma iniciativa brasileira para a realidade do cenário de desenvolvimento de *software* nacional e de implantação mais simplificada, granularizada e de menor custo financeiro de investimento (KALINOWSKI; SANTOS; REINEHR, 2010), mas que ainda sim é aderente a outros padrões internacionais, como o CMMI.

## 5.1 Modelo do Processo

Será utilizada a notação BPMN para a modelagem do processo da proposta de integração entre agilidade e qualidade no desenvolvimento de *software*. A ferramenta utilizada para o mapeamento BPMN é o BizAgi Modeler (BIZAGI, 2015), ferramenta de modelagem gratuita, amplamente adotada em organizações e de simples utilização.

### 5.1.1 Processo de Demandas

Antes de propriamente apresentar o modelo do processo ágil, vale focar em como que as solicitações de alteração de *software* chegam para serem atendidas. Esse é o fluxo anterior ao desenvolvimento e é exposto na Figura 4, visando representar como as demandas de desenvolvimento são geradas.

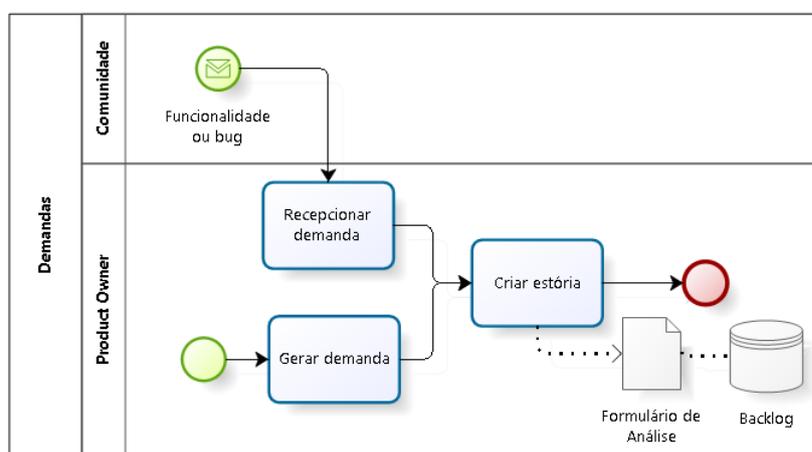


Figura 4: Processo de recepção de novas demandas para atualização do *software*.

Como pode-se observar as demandas chegam através da comunidade, em geral os usuários que utilizam o *software*, ou ainda pelo próprio *Product Owner*, como exposto na Seção 2.1.1, o dono do produto representante do negócio. Falhas encontradas no *software* que necessitam ser corrigidas, pequenas adaptações e ajustes para sua melhoria, ou ainda novas funcionalidades e módulos devem ser recepcionadas e/ou registradas pelo P.O. na forma de Estória, essas irão compor o *Backlog*. Assim sendo, acionado pela “Comunidade” através de uma “Funcionalidade ou Bug” cabe ao P.O. “Recepcionar demanda” ou ainda “Gerar demanda”, sendo que ele deve “Criar estória” através do “Formulário de Análise” que será armazenado no “Backlog”.

A Estória é um relato não técnico que descreve o que deve ser feito no *software*, como um requisito do sistema. Para a descrição da alteração do *software* é proposto um formulário para registro dessa solicitação intitulado Formulário de Análise, visando atender a disciplina de Gerência de Requisitos, uma das áreas associada à qualidade do processo de desenvolvimento de *software* e gerenciamento de projetos que será descrito na

Seção 5.1.8 e consta no Apêndice A - Formulário de Análise. Com esse documento em mãos o time de desenvolvimento o divide em tarefas técnicas para que o requisito seja atendido. Após registrada a Estória, o Formulário de Análise é cadastrado no *Backlog*, que são todas as alterações requisitadas para o *software* e servirá como entrada para o processo proposto de metodologia ágil de desenvolvimento de *software* que será apresentado a seguir.

Assim sendo, de maneira mais sucinta, o fluxo se dá da seguinte forma:

1. A comunidade pode sugerir uma nova funcionalidade ou reportar um *bug* no *software* ao P.O.;
2. Cabe ao P.O. recepcionar essa manifestação da comunidade, criar uma estória e registrá-la no formulário de análise;
3. O P.O. pode gerar uma demanda, ou seja, sugerir uma nova funcionalidade ou reportar um *bug*, também devendo criar uma estória e registrá-la no formulário de análise;
4. O formulário de análise irá compor o Backlog, que são todas as demandas do projeto à serem alteradas no *software*.

Na Figura 5 se encontra o modelo da proposta de processo ágil de desenvolvimento de *software* para o setor público agregando Agilidade e Qualidade e utilizando a notação BPMN.

De forma geral, o processo se inicia com a reunião de Planning, que gera as demandas a serem desenvolvidas, Formulário de Análise priorizados. Durante o dia a dia dos trabalhos são gerados documentos para registrar a execução das atividades. Ao final há uma última reunião para avaliação e melhoria do processo, e o registro dessas ações para aproveitamento dessas informações em outros projetos. Nas próximas seções serão detalhados cada elemento do processo.

### 5.1.2 Processo da Metodologia Ágil de Desenvolvimento de Software para o Setor público

Como dito, na Figura 5 consta o modelo do processo da proposta de metodologia ágil para o setor público. A metodologia é iterativa e incremental, seu *pool* representa um ciclo, denominado Sprint, assim sendo se inicia com uma reunião de planejamento, tendo com o entrada as tarefas do Backlog e as Lições Aprendidas. Durante suas atividades são produzidos artefatos para a documentação da execução do processo. Ao seu final, por ser cíclico e incremental, o processo recomeça novamente com um novo Sprint, sendo retroalimentado com as Lições Aprendidas dos Sprints anteriores.



As atividades e artefatos desse processo visam atender a burocracia intrínseca do setor público e seus requisitos de documentação para cobrir os riscos apontados na Seção 2.3, de forma a ser considerada uma burocracia necessária, aquela que otimiza os fluxos e trabalhos em busca da melhoria contínua. Ou seja, hoje a implantação desse processo de desenvolvimento de *software* em uma instituição pública promoveria a desburocratização (BELTRÃO, 1982) da parte do setor de TI responsável pela análise e desenvolvimento de sistemas de informação. E em especial, em uma instituição de ensino, devido a gama de áreas que os sistemas de informação devem atender devido a transversalidade de seus processos.

### 5.1.3 Papéis

Os papéis envolvidos no processo são representados nas *lanes*, raias, divisões horizontais da Figura 5. Para o modelo do processo são três papéis baseados na metodologia Scrum, conforme descritos a seguir:

#### **Product Owner**

O Product Owner ou P.O. é o dono do produto e representante da área do negócio. Por exemplo, um sistema para automatização de estoque o P.O. seria o Coordenador de Almoxarifado, um *software* para gerenciamento de acervo de biblioteca, um bibliotecário responsável. O P.O. prioriza as demandas, soluciona dúvidas quanto as regras de negócio e acompanha o dia a dia do desenvolvimento das soluções. Ele é a interface entre os usuários e o Time de desenvolvimento, descrito a seguir.

#### **Time**

A equipe de analistas, técnicos, programadores e outros profissionais que trabalharão na implantação da solução é denominada Time de desenvolvimento. Essa equipe em geral é da área de Tecnologia de Informação e ela que atenderá as demandas do projeto, por exemplo desenvolvedores, testadores, web-designers, dentre outros. Esses também devem sugerir soluções e comunicar qualquer intercorrência que ocorrer durante o desenvolvimento do projeto e que apresentem riscos à esse. Isso para que haja ciência e deliberações a fim de mitigá-los ou minimizar seus impactos. Faz parte do Time o Scrum Master, sendo suas atribuições apresentadas no próximo parágrafo .

#### **Scrum Master**

O Scrum Master é um membro do time conforme exposto anteriormente. Ele é o responsável pelo processo e deve garantir que todas suas atividades e artefatos sejam cumpridos. Também é ele que zela pelo andamento dos trabalhos, para tal deve fomentar debates entre os envolvidos e viabilizar meios para que as tarefas sejam executadas para o

desenvolvimento do *software* dentro do esperado pelo P.O. Cabe ao Scrum Master preencher os formulários que serão expostos na Seção 5.2 e ser o principal canal de comunicação entre o Time e o P.O.

#### 5.1.4 Fases

São quatro as fases do processo, representadas pelos *milestones*, colunas, divisões verticais da Figura mais detalhadamente a seguir:

##### **Planning**

O Planning é a primeira etapa do Sprint, destinada ao planejamento do que ocorrerá durante o Sprint atual. Nele, baseado nas Lições Aprendidas do Sprint anterior, ocorre a Reunião de Planejamento, que se seleciona um conjunto de melhorias no *software* a serem desenvolvidas pelo Time na próxima fase, o Game.

##### **Game**

A etapa onde ocorre a execução das tarefas, ou seja as atividades de desenvolvimento e programação é denominada Game. As atividades do Game são repetidas diariamente até que seja o último dia do Sprint. Nessa fase o conjunto de melhorias planejado na fase anterior é implementado pelo Time. É durante o Game que ocorrem as reuniões diárias para o acompanhamento dos andamentos dos trabalhos e que possibilitam a identificação e tratamento dos riscos que ameaçam a entrega da nova versão do *software* na Review.

##### **Review**

É na fase de Review que é entregue a nova versão do *software* com o incremento das funções desenvolvidas pelo time durante o Game e definidas durante o Planning. Nessa etapa a nova *release* do sistema é apresentada ao solicitante, o P.O., que deve aceitá-la ou rejeitá-la, cabendo ao Scrum Master documentar a entrega.

##### **Retrospectiva**

A última etapa do Sprint, a Retrospectiva, tem a função de realizar a melhoria contínua do processo. A equipe se reúne a fim de melhorar o processo, onde as experiências mais impactantes, positivas ou negativas, vivenciadas durante as fases anteriores de Planning, Game, Review e até durante a própria Retrospectiva, são levantadas, discutidas e registradas em uma base de conhecimento. Essa base, denominada Lições Aprendidas, possui a finalidade de retroalimentar a fase de Planning dos próximos Sprints e tam-

bém servir de referência em outros projetos para que não passem, ou ao menos tenham conhecimento, das experiências negativas, e fomentem as experiências positivas.

### 5.1.5 Eventos

São cinco os eventos no processo, todos estes tem data, hora e duração fixas, seguindo os conceitos das *time-boxes* do Scrum. A seguir seguem a descrição dos eventos que ocorrem durante o processo da metodologia proposta.

#### **Início do Sprint**

O Início do Sprint, como o próprio nome diz, inicia o Sprint, com o planejamento do que será feito. Esse evento é o inicial do processo, ocorre periodicamente conforme intervalo fixo acordado e definido por todos envolvidos no processo. Ele começa a Reunião de Planejamento que dispara todas as outras atividades do processo.

#### **Diariamente**

O evento intermediário Diariamente inicia as reuniões de Daily. Esse evento deve ocorrer incondicionalmente todo dia no mesmo horário para iniciar a reunião de alinhamento e acompanhamento do projeto.

#### **Data da Entrega**

O incremento de *software* é apresentado ao solicitante quando se inicia o evento intermediário Data da Entrega. Já pré-agendado, esse evento ocorre para que seja entregue a nova versão do sistema.

#### **Data da Retrospectiva**

O dia em que o Time se reúne para discutir o Sprint é iniciado pelo evento intermediário Data da Retrospectiva. Esse evento é iniciado no último dia do Sprint com data fixa e ocorre independente do sucesso ou não da entrega que ocorreu na Data de Entrega. É a Data da Retrospectiva que inicia o último dia e a última reunião do Sprint.

#### **Fim do Sprint**

O evento final Fim do Sprint marca o final do processo. Por ser um processo cíclico, iterativo e incremental, o novo Sprint é iniciado com um novo Planning com o evento inicial Início do Sprint, isso no próximo Sprint.

### 5.1.6 Fluxos

Representados pelo losango, o *gateway* do BPMN, controlam iterações, criam caminhos alternativos, paralelos ou unificam fluxos para continuação em uma mesma sequência de atividades dentro do modelo do processo. São quatro os *gateways* do processo:

#### **Game**

O Game é um *gateway* paralelo que inicia uma série de atividades a serem executadas ao mesmo tempo. É a partir dele que há o desenvolvimento da nova versão do *software* baseado nas necessidades definidas na Reunião de Planejamento e que se aguardam as datas das Reuniões de Daily, Entrega e Retrospectiva.

#### **Último dia do sprint?**

Para a verificação da condição para a continuação do Game, ou para que ocorra a última reunião, a Retrospectiva, há o *gateway* de decisão “Último dia do sprint?”. Assim sendo, ele que controla o fluxo de quando continuar o Game com seus trabalhos de desenvolvimento, ou quando o Sprint está chegando ao seu fim e é o dia de executar as atividades para encerrá-lo.

#### **Algum problema?**

O *gateway* de controle de fluxo “Algum problema?” é verificado diariamente após a reunião de Daily para caso seja identificado riscos, esses sejam tratados. Assim sendo, se não há problemas, o Time continua com seus trabalhos de desenvolvimento. Se há algum problema, após identificá-lo no Daily, esse deve ser discutido com a equipe e registrado para dar sequência aos trabalhos.

#### **Aguardando retrospectiva**

O *gateway* de junção “Aguardando retrospectiva” visa unir os fluxos divididos pelo *gateway* paralelo Game em um único fluxo. Esse novo fluxo é iniciado pelo evento intermediário Data da Retrospectiva, onde se iniciam as últimas atividades para o fim do Sprint.

### 5.1.7 Atividades

As atividades propostas no modelo do processo visam garantir a aderência tanto aos princípios da metodologia ágil Scrum, quanto à requisitos de qualidade do MPS.BR.

O tempo envolvido nas atividades do processo ágil proposto visam uma burocracia mínima, considerando o tempo investido na execução dessas atividades como um retorno

onde o custo-benefício seja alto em forma de qualidade no processo de desenvolvimento, refletindo diretamente para a melhoria de qualidade do *software*.

### **Reunião de Planejamento**

A atividade Reunião de Planejamento ocorre na etapa do Planning e é a primeira do Sprint, essa ocorre periodicamente com intervalo e hora fixos. Dessa reunião participam o Product Owner e o Time, incluindo o Scrum Master. Iniciada pelo evento Início do Sprint, ela tem como entradas o Backlog, conjunto de todas as Estórias, que são as alterações necessárias no *software*, e a base de conhecimento Lições Aprendidas de outros Sprints que já ocorreram.

Assim sendo, baseado em todas as necessidades do sistema apontadas pelo P.O. e experiências anteriores é feito o planejamento do Sprint, ou seja, é definido o subconjunto de Estórias que será atendido no Sprint em questão. Esse subconjunto é o Sprint Backlog, Estórias priorizadas de acordo com os critérios do P.O. e do Time considerando o esforço para cada necessidade e quais agregam mais valor ao P.O. Dessa forma se tem uma meta a atingir baseado no Sprint Backlog, que é a entrada das próximas atividades, Desenvolver e Gerenciar *Kanban*.

### **Gerenciar *Kanban***

Essa atividade pertence a etapa do Game e é executada repetidamente até que seja o último dia do Sprint. Gerenciar *Kanban* possui como entrada o Sprint Backlog, o conjunto de Estórias priorizadas a serem desenvolvidas. Inicialmente essas Estórias serão divididas em Tarefas técnicas, uma vez executada todas as Tarefas de uma Estória, então o requisito descrito por essa Estória está cumprido. Assim sendo, as Estórias e suas respectivas Tarefas devem ser dispostas no *Kanban* e terem sua fase atualizada diariamente conforme seu cumprimento e situações que se encontram.

### **Desenvolver**

Na atividade de Desenvolver o Time realiza a programação, testes, implantação, documentação, revisão de manuais, dentre outras mais atividades que envolvam as fases que constam no *kanban* para as Tarefas, e conseqüentemente o atendimento das necessidades descritas nas Estórias do Sprint Backlog. Ou seja, é nessa atividade do Game que se tem o trabalho de desenvolvimento para que se desenvolva a nova versão do *software*. O progresso e dificuldades encontradas durante essa atividade serão apontadas e discutidas na próxima fase, a Reunião de Daily.

## Reunião de Daily

A Reunião de Daily ocorre diariamente sempre na mesma hora e local devendo durar no máximo 15 minutos, por ser uma atividade do Game, essa se repete até o último dia do Sprint. A pauta dessa reunião é bem objetiva: os membros do Time relatam o que fizeram desde a última reunião de Daily e o que vão fazer até a próxima. A pauta curta é para que todos saibam como está o andamento dos trabalhos e não se perca o foco e objetividade, assim não consumindo muito tempo da equipe.

Esse acompanhamento diário possibilita que sejam identificados mais rapidamente riscos ao projeto, sendo esses discutidos e tratados. Caso os trabalhos estejam de acordo com o planejado deve-se diretamente atualizar o quadro na atividade Gerenciar o *Kanban*, caso haja algum problema a próxima atividade é de Registrar Risco, por fim se for o último dia do Sprint, então o Time deve aguardar a Reunião de Retrospectiva.

## Registrar Risco

Caso seja identificado durante a Reunião de Daily algum risco ao cumprimento da meta do Sprint, deve-se executar a atividade Registrar Risco. Essa atividade é de responsabilidade do Scrum Master e consiste no preenchimento do Formulário de Risco, que será descrito na Seção 5.1.8 e seu modelo consta no Apêndice B - Formulário de Risco. Com essa atividade se cobre a disciplina de Gerência de Risco, uma das áreas associadas à qualidade do processo de desenvolvimento de *software* e gerenciamento de projetos.

Dessa forma, o preenchimento do Formulário de Risco abrange a identificação do risco, a tarefa que lhe originou, medidas para preveni-lo e ações corretivas caso o risco se concretize tornando-se um incidente. Após essa atividade executa-se a atividade Gerenciar *Kanban*.

## Reunião de Entrega

Na Reunião de Entrega o Time apresenta a nova versão do *software* ao Product Owner, sendo que esse incremento de *software* deve atender a meta definida na fase do Planning. Essa atividade inicia a fase de Review com data e hora pré-definidas, e devem ser apresentadas todas as mudanças que foram realizadas a partir da versão anterior do sistema. Na sequência deve-se documentar como ocorreu a entrega, na atividade Registrar Entrega.

## Registrar Entrega

A atividade Registrar Entrega atende a disciplina de Gerência de Configuração, uma das áreas associadas à qualidade do processo de desenvolvimento de *software* e gerenciamento de projetos. Essa atividade é de responsabilidade do Scrum Master e consiste

no preenchimento do Formulário de Configuração, que será descrito na Seção 5.1.8 e seu modelo consta no Apêndice C - Formulário de Configuração.

Assim sendo, o preenchimento do Formulário de Configuração abrange na identificação do sistema, módulos e funcionalidades que sofreram algum tipo de alteração. Bem como a aceitação da solução, observações e a identificação de quem testou e homologou a nova versão.

### Reunião de Retrospectiva

A última etapa do Sprint, a Retrospectiva, é iniciada pela Reunião de Retrospectiva. Essa deve ocorrer no último dia do Sprint tendo como participantes o Time, incluindo o Scrum Master. O objetivo dessa reunião é realizar a melhoria contínua do processo e da equipe.

Tendo como entrada os Formulários de Risco produzidos durante a etapa de Game do Sprint, e o Formulário de Configuração, resultado da etapa de Review, o Time indica os pontos negativos e positivos que ocorreram durante o Sprint e atua-se de forma indicar ações a fim de corrigi-los e melhorá-los, respectivamente. Por exemplo, a descoberta de uma tecnologia é um ponto a se melhorar estudando-a mais a fundo, já uma entrega bem sucedida, mas com ressalvas, são pontos a se corrigir para resolvê-los ou evitá-los futuramente.

### Registrar Memória

A última atividade da Retrospectiva, e conseqüentemente do Sprint, consiste em Registrar a Memória. Essa atividade atende as disciplinas de Implantação de Inovações na Organização e de Análise de Causas e Resolução, áreas associadas à qualidade do processo de desenvolvimento de *software* e gerenciamento de projetos.

Nessa atividade cabe ao Scrum Master preencher as informações necessárias no Formulário de Memória (descrito na Seção 5.1.8 e seu modelo consta no Apêndice D), que indicam os pontos negativos, positivos e melhorias para tais. Ou seja, baseado nos apontamentos, sejam eles prós ou contras, deve-se indicar as ações e encaminhamentos a fim de que no próximo Sprint os pontos negativos sejam corrigidos ou minimizados, e os positivos sejam mantidos ou maximizados. Uma vez preenchido, o Formulário de Memória irá compor a base de conhecimento de Lições Aprendidas, que servirá de entrada para o próximo Sprint na Reunião de Planejamento.

## 5.1.8 Artefatos

Como artefatos são propostos elementos necessários para garantir a qualidade do processo e do *software*. A intenção desses é documentar requisitos, versões do *software*,

ameaças ao projeto e o registro de problemas e soluções para a melhoria contínua, tornando o processo de desenvolvimento cada vez mais eficiente.

A seguir são apresentados sete artefatos, sendo quatro formulários que serão mais detalhados na Seção 5.2 “Formulários”:

### **Formulário de Análise**

O Formulário de Análise é o registro de funcionalidades de *software*. Ele é um produto do processo Demandas, mais especificamente da atividade de “Criar estória” (Figura 4) e compõe o “Backlog” do projeto. Esse deve ser preenchido pelo P.O. e registra uma Estória do Scrum.

Por compor o “Backlog” do projeto esse se tornará entrada da atividade “Reunião de Planejamento” na fase de Planning e será parte do “Sprint Backlog”, produto dessa atividade e entrada da atividade de “Gerenciar *Kanban*”.

Na Seção 5.2.1 é exposta uma proposta para o formulário de análise, detalhando sua estrutura e descrevendo seus campos de preenchimento.

### **Backlog**

O Backlog é o conjunto de todas as Estórias a serem desenvolvidas no *software*. Ou seja, ajustes, adaptações e novas funcionalidades descritas nos Formulários de Análise são agrupadas, podendo estar em uma base de dados, por exemplo. Essas funcionalidades serão priorizadas e atendidas, formando assim o Sprint Backlog, descrito a seguir.

### **Sprint Backlog**

O Sprint Backlog é um conjunto das Estórias priorizadas pelo Product Owner em conjunto com o Time, e que esse se compromete a atender no Sprint em questão. O Sprint Backlog é um subconjunto do Backlog, assim sendo são Formulários de Análise que passaram por um processo de decisão para que fossem selecionados, ou seja as Estórias priorizadas. Por sua vez esses formulários serão transformados em tarefas técnicas de forma a atender as necessidades levantadas pelo P.O. e gerar uma nova versão do *software* que agregue valor para a comunidade.

### **Formulário de Risco**

O Formulário de Risco é um produto da atividade “Registrar risco” da fase de Game e entrada para a atividade “Reunião de retrospectiva” da fase de Retrospectiva. Esse deve ser preenchido após a reunião de Daily pelo Scrum Master conforme seus encaminhamentos, podendo gerar mais de um formulário de acordo com o número de riscos

identificados. Esses formulários serão apresentados e discutidos ao final do Sprint para a melhoria contínua do processo e do projeto.

Na Seção 5.2.2 é descrita uma proposta para o formulário de risco, detalhando sua estrutura, apresentando seus campos de preenchimento e um cálculo para a quantificação de impacto dos riscos para a tomada de decisões quanto ao tratamento desses.

### **Formulário de Configuração**

Como produto da atividade “Registrar entrega” da fase Review e entrada para a atividade “Reunião de retrospectiva” da fase da Retrospectiva, o Formulário de Configuração registra as funcionalidades contempladas pela nova versão do *software* desenvolvidas durante o Sprint. A responsabilidade de seu preenchimento é do Scrum Master.

Na Seção 5.2.3 é descrita uma proposta para o formulário de configuração, detalhando sua estrutura e justificando seus campos de preenchimento.

### **Formulário de Memória**

O Formulário de Memória é produto da atividade “Registrar Memória” da fase de Retrospectiva. Esse deve ser preenchido pelo Scrum Master registrando os principais pontos discutidos na reunião de Retrospectiva pelo Time.

Ele irá compor a base de Lições Aprendidas, descrita a seguir. Na Seção 5.2.4 é descrita uma proposta para o formulário de memória, detalhando sua estrutura e explicando seus campos de preenchimento.

### **Lições Aprendidas**

O artefato Lições Aprendidas é o agrupamento dos Formulários de Memória, podendo compor um banco de dados, por exemplo. Assim sendo, é uma Base de Conhecimento para que em futuros Sprints, podendo se estender a outros projetos, as dificuldades não se repitam, e caso ocorram já se tenham soluções e a informação do sucesso ou não de suas adoções, bem como para que os pontos positivos sejam repetidos ou tomadas ações para maximizá-los.

## **5.2 Formulários**

Conforme descrito anteriormente na Seção 5.1.8 “Artefatos”, os formulários propostos são artefatos do processo da metodologia ágil de desenvolvimento de *software* para o setor público, assim sendo são produtos desse processo.

Esses formulários tem por função documentar e garantir a qualidade do processo e do *software*, assim registrando requisitos, versões do *software*, ameaças ao projeto e

o registro de problemas e soluções para a melhoria contínua, tornando o processo de desenvolvimento cada vez mais eficiente.

Ainda possibilita rastrear uma mudança, identificando sua origem, como, porquê, quando e quem a motivou. Isso tanto no *software*, por exemplo uma mudança de requisito, como também na metodologia, ou seja, a mudança nas atividades rotineiras, ambiente de trabalho dentre outros fatores que envolvam e influenciem o processo de desenvolvimento.

Cada formulário tem por objetivo contemplar alguns processos de determinados níveis de maturidade do MPS.BR. A seguir esses formulários são detalhados vinculando suas características às disciplinas de qualidade atendidas.

### 5.2.1 Proposta para Formulário de Análise

O Formulário de Análise visa atender as disciplinas de Gerência do Projeto e Gerência de Requisitos do nível G de maturidade do MPS.BR, e objetiva de forma simples e sucinta descrever claramente o que o solicitante deseja para que a equipe de desenvolvimento implante no *software*. Dessa forma, ajustes, adaptações, novas funcionalidades, dentre outras solicitações de alteração da versão corrente do *software* devem ser descritas nesse formulário, sendo esse análogo à Estória do Scrum.

A primeira parte do Formulário de Análise é destinada à identificação, contendo os seguintes campos:

**Nome da Organização** no cabeçalho é destinado à identificação da organização, podendo conter o nome da instituição, seu logotipo, setor, dentre outros elementos que identifiquem à quem pertence o formulário.

**Sistema** é o nome do sistema, indica a qual *software* o projeto que a metodologia contempla está atendendo.

**Módulo** para caso seja um sistema constituído por diversos módulos, nesse campo deve ser indicado a qual módulo a necessidade do solicitante se aplica.

**Solicitante** deve ser preenchido com o nome de quem indicou a necessidade. Essa identificação é necessária para o rastreamento da configuração do *software*, possibilitando verificar quem e qual solicitação gerou determinada alteração no *software*.

**E-mail** consta o endereço do correio eletrônico do solicitante. Caso a equipe tenha alguma dúvida em relação a interpretação do formulário ou durante a implantação algum por menor não detalhado da necessidade, se consiga contatar o requisitante para esclarecimento.

**Telefone** com o número de telefone do solicitante. Assim como no campo E-mail, o Telefone é para que a equipe contate o requisitante caso haja alguma dúvida.

**Data** com o dia que o formulário foi preenchido. Essa informação é importante para critérios de ordenação e priorização. Como também pode ser útil se identificado que a Data do preenchimento do formulário é antiga talvez seja necessário o contato (através do E-mail ou Telefone) com o requisitante para a verificação se houve alteração nos requisitos da necessidade, ou ainda se a seu atendimento ainda é necessário. Já que devido à demora para o cumprimento da demanda o problema não mais exista ou tenha sido solucionado de uma outra forma, não mais tendo validade o formulário em questão.

Após as identificações há uma área destinada à listagem das necessidades demandadas que o sistema/módulo deve contemplar, denominada “Funcionalidades/Fluxos”. Para cada item (linha) dessa lista devem ser preenchidos os seguintes campos (colunas):

**Identificação Numérica** referencia a funcionalidade/fluxo listado anteriormente.

**Funcionalidade/Fluxo** indica o título da funcionalidade/fluxo que será detalhado os campos a seguir.

**Existe** indica se a funcionalidade é nova. Ou seja, não deve ser marcado esse campo caso o sistema ainda não contemple e deva ser implementado, e marcado caso o fluxo já exista e deva ser alterado. Esse campo auxilia na priorização de determinada demanda, por exemplo para uma funcionalidade que já existe provavelmente o esforço para atendê-la seja menor, pois é mais simples de ser realizada já que deve ser feito um ajuste ao invés de algo totalmente novo.

**Crítico** indica o impacto para a organização. Deve ser marcado se a Funcionalidade/Fluxo é crítica para o negócio, sendo considerada como crítica aquela de alto impacto para o órgão que utiliza o *software*, ou seja, caso não seja atendida impossibilite que entre uma nova versão do sistema. Assim como no campo Existe, também auxilia na priorização da demanda, por exemplo para uma demanda considerada crítica deve ser levado em conta que seu atendimento impacta no andamento das atividades da organização.

Assim sendo, pode-se concluir que demandas Funcionalidades/Fluxos que já existem no sistema e que são críticos para a organização possuem grandes chances de serem priorizadas. Na sequência da listagem, para cada item levantado anteriormente deve ser criado um bloco com informações que constam na Estória do Scrum descrevendo de forma sucinta as Funcionalidades/Fluxos, sendo esses:

**Identificação Numérica** visa identificar de maneira única a mudança requisitada, a funcionalidade/fluxo. Esse pode ser um número sequencial, por exemplo.

**Funcionalidade/Fluxo** deve ser preenchido com o título da mudança. Esse título deve ser coeso e descrever objetivamente qual o escopo da solicitação. Esse campo é o título de uma estória do Scrum.

**Como quem** descreve o papel de quem irá interagir com determinada Funcionalidade/Fluxo. Esse pode ser o cargo ou função de quem irá utilizar a funcionalidade descrita. Isso para que se saiba qual a visão que o solicitante tem em relação ao sistema. Por exemplo, em um fluxo intitulado Visualizar Requisições, um papel de Operador se interessa em ver as requisições que ele está envolvido, já um Gestor gostaria de ver todas as requisições feitas para sua área de incumbência.

**Como é** deve ser preenchido caso a Funcionalidade/Fluxo já exista no sistema. Esse campo descreve as características do que deve ser alterado, ou seja como o sistema funciona atualmente. Nesse pode-se:

- Colocar uma imagem da captura de tela do sistema que deve ser ajustado;
- Anexar o relatório a ser alterado;
- Descrever o comportamento atual do sistema;
- Indicar o caminho até acessar a referida funcionalidade;
- URL da tela que deseja alterar;
- Dentre outras características que auxiliem a identificar o que deve ser alterado.

**Porque mudar** deve ser preenchido com a motivação da alteração. Por exemplo:

- Citar e/ou anexar o instrumento regulatório que motiva a mudança;
- Justificar a razão da alteração;
- Enumerar os benefícios que a mudança traz;
- Citar os prejuízos que a não implantação causa;
- Dentre outras causas decorrentes da implantação da funcionalidade/fluxo.

**Alteração** descreve qual a expectativa do solicitante quanto ao comportamento que o sistema deve ter para que atenda sua necessidade. Esse campo é uma sugestão de solução, caso não seja possível atender à solicitação da forma sugerida, por uma inviabilidade técnica por exemplo, baseado em “Porque mudar” que motiva a alteração, o Time entrará em contato com o solicitante através do Telefone ou E-mail para propor outro método que atenda e solucione sua necessidade. Por exemplo, se sugerida uma caixa de seleção para o sistema filtrar dados de um relatório, porém devido a um impedimento como a grande quantidade de registros por exemplo, o Time sugere que esse seja dividido em diversos relatórios, cada qual listando as informações já separadas pelos mesmos critérios de filtro solicitados originalmente. Com isso, no preenchimento desse campo pode-se:

- Fazer observações baseadas na imagem capturada da tela;
- Esboçar a tela a ser adicionada ou alterada no sistema;
- Definir as informações de um determinado relatório;
- Detalhar o novo o comportamento do sistema;
- Listar definições de permissões de acesso;
- Mapear o fluxo a ser tratado;
- Outras características a ser adicionadas/alteradas para atender à necessidade.

E na sequência, para cada item listado na área “Funcionalidades/Fluxos”, deve ser preenchido com os campos listados anteriormente. Dessa forma a quantidade de Funcionalidades/Fluxos deve ser igual ao número de blocos dos campos anteriores. No Apêndice A - Formulário de Análise se encontra o modelo do Formulário de Análise bem como um exemplo de seu preenchimento.

O conjunto desses formulários compõem o Backlog, que são as demandas a serem priorizadas e que formarão o Sprint Backlog que por sua vez serão atendidas gerando uma nova versão do *software*.

### 5.2.2 Proposta para Formulário de Risco

O Formulário de Risco visa atender a disciplina de Gerência de Risco do nível C de maturidade do MPS.BR, e é preenchido após a reunião de Daily pelo Scrum Master, podendo gerar mais de um formulário de acordo com o número de riscos identificados.

Assim sendo, primeiramente o Scrum Master deve identificar o formulário preenchendo os seguintes campos:

**Nome da Organização** no cabeçalho é destinado a identificação da organização, podendo conter o nome da instituição, seu logotipo, setor, dentre outros elementos que identifiquem à quem pertence o formulário.

**Sprint** indica à qual Sprint se refere o risco. Preenchido com o número da iteração do Sprint do projeto.

**Data** com o dia da reunião do Daily que o risco foi identificado.

**Título** deve ser preenchido com o nome do risco identificado. Esse nome deve ser coeso e objetivo.

**Fator** de risco é calculado para que seja priorizado e se tome a decisão de como o Time se comportará mediante à ameaça. Para esse cálculo deve-se multiplicar a pontuação

da Probabilidade pela pontuação do Impacto. Sendo uma escala crescente de um a três para cada variável conforme indicado no Quadro 7.

**Probabilidade** deve ser preenchido com o valor de 1 a 3 indicando se a probabilidade do risco ocorrer é 1 - Improvável, 2 - Ocasional e 3 - Muito provável, conforme indicado no Quadro 7.

**Impacto** é preenchido com o valor de 1 a 3 indicando se o impacto do risco, caso aconteça e se torne um incidente, é 1 - Desprezível, 2 - Crítico e 3 - Gravíssimo, conforme indicado no Quadro 7.

Quadro 7: Valores da pontuação para cálculo do Fator de Risco.

| Pontuação            | 1           | 2              | 3              |
|----------------------|-------------|----------------|----------------|
| <b>Probabilidade</b> | improvável  | ocasionalmente | muito provável |
| <b>Impacto</b>       | desprezível | crítico        | gravíssimo     |

Dessa forma, seguindo o Quadro 7 por exemplo, para uma Probabilidade ocasional (pontuação dois) e um Impacto crítico (pontuação dois), o fator será a multiplicação dessas pontuações, ou seja o fator de risco será igual à quatro.

Uma vez identificado e quantificado o risco, deve-se preencher os seguintes campos:

**Funcionalidade/Fluxo** indica no atendimento de qual funcionalidade que o risco em questão foi identificado, podendo se for o caso especificar a tarefa associada a essa Estória.

**Risco** deve listar as consequências resultantes, que caso o risco ocorra se tornando um incidente, irá ocasionar, assim:

- Descrever os impactos do risco;
- Citar os afetados;
- Exemplificar como o risco, se ocorrer, afetará a organização;
- Dentre outros detalhamentos do risco.

**Medidas preventivas** devem ser elencadas ações para mitigar, eliminar ou transferir o risco identificado. Por exemplo:

- Descrever a linha de ação deliberada pelo Time para prevenção;
- Explicar as consequências da linha de ação escolhida;
- Elencar os pontos que serão solucionadas com a medida preventiva;
- Listar os pontos que ficarão descobertos com a medida preventiva escolhida;
- Dentre outras características da medida preventiva.

**Ações corretivas** a serem tomadas caso o risco se concretize e se torne um incidente.

Dessa forma pode conter:

- Comunicar os interessados e envolvidos com o impacto;
- Listar medidas paliativas para contornar o problema;
- Dentre outras atividades para a correção dos impactos do risco.

No Apêndice B - Formulário de Risco é apresentado o modelo do Formulário de Risco com um exemplo de seu preenchimento.

### 5.2.3 Proposta para Formulário de Configuração

O Formulário de Configuração visa atender a disciplina de Gerência de Configuração do nível F de maturidade do MPS.BR, e tem a função de documentar a reunião de entrega da fase de Review, bem como registrar todas as funcionalidades que a nova versão do *software* atende. Como há um relacionamento com o Formulário de Análise permite o rastreamento do que motivou determinada mudança, possibilitando auditoria e identificando o solicitante, o porquê da mudança e quando ocorreu.

Inicialmente deve-se identificar o formulário preenchendo os seguintes campos:

**Nome da Organização** no cabeçalho é destinado a identificação da organização, podendo conter o nome da instituição, seu logotipo, setor, dentre outros elementos que identifiquem à quem pertence o formulário.

**Sprint** indica a qual Sprint se refere a versão de *software* a ser entregue. Preenchido com o número da iteração do Sprint do projeto.

**Data** com o dia que a reunião de entrega da Review ocorreu para a apresentação da nova versão do *software*.

**Sistema** é o nome do sistema, indica à qual *software* se refere a nova versão.

**Versão** é o código da versão do *software* que está sendo entregue, a *release* do sistema.

**Participantes** lista todas as pessoas que participaram da reunião de entrega da Review.

Após identificado, na próxima área devem ser listadas todas as funcionalidades contempladas pela nova versão do sistema. Para tal cada item implantado (linha) deve ser preenchidos os seguintes campos (colunas):

**Módulo** à qual a funcionalidade/fluxo alterado pertence. Deve ser preenchido caso a arquitetura do sistema for modular.

**Tipo de Alteração** indica qual a intervenção aplicada na nova versão do *software*, no módulo indicado anteriormente se houver. Os possíveis tipos são: (i) “Inclusão” de uma funcionalidade, (ii) “Alteração” para o ajuste ou adaptação de uma funcionalidade ou fluxo já existente, (iii) “Correção” para o conserto de *bugs*, falhas do sistema e (iv) “Exclusão” para a remoção da funcionalidade citada.

**Funcionalidade/Fluxo** é o título da mudança. O mesmo apontado no Formulário de Análise e que foi contemplado da nova versão do sistema.

Ao final do Formulário de Configuração há a área destinada ao “Aceite”. Nessa devem ser preenchidos os seguintes campos:

**Responsável** indicando quem é o responsável pela homologação e aceite da nova versão para que essa seja colocada em produção.

**E-mail** do responsável pelo aceite para contato em caso de dúvidas quanto as informações do preenchimento ou quanto à homologação da nova versão.

**Aceito** deve ser indicado se a versão foi aceita ou recusada. Ou seja, se pode ser disponibilizada para o uso de todos em produção, ou se continuará com a atual versão do *software* sem as funcionalidades descritas na sessão “Funcionalidades/Fluxos”.

**Observações** deve ser preenchido caso o responsável pelo aceite julgue necessário indicar detalhes da entrega da nova versão do *software*. Dessa forma podendo:

- Listar ressalvas caso seja aceite;

- Pontuar as justificativas para a recusa;

- Dentre outras informações como sugestões, críticas ou elogios que julgar pertinente.

O modelo do Formulário de Configuração e o exemplo de seu preenchimento são apresentados no Apêndice C - Formulário de Configuração.

#### 5.2.4 Proposta para Formulário de Memória

O Formulário de Memória visa atender as disciplinas de Implantação de Inovações na Organização e de Análise de Causas e Resolução do nível A de maturidade do MPS.BR, assim tem por objetivo documentar os principais pontos discutidos na reunião de Retrospectiva.

As primeiras informações a serem preenchidas são referentes a identificação, contempladas pelos campos:

**Nome da Organização** no cabeçalho é destinado a identificação da organização, podendo conter o nome da instituição, seu logotipo, setor, dentre outros elementos que identifiquem à quem pertence o formulário.

**Sprint** indica à qual Sprint se refere a reunião de retrospectiva. Deve ser preenchido com o número da iteração do Sprint do projeto.

**Data** é referente ao dia que a reunião de retrospectiva da fase de Retrospectiva ocorreu.

**Participantes** lista todas as pessoas do Time que participaram da reunião de retrospectiva.

Na sequência há três áreas para preenchimento indicando os pontos mais importantes da reunião. Recomenda-se que essas áreas sejam preenchidas na estrutura de tópicos curtos para que o texto seja objetivo e conciso, sendo esses campos:

**Pontos Negativos** deve ser preenchido com os “contras” do Sprint, ou seja, fatos que o Time julga que trouxeram malefícios para o dia a dia dos trabalhos do Time durante a vigência do Sprint corrente. Por exemplo:

- Dificuldades do dia a dia;
- Problemas com membros da equipe;
- Causas ambientais;
- Impedimentos técnicos;
- Dentre outros contras.

**Pontos Positivos** deve ser preenchido com os “prós” do Sprint, ou seja, fatos que o Time julga que trouxeram benefícios para os trabalhos do Time durante o Sprint atual. Assim sendo:

- Procedimentos que melhoraram a rotina de trabalho;
- Descoberta de tecnologias que facilitaram as atividades;
- Fatores ambientais favoráveis;
- Fatos que o Time considera benéficos;
- Dentre outros prós.

**Melhorias** devem ser descritas as ações a serem tomadas pelo Time no próximo Sprint para solucionar, evitar ou minimizar os Pontos Negativos e manter ou maximizar os Pontos Positivos. Podendo:

- Listar atitudes que mitiguem riscos;
- Descrever ações para minimizar o impacto de algum ponto negativo;

- Exemplificar condutas para manter os pontos positivos;
- Sugerir propostas para maximizar o impacto favorável;
- Outros encaminhamentos para melhorar o próximo Sprint.

O modelo do Formulário de Memória juntamente com exemplo de seu preenchimento pode ser observado no Apêndice D - Formulário de Memória. O conjunto desses formulários compõem a base de conhecimento de Lições Aprendidas para ser utilizados em futuros Sprints, podendo se estender a outros projetos.

## 6 Conclusão

O termo “metodologia ágil” causa estranheza no setor público, devido à percepção que a população tem do serviço público e considerá-lo algo lento. Ágil não significa que as coisas serão mais rápidas, mas que os esforços de trabalho se concentrarão em ações que representarão grandes impactos na utilidade do produto, ou seja, ter foco no que é realmente necessário. O objetivo do ágil é buscar a melhor relação custo benefício entre clientes e a equipe de desenvolvimento, tendo como premissa que requisitos sempre mudam então a insatisfação sempre existirá. Mas essa instabilidade pode ser contornada e convertida como diferencial de adaptabilidade.

Essa relação também leva em conta a burocracia inerente do setor público que carrega a conotação de algo ineficiente, moroso, indiferente às necessidades das pessoas, e muitas vezes relacionada a questão de muitos papéis, assinaturas, carimbos, enfim excesso de formalidades. Em sua concepção, a burocracia prega uma estrutura organizacional formal, com objetivos e hierarquia definidos, tendendo a formalizar e padronizar processos. Como as organizações da Administração Pública são essencialmente regidas pela lei, o sistema burocrático é o ideal. Porém, a burocracia passa a ser sinônimo de suas disfunções deixando de ser um sistema de otimização para ser um gargalo, isso pois na tentativa de regulamentar e normatizar os administradores criam e aumentam cada vez mais os entraves burocráticos.

Portanto, esse trabalho propôs um processo que traz o equilíbrio, uma boa relação custo benefício entre a necessidade burocrática inerente do setor público, atendendo os requisitos de documentação e os riscos apontados na Seção 2.3, e a necessidade do desenvolvimento de *software* atender os anseios da comunidade. Sendo o processo proposto considerado uma burocracia necessária, aquela que otimiza os fluxos e trabalhos em busca da melhoria contínua. Ou seja, hoje a implantação desse processo de desenvolvimento de *software* em uma instituição pública promoveria a desburocratização (BELTRÃO, 1982) da parte do setor de TI responsável pela análise e desenvolvimento de sistemas de informação. E em especial, em uma instituição de ensino, devido a gama de áreas que os sistemas de informação devem atender devido a transversalidade de seus processos.

A seguir serão expostos as conclusões e considerações acerca do trabalho desenvolvido bem como ideias para a continuidade e expansão para futuros trabalhos tendo esse como base.

## 6.1 Considerações Finais

Tendo a TI como um setor estratégico dentro das organizações é importante que essa participe do planejamento e decisões junto à alta gestão, estando alinhada com as diretrizes e metas organizacionais. No que tange o desenvolvimento de sistemas, estabelecer uma metodologia de desenvolvimento de *software* traz a padronização do fluxo de desenvolvimento, aumentando a qualidade final do produto, proporcionando bom ambiente organizacional e a melhoria na satisfação do cliente e/ou usuário final, no caso do setor público, a sociedade.

Visando o alcance desses benefícios, a utilização de metodologias ágeis de desenvolvimento de *software* no setor público, mesmo que contrariando a literatura é possível, isso através da padronização no desenvolvimento em que durante seu processo sejam incorporadas atividades a fim de se gerar documentação baseadas em padrões de qualidade de gerenciamento de projetos e de desenvolvimento do *software* a fim de atender suas características.

Essa documentação busca uma boa relação custo benefício entre eficácia e burocracia para um serviço público mais eficiente, de qualidade e a um baixo custo financeiro através da gestão da tecnologia de informação, mais especificamente atuar na gestão do desenvolvimento de sistemas de informação, para diretamente refletir e melhorar a gestão pública.

Assim, a proposta do modelo conceitual de um processo de desenvolvimento de *software* para o setor público uma metodologia ágil, visa melhorar a qualidade dos serviços públicos para a população provendo os benefícios que o paradigma da Agilidade proporciona e que faz que diversas organizações privadas o adotem. Então, tendo como base o Scrum como metodologia ágil foi feita a integração de alguns processos do padrão de qualidade MPS.BR assim realizando a junção da Agilidade com a Qualidade a fim de atender as especificidades da Administração Pública Brasileira.

A modelagem de tal processo, unindo Agilidade e Qualidade, foi concluída com êxito permitindo visualizar o processo e entender melhor como se dá a sequência de atividades, bem como o detalhamento dessas com seus fluxos e artefatos envolvidos. Em especial os artefatos, os formulários elaborados a serem gerados durante o processo, visam a documentação de fatores importantes do projeto. Para a elaboração de tais formulários foi posto como premissa de estes serem ótimos no que diz respeito a terem o mínimo necessário, de forma a conter somente as informações mais relevantes e serem de fácil e rápido preenchimento e compreensão.

Dessa forma, as atividades e artefatos cobrem os seguintes níveis de maturidade e respectivos processos do MPS.BR conforme pode ser observado no Quadro 8:

Observa-se no Quadro 8 que modelo conceitual proposto contempla totalmente

Quadro 8: Relação entre os formulários, nível de maturidade e processos.

| Formulário   | Nível de Maturidade       | Processo   |
|--------------|---------------------------|--|
| Memória      | A Em otimização           | Implantação de Inovações na Organização<br>Análise de Causas e Resolução |
| Risco        | C Definido                | Gerência de Risco  |
| Configuração | F Gerenciado              | Gerência de Configuração   |
| Análise      | G Parcialmente Gerenciado | Gerência do Projeto<br>Gerência do Requisito                             |

o Nível G - Parcialmente Gerenciado com seus dois processos: Gerência de Projeto e Gerência de Requisitos. Atende dois processos dos níveis F e C, Gerência de Configuração e Gerência de Risco, respectivamente. E por fim todos os dois processos do Nível A, sendo eles Análise de Causas e Resolução e Implantação de Inovações na Organização.

Tendo como base os níveis de maturidade em qualidade do MPS.BR apresentados anteriormente no Quadro 1 e comparando com os do modelo conceitual no Quadro 8 percebe-se que o processo é totalmente aderente ao Nível G de maturidade, porém embora implante todos os processos do Nível A, para que seja considerado nessa maturidade é preciso que contemple todos os processos dos níveis anteriores.

Por fim, ainda que o processo seja classificado como Nível G, o mais baixo, ele ainda atende a outros processos de outros níveis, inclusive os processos do Nível A, o mais alto. Porém não pode receber uma classificação superior, pois para tal teria que atender todos os processos do Nível F, por exemplo. Mas ainda que no menor nível e atendendo a outros processos, considera-se esses o mínimo necessário para que as necessidades do setor público sejam satisfeitas.

Assim sendo, a maior contribuição desse trabalho consiste no processo de desenvolvimento de *software* a ser aplicado em uma instituição pública de educação, ou até mesmo em uma organização pública no âmbito federal, para que os benefícios dessa padronização reflita na comunidade atendida pela instituição e em toda sociedade. E que tal modelagem contribua para a área acadêmica, disponibilizando o modelo de processo para a comunidade e esperando contribuições em forma de esperando *feedbacks* e intervenções para ajustes e adaptações, para a melhoria do processo proposto.

## 6.2 Trabalhos Futuros

Como proposta para a continuidade e extensão dessa monografia e possíveis trabalhos futuros tendo esse como base, sugere-se as seguintes frentes e linhas de atuação:

**Estudo de caso** A aplicação do modelo do processo ágil de desenvolvimento de *soft-*

*ware* elaborado no setor de TI de uma instituição de educação pública, ou até em organização pública do âmbito federal. Isso para que seja verificada sua aderência na prática, assim validando o modelo, e caso haja, a indicação de ajustes e adaptações para a contribuição da melhoria do processo. Por exemplo, inclusão de campos nos formulários para maior completude das informações ou exclusão para tornar seu preenchimento menos burocrático. Ou a edição de atividades e fluxos do processo visando aproximar o modelo às necessidades do setor público, atentando para que nessas alterações não se percam as premissas dos métodos ágeis descritos no Manifesto Ágil (BECK; BEEDLE; BENNEKUM, 2001).

**Métrica e Estimativa** Nas primeiras etapas do processo, agregar à fase de levantamento de requisitos e no processo de deliberação das prioridades a serem desenvolvidas técnicas de métrica de *software*, por exemplo pontos de função, pontos de caso de uso, planning poker<sup>1</sup>, dentre outros. Essas métricas permitem entender, avaliar, controlar e prever o esforço e tempo que serão despendidos para a conclusão do desenvolvimento dos incrementos de *software* necessários para atender as solicitações do requisitante. Assim permitindo estimar e possibilitando o acompanhamento do andamento dos trabalhos da equipe.

**Transparência** Uma das atuais campanhas que a administração pública brasileira tem atacado com veemência é a questão da transparência pública, tendo como mote que “*acesso a informação é a regra e o sigilo, a exceção*”. Baseado nessa diretriz, tornar os documentos e bases de dados resultantes do processo proposto públicos, ou ainda no processo de desenvolvimento incluir atividades que garantam o cumprimento de tal diretiva. Atividades essas de comunicação para que todo e qualquer interessado da sociedade possa ter acesso a informações do projeto e do processo para o cumprimento de tal conduta, possibilitando a sociedade atuar como um agente público de fiscalização.

**DevOps** Essa é uma metodologia de desenvolvimento ágil de *software* que integra desenvolvedores e profissionais de TI responsáveis pela operação, geralmente os que atuam no suporte junto ao usuário final. O DevOps auxilia organizações a produzir *software* e serviços mais rapidamente e visa automatizar a maior quantidade possível de processos operacionais. Aproximar de alguma forma o usuário final, ou seja a comunidade à equipe de desenvolvimento, isso traria uma resposta mais rápida e certa aos anseios e necessidades da sociedade. Para agregar o DevOps à metodologia ágil elaborada, é necessário agregar atividades à esse e adicionar uma nova

<sup>1</sup> planning poker é uma técnica utilizada no Scrum para estimar o esforço de desenvolvimento de um requisito. Essa avaliação é acordada entre o Time, que jogam cartas numeradas sem mostrar seus valores em vez de falar em voz alta. As cartas são reveladas e o valor da estimativa é discutido até que haja consenso. Ao esconder a carta o Time evita o efeito manada onde o primeiro a falar influencia os demais.

Lane com o papel o Ops (operação), ou até um novo processo que interaja com o modelo proposto para que de alguma forma, via o Suporte, as necessidades cheguem ao Time para que o *software* atenda com maior completude as necessidades da coletividade.

**Evolução da Maturidade** Incluir atividades e artefatos, de preferência no formato de formulários zelando pela burocracia mínima necessária, para que o processo seja aderente a mais níveis do MPS.BR e seus respectivos processos, como por exemplo o processo Avaliação e Melhoria do Processo Organizacional (AMP) do nível E (Parcialmente Definido), e o processo Gerência Quantitativa do Projeto (GQP) do nível B (Gerenciado Quantitativamente). O último em especial, uma das ações a se tomar é a frente de trabalhos futuros para essa monografia “Métrica e Estimativa” descrita anteriormente, que fornecerá números para a gerência quantitativa do andamento dos trabalhos para controle do projeto.

**Informatização do Processo** Pesquisar sistemas computacionais para adaptações ou desenvolver uma solução informatizada para que toda a execução do processo seja feita digitalmente. O fluxo das atividades controlados por permissões de acesso, as informações dos formulários centralizadas em banco de dados bem como o Backlog e as Lições Aprendidas, ferramentas como filtros, gráficos e relatórios, permitiriam um melhor gerenciamento do projeto, podendo até ampliar para que sejam gerados indicadores de desempenho do processo de desenvolvimento de *software* para melhor gestão.

## Referências

- ABNT. *NBR ISO 9000: 2000. Sistema de gestão da qualidade. Fundamentos e vocabulário*. [S.l.]: ABNT, 2000. Citado na página 19.
- ALVES, B. P. Plano de Intervenção para o Setor de Tecnologia de Informação do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo. Título de Especialista em Gestão Pública. 2013. Citado na página 23.
- BALLARD, M. Agile will fail GovIT, says corporate lawyer. *Disponível em:* <[www.computerweekly.com/blogs/public-sector/2011/04/agile-will-fail-govit-says-corpor.html](http://www.computerweekly.com/blogs/public-sector/2011/04/agile-will-fail-govit-says-corpor.html)>. Acesso em 12/2014, 2011. Citado 2 vezes nas páginas 11 e 34.
- BARBIERI, C. *PósGraduação em Engenharia e Qualidade de Software com Modelo MPS*. 2011. Disponível em: <[slideplayer.com.br/slide/44839/](http://slideplayer.com.br/slide/44839/)>. Acesso em 03/2016. Citado 2 vezes nas páginas 7 e 20.
- BECK, K.; BEEDLE, M.; BENNEKUM, A. V. Manifesto for Agile Software Development. *Disponível em:* <[agilemanifesto.org](http://agilemanifesto.org)>. Acesso em 12/2014, 2001. Citado 3 vezes nas páginas 15, 16 e 60.
- BELTRÃO, H. *Desburocratização: idéias fundamentais*. [S.l.]: Presidência da República, Programa Nacional de Desburocratização, 1982. Citado 3 vezes nas páginas 23, 39 e 57.
- BENITES, M.; PAIVA, D. M. B.; FERNANDES, P. Implantação de Resultados Esperados do Processo Gerência de Projetos com o Apoio do Scrum no Setor Público. *SEGeT – Simpósio de Excelência em Gestão e Tecnologia*, v. 8, p. 19, 2011. Citado 2 vezes nas páginas 11 e 34.
- BENITES, M. G.; CAGNIN, M. I.; PAIVA, D. Iamps: An process to support the mps. br implementation together with agile methods. In: IEEE. *Computing Conference (CLEI), 2014 XL Latin American*. [S.l.], 2014. p. 1–9. Citado na página 33.
- BIZAGI. *Bizagi. Model - Muild - Run*. 2015. Disponível em: <<http://www.bizagi.com/>>. Citado na página 36.
- CAELUM. *PM-83 Gerenciamento Ágil de Projetos de Software com Scrum*. 2011. Citado 2 vezes nas páginas 16 e 17.
- CARVALHO, A. A. d. Scrummps 2.0: Evolução de uma Ferramenta Interativa para Suporte ao Scrum e Mps.BR. 2013. Citado na página 32.
- CARVALHO, E. A. *Engenharia de Processos de Negócios e a Engenharia de Requisitos: Análise e Comparações de Abordagens e Métodos de Elicitação de Requisitos de Sistema Orientada por Processos de Negócio*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro - UFRJ, 2009. Citado na página 25.
- CARVALHO, M. M.; PALADINI, E. P.; BOUER, G. Gestão da Qualidade. *Teoria e casos*, v. 2, 2012. Citado na página 15.

- CATUNDA, E.; NASCIMENTO, C.; CERDEIRAL, C. Implementação do Nível F do MR-MPS com Práticas Ágeis do Scrum em uma Fábrica de Software. *SBQS2011*, 2011. Citado na página 13.
- CHAVES, N.; TAKADA, L.; MACIEIRA, A. *Coletânea de Casos em Gerenciamento de Processos na Administração Pública*. [S.l.]: ABPMP Brazil, 2014. Citado na página 22.
- CMMI. CMMI for Development, Version 1.3, Improving processes for developing better products and services. no. *CMU/SEI-2010-TR-033*. *Software Engineering Institute*, 2010. Citado na página 19.
- FAGUNDES, P. B.; DETERS, J. I.; SANTOS, S. d. S. Comparação entre os Processo dos Métodos ágeis: Xp, scrum, fdd e asd em Relação ao Desenvolvimento Iterativo Incremental. *Atualidades Tecnológicas para Competitividade Industrial, Florianópolis*, 2008. Citado 2 vezes nas páginas 16 e 35.
- FERNANDES, A. A.; ABREU, V. F. *Implantando a Governança de TI : da Estratégia à Gestão de Processos e Serviços*. [S.l.]: Brasport, 2014. Citado na página 11.
- FERREIRA, R. B.; LIMA, F. P. A. Metodologias Ágeis: Um Novo Paradigma de Desenvolvimento de Software. In: *II Workshop Um Olhar Sociotécnico sobre a Engenharia de Software - WOSES*. [S.l.: s.n.], 2006. Citado na página 11.
- GERHARDT, T. E.; SILVEIRA, D. T. *Metodos de Pesquisa*. 2009. Citado na página 30.
- GIL, A. C. Como Elaborar Projetos de Pesquisa. *São Paulo*, v. 5, p. 61, 2002. Citado na página 30.
- GUARDATI, S.; PONCE, A. *Guía de Pruebas de Software (GPS) para MoProSoft*. [S.l.]: Comunidad MoProSoft, 2010. Citado na página 19.
- JUNIOR, J. M.; PINHEIRO, T. H. Introdução à Gestão de Processos - Módulo 1: Introdução e Conceitos Básicos. In: ENAP - FUNDAÇÃO ESCOLA NACIONAL DE ADMINISTRAÇÃO PÚBLICA. *Introdução à Gestão de Processos*. [S.l.], 2015. Citado 2 vezes nas páginas 25 e 35.
- JUNIOR, J. M.; PINHEIRO, T. H. Introdução à Gestão de Processos - Módulo 2: Como Gerir e Melhorar Processos. In: ENAP - FUNDAÇÃO ESCOLA NACIONAL DE ADMINISTRAÇÃO PÚBLICA. *Introdução à Gestão de Processos*. [S.l.], 2015. Citado na página 25.
- KALINOWSKI, M.; SANTOS, G.; REINEHR, S. MPS.BR: Promovendo a Adoção de Boas Práticas de Engenharia de Software pela Indústria Brasileira. In: *XIII Congreso Iberoamericano en "Software Engineering"(CIBSE)*. *Universidad del Azuay (in Portuguese), Cuenca, Equador, ISBN*. [S.l.: s.n.], 2010. p. 978-9978. Citado na página 35.
- KAMEI, F. K.; QUEIROZ, F. B.; FILHO, R. R. G. N. Scrum no Serviço Público: um Relato de Implantação nas Secretarias Estaduais da Fazenda e da Gestão Pública do Estado de Alagoas. *Simpósio Brasileiro de Qualidade de Software - SBC*, v. 5, 2012. Citado 2 vezes nas páginas 11 e 34.

- KNIBERG, H.; SKARIN, M. *Kanban e Scrum - obtendo o melhor de ambos*. [S.l.]: C4Media, 2009. Citado 2 vezes nas páginas 7 e 18.
- LÜBECK, R. M.; WITTMANN, M. L.; GOMES, C. M. Inovação na Gestão da Informação: Evidências Empíricas no Setor de Transporte Público Urbano. *Revista de Administração e Inovação - RAI*, v. 9, n. 4, 2012. Citado na página 13.
- MARTINS, A. L. A.; ANDRADE, J.; GONÇALEZ, F. F. Qualidade de Software e a Inserção do Modelo MPS.BR nas Empresas. *Revista Tecnologia em Projeção*, v. 3, n. 2, p. 43–49, 2013. Citado na página 19.
- MONTEIRO, J. M. *Levantamento sobre Aplicação de Metodologias Ágeis em Desenvolvimento de Software*. 2013. Citado 2 vezes nas páginas 23 e 24.
- MORAES, L. *A Vida com Scrum*. 2010. Disponível em: <alemdati.wordpress.com>. Acesso em 04/2016. Citado na página 16.
- OLIVEIRA, J. F. Gestão de Tecnologias da Informação e da Comunicação na Saúde: uma análise sobre o uso do prontuário eletrônico. *INTERFACE*, v. 9, n. 1, 2013. Citado na página 12.
- PARENTE, P. Diferenças entre Gestões Públicas e Privadas. In: *Congresso de Entidades Filiadas à Federasul*. [S.l.: s.n.], 2009. Citado 2 vezes nas páginas 8 e 22.
- PRESSMAN, R. S. *Engenharia de Software*. [S.l.]: McGraw Hill Brasil, 2011. Citado na página 15.
- ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C. *Qualidade de Software*. [S.l.]: São Paulo: Prentice Hall, 2001. Citado 2 vezes nas páginas 11 e 19.
- ROCHA, D. S. Avaliação do Método Ágil Scrum para uma Instituição Pública. Especialização em Gestão de Projetos em Desenvolvimento de Sistemas de Software. 2015. Citado na página 34.
- ROCHA, Y. C. C.; OLIVEIRA, E. A. A. Q.; OLIVEIRA, A. L. Inclusão Digital do Governo Eletrônico de São Paulo. *Revista Científica on-line - Tecnologia, Gestão e Humanismo*, v. 2, n. 1, 2013. Citado na página 13.
- RUBIN, K. S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. [S.l.]: Addison-Wesley, 2012. Citado na página 16.
- SALGADO, A.; MELCOP, T.; ACCHAR, J. Aplicação de um Processo Ágil para Implantação de Processos de Software Baseado em Scrum na Chemtech. *IX Simpósio Brasileiro de Qualidade de Software, Belém, Brasil*, p. 351–358, 2010. Citado na página 16.
- SCHWABER, K.; BEEDLE, M. *Agile Software Development with Scrum*. [S.l.]: Pearson, 2002. Citado na página 16.
- SOFTEX. MPS.BR-Melhoria de Processo do Software Brasileiro. 2012. Citado 3 vezes nas páginas 8, 20 e 21.

- SZIMANSKI, F.; ALBUQUERQUE, J.; FURTADO, F. Implementando Maturidade e Agilidade em uma Fábrica de Software Através de Scrum e MPS.BR Nível G. *XI Encontro de Estudantes de Informática do Tocantins. Centro Universitário Luterano de Palmas*, p. 161–170, 2009. Citado na página [33](#).
- TELES, V. M. Extreme Programming: Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. *Novatec*, 2004. Citado na página [13](#).
- TRECCANI, P. J.; SOUZA, C. R. Utilização de Metodologias ágeis no Desenvolvimento de Software: Resultados de um Estudo Empírico. In: *Experimental Software Engineering Latin Workshop(ESELAW)*. [S.l.: s.n.], 2010. v. 7, p. 50–59. Citado na página [15](#).
- WHITE, S. A. Introduction to BPMN. *IBM Cooperation*, v. 2, n. 0, p. 0, 2004. Citado 4 vezes nas páginas [25](#), [26](#), [27](#) e [28](#).

# Apêndices

# APÊNDICE A – Formulário de Análise

[NOME DA ORGANIZAÇÃO]

| Formulário de Análise  |                                     |                     |   |
|--|-------------------------------------|---------------------|---|
| <b>Informações Gerais</b>  |                                     |                     |   |
| <i>Sistema</i>   | Sistema Integrado de Gestão         | <i>Módulo</i>       | Requisições   |
| <i>Solicitante</i>   | Brunno dos Passos Alves             | <i>E-mail</i>       | brunno.ifsp@gmail.com   |
| <i>Data</i>  | 21/03/2016                          | <i>Telefone</i>     | (11) 234-5678   |
| <b>Funcionalidades/Fluxos</b> <i>(Alterações – se já existe no sistema e se sua implantação é crítica para o negócio da organização)</i>   |                                     |                     |   |
|  |                                     | Existe              | Crítico   |
| 1  | Relatório de totalização por status |                     | X   |
| 2  | Ordenar por data de criação         | X                   |   |
| <b>1 Relatório de totalização por status</b>   |                                     |                     |   |
| <b>Como quem</b> <i>(papel/perfil do usuário)</i>  |                                     |                     |   |
| Gerente  |                                     |                     |   |
| <b>Como é</b> <i>(atual funcionamento do sistema)</i>  |                                     |                     |   |
| <ul style="list-style-type: none"> <li>Hoje não há a informação de totalizadores para quantificar o número de requisições.</li> <li>Quando há a necessidade dessa informação é coletada individualmente consultando todas as requisições.</li> </ul>                               |                                     |                     |   |
| <b>Porque mudar</b> <i>Descrição do porquê mudar</i>   |                                     |                     |   |
| <ul style="list-style-type: none"> <li>A informação de totalização é útil para a prestação de contas ao usuário final e necessária para o planejamento.</li> <li>O tempo para compilar manualmente esses dados é muito grande, bem como montar as planilhas e gráficos.</li> </ul> |                                     |                     |   |
| <b>Alteração</b> <i>Descrição de como deseja a mudança, como o sistema deveria se comportar</i>  |                                     |                     |   |
| <ul style="list-style-type: none"> <li>Criar um relatório onde consiga visualizar as requisições com gráfico diferenciando a situação aberto, em atendimento e concluído.</li> <li>Exemplo:</li> </ul>   |                                     |                     |   |
|  |                                     |                     |   |
| <b>Número</b>  | <b>Setor</b>                        | <b>Requisitante</b> | <b>Status</b>   |
| 156165   | RH                                  | Maria               | Aberto  |
| 87941  | RH                                  | Fernando            | Concluído   |
| 870231   | Finanças                            | Cláudio             | Concluído   |
|  |                                     |                     |  |
| <b>2 Ordenar por data de criação</b>   |                                     |                     |   |
| <b>Como quem</b> <i>(papel/perfil do usuário)</i>  |                                     |                     |   |
| ...  |                                     |                     |   |
| <b>Como é</b> <i>(atual funcionamento do sistema. Para novas funcionalidades não se aplica)</i>  |                                     |                     |   |
| ...  |                                     |                     |   |
| <b>Porque mudar</b> <i>Descrição do porquê mudar</i>   |                                     |                     |   |
| ...  |                                     |                     |   |
| <b>Alteração</b> <i>Descrição de como deseja a mudança, como o sistema deveria se comportar</i>  |                                     |                     |   |
| ...  |                                     |                     |   |

# APÊNDICE B – Formulário de Risco

[NOME DA ORGANIZAÇÃO]

| Formulário de Risco  |                                 |                      |                |                |   |
|--|---------------------------------|----------------------|----------------|----------------|---|
| <b>Risco</b>   |                                 |                      |                |                |   |
| <i>Sprint</i>  | 3                               |                      | <i>Data</i>    | 11/04/2016     |   |
| <i>Título</i>  | Valores nulos no banco de dados |                      |                |                |   |
| <i>Fator (Probabilidade x Risco)</i>   | 6                               | <i>Probabilidade</i> | 3              | <i>Impacto</i> | 2 |
| <b>Pontuação</b>   | <b>1</b>                        | <b>2</b>             | <b>3</b>       |                |   |
| <i>Probabilidade</i>   | improvável                      | ocasionalmente       | muito provável |                |   |
| <i>Impacto</i>   | desprezível                     | crítico              | gravíssimo     |                |   |
| <b>Funcionalidade/Fluxo</b> ( <i>Funcionalidade/Fluxo e tarefa que gerou o risco</i> )   |                                 |                      |                |                |   |
| <ul style="list-style-type: none"> <li>Relatórios totalizadores – consulta ao banco de dados</li> </ul>  |                                 |                      |                |                |   |
| <b>Risco</b> ( <i>consequências do risco</i> )   |                                 |                      |                |                |   |
| <ul style="list-style-type: none"> <li>Os quantitativos podem não representar a realidade.</li> <li>Se não tratado pode ocorrer um erro para o usuário ao ser solicitado o relatório.</li> <li>O gráfico pode ser exibido com alguma desconfiguração gráfica.</li> </ul> |                                 |                      |                |                |   |
| <b>Medidas Preventivas</b> ( <i>ações a serem tomadas para mitigar, excluir ou transferir o risco</i> )  |                                 |                      |                |                |   |
| <ul style="list-style-type: none"> <li>Tratar valores nulos e exibir mensagem para o usuário quando isso acontecer.</li> <li>Completar com valores válidos os dados nulos.</li> <li>Tornar obrigatório o preenchimento das informações.</li> </ul>                       |                                 |                      |                |                |   |
| <b>Ações Corretivas</b> ( <i>atividades a serem tomadas caso o risco se torne um incidente</i> )   |                                 |                      |                |                |   |
| <ul style="list-style-type: none"> <li>Retirar módulo do ar.</li> <li>Alertar aos usuários sobre o erro.</li> </ul>  |                                 |                      |                |                |   |



# APÊNDICE D – Formulário de Memória

[NOME DA ORGANIZAÇÃO]

| Formulário de Memória  |  |             |            |
|--|--|-------------|------------|
| <b>Retrospectiva</b>   |  |             |            |
| <i>Sprint</i>  | 3  | <i>Data</i> | 24/04/2016 |
| <i>Participantes</i>   | <ul style="list-style-type: none"> <li>• Brunno Alves</li> <li>• Fulano Rocha</li> <li>• Beltrano da Silva</li> <li>• Elaine Vieira</li> </ul> |             |            |
| <b>Pontos Negativos</b> ( <i>tópicos com os pontos negativos levantados pelo Time durante o Sprint</i> )   |  |             |            |
| <ul style="list-style-type: none"> <li>• Queda constante de energia elétrica.</li> <li>• Falta de conhecimento no framework utilizado.</li> <li>• Problemas de comunicação com o DBA.</li> </ul>   |  |             |            |
| <b>Pontos Positivos</b> ( <i>tópicos com os pontos positivos levantados pelo Time durante o Sprint</i> )   |  |             |            |
| <ul style="list-style-type: none"> <li>• Happy hour motivou mais a equipe.</li> <li>• Home Work rendeu mais nos resultados dos trabalhos</li> </ul>  |  |             |            |
| <b>Melhorias</b> ( <i>tópicos das ações baseadas nos pontos negativos e positivos para o próximo Sprint</i> )  |  |             |            |
| <ul style="list-style-type: none"> <li>• Viabilizar a compra de no break.</li> <li>• Verificar material para treinamento no framework</li> <li>• DBA se comprometeu a responder os e-mails mais rapidamente.</li> <li>• Marcar happy hour com a equipe com mais frequência.</li> </ul> |  |             |            |

# Anexos

# ANEXO A – Manifesto Ágil

## A.1 Manifesto para o Desenvolvimento Ágil de Software

Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar:

**Indivíduos e interação entre eles** mais que processos e ferramentas

**Software em funcionamento** mais que documentação abrangente

**Colaboração com o cliente** mais que negociação de contratos

**Responder a mudanças** mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

|                  |               |                   |                   |
|------------------|---------------|-------------------|-------------------|
| Kent Beck        | Mike Beedle   | Arie van Bennekum | Alistair Cockburn |
| Ward Cunningham  | Martin Fowler | James Grenning    | Jim Highsmith     |
| Andrew Hunt      | Ron Jeffries  | Jon Kern          | Brian Marick      |
| Robert C. Martin | Steve Mellor  | Ken Schwaber      | Jeff Sutherland   |
| Dave Thomas      |               |                   |                   |

## A.2 Os Doze princípios do Software Ágil

Princípios por trás do manifesto ágil.

Nós seguimos os seguintes princípios:

1. Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
3. Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
4. Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.

5. Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
6. O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
7. Software funcional é a medida primária de progresso.
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
9. Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
10. Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
11. As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
12. Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.