

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS  
UNIDADE ACADÊMICA DE EDUCAÇÃO CONTINUADA  
MBA EM GESTÃO ESTRATÉGICA DE NEGÓCIOS

ISMAEL DA SILVA DE MELO

GERENCIAMENTO, CONTRATAÇÃO E DESENVOLVIMENTO DE PROJETOS DE  
SOFTWARE ENXUTO COM PONTOS DE FUNÇÃO: UMA PROPOSTA PARA  
MELHORIA DE PROCESSOS NO ESTUDO DE CASO DA EMPRESA FINA S/A

PORTO ALEGRE

2016

ISMAEL DA SILVA DE MELO

GERENCIAMENTO, CONTRATAÇÃO E DESENVOLVIMENTO DE PROJETOS DE  
SOFTWARE ENXUTO COM PONTOS DE FUNÇÃO: UMA PROPOSTA PARA  
MELHORIA DE PROCESSOS NO ESTUDO DE CASO DA EMPRESA FINA S/A

Projeto de Pesquisa apresentado como requisito parcial para obtenção do título de Especialista em Gestão Estratégica de Negócios, pelo MBA em Gestão e Negócios da Tecnologia da Informação da Universidade do Vale do Rio dos Sinos – UNISINOS.

Orientador: Prof. Dr. Emir José Redaelli

Porto Alegre

2016

## **AGRADECIMENTOS**

À Deus, o que seria de mim sem a fé que eu tenho nele.

Aos meus pais, Nilo e Célia Melo, minha esposa Débora Melo (que como sempre, me deu apoio integral durante a realização desta pesquisa), minhas filhas Quezia, Luiza e Lívia, e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

À Unisinos, pela oportunidade para eu ampliar meus conhecimentos e por proporcionar o relacionamento interpessoal.

Ao meu orientador, Prof. Dr. Emir José Redaelli, pela confiança depositada no propósito do meu trabalho, pelo apoio e dedicação, sempre que necessário. E principalmente, quando do desafio que a ele impus durante o curto, porém intenso período de orientação a que o submeti.

À empresa onde trabalho e meus colegas de trabalho, por me proporcionar o conhecimento sobre seus processos para a perfeita realização do trabalho.

E, por fim, a todos os demais que direta ou indiretamente colaboraram para a minha formação, o meu obrigado.

## RESUMO

A organização FINA S/A, visando evoluir seus processos e reduzir o tempo de entrada de novos produtos e funcionalidades no mercado, tornando assim a empresa mais competitiva no seu ramo de atuação perante o seu público-alvo, adotou recentemente novos métodos de trabalho, definidos como mais ágeis, colaborativos, enxutos e mensuráveis em relação ao contexto tradicional, outrora consolidado. Com este propósito, e com base em novas práticas sendo adotadas na indústria de software, destaca-se que foi concebido e implantado um modelo ágil inicial, com o intuito de evoluí-lo gradualmente ao longo do tempo. Ao mesmo passo, iniciaram-se iniciativas para revisão do ciclo de desenvolvimento tradicional, contemplando também sua adaptação a uma abordagem mais enxuta.

Com base no cenário apresentado, este estudo apresenta uma proposta de melhoria de processos de gerenciamento, contratação e desenvolvimento de sistemas na empresa FINA S/A, os quais foram construídos conforme preconizam os princípios da metodologia ágil e os métodos tradicionais utilizados pela organização (estes orientados a um contexto de maior certeza), ambos analisados em uma perspectiva enxuta.

Salienta-se, que esta pesquisa foi realizada com base em uma análise bibliográfica vinculada ao tema, transpondo e demonstrando um conjunto de boas práticas de desenvolvimento. E posteriormente, foi realizada a coleta e análise de dados com base nos métodos científicos selecionados, onde se destaca o estudo de caso realizado.

Ao final da pesquisa, foi possível identificar oportunidades de melhorias nos já referidos processos da organização, inclusive contemplando uma proposta de possíveis derivações da análise realizada.

**Palavras-chave:** Desenvolvimento de Software. Métodos Ágeis. Software Enxuto. Lean. Pontos de Função.

## LISTA DE FIGURAS

<b>Figura 1 – Antítese e a Cultura Organizacional .....</b>	<b>17</b>
<b>Figura 2 – Design Thinking como um Processo.....</b>	<b>20</b>
<b>Figura 3 – Laço de Aprendizagem do Lean Startup .....</b>	<b>23</b>
<b>Figura 4 – Etapas do Desenvolvimento <i>Waterfall</i>.....</b>	<b>26</b>
<b>Figura 5 – Ciclo de Desenvolvimento Scrum.....</b>	<b>34</b>
<b>Figura 6 – Template de Visão do Produto .....</b>	<b>38</b>
<b>Figura 7 – Exemplo de Medição de Velocidade com Story Points.....</b>	<b>39</b>
<b>Figura 8 – Exemplo da Definição de Preparado .....</b>	<b>39</b>
<b>Figura 9 – Fluxo do Item: Definição de Preparado e Definição de Pronto .....</b>	<b>40</b>
<b>Figura 10 – Quadro de Tarefas do Scrum .....</b>	<b>41</b>
<b>Figura 11 – Exemplo da Definição de Pronto.....</b>	<b>42</b>
<b>Figura 12 – Exemplo de um Quadro de Tarefas ao Final da Sprint .....</b>	<b>44</b>
<b>Figura 13 – Quadro de Retrospectivas .....</b>	<b>45</b>
<b>Figura 14 – Gráfico de Release Burndown.....</b>	<b>46</b>
<b>Figura 15 – Gráfico de Sprint Burndown.....</b>	<b>47</b>
<b>Figura 16 – Gráfico de Release Burnup .....</b>	<b>47</b>
<b>Figura 17 – Metodologia Ágil do SISP .....</b>	<b>49</b>
<b>Figura 18 – MVP Canvas (exemplo).....</b>	<b>55</b>
<b>Figura 19 – Exemplo do Desenvolvimento Incremental com MVPs .....</b>	<b>56</b>
<b>Figura 20 – Agenda da Concepção Enxuta .....</b>	<b>58</b>
<b>Figura 21 – Agenda Burnup.....</b>	<b>58</b>
<b>Figura 22 – Sala de Guerra .....</b>	<b>59</b>
<b>Figura 23 – Parque em Lote (Parking Lot).....</b>	<b>60</b>
<b>Figura 24 – Template da Visão do Produto .....</b>	<b>62</b>
<b>Figura 25 – É - Não é - Faz - Não faz .....</b>	<b>62</b>
<b>Figura 26 – Objetivos do Produto .....</b>	<b>63</b>
<b>Figura 27 – Template de Personas .....</b>	<b>65</b>
<b>Figura 28 – Mapa de Empatia .....</b>	<b>65</b>
<b>Figura 29 – Exemplo de Features, Objetivos e Personas, .....</b>	<b>67</b>
<b>Figura 30 – Exemplo de Priorização de Features .....</b>	<b>68</b>
<b>Figura 31 – Exemplo de Features com Marcações de Esforço .....</b>	<b>69</b>

<b>Figura 32 – Entendimento de Negócio x Técnico e Esforço x Retorno</b> .....	70
<b>Figura 33 – Features depois das atividades</b> .....	70
<b>Figura 34 – Exemplo de "jornadas do usuário"</b> .....	71
<b>Figura 35 – Exemplo de Features x Jornadas</b> .....	73
<b>Figura 36 – Exemplo de Features vinculadas às Jornadas</b> .....	73
<b>Figura 37 – Exemplo de Canvas MVP</b> .....	75
<b>Figura 38 – Contexto de Administração das Aquisições</b> .....	82
<b>Figura 39 – Processo de Medição Funcional</b> .....	95
<b>Figura 40 – Processo de Avaliação Não Funcional</b> .....	103
<b>Figura 41 – Região Impossível e a Relação de Custo <i>versus</i> Esforço</b> .....	106
<b>Figura 42 – Portfólio de Projetos e os Demais Processos Organizacionais</b> ....	120
<b>Figura 43 – Relação entre Projetos, Programa e Portfólio</b> .....	121
<b>Figura 44 – Critérios de Priorização de Portfólio (exemplo)</b> .....	123
<b>Figura 45 – Exemplo de Planilha de Acompanhamento de Portfólio</b> .....	124
<b>Figura 46 – Backlog do Produto e Sustentação</b> .....	132
<b>Figura 47 – Processo de Portfólio de Projetos de TI</b> .....	145
<b>Figura 48 – Fases de Gerenciamento de Projetos</b> .....	146
<b>Figura 49 – Fases de Projetos e Etapas de Desenvolvimento</b> .....	148
<b>Figura 50 – Etapas do Método Tradicional</b> .....	149
<b>Figura 51 – Etapas do Método Ágil</b> .....	149
<b>Figura 52 – Etapas do Método Híbrido</b> .....	150
<b>Figura 53 – Pontos de Estimativa ou Medição</b> .....	153

## LISTA DE QUADROS

<b>Quadro 1 – Pontos-chave do Scrum e papéis relacionados .....</b>	<b>36</b>
<b>Quadro 2 – Objetos Contratados em Órgãos Públicos com Métodos Ágeis .....</b>	<b>78</b>
<b>Quadro 3 – Tipos de Função na APF .....</b>	<b>96</b>
<b>Quadro 4 – Complexidade em Funções de Dados .....</b>	<b>97</b>
<b>Quadro 5 – Complexidade do Tipo de Função EE na APF .....</b>	<b>97</b>
<b>Quadro 6 – Complexidade dos Tipos de Função CE e SE na APF .....</b>	<b>97</b>
<b>Quadro 7 – Expoente t por tipo de Projeto.....</b>	<b>105</b>
<b>Quadro 8 – Estimativas de Prazo de Projetos menores que 100 PF.....</b>	<b>106</b>
<b>Quadro 9 – Questões da Entrevista.....</b>	<b>134</b>
<b>Quadro 10 – Questões da Enquete (Survey).....</b>	<b>135</b>
<b>Quadro 11 – Respostas: Gerenciamento de Portólio de Projetos .....</b>	<b>145</b>
<b>Quadro 12 – Respostas: Gerenciamento de Projetos.....</b>	<b>148</b>
<b>Quadro 13 – Respostas: Processo de Desenvolvimento.....</b>	<b>151</b>
<b>Quadro 14 – Respostas: Homologação de Software .....</b>	<b>152</b>
<b>Quadro 15 – Respostas: Contratação de Fornecedores.....</b>	<b>154</b>
<b>Quadro 16 – Respostas: Contratação de Fornecedores.....</b>	<b>156</b>
<b>Quadro 17 – Respostas: Contratações de Software .....</b>	<b>157</b>
<b>Quadro 18 – Respostas: Gestão de Fornecedores .....</b>	<b>159</b>
<b>Quadro 19 – Respostas: Pessoas e Culturas .....</b>	<b>160</b>
<b>Quadro 20 – Respostas: Boas Práticas.....</b>	<b>161</b>
<b>Quadro 21 – Respostas: Arquitetura e Tecnologias .....</b>	<b>162</b>
<b>Quadro 22 – Respostas: Ferramentas Internas .....</b>	<b>163</b>
<b>Quadro 23 – Propostas de Melhorias .....</b>	<b>166</b>

## LISTA DE ABREVIATURAS

APF	Análise de Pontos de Função
ALM	<i>Application Lifecycle Management</i>
CIO	<i>Chief Information Officer</i>
CPM	<i>Counting Practices Manual</i>
COSMIC	<i>Common Software Measurement International Consortium</i>
EA	<i>Enterprise Architect</i>
IFPUG	<i>International Function Points Users Group</i>
PMI	<i>Project Management Institute</i>
PMO	<i>Project Management Office</i>
SISP	Secretaria de Logística e Tecnologia de Informação
SLA	<i>Service Level Agreement</i>
SNAP	<i>Software Non Functional Process</i>
TI	Tecnologia da Informação

# Sumário

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
<b>1.1 Definição do Problema</b> .....	<b>13</b>
<b>1.2 Objetivos</b> .....	<b>15</b>
1.2.1 Objetivo Geral .....	15
1.2.2 Objetivos Específicos .....	15
<b>1.3 Justificativa</b> .....	<b>16</b>
<b>2 REVISÃO BIBLIOGRÁFICA</b> .....	<b>17</b>
<b>1.1 Cocriação &amp; Pensamento Enxuto</b> .....	<b>17</b>
1.1.1 Design Thinking.....	18
1.1.2 Lean Thinking.....	20
1.1.3 Lean Startup.....	22
<b>1.2 Desenvolvimento de Projetos de Software</b> .....	<b>24</b>
1.2.1 Abordagem Tradicional .....	25
1.2.2 Abordagem Ágil.....	27
<b>1.3 Projetos Ágeis e Enxutos</b> .....	<b>32</b>
1.3.1 Software Ágil com Scrum .....	33
1.3.2 A Metodologia Ágil do SISP .....	48
1.3.3 Concepção Enxuta (Direto ao Ponto).....	54
<b>1.4 Estratégias de Contratação de Software</b> .....	<b>75</b>
1.4.1 Estratégias de Terceirização .....	76
1.4.2 Planejamento, Execução e Gestão de Contratos.....	84
1.4.3 Instrumentos.....	89
<b>1.5 Estimativas e Medição de Software</b> .....	<b>93</b>
1.5.1 Abordagem do IFPUG .....	95
1.5.2 Abordagem do COSMIC.....	110
<b>1.6 Escalando o Desempenho Organizacional</b> .....	<b>118</b>
1.6.1 Inovação em Camadas de Ritmo .....	118
1.6.2 Portfólios de Projetos (para o SISP).....	120
1.6.3 Avaliação de Desempenho.....	126

<b>3 MÉTODOS E PROCEDIMENTOS .....</b>	<b>129</b>
3.1 Delineamento da Pesquisa .....	129
3.2 Unidade-Caso e Sujeitos da Pesquisa.....	130
3.3 Coleta de Dados .....	133
3.4 Análise de Dados .....	137
3.4 Análise de Documentos.....	138
3.5 Limitações do Método.....	139
<b>4 DESENVOLVIMENTO E RESULTADOS .....</b>	<b>140</b>
<b>4.1 Perfil da Organização.....</b>	<b>140</b>
4.1.1 Departamento de Negócio.....	140
4.1.1.1 Áreas de Negócio.....	141
4.1.1.2 Escritório de Projetos Corporativo.....	141
4.1.2 Departamento de TI .....	141
4.1.2.1 Áreas de Projetos de Software.....	141
4.1.2.2 Áreas de Sistemas .....	142
4.1.2.3 Sustentação de TI .....	142
4.1.2.4 Qualidade de TI.....	142
4.1.2.5 Estratégia de TI e Arquitetura.....	143
4.1.2.6 Infraestrutura e Serviços de TI .....	143
4.1.2.7 Outsourcing de TI.....	143
<b>4.2 Estudo de Caso: Apresentação e Análise.....</b>	<b>144</b>
4.2.1 Processo de Portfólio de Projetos .....	144
4.2.2 Processo de Gerenciamento de Projetos.....	146
4.2.3 Processo de Desenvolvimento de Software .....	148
4.2.6 Processo de Homologação de Software .....	151
4.2.7 Processo de Estimativas de Software .....	152
4.2.6 Processo de Contratação de Fornecedores .....	156
4.2.7 Processo de Contratações de Software .....	157
4.2.8 Processo de Gestão de Fornecedores.....	157
4.2.9 Aspectos Gerais da Organização.....	159
4.2.9.1 Pessoas e Cultura .....	159
4.2.9.2 Processos e Boas Práticas.....	160

<i>4.2.9.3 Arquitetura e Tecnologias</i> .....	161
<i>4.2.9.4 Ferramentas Internas</i> .....	162
<b>5 CONSIDERAÇÕES FINAIS</b> .....	<b>164</b>
<b>5.1 Oportunidades de Melhoria</b> .....	<b>165</b>
<b>5.2 Limitações do Estudo</b> .....	<b>168</b>
<b>5.3 Desdobramentos Futuros</b> .....	<b>168</b>
<b>REFERÊNCIAS</b> .....	<b>169</b>

## 1 INTRODUÇÃO

Este estudo objetiva apresentar e propor melhorias para o modelo estratégico utilizado pela empresa FINA S/A, na gestão, contratação e desenvolvimento de software com base em métodos ágeis e tradicionais, frente a métodos de trabalho mais enxutos, atualmente no início de sua utilização, e engajados na abordagem de mensuração dos serviços com pontos de função.

Para cumprir com o objetivo citado, foram revistos conceitos e técnicas relacionadas ao *Design Thinking*, *Lean Thinking*, *Lean Thinking* e principalmente o *Lean Startup*, muito utilizados na indústria de manufatura tradicional, e que têm se mostrado também eficazes para orientar o desenvolvimento de projetos de software da empresa em estudo, tendo comprovado em sua experiência, que eles facilitam a descoberta, construção e entrega de produtos e serviços, com maior foco na agregação de valor ao negócio e na redução/ eliminação de desperdícios relacionados ao processo produtivo.

É neste contexto, que a utilização de métricas de software visa suportar e monitorar o processo de desenvolvimento de software, tonando a operação enxuta mais transparente, eficaz e eficiente:

- Transparente, devido a maior tangibilidade, previsibilidade e comparabilidade das contratações realizadas;
- Eficaz, devido à obtenção de resultados de forma condizente com a expectativa dos seus usuários, e também consistente durante a evolução do produto sendo desenvolvido; e
- Eficiente, pela capacidade de monitoramento e acompanhamento dos resultados com melhor uso dos recursos, ao tratar as fontes de desperdício.

O conjunto de informações abordado neste estudo é de certa forma, difícil de ser vislumbrado no mercado, principalmente no tocante às abordagens ágil e híbrida que serão apresentadas (que utilizam o método “Direto ao Ponto”), então há uma expectativa de que ele seja de grande utilidade principalmente aos profissionais atuantes no contexto de gestão de *outsourcing*, contratos e métricas de software

(em diferentes organizações), e que possa derivar outras pesquisas relacionadas de modo a enriquecer a contribuição do autor à comunidade de Engenharia de Software.

### **1.1 Definição do Problema**

Cada vez mais organizações têm optado pela utilização de métodos de desenvolvimento que privilegiam a entrega de valor, o mais cedo possível e com o menor consumo relativo de recursos, evitando o desperdício de tempo e dinheiro com tarefas e artefatos que não trazem benefícios reais ao negócio. Alinhados a estes objetivos, principalmente nos últimos anos, os chamados “métodos ágeis” de desenvolvimento de software têm se destacado na indústria de software. Entretanto, há a percepção de que existem muitas iniciativas envolvendo métodos ágeis que não têm obtido os resultados esperados, por inúmeras razões.

Embora ainda pouco exploradas no uso conjunto com a metodologia ágil, o estudo de caso apresentando neste trabalho, realizado com base na realidade da empresa FINA S/A, mostra que métricas de software podem auxiliar organizações que almejam um acompanhamento mais efetivo para viabilizar a gestão estratégica dos seus aspectos de tecnologia de informação. Além disso, mostra que é possível conviver com diferentes estratégias de contratação e desenvolvimento de software, mas que a tarefa é árdua e dispendiosa (em relação ao engajamento necessário).

Cabe ressaltar, que o Brasil é atualmente um dos países que mais utilizam métricas para suportar contratos de software, em especial a Análise de Pontos de Função (APF), sendo sua aplicação já bastante comum nas concorrências e contratos de pequenas, médias e grandes empresas dos setores público e privado, onde há inúmeros casos de implantações bem sucedidas, porém o mercado tem encontrado dificuldades em “casar” o modelo ágil com abordagens de aferição. Neste contexto, um ponto de função é normalmente valorado em moeda corrente, assim como o CUB (Custo Unitário Básico) o é no ramo de construção civil quando da precificação de unidades que devem ser construídas, facilitando desta forma, a realização de atividades de gestão e acompanhamento, e simplificando cálculos associados aos projetos.

O FINA S/A, a saber, é uma organização do ramo bancário/ financeiro, com atuação nacional e com mais de dois milhões de clientes. A empresa possui um portfólio de serviços direcionado a todos os tipos de pessoas que desejam ter um relacionamento bancário, porém com foco em moradores e empresários de zonas rurais, onde está a maior parte dos seus clientes e também onde se concentra sua estratégia de negócios atual. No entanto, a concorrência neste nicho tem aumentado e conseqüentemente a lucratividade tem diminuído e impulsionado uma busca por um maior espaço no mercado no que tange ao meio urbano.

Principalmente no meio urbano, como diagnóstico em uma pesquisa realizada internamente, a empresa FINA S/A constatou que seus concorrentes têm desenvolvido novos serviços com qualidade e velocidade superior à sua. Além disso, o custo de manutenção do portfólio de projetos tem crescido ano a ano, porém sem uma visibilidade de que o consumo de recursos realmente tem sido adequado à necessidade e realidade da organização, ou seja, não se sabe se os recursos foram utilizados com base no princípio da eficiência, com pouco desperdício.

Entendendo serem os pontos citados os principais motivadores para o baixo desempenho obtido, a organização reestruturou recentemente o seu ciclo de desenvolvimento com base em diferentes abordagens, onde se privilegia a metodologia ágil e enxuta, com entregas rápidas que agregam valor e são associadas ao conceito de MVP (*Minimum Viable Product*), oriundo do *Lean Startup*. Estas medidas compreendem compromissos estratégicos da organização, e têm contribuído para o lançamento mais rápido do software solicitado pelo negócio.

Desta forma, para suportar o crescimento esperado, melhorando a gestão e o processo de desenvolvimento de software, deverá tornar-se cada vez mais comum a utilização da metodologia ágil, o que remete a uma afirmação de GOMES (2013):

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

No entanto, a metodologia ágil vislumbrada, no entendimento dos *stakeholders*, consensualmente ainda não é madura o suficiente e não atende

plenamente a necessidade da empresa, que por este motivo busca um meio de conviver com métodos diferentes, dando-lhes um direcionamento em alto nível.

Desta forma, a proposta desta pesquisa é definida com o intuito de utilizar o conhecimento adquirido na empresa de realização do estudo de caso, sobre desenvolvimento de software enxuto e métricas de software, com foco na seguinte questão:

- Como identificar, na empresa FINA S/A, os requisitos e procedimentos necessários para evoluir os seus processos de gerenciamento, contratações e desenvolvimento de software, de forma enxuta e mensurável (com pontos de função)?

## 1.2 Objetivos

Os objetivos de uma pesquisa científica caracterizam um objeto a ser investigado, direcionam o trabalho do pesquisador e servem como uma espécie de linha de chegada para o trabalho proposto. Se a definição do problema é uma questão a investigar, o objetivo é um resultado a ser alcançado, logo um objetivo geral.

O objetivo geral, por sua vez, responde ao problema de pesquisa, sendo os objetivos específicos, as metas que devem ser atingidas para responder plenamente ao objetivo geral (SILVA, MACHADO, SACCOL & AZEVEDO, 2012).

### 1.2.1 Objetivo Geral

Analisar, no contexto da empresa FINA S/A, os requisitos e procedimentos necessários para melhorar os processos de gestão, contratação e desenvolvimento de software em uma perspectiva enxuta e mensurável (com pontos de função).

### 1.2.2 Objetivos Específicos

Visando alcançar o objetivo geral, foram estabelecidos os seguintes objetivos intermediários:

- a) Revisar a bibliográfica dos conceitos vislumbrados no escopo deste trabalho, tais como *Design Thinking*, *Lean Thinking*, *Lean Startup* e outros conceitos e práticas de mercado relacionadas ao desenvolvimento, contratação e gestão de projetos de software;
- b) Realizar na empresa do estudo de caso, um diagnóstico da situação atual das principais unidades de TI e áreas vinculadas;
- c) Mapear e descrever o processo e os modelos de desenvolvimento de software atual da empresa do estudo de caso;
- d) Identificar os principais processos atuais de planejamento, gerenciamento e contratação, em torno do processo de desenvolvimento de software;
- e) Apresentar propostas para melhoria dos processos analisados frente aos princípios enxutos e utilizando estimativas por pontos de função, visualizando a possível aderência à bibliografia pesquisada.

### 1.3 Justificativa

O autor deste trabalho atualmente labora na instituição financeira que configura o objeto de estudo desta pesquisa, sendo também especialista em métricas de software certificado CFPS (*Certified Function Point Specialist*) e CSP (*Certified SNAP Practitioner*) pelo órgão IFPUG (*International Function Point Users Group*), do qual é membro. O mesmo é membro também, do Comitê Consultivo Internacional (*International Advisory Council*, em inglês) do órgão COSMIC (*Common Software Measurement International Consortium*); e durante suas atividades, identificou a necessidade de evoluir e compartilhar o entendimento sobre o assunto em questão, nas comunidades de interesse relacionado à “Desenvolvimento de Software” e “Estratégia/ Gestão de TI”, passando inicialmente pelo entendimento completo do contexto onde está inserido.

A expectativa do autor é que os resultados obtidos, tais como as propostas de melhoria oriundas deste estudo, sejam suficientemente claras e úteis para que a organização possa decidir por dar-lhes andamento.

## 2 REVISÃO BIBLIOGRÁFICA

A seguir serão apresentados os fundamentos que sustentam a elaboração da presente pesquisa, assim como a relação de conceitos relevantes ao completo entendimento e contextualização do estudo de caso que será apresentado.

Ressalta-se, que o estudo abordará tópicos inerentes às estratégias de colaboração e cocriação, assim como os fundamentos do “pensamento enxuto”. Em seguida serão também vislumbrados os conceitos relacionados a métodos ágeis e métricas de software. Sendo que, ao final do capítulo, serão apresentadas as teorias e práticas relacionadas à aquisição de software, com base em um processo estruturado de gestão de fornecedores e portfólio de projetos.

### 1.1 Cocriação & Pensamento Enxuto

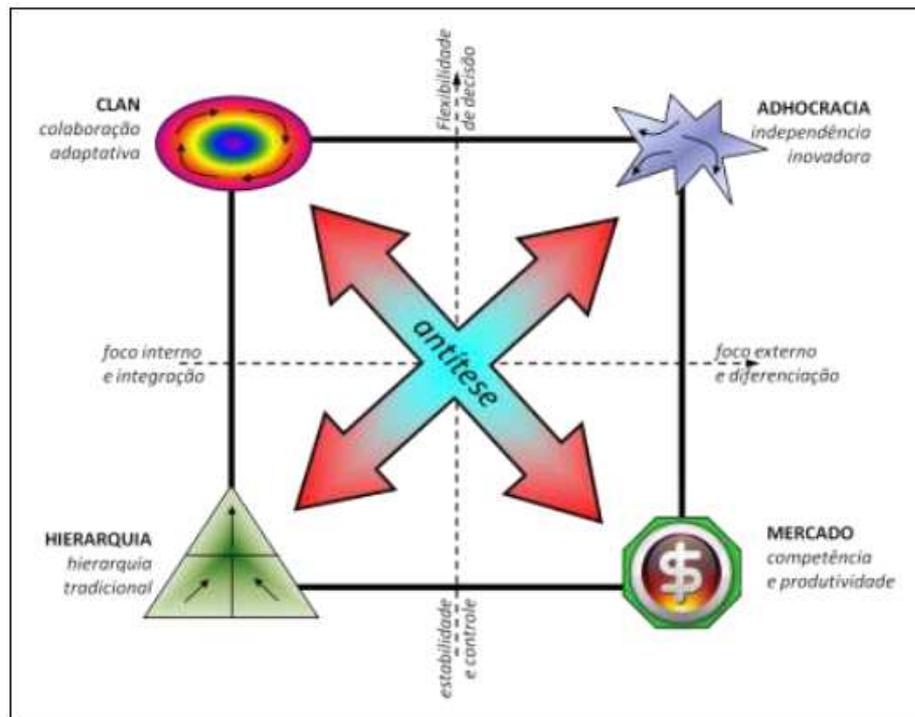
Segundo PRATES (2015), as práticas participativas para o desenvolvimento de novos negócios e produtos ampliam fronteiras sobre como pensamos nas escolhas das soluções mais inovadoras, o que proporciona transformar ideias inéditas em novos negócios.

Todavia, conforme BORJA (2011, Pág. 43), a cultura organizacional liga-se intimamente à maneira como as decisões são tomadas, sendo também um dos principais fatores para viabilizar a tomada de seus benefícios.

Desta forma, de acordo com o Mapa de Antítese (na **Figura 1**), as decisões ou são tomadas pelas posições mais altas em detrimento da opinião daqueles que efetivamente trabalham na operação (devido à colaboração ou independência) ou são delegadas, de forma que todos possam contribuir com o direcionamento estratégico (pela hierarquia ou competências). E é com base nesta afirmação, que serão abordados a seguir os conceitos relacionados a algumas práticas de cocriação para o desenvolvimento de produtos enxutos, todos fortemente estruturados em meios de trabalho colaborativos, criativos e participativos.

#### **Figura 1 – Antítese e a Cultura Organizacional**

Classificação da Informação: **Uso Irrestrito**



Fonte: BORIA (2013, Pág. 43)

### 1.1.1 Design Thinking

O “pensamento do desenho” ou *Design Thinking* (em inglês), é um processo colaborativo de pensamento crítico e criativo que permite organizar informações e ideias, para tomar decisões, aprimorar situações e adquirir conhecimento (VIANNA, 2012), ou seja, ele é um método que trabalha essencialmente a cocriação.

Como filosofia, o *Design Thinking* tem também o objetivo de promover o bem-estar na vida das pessoas com base no comportamento do consumidor de um produto, serviço ou informação, considerando suas experiências (PRATES, 2015). A sua utilização abrange diferentes ramos da indústria, podendo ser utilizado como uma ferramenta estratégica que fomenta a criação conjunta a partir de uma visão por processos (**Figura 2**).

O processo de *Design Thinking* é definido conforme as seguintes fases (VIANNA, 2012):

- Imersão (entendimento) é a fase que envolve a contextualização do projeto, a partir de três etapas (sendo a última opcional):
  - “Imersão Preliminar”, etapa ou subdivisão da fase que visa o entendimento inicial do problema em foco (e caso necessário, o seu reenquadramento), podendo também ser executada uma “Pesquisa Exploratória” ou “Pesquisa *desk*”;
  - “Imersão em Profundidade”, que consiste em uma etapa onde se identificam necessidades dos *stakeholders* e prováveis oportunidades que possam emergir do entendimento de suas experiências, utilizando técnicas tais como “Entrevistas”, “Cadernos de Sensibilização”, “Sessões Generativas”, técnica “Um Dia na Vida” e técnica “Sombra”;
  - “Análise e Síntese”, uma consolidação visual do conteúdo identificando padrões de contexto e estabelecendo ferramentas de apoio, tais como: “Cartões de *Insights*”, “Diagrama de Afinidades”, “Mapa Conceitual”, definição de critérios norteadores, identificação de “Personas”, “Mapa de Empatia”, “Jornada do Usuário” e “*Blueprint*”.
- Ideação (criação) é a fase que tem o intuito de gerar ideias inovadoras para o tema de projeto, utilizando-se de ferramentas de síntese aplicadas às informações oriundas da etapa de “Análise e Síntese”, estimulando a criatividade e gerando soluções conforme o contexto trabalhado – são algumas das ferramentas utilizadas: “*Brainstorm*”, “*Workshop* de Cocriação”, “Cardápio de Ideias” e “Matriz de Posicionamento”.
- Prototipação (teste) é a fase que tem como função validar hipóteses a partir da formulação de questões, criação de protótipos, testes, avaliações e conclusão, podendo ocorrer concomitantemente com as fases de “Imersão e Ideação”, o que proporciona selecionar e refinar ideias, tangibilizando-as e avaliando-as e validando-as interativamente junto a uma amostra do público para antecipar a visibilidade sobre eventuais gargalos e problemas, o que ajuda a reduzir riscos e otimizar custos – a

prototipação normalmente utiliza-se de recursos rudimentares, tais como “Protótipos de Papel”, “Modelos de Volume”, “Encenação” ou “*Storyboards*”, “Protótipos de Serviços”, entre outros.

- Desenvolvimento (aplicação) é o momento dedicado ao uso do conhecimento adquirido por meio do processo criativo, permitindo transformar as soluções identificadas em negócios, estratégias e/ou oportunidades concretas, ou seja, é o ponto em que o processo de *Design Thinking* foi finalizado de forma com que algo seja passível de ser desenvolvido e disponibilizado na produção.

**Figura 2 – Design Thinking como um Processo**



Fonte: autor desconhecido.

### 1.1.2 Lean Thinking

A indústria automobilística foi o berço da filosofia *Lean*, difundindo mais tarde este conceito para outros setores da economia.

Também conhecido como *Lean Manufacturing*, ou ainda “Sistema Toyota de Produção”, o *Lean* remete à obtenção dos materiais corretos, no local correto, na quantidade correta e com o mínimo desperdício, para manter a flexibilidade e a

adaptabilidade do sistema produtivo. Neste contexto, podem ser destacados como pontos chave do *Lean Manufacturing*:

- Qualidade total imediata – atuação na detecção e eliminação de defeitos, com a solução dos problemas na sua origem;
- Minimização do desperdício – eliminação de tarefas que não acrescentam valor para o negócio e/ou levam à insatisfação do cliente e demais *stakeholders*, isto é, são aquelas tarefas que possuem relação com recursos usados indevidamente e que contribuem para o aumento de custos e tempo;
- Melhoria contínua – busca contínua por redução de custos, melhoria da qualidade, aumento da produtividade e compartilhamento da informação;
- Processos "pull" – os produtos devem ser retirados pelo cliente final, e não empurrados para o fim da cadeia de produção;
- Flexibilidade – produção rápida de diferentes lotes e grande variedade de produtos, sem comprometimento da eficiência que é esperada do processo;
- Relação de longo prazo com os fornecedores – tomam-se acordos para compartilhar o risco, custos e a informação.

O termo “Pensamento Enxuto”, em inglês “*Lean Thinking*”, atualmente em voga, é derivado do conceito “*Lean Manufacturing*”, sendo estes muitas vezes citados como sinônimos. Ele foi usado pela primeira vez por James Womack e Daniel Jones, na obra de mesmo nome (WOMACK, 2004), e desde então é utilizado para referir-se à filosofia de liderança e gestão que tem como objetivo a eliminação do desperdício e a criação de valor.

As características centrais do *Lean Thinking* são elencadas abaixo:

- Organização baseada em equipes envolvendo pessoas flexíveis, com múltipla formação, com elevada autonomia e responsabilidade nas suas áreas de trabalho;

- Estruturas de resolução de problemas ao nível das áreas de trabalho, em sintonia com uma cultura de melhoria contínua;
- Operações Lean, o que leva os problemas a revelarem-se, e a serem posteriormente corrigidos;
- Políticas de recursos humanos voltadas aos valores e a capacidade de comprometimento, encorajando sentimentos de pertença, partilha e de dignidade;
- Relações de grande proximidade com fornecedores;
- Equipes de desenvolvimento multifuncionais;
- Grande proximidade com o cliente.

Dentre os princípios do “pensamento enxuto” podem ser destacados, portanto: a eliminação de desperdícios; o respeito às pessoas; a busca pela qualidade; a busca pela simplicidade; o aperfeiçoamento do todo e entregas rápidas (POPPENDIECK; POPPENDIECK, 2011). Ressalta-se que todos os princípios citados remetem bem à ideologia que suporta o referido modelo de gestão.

### 1.1.3 Lean Startup

Conceito derivado da filosofia *Lean* e bastante utilizado na indústria tradicional e também na indústria de software, o *Lean Startup*, ou “Startup Enxuta” em português, é uma abordagem que proporciona a idealização de produtos e serviços com foco na geração de aprendizado sobre mercado, produto e cliente, a partir da validação de hipóteses.

No *Lean Startup*, as hipóteses são formadas por crenças construídas com base na experiência e observações dos envolvidos, e que devem ser validadas para serem transformadas em fatos (SABBAGH, 2013).

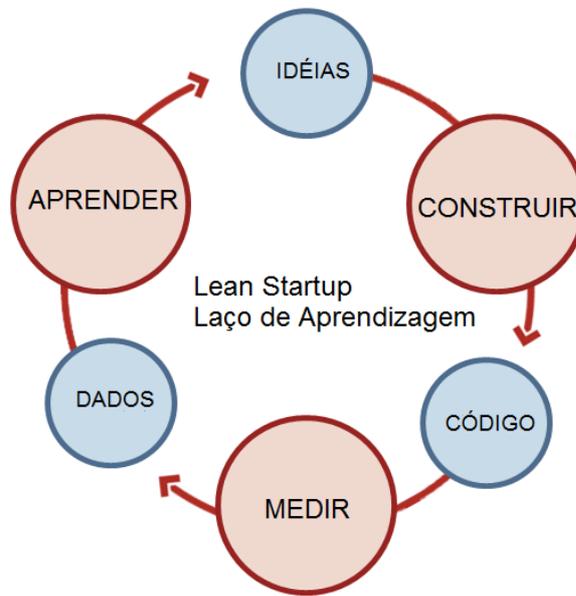
O método promove um novo jeito de pensar e de construir produtos inovadores para negócios sustentáveis, a partir da identificação e eliminação

sistemática de desperdícios e de uma combinação de ideias relacionadas a marketing, tecnologia e gestão (RIES, 2012). Estas características, no entanto, residem em cinco princípios básicos:

- Empreendedores estão em toda parte – aplicação independente de especificidades ou estereótipos, especialmente voltada cenários de extrema incerteza;
- Empreendedorismo é administração – gestão focada na extrema incerteza, como uma instituição a ser gerida;
- Aprendizagem validada – busca pela validação científica dos experimentos, podendo testar cada elemento do modelo de negócios;
- Construir-medir-aprender – foco na transformação de ideias em produtos, medindo como os consumidores reagem e decidindo sobre mudar ou perseverar;
- Contabilidade para inovação – busca por melhores resultados e atribuição de responsabilidade aos empreendedores, atentando para a medição de progresso, definição de marcos e priorização de trabalho.

Para a construção de produtos de software, o *Lean Startup*, segundo SABBAGH (2013), mostra-se muito útil, pois considera um laço de aprendizagem com ênfase em etapas de aprendizado, construção e medição (**Figura 3**), permitindo assim validar hipóteses, potencializadas através do conceito de produto mínimo viável (MVP, do inglês: *Minimum Viable Product*), ou seja: a versão mais simples de um produto que pode ser lançada com uma quantidade mínima de esforço e tempo de desenvolvimento, em um processo de aprendizagem contínuo e que possibilita o teste de hipóteses fundamentais ao negócio (RIES, 2011, págs. 77 e 93).

### **Figura 3 – Laço de Aprendizagem do Lean Startup**



Fonte: autor desconhecido.

## 1.2 Desenvolvimento de Projetos de Software

ARCHIBALD (2003) desenvolveu um amplo estudo de categorização de projetos, visando aperfeiçoar as características de cada categoria, conforme a relação abaixo:

- Projetos de Defesa, Segurança e Aeroespacial;
- Projetos de mudanças organizacionais e de melhorias em resultados operacionais e financeiros;
- Projetos de Sistemas de Comunicações;
- Projetos de Eventos;
- Projetos de Construção e Montagem (Empreendimentos, Investimentos, Construção, Demolição, Manutenção e Modificação, *Design/ Procurement/ Construction*);
- Projetos de Sistemas de Informação (software);
- Projetos de Desenvolvimento Regional/ Internacional;
- Projetos de Mídia e Entretenimento;

- Projetos de Desenvolvimento de Novos Produtos e Serviços;
- Projetos de Pesquisa e Desenvolvimento.

Com base na categorização demonstrada, este estudo mostra interesse apenas pelos projetos da categoria “Projetos de Sistemas de Informação”, ou seja, àqueles que contemplam um conjunto de atividades relacionadas ao desenvolvimento e manutenção de software, cenário em que não estão inseridas as necessidades de desenvolvimento e manutenção de produtos de equipamentos de informática (hardware).

Ressalta-se ainda, que os projetos de sistemas de informação podem ser desenvolvidos de diversas maneiras, mas que o estudo aborda apenas duas de suas abordagens principais: o modelo tradicional de desenvolvimento de software e o modelo ágil de desenvolvimento de software.

### 1.2.1 Abordagem Tradicional

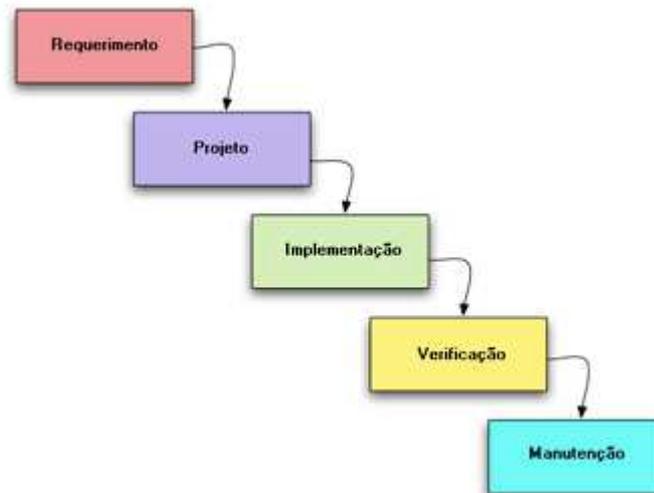
Os projetos de software no modelo tradicional de desenvolvimento de software englobam paradigmas derivados da abordagem conhecida como *waterfall* (**Figura 4**), ou “cascata” em português, tais como os seus conceitos relacionados (SABBAGH, 2013). Inicialmente descrita por ROYCE (1970), este método segue uma linha determinística<sup>1</sup> para o desenvolvimento de produtos e é tipicamente sequencial e composto por um conjunto de atividades fortemente dependentes (PRESSMAN 2006), similar ao cenário encontrado nas linhas de produção de fábricas de manufatura. Há desta forma, pouca adaptabilidade e baixa propensão a mudanças.

---

<sup>1</sup> Segundo POPPENDIECK (2011, pág. 45), o processo de transformação em ideias em produtos, quando segue a linha proposta pela escola de pensamento determinístico, começa criando uma definição completa do produto e então cria uma realização a partir desta definição. A escola de pensamento empírico, no entanto, começa com um conceito de produto em alto nível e então estabelece ciclos de *feedback* bem definidos, que ajustam as atividades para criar uma interpretação ótima do conceito.

No método cascata, os papéis, artefatos e responsabilidades dos profissionais são bastante claros e as etapas são bem definidas, durante todo o processo de desenvolvimento de software (SABBAGH, 2013, pág. 19).

**Figura 4 – Etapas do Desenvolvimento *Waterfall***



*Fonte: ROYCE (1970).*

Todavia, ROYCE (1970) criticava a aplicação do método “cascata” para atividades de desenvolvimento de software, afirmando que seu uso era arriscado neste cenário, uma visão compartilhada por SABBAGH (2013). Ele considera que os métodos tradicionais são fortemente prescritivos e de forma geral, caracterizados pelo foco em planos detalhados definidos no início do projeto, preocupando-se com o custo, escopo e um cronograma detalhado, com micro gerenciamento e poder centralizado, em uma estrutura de processos complicados e prevendo extensa documentação.

POPPENDIECK (2011, pág. 45) também defende que o software, por sua própria natureza, deveria ser projetado para se adaptar às mudanças durante todo o seu ciclo de vida – e que por isso, o processo de desenvolvimento não deveria ser determinístico (como nos métodos tradicionais), mas empírico, como na metodologia ágil, que será introduzida no tópico subsequente.

## 1.2.2 Abordagem Ágil

Devido a crescente complexidade e necessidade do mercado de inovar para conceber determinados tipos de produtos, no ano de 2001, surgiu um movimento na comunidade de desenvolvimento de software que culminou com a publicação de importantes críticas às práticas de desenvolvimento e gerenciamento de projetos software, até então essencialmente executados no modelo tradicional.

A referida publicação foi realizada por um grupo de desenvolvedores e engenheiros de software, com a justificativa de que o modelo cascata já não mais condizia com a realidade das empresas. E este apanhado ficou conhecido como “Manifesto Ágil” (BECK et al, 2001).

A proposta ágil foi definida com base em quatro direcionadores principais:

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

Entretanto, além dos direcionadores citados, o Manifesto Ágil (BECK et al, 2001) conta ainda com 12 princípios, os quais são relacionados abaixo:

- Priorizar a satisfação do cliente através da entrega contínua e adiantada, com valor agregado;
- Permitir mudanças nos requisitos, mesmo tardiamente, visando vantagem competitiva ao cliente;
- Entregar frequentemente software funcionando, preferindo a menor escala de tempo possível;
- Unir as pessoas do negócio e os desenvolvedores diariamente no trabalho despendido durante todo o projeto;
- Construir projetos em torno de indivíduos motivados, com ambiente, suporte e confiança necessária para executar o trabalho;
- Buscar o relacionamento direto entre as pessoas;
- Primar pelo software funcionando como medida primária de progresso;

- Manter um ritmo constante, indefinidamente e de forma sustentável;
- Aumentar a agilidade através de contínua atenção, excelência técnica e bom *design*;
- Buscar a simplicidade, maximizando a quantidade de trabalho não realizado;
- Conceber equipes auto organizáveis, o que leva à emergir melhores arquiteturas, requisitos e designs;
- Possibilitar que a equipe reflita sobre sua eficácia, em intervalos regulares, refinando e ajustando seu comportamento conforme necessário.

Reforçando a importância da abordagem ágil e derivando seus princípios, HIGHSMITH (2004) sugeriu a aplicação de técnicas de gerenciamento ágil em projetos que tratam de novos produtos, enfatizando que as empresas precisam desenvolver uma cultura que promova a adaptabilidade para absorver mudanças, com direcionadores bem definidos para encorajar os aspectos relacionados à auto-organização, de forma combinada à capacidade necessário do time para a autogestão, colaboração e interação dentre os seus membros.

Salienta-se, que métodos ágeis são fortemente estruturados sob aspectos de colaboração com o cliente e em métodos de trabalho definidos à luz dos valores e princípios do “Manifesto Ágil” (BECK et al, 2001). Senso assim, são suas premissas:

- O foco na agregação de valor ao negócio; e
- A valorização das pessoas envolvidas no processo produtivo, das quais, por sua vez, espera-se comprometimento e capacidade de autogestão, para que possam colaborar entre si e alcançar os objetivos comuns mais facilmente.

Contudo, na interpretação de POPPENDIECK (2011, pág. 44), as práticas enxutas de produção e de gerenciamento da cadeia de suprimentos não são facilmente “traduzidas” para o desenvolvimento de software, pois tanto o software como o desenvolvimento são bastante desformes em relação ao contexto de operações e logísticas. E exatamente por este motivo, o presente estudo tratará algumas adaptações sugeridas por ele.

### 1.2.2.1 *Extreme Programming*

A metodologia ágil de desenvolvimento chamada “*Extreme Programming*” (XP) foi criada por Kent Beck nos anos 1990 e cobre diversos aspectos técnicos do desenvolvimento de software, tais como codificação, design e testes (GOMES, 2013). FRANCO (2007) complementa esta visão, afirmando que o XP surgiu como uma tentativa de solução para os problemas causados pelos longos e morosos ciclos de desenvolvimento dos modelos de desenvolvimento tradicionais e que por isso consolidou-se no mercado.

O XP é então considerado uma proposta de desenvolvimento de software simples, que visa entregar o que o cliente solicitou, no prazo acordado com ele. Ele define uma equipe única de trabalho, ao contrário do Scrum que preconiza um time de desenvolvimento a parte dos demais envolvidos no projeto. Ele define ainda, que gerentes, clientes e desenvolvedores devem também formar uma equipe única dedicada, destinada a entregar software com mais qualidade (AUDY e PRIKLANDNICKI, 2007).

Segundo HIGHSMITH (2002), o XP foca em interações de cliente e desenvolvedor que devem ocorrer em um ambiente apropriado para este fim: times dividindo o mesmo espaço físico, compostos por dez ou menos desenvolvedores e com um cliente no local, dedicado integralmente ao projeto. Como outras metodologias ágeis, o desenvolvimento com XP ocorre com base em iterações curtas (de três semanas ou menos).

### 1.2.2.2 *Desenvolvimento de Software Enxuto*

POPPENDIECK (2011) consolida uma ampla variedade de teorias e experiências “traduzindo” o *Lean* da indústria de manufatura (*Lean Manufacturing*) para a sua aplicação no contexto de desenvolvimento de software.

Segundo POPPENDIECK (2011), os problemas fundamentais de produção são também problemas que ocorrem com desenvolvimento de software, tais como:

- Lidar com incerteza e mudança;
- Melhorar continuamente os processos; e
- Entregar valor aos clientes.

O autor defende também, que a conversão das teorias enxutas para o desenvolvimento de software impacta em sete princípios e sete desperdícios, sendo demonstrados abaixo os “Sete Princípios”:

- Eliminar o desperdício;
- Integrar qualidade;
- Criar conhecimento;
- Adiar comprometerimentos;
- Entregar rápido;
- Respeitar as pessoas;
- Otimizar o todo.

A perspectiva dos “Sete Desperdícios” no desenvolvimento de software, por sua vez, identifica os seguintes pontos a serem atentados e trabalhados (POPPENDIECK, 2011, pág. 93):

- Trabalho inacabado – o estoque do desenvolvimento de software é o trabalho inacabado;
- Funcionalidades extras – considerado o pior item desta lista, é a ação de adicionar funcionalidades que não são realmente necessárias ao cliente e são adicionadas no escopo de desenvolvimento;
- Reaprendizagem – desperdício de conhecimento devido a ignorar a participação, contribuição e engajamento das pessoas;
- Transferência de controle – evitar realizar transferência do controle do trabalho;
- Troca de tarefas – evitar trocar tarefas;
- Atrasos;
- Defeitos.

POPPENDIECK (2011, págs. 82, 83, 87 e 89, 91) define ainda uma série de questões que devem ser levadas em conta ao se desenvolver software, tais como as citadas abaixo:

- Projetos e produtos são coisas diferentes, visto que projetos possuem início, meio e fim e produtos possuem início, mas não necessariamente um fim;
- Evitar a complexidade, pois esta é uma das principais aliadas à dívida técnica e do desperdício – cujo custo associado é exponencial;
- Optar por times perenes, embora exista a tendência de constituir uma nova equipe de desenvolvimento a cada novo projeto;
- Enxergar-se como uma empresa de software, tendo o negócio como seu cliente;
- Cada funcionalidade desenvolvida deveria passar por uma prova de valor, justificando o porquê da sua existência para enxergar o real valor para o negócio;
- Buscar dividir o software em conjuntos mínimos de funcionalidades úteis, modo como o negócio passa a utilizar o software mais depressa;
- Não automatizar a complexidade, primeiro o processo candidato à automação deve ser esclarecido e simplificado.

### 1.2.2.3 DevOps e Entrega Contínua

Completamente aderente ao processo de desenvolvimento ágil e enxuto, SATO (2013) define *DevOps* como um movimento cultural e profissional que está tentando quebrar barreiras para prover a entrega de software em produção. A utilização das suas práticas têm se tornado cada vez mais comum, pois possibilitam entregar software em produção com maior frequência e com maior estabilidade e robustez – o que conforme SATO (2013), muitas empresas de sucesso na internet vêm fazendo – como Google, Amazon, Netflix, Flickr, GitHub e Facebook.

Conforme SATO (2013), apenas nos últimos anos as empresas vem percebendo que a tecnologia pode ser usada a seu favor e que o atraso em um *deploy* para produção significa muitas vezes atrasar sua habilidade de competir e se adaptar a mudanças no mercado; sendo hoje comum organizações realizarem por vezes dezenas ou até centenas de *deploys* por dia – uma abordagem conhecida como “Entrega Contínua” e faz parte de uma estratégia de *DevOps* moderna e competitiva.

### 1.3 Projetos Ágeis e Enxutos

HIGHSMITH (2004) define o Gerenciamento de Projetos Ágeis, ou em inglês, *Agile Project Management (APM)*, como um conjunto de valores, princípios e práticas que auxiliam a equipe de projeto a entregar produtos ou serviços de valor em um ambiente complexo, instável e desafiador – onde os valores e princípios referem-se aos conceitos e as práticas voltadas para o “como” realizá-lo.

O APM é simples, flexível e iterativo, e possui grande capacidade de adaptação às práticas de gerenciamento existentes, podendo ser aplicado em ambientes dinâmicos cujas especificações são regidas pela inovação, em complemento às teorias ditas “tradicionais” (AMARAL, 2011, págs. 2, 3 e 8).

AUGUSTINE (2005) afirma que o gerenciamento ágil de projetos deve ser constituído pela energização, capacitação e habilitação da equipe envolvida no projeto, para prover entregas mais rápidas e com maior valor para o negócio, fazendo isso através da integração dos clientes, num processo de aprendizagem contínua, com adaptabilidade a mudanças e de acordo com suas necessidades e ambiente.

Como abordado anteriormente, os Projetos de Sistemas de Informação, uma categorização conferida por ARCHIBALD (2003), possuem características específicas; e em um contexto ágil, estas características também devem ser levadas em conta, pois tendem a ser por vezes específicas, e com algum tratamento diferenciado através de métodos de gestão e execução complementares – alguns

destes serão apresentados a seguir (em especial, referindo-se aos métodos Scrum e Direto ao Ponto).

### 1.3.1 Software Ágil com Scrum

O SCRUM (2013, p. 31) é um *framework* de gestão de projetos ágeis, que vem sendo utilizado desde a década de 1990, principalmente na construção de produtos complexos. O seu papel (SCRUM, 2013, p. 31) é fazer transparecer a eficácia relativa das suas práticas, de modo que elas possam ser melhoradas por quem as utiliza, agregando outros processos e técnicas com naturalidade. E ele faz isso através de uma abordagem de desenvolvimento iterativa e incremental, com o intuito de alavancar a previsibilidade e reduzir os riscos associados (SCHWABER, 2009, p. 3).

A saber, o *framework* do Scrum é composto por um conjunto de recursos, formado por (SCHWABER, 2009, p. 4): times Scrum e seus papéis associados (Scrum Master, *Product Owner* e Time), eventos com duração fixa (*time boxes*), artefatos e regras. Ele é constituído também por três pilares principais (SCRUM, 2013), que sustentam processos empíricos:

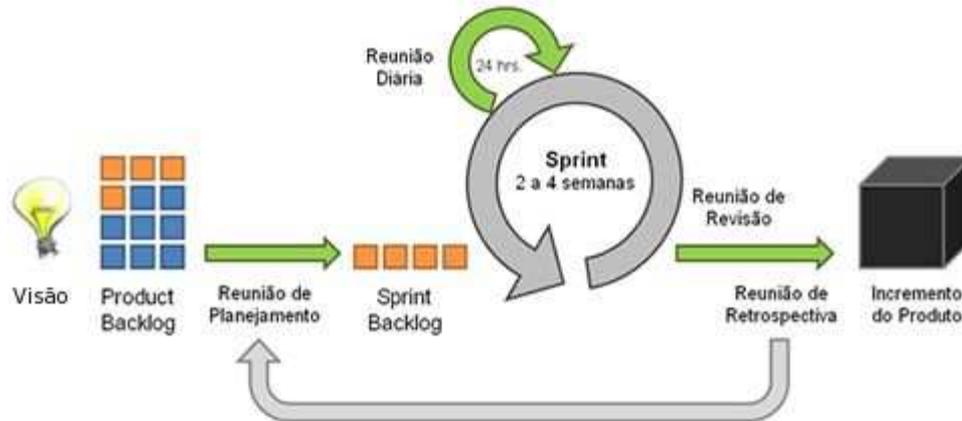
- Transparência – os resultados devem ser visíveis para aqueles que gerenciam os resultados;
- Inspeção – aspectos do processo devem ser revisitados, visando mitigar a existência de desvios que não são seriam aceitáveis;
- Adaptação – se um ajuste se fizer necessário, deve sê-lo feito o mais rápido possível, visando minimizar desvios posteriores.

Destaca-se que, para os eventos de duração fixa, o método emprega o conceito de *Sprint*, que é considerada a sua principal característica. O termo “*Sprint*” (SCHWABER, 2009, p. 10) designa uma iteração de curta duração (um mês ou menos), cujo esforço é direcionado ao desenvolvimento de um pedaço de software, com base em metas previamente estabelecidas e buscando gerar um incremento

potencialmente entregável. E este incremento, a critério do usuário, poderá ou não ser disponibilizado em produção.

Desta forma, um projeto de software utilizando Scrum pode conter diversas *Sprints*, as quais devem ser sequencialmente executadas como miniciclos de desenvolvimento (vide **Figura 5**).

**Figura 5 – Ciclo de Desenvolvimento Scrum**



*Fonte: autor desconhecido.*

Segundo SABBAGH (2013), o Scrum atualmente é a forma mais comum de se trabalhar em projetos de software, tendo sido adotado por organizações de diversos tamanho e tipos, sejam multinacionais, *startups*, empresas privadas ou públicas. Ele ressalta ainda, que ao utilizar Scrum, existem termos que frequentemente serão citados e que dizem muito sobre a natureza do método, tais como: facilitação e trabalho em equipe, auto-organização, metas de negócio, motivação, relacionamento com os clientes – e que alguns dos principais benefícios do método são os que seguem (SABBAGH, 2013):

- Entregas frequentes de retorno ao investimento dos clientes – partes do produto funcionando podem ser entregues desde cedo e frequentemente, através de ciclos curtos de desenvolvimento, o que reduz o prazo de introdução do produto no mercado;
- Redução dos riscos do projeto – há *feedback* constante dos clientes e demais partes interessadas, estabelecendo uma melhor relação entre o que foi solicitado e aquilo que efetivamente foi recebido como produto,

embora também seja importante a presença do time multidisciplinar, para redução de dependências;

- Maior qualidade do produto gerado – o próprio time de desenvolvimento é encarregado de garantir a qualidade do produto entregue e de validá-lo junto ao requisitante (como isso ocorre com maior frequência, diminui-se o risco de problemas acumularem-se as entregas);
- Mudanças utilizadas como vantagem competitiva – na verdade são vistas como oportunidades e não como empecilhos, como no método tradicional, muitas vezes, são até mesmo uma consequência do ciclo de aprendizado;
- Visibilidade do progresso do projeto: o conjunto de práticas e artefatos do Scrum visa garantir a visibilidade e transparência do progresso do projeto para seus participantes e envolvidos, além de prover uma sensação de progresso perante os clientes, que recebe frequentemente partes do produto funcionando;
- Redução do desperdício – o desperdício é reduzido e evitado na busca pela simplicidade, já que o time que usa Scrum tende a produzir e utilizar apenas o que é necessário e suficiente (produzindo apenas o que os usuários irão utilizar, planejando apenas com o nível de detalhe possível e utilizando apenas os artefatos necessários e suficientes);
- Aumento de produtividade – diversos fatores potencializam a produtividade de times que utilizam Scrum, dentre eles: o trabalho da equipe e a autonomia do time na realização desse trabalho, a existência de facilitação e de remoção de impedimentos, a melhoria contínua dos processos de trabalho, além da busca incessante por um ritmo sustentável de trabalho e maior motivação do time.

#### *1.3.1.1 Papéis do Scrum*

Segundo SABBAGH (2013), existem quatro papéis básicos vinculados ao Scrum; são eles:

- Time de Scrum:

Classificação da Informação: **Uso Irrestrito**

- Product Owner (PO);
- Scrum Master (SM); e
- Time de Desenvolvimento.

Embora a nomenclatura assemelhe-se, é importante destacar que enquanto o Time de Scrum contempla todas as pessoas que integram o projeto de software, o Time de Desenvolvimento realiza o trabalho de desenvolvimento do produto propriamente dito, possuindo todo o conhecimento necessário para realizá-lo.

O Product Owner (PO), por sua vez, é um papel único, que se responsabiliza por definir, comunicar e manter a Visão do Produto relativamente constante ao longo do projeto. É fundamental que ele aja como um representante do cliente perante o Time de Desenvolvimento e Scrum Master, papel o qual é caracterizado pela sua neutralidade e pela atuação na remoção dos impedimentos, tal como na difusão do conhecimento relacionado ao Scrum.

No Scrum, como por padrão não existe um gerente de projetos, um papel tradicional, o trabalho de gerir o projeto é distribuído pelos seus papéis, sendo definidos pontos chave que cabem a cada um destes papéis (**Quadro 1**).

Conforme SABBAGH (2013), com a divisão, na prática, o Time de Scrum compartilha a responsabilidade sobre o escopo, custo, tempo, qualidade, risco e gestão do trabalho de desenvolvimento do produto, ou seja, os seus integrantes trabalham de forma colaborativa e são igualmente comprometidos com a satisfação dos clientes do projeto (SABBAGH, 2013, págs. 107 e 108).

### **Quadro 1 – Pontos-chave do Scrum e papéis relacionados**

Ponto-chave a ser gerenciado	Product Owner	Time de Desenvolvimento	ScrumMaster
Retorno sobre o investimento	X		
Necessidades/objetivos de negócios	X		
Clientes e demais partes interessadas	X		
Visão do Produto	X		
Releases	X		
Tarefas de desenvolvimento do produto		X	
Qualidade interna do produto		X	
Qualidade externa do produto	X	X	
Estimativas ou previsões		X	
Processos (funcionamento do Scrum)			X
Impedimentos no trabalho			X
Relacionamento e motivação do Time		X	X
Riscos	X	X	X
Comunicação	X	X	X

Fonte: SABBAGH (2013, pág. 108).

### 1.3.1.2 Backlog do Produto

O Scrum é iniciado com a criação do “*Backlog do Produto*”, uma lista priorizada de itens que são derivados de requisitos, melhorias e correções que devem ser feitas no produto (SCHWABER; SUTHERLAND, 2013).

Os itens do *Backlog* representam uma “Visão do Produto”, sendo dinâmicos e incompletos, e ordenados com o propósito de maximizar o retorno ao investimento dos clientes, já que aqueles itens menores, mais detalhados e com maior importância para o negócio ficam no topo da lista.

A “Visão de Produto”, cujo template sugerido pode ser encontrado na **Figura 6**, embora não integre o *framework* do Scrum, representa aquilo que se acredita que precisará ser desenvolvido pelo Time de Desenvolvimento no decorrer do projeto de software, tomando por base um objetivo ou necessidades de negócio de alto nível, e que fornecem o contexto, orientação, motivação e inspiração para o desenvolvimento do produto.

**Figura 6 – Template de Visão do Produto**

“Para (cliente-alvo),  
 que (problema ou oportunidade),  
 o (nome do produto) é um (categoria do produto)  
 que (benefício-chave, razão convincente para utilizar).  
 Ao contrário de (alternativa primária competidora),  
 nosso produto (diferenciação primária).”

Fonte: SABBAGH (2013, pág. 168) e HIGHSMITH (2004).

É importante citar uma estratégia muito utilizada para o planejamento de um *Backlog* de Produto, que é aquela onde o Time de Desenvolvimento estima cada um de seus itens, visualizando assim o trabalho que será despendido em uma próxima *Sprint* ou *Release* do desenvolvimento do produto.

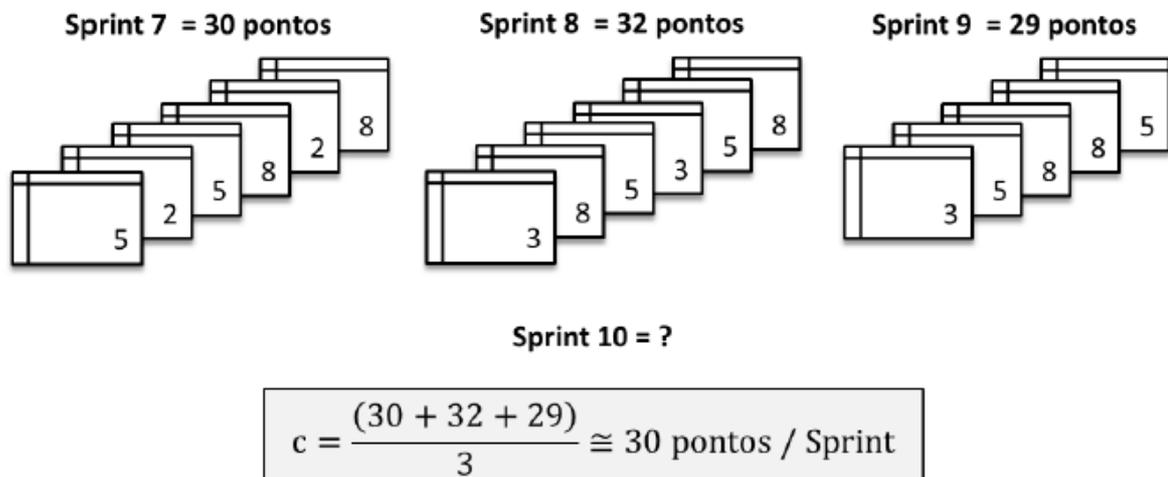
Tipicamente, as estas estimativas geradas pelo time baseiam-se em técnicas adaptativas, considerando um contexto e necessidade, onde comumente são utilizados os seguintes tipos de unidades: “tempo real”, em dias ou horas de trabalho real; “tempo ideal”, considerando dias ou horas de trabalho ideal; ou “*Story Points*”, uma unidade relativa de tempo criada pelo time de desenvolvimento – sendo esta a mais utilizada (SABBAGH, 2013).

Segundo JEFRIES (2000), a técnica de *Story Points* foi criada com o propósito de burlar estimativas convencionais em unidades de tempo (em horas ou dias, por exemplo), pois a gerência frequentemente as entendia como promessas e não como estimativas, uma prática que prejudicava o bom andamento das tarefas do time, dado o foco da personalidade nas cobranças. Sendo assim, *Story Points* reconhece que estimativas são necessárias ao planejamento, mas ao mesmo tempo, evita o comprometimento com sua alta precisão.

Conforme SABBAGH (2013) existem estimativas derivadas de *Story Points* e por vezes também utilizadas no planejamento das *Sprints*, como por exemplo: a “Velocidade do Time”, uma variável calculada pela média de pontos produzidos pelo Time de Desenvolvimento, por *Sprint* (exemplo na **Figura 7**). Deste modo, a

definição de “ponto” pode assumir diversas formas, não sendo, portanto, uma medida que comparável entre diferentes times de desenvolvimento.

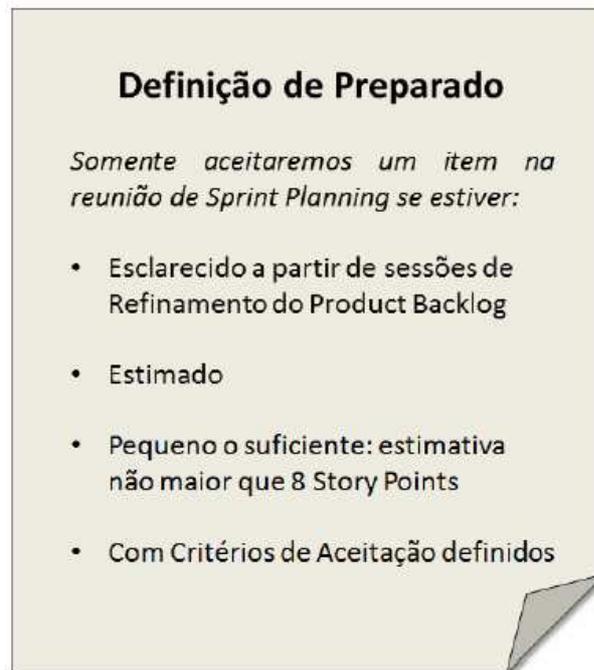
**Figura 7 – Exemplo de Medição de Velocidade com Story Points**



Fonte: SABBAGH (2013, pág. 132).

Em relação ao fluxo dos itens, é importante frisar, que para um item migrar do *Backlog* do Produto para o Planejamento da *Sprint* (logo, constando no *Backlog da Sprint*), este deverá ser avaliado conforme a “Definição de Preparado” cuja elaboração dar-se-á nas sessões de replanejamento do *Backlog* do Produto, realizadas entre o PO e o Time de Desenvolvimento (exemplo na **Figura 8**).

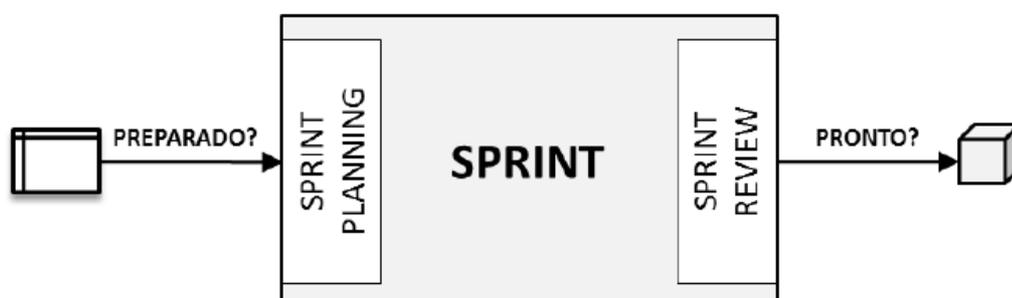
**Figura 8 – Exemplo da Definição de Preparado**



*Fonte: SABBAGH (2013, pág. 190).*

A “Definição de Preparado” é o artefato utilizado para garantir que os itens a serem considerados na “Reunião de Planejamento da Sprint” estarão realmente preparados de acordo com um critério bem definido, como um acordo formal entre o PO e o Time de Desenvolvimento. Ressalta-se que este é um direcionamento similar ao que é definido para o item a ser migrado da *Sprint Review* para o status de “pronto” (**Figura 9**) e que este mecanismo é um cuidado do método para mitigar o risco de tornar a reunião muito longa, cansativa e/ou ineficiente, o que poderia levar ao seu fracasso e prejudicar o bom andamento da *Sprint* (SABBAGH, 2013).

**Figura 9 – Fluxo do Item: Definição de Preparado e Definição de Pronto**



*Fonte: SABBAGH (2013, pág. 188).*

### 1.3.1.3 Reunião de Planejamento da Sprint

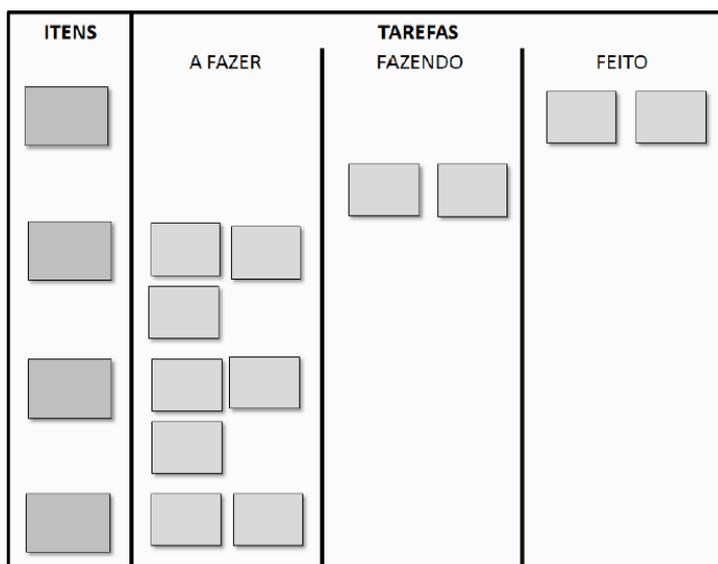
Após a formação do *Backlog* do Produto e no início de cada *Sprint*, faz-se necessária a realização de uma “Reunião de Planejamento da *Sprint*” (*Sprint Planning*), cerimônia que tem como objetivo planejar o ciclo de desenvolvimento que se inicia (*Sprint*), já no seu primeiro dia, e prevendo uma duração máxima proporcional à 8 horas para *Sprints* de um mês.

Na referida cerimônia, deverá ser definida uma meta para a *Sprint* e a equipe se comprometerá a completar uma quantidade específica de tarefas elaboradas pelo PO, Scrum Master e Time de Desenvolvimento, conforme originadas a partir dos itens do *Backlog* do Produto. Destaca-se que estas tarefas também podem ser estimadas e relatadas no *Sprint Backlog* (SCHWABER; SUTHERLAND, 2013).

De acordo com SABBAGH (2013), a “Meta da Sprint” é o objetivo de negócio a ser alcançado pela *Sprint* e o *Sprint Backlog* é uma lista de itens selecionados do alto do *Product Backlog*, onde se destacarão: os itens menores, mais detalhados e importantes para o negócio; e os acordos realizados entre o Time de Desenvolvimento e o *Product Owner* para a sua composição no desenvolvimento de um incremento de produto no *Sprint* (“o quê”), adicionado de um plano do trabalho que será realizado (“como”). É importante frisar, que nos acordos citados, a “Velocidade do Time” deve ser levada em consideração como um critério de análise de capacidade, para que seja respeitado o limite de trabalho que “cabe” dentro da *Sprint*, com vistas ao seu planejamento.

O plano de trabalho da *Sprint*, por sua vez, que deve ser gerado e criado para fins de acompanhamento, é geralmente expresso por um determinado conjunto de tarefas correspondente a cada item, além da indicação do andamento de cada tarefa conforme sua situação (exemplo na **Figura 10**). Ele pode ainda, ser relacionado às informações sobre as estimativas realizadas, para que o time possa acompanhar o seu progresso em relação ao final da *Sprint*.

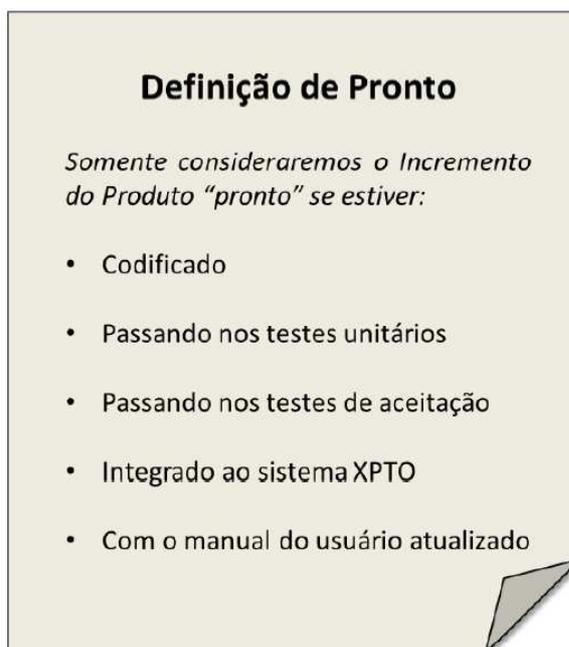
#### **Figura 10 – Quadro de Tarefas do Scrum**



Fonte: SABBAGH (2013, pág. 149).

Depois de definir “o quê” e “como” algo será realizado durante a *Sprint*, o Time de Desenvolvimento elaborará então a “Definição de Pronto”, que pode variar de time para time e de projeto para projeto. Ela deverá ser compreendida e compartilhada por todos os membros da equipe (a **Figura 11** ilustra um exemplo relacionado).

**Figura 11 – Exemplo da Definição de Pronto**



Fonte: SABBAGH (2013, pág. 156).

#### 1.3.1.4 Execução da Sprint

Após a “Reunião de Planejamento da Sprint”, inicia-se o trabalho de desenvolvimento dos itens do *Sprint* Backlog (SABBAGH, 2013), na ordem do item mais importante para menos importante. Então ao final da *Sprint*, o Time de Desenvolvimento entregará um “Incremento do Produto”, ou seja: o total dos itens completos no *Sprint* conforme a “Definição de Pronto”, os quais constituem, portanto, novas funcionalidades e melhorias daquilo que foi produzido anteriormente.

Espera-se que um Incremento de Produto seja potencialmente entregável, de forma que o *Product Owner* possa decidir por efetivá-lo ou não como *release* do produto, ou seja, uma versão que será fornecida aos clientes do projeto logo ao final da *Sprint*. Em alguns casos, no entanto, isso pode não fazer sentido para o negócio frente as suas metas e/ ou pode representar uma visão de produto mínimo incompleta, sendo necessário, portanto, o acúmulo de alguns incrementos até que ocorra a *release*.

#### 1.3.1.5 Reunião Diária

Segundo SABBAGH (2013), a cada dia de trabalho, os membros do Time de Desenvolvimento se encontram em uma reunião curta, de no máximo quinze minutos, que deve ser realizada preferencialmente no mesmo horário e local. Esta reunião é chamada *Daily Scrum* e é responsável por promover no time a capacidade de auto-organização. Ela tem também como propósito proporcionar a todos a visibilidade do trabalho realizado e a realizar, facilitando a comunicação e conferindo visibilidade sobre os obstáculos. Contudo, isso também serve como oportunidade para a tomada de decisões rápidas intrínsecas ao progresso da *Sprint*.

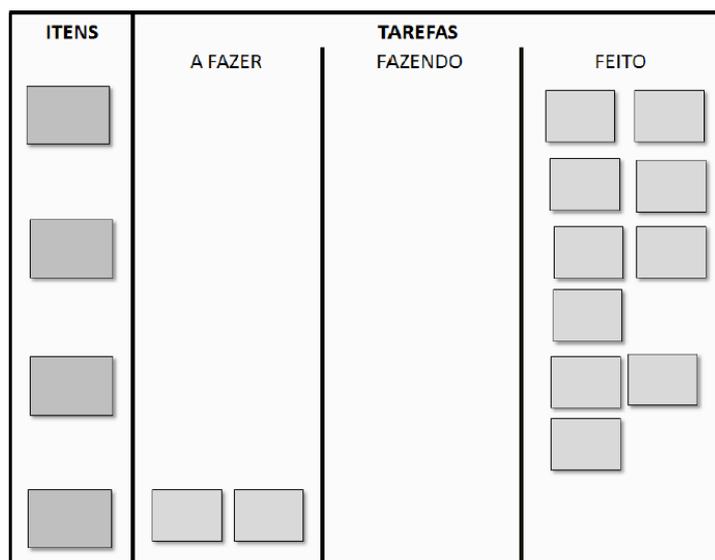
### 1.3.1.6 Reunião de Revisão da Sprint

A “Reunião de Revisão da *Sprint*” é uma cerimônia dedicada à inspeção e adaptação de produto, para obtenção de *feedback* sobre o “Incremento do Produto” desenvolvido na *Sprint* (SABBAGH, 2013). Nesta reunião, que é realizada no último dia da *sprint*, os itens desenvolvidos são confrontados frente à “Definição de Pronto” previamente estabelecida (conforme ilustração na **Figura 9**). Ou seja, isso significa que se um item for considerado aderente à “Definição de Pronto”, na “Reunião de Revisão de *Sprint*”, ele será de fato aceito e passará a compor o “Incremento do Produto” gerado na *Sprint*.

Em geral, durante a referida reunião, o “Incremento do Produto” construído é mostrado funcionando perante o cliente, em uma atividade cooperativa executada pelo PO e pelo Time de Desenvolvimento, demonstrando assim, o valor agregado gerado.

Em caráter ilustrativo, segue um exemplo do Quadro de Tarefas disponível no final de uma *Sprint*, representado pela **Figura 12**. O exemplo demonstra possivelmente uma *Sprint* bem sucedida, mesmo que duas de suas tarefas não tenham sido concluídas a tempo.

**Figura 12 – Exemplo de um Quadro de Tarefas ao Final da Sprint**



Fonte: SABBAGH (2013, pág. 223).

### 1.3.1.7 Retrospectiva da Sprint

Durante a “Retrospectiva da Sprint”, faz-se a inspeção e adaptação da forma de trabalho (SABBAGH, 2013), com o objetivo prover a melhoria contínua com base na reflexão da equipe em relação a como ela pode se tornar mais efetiva a partir de ajustes em aspectos do seu comportamento.

A “Retrospectiva da Sprint” é uma reunião que deve ser realizada no último dia de cada Sprint, após a “Reunião de Revisão da *Sprint*”, possuindo duração máxima proporcional de três horas para *Sprints* de um mês. Nesta cerimônia, o time pode visualizar tanto os aspectos que foram bem, como aqueles que devem melhorar (**Figura 13**), transformando-os em planos de ação.

**Figura 13 – Quadro de Retrospectivas**



Fonte: SABBAGH (2013, pág. 231).

### 1.3.1.8 Acompanhamento do Trabalho

O Scrum privilegia ferramentas visuais para o acompanhamento do progresso do cumprimento da quantidade mensurável estimativa de trabalho para um tempo previamente determinado, permitindo uma visualização rápida e com vistas à identificar problemas mais cedo, reduzindo assim os riscos associados ao projeto

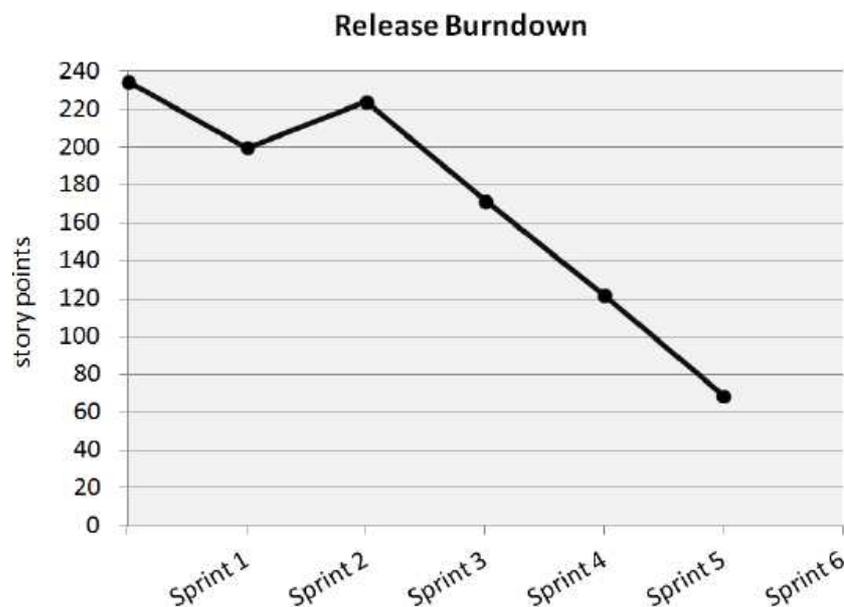
Classificação da Informação: **Uso Irrestrito**

(SABBAGH, 2013). São comumente utilizados os seguintes gráficos com este propósito:

- *Release Burndown*;
- *Sprint Burndown*;
- *Release Burnup*;
- *Sprint Burnup*.

Atenta-se, que da relação acima, o gráfico “*Release Burndown*”, embora não original do Scrum (assim como o *Release Burnup*), mostra-se muito útil ao planejamento de *Release* e representa o trabalho “queimado” em direção ao seu objetivo, ou seja, ele deve apresentar uma linha decrescente em relação ao objetivo do projeto. Este gráfico é mantido pelo *Product Owner* e pelo Time de Desenvolvimento, com o propósito de mapear o progresso no desenvolvimento em direção à entrega de *release* e apoiar mudanças de rumo ou metas. Abaixo, na **Figura 14**, pode ser visualizado um cenário hipotético do seu uso, representando a penúltima *Sprint* da *Release* disponibilizada por um projeto.

**Figura 14 – Gráfico de Release Burndown**

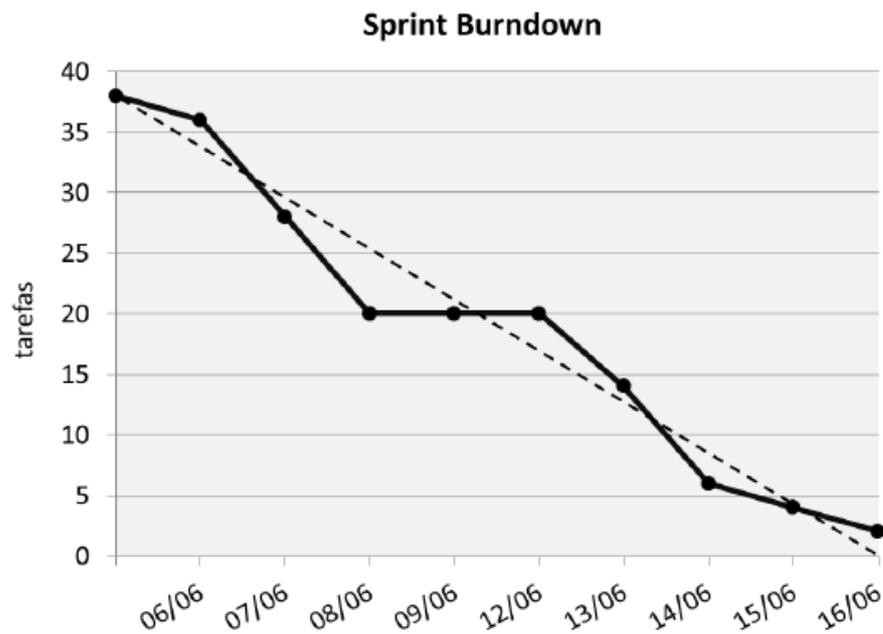


Fonte: SABBAGH (2013, pág. 173).

O gráfico “*Sprint Burndown*” é similar ao gráfico anteriormente apresentado, porém apresenta no seu eixo “X” uma outra unidade de tempo (por exemplo: dias ou

semanas) e normalmente exibe uma visão de tarefas. O exemplo da **Figura 15** mostra este tipo de gráfico contendo ainda uma linha de tendência, o que também pode constar nos demais tipos de gráficos sendo apresentados, a depender também da necessidade específica do tipo de acompanhamento sendo realizado.

**Figura 15 – Gráfico de Sprint Burndown**

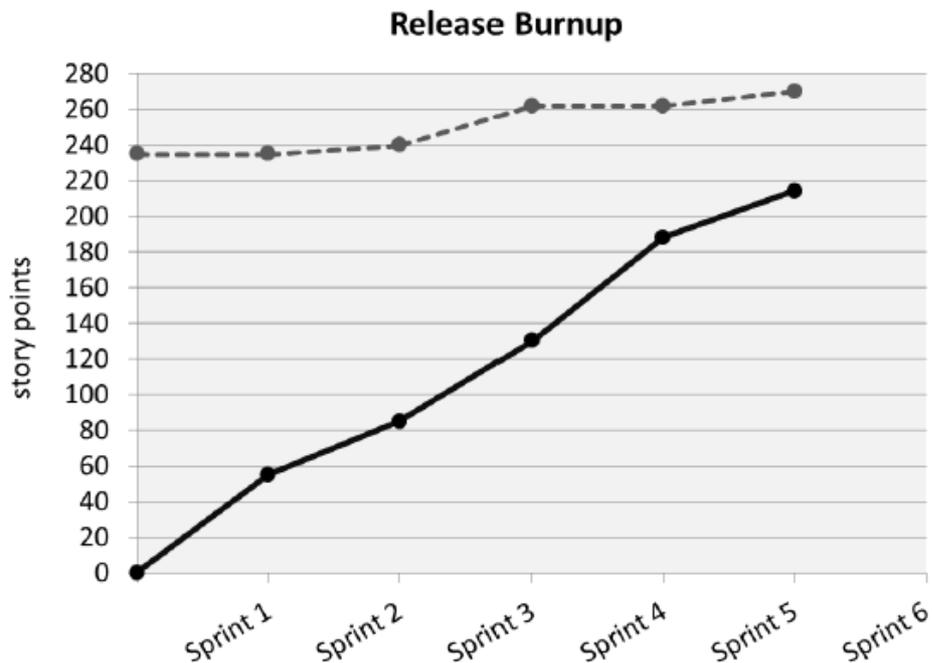


Fonte: SABBAGH (2013, pág. 185).

Os gráficos do tipo “*Burnup*” também são aplicáveis, e apresentam uma perspectiva crescente, contrária àquelas dos gráficos do tipo “*Burndown*”. Eles também são mantidos pelo *Product Owner* e pelo Time de Desenvolvimento, para o monitoramento de progresso, podendo assim representar uma forma de acompanhar os itens do *backlog* conforme sua estimativa ou valor de negócio atribuído, tipo de demonstração que é realizada com base em uma linha do tempo e conforme o trabalho já realizado.

Abaixo, na **Figura 16**, há um exemplo de utilização do gráfico de “*Burnup de Release*”, aplicando-se o mesmo conceito para o gráfico de “*Burnup de Sprint*”, apenas considerando-se outros propósitos e unidades de tempo menores.

**Figura 16 – Gráfico de Release Burnup**



*Fonte: SABBAGH (2013, pág. 177).*

### 1.3.2 A Metodologia Ágil do SISP

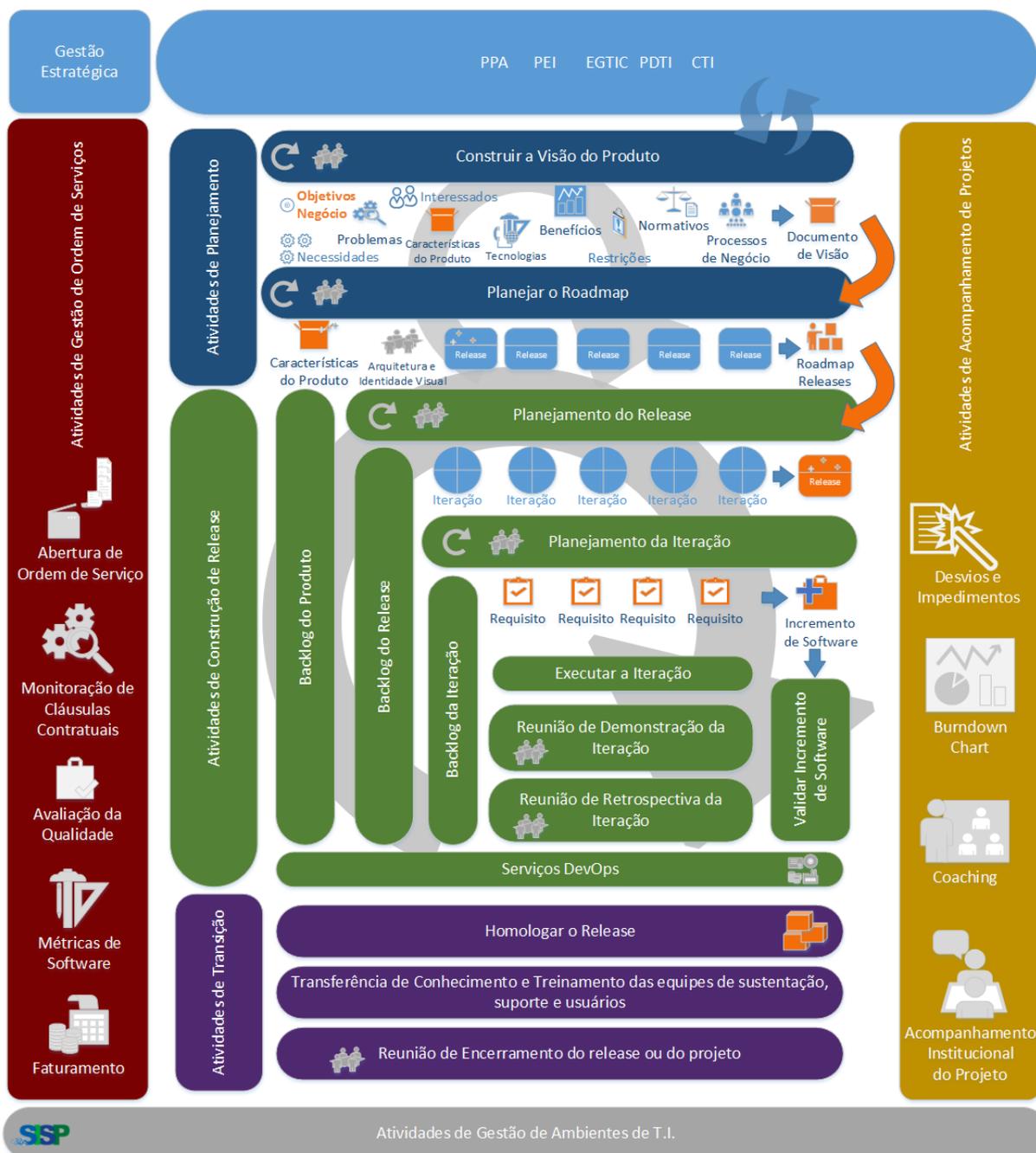
Segundo o SISP (2015b, pág. 16), muitas instituições iniciam o uso de métodos ágeis por meio da adoção do framework fornecido pelo Scrum, porém isso é apenas uma parte do que é necessário para oferecer soluções sofisticadas para seus *stakeholders*, já que, inevitavelmente, as equipes precisam olhar para outros métodos para preencher lacunas que o Scrum propositalmente não considera no seu arcabouço.

Desta forma, a proposta do SISP (2015b) para a metodologia ágil, ilustrada na **Figura 17**, considera que as atividades de desenvolvimento do projeto envolvem basicamente três fases (que por sua vez, englobam um conjunto de métodos):

- Planejamento – fase que define um roadmap do produto com base na concepção de um Documento de Visão;
- Construção do release – fase que utiliza diversos métodos para desenvolver software de forma iterativa e incremental;

- Transição – fase para implantação definitiva do produto desenvolvido, assim como aspectos de suporte e avaliação.

**Figura 17 – Metodologia Ágil do SISP**



Fonte: SISP (2015b)

Estas fases são compostas por grupos de atividades de gestão de ordem de serviço, acompanhamento do projeto e gestão dos ambientes de TI, onde os ciclos de entrega envolvem os grupos de atividades de planejamento, construção do Classificação da Informação: **Uso Irrestrito**

release e transição, um para cada release. E após a finalização do grupo de atividades de construção do release, é feita a transição dos produtos para o ambiente de produção.

O grupo de atividades de “Gestão de Ordem de Serviço” reúne mecanismos de controle, fiscalização e monitoração da execução e também dos produtos entregues pela contratada de desenvolvimento de software. Por meio deste instrumento é possível mitigar riscos da contratação, tais como falta de qualidade técnica e funcional, pagamento sem produto de software funcional, pagamento sem métrica objetiva, entre outros (SISP, 2015b).

O grupo de atividades de “Acompanhamento do Projeto” contempla atividades para monitorar os projetos em execução pela contratada, identificando e mitigando riscos e desvios, tratando impedimentos e descumprimento de cronograma, entre outros aspectos.

E finalmente, as “Atividades de Gestão de Ambientes de TI” contemplam tarefas para viabilizar a criação de ambientes e facilitar a sua disponibilidade, para construir e implantar cada *release* do produto de software.

### *1.3.2.1 Papéis e Responsabilidades*

Na metodologia proposta pelo SISP (2015b), elencou-se o seguinte conjunto de papéis para a equipe da empresa contratante:

- Analista de Infraestrutura de TI – representante da área de infraestrutura de TI para um projeto específico, sendo uma de suas responsabilidades principais a disponibilização dos recursos de hardware, software e serviços necessários, relacionados aos ambientes de Testes, Homologação e Produção, além de outras atividades correlacionadas;
- Analista de Qualidade – representa diferentes perfis ligados a avaliação de qualidade de software, tais como: Analista de Teste, Arquiteto de Software, Analista de Banco de Dados, Analista de Configuração e Mudança e outros, cuja responsabilidade é avaliar e definir processos de

verificação da qualidade dos produtos entregues pela empresa contratada para o desenvolvimento – esta atividade pode ser parcialmente delegada a uma outra empresa contratada que possa exercer esta função, o que é comum em algumas empresas públicas;

- Analista de Métricas – é o responsável pelos serviços de medição de software, utilizando a métrica definida no contrato firmado com a empresa de desenvolvimento de software – esta atividade também pode ser parcialmente delegada para outra empresa contratada que pode cumprir esta função, algo bastante comum nas contratações públicas;
- Cliente (da área de negócio) – representante da área de negócio e responsável pela solicitação e homologação dos serviços de desenvolvimento de software;
- Coach (este papel pode ser exercido pelo Líder de Projeto) – profissional experiente e com sólidos conhecimentos nas práticas do processo ágil, devendo zelar pela sua correta execução, com vistas a guiar os outros envolvidos no projeto a executar o processo e práticas de forma adequada;
- Product Owner – é o representante da área de negócio (cliente) com conhecimento suficiente para definir e priorizar requisitos do negócio e responder aos questionamentos da equipe de desenvolvimento;
- Líder de Projeto – é responsável pelo projeto, que se relaciona com todos os envolvidos e atividades do projeto;
- Usuário – tipo de *stakeholder* que usa o produto de software, seja interno ou externo à organização;
- Outros (legais, no contexto dos órgãos públicos) – devido à atribuições legais ou organização interna da contratante, podem ser atribuídos outros papéis, tais como um Gestor do Contrato e fiscais de contrato.

Complementarmente, os papéis da equipe contratada, no modelo de referência do SISP (2013), são os seguintes:

- Preposto – um responsável técnico pelos serviços prestados, atua como um gestor técnico e ponto focal do time da contratada;

- Scrum Master – um dos responsáveis pelo time da contratada, este papel trabalha para o sucesso do projeto através da entrega de valor para o cliente, viabilizada pela atuação eficaz da equipe de desenvolvimento e apoio ao dono do produto;
- Time de Desenvolvimento – é um grupo de cinco a nove integrantes da contratada de desenvolvimento de software, que deve possuir uma característica multifuncional, auto-organizados, contando com analistas, arquitetos, programadores, testadores, administrador de banco de dados entre outros;
- Gerente de Relacionamento (opcional) – realiza o acompanhamento de todo o ciclo de atendimento da ordem de serviço garantindo as entregas conforme compromissos firmados.

### 1.3.2.2 Planejamento do Produto

Conforme o SISP (2015b), esta fase contempla a concepção do Documento de Visão, artefato em que são descritos os objetivos de negócio e as características-chave (*features*) do produto. E após a concepção deste, inicia-se então o planejamento de cada entrega (release), que consiste no agrupamento de objetivos de negócio e aspectos do produto definidos através de critérios de priorização e importância. Ao final do Planejamento, obtêm-se, portanto, um *roadmap* do produto.

Desta forma, são as principais atividades do Planejamento:

- Construir a Visão do Produto;
- Planejar o Roadmap.

### 1.3.2.3 Construção do Produto

A fase de construção do produto contempla o planejamento, especificação e implementação dos objetivos de negócio e das características-chave do produto

sendo desenvolvido, contidas em releases e construídas iterativamente (Sprints). Segundo o SISP (2015b), estas atividades são definidas a partir de algumas práticas de métodos ágeis, tais como: Scrum, XP e *Lean*.

Dada a natureza ágil de incentivo às mudanças constantes, durante a construção o SISP (2015b) define dois tipos de alterações em funcionalidades:

- Refinamento – resultante da evolução natural dos requisitos como um aprimoramento do entendimento das histórias de usuário (telas/funcionalidades), com o aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento ágil;
- Mudança de Escopo – resultante da evolução das políticas, estratégias, legislação, objetivos específicos de negócio e necessidades ou processos de negócio da empresa.

Também visando fornecer um melhor entendimento acerca do assunto, é importante também destacar na “Construção do Produto” as principais atividades de construção do *release* e desenvolvimento de iterações:

- Release
  - Elaborar o *Backlog* do Produto;
  - Planejar o *Release*.
- Iterações
  - Reunião de Planejamento da Iteração;
  - Executar a Iteração;
  - Reunião de Demonstração da Iteração;
  - Reunião de Retrospectiva;
  - Validar Incremento de Software.

### 1.3.2.4 Transição do Produto

A fase de “Transição do Produto” contempla as atividades que visam garantir a implantação de cada *release*, contemplando também a avaliação dos resultados obtidos e condições gerais de entrega, com suporte ao produto do projeto.

Para o SISP (2015b), cada *release* é homologado e liberado nas atividades de transição na forma de uma versão funcional, com produto pronto para disponibilização em produção. Desta forma, as atividades de transição visam garantir que clientes e usuários tenham condições de usar o produto de software.

São atividades de Transição do Produto:

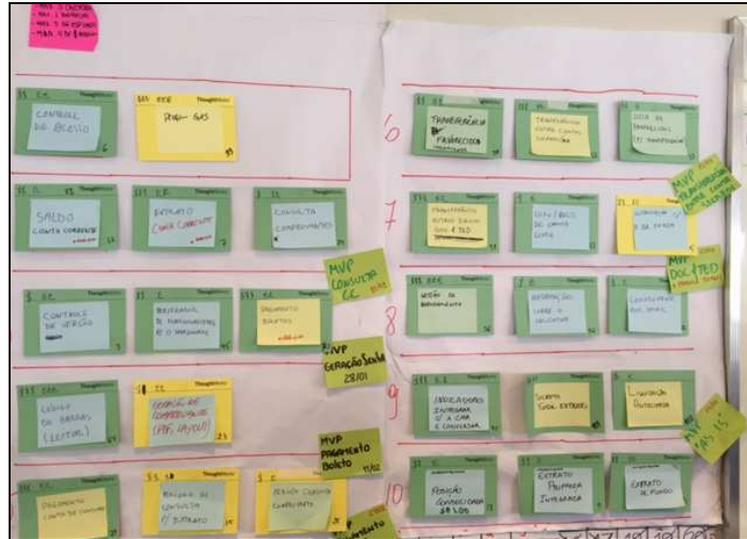
- Homologar o Release;
- Preparar e Realizar Treinamentos;
- Realizar reunião de Encerramento do Release ou do Projeto.

### 1.3.3 Concepção Enxuta (Direto ao Ponto)

Segundo CAROLI (2015) o MVP Canvas (**Figura 18**) é uma visão consolidada do planejamento de *releases* de um produto sendo trabalhado, chamadas MVPs e que configuram porções de código que incrementam e/ou aumentam o produto, facilitando sua compreensão pelos envolvidos em um projeto de software desenvolvido colaborativamente.

Um MVP Canvas contém os recursos e a proposição de valor a ser entregue em cada porção parcial de software que é potencialmente entregável, tal como a sua ordem de liberação. Estas porções são consideradas incrementos, que constituem conjuntos de hipóteses validadas e orientadas pelos resultados iniciais, sendo que a sua correção ou a confirmam um curso que será utilizado para guiar os próximos incrementos. E estes são novos produtos mínimos adicionados aos produtos mínimos já validados; mais uma vez, produtos mínimos, entretanto viáveis para fazer novas verificações sobre o direcionamento.

**Figura 18 – MVP Canvas (exemplo)**



Fonte: CAROLI (2015b).

O MVP Canvas configura-se assim, no principal resultado obtido em uma Concepção Enxuta (“*Lean Inception*”), termo criado por CAROLI (2015) e que remete à etapa executada de desenvolvimento de produtos executada com base em um método de sua autoria, chamado “Direto ao Ponto”. Este método utiliza um conjunto de práticas provenientes de outros *frameworks* ligados aos métodos de cocriação (tais como o *Design Thinking*, já abordado anteriormente), métodos ágeis e *Lean*.

Segundo CAROLI (2015), “Direto ao Ponto” é uma técnica que age como uma “receita” (como o autor prefere chamar), para possibilitar o entendimento e planejamento da entrega incremental de MVPs, organizando suas ideias e recursos em torno de um modelo que busca entender a principal finalidade do produto, considerando as jornadas dos usuários para realizar as entregas incrementais de produtos viáveis.

A “receita” do Direto ao Ponto é definida por uma sequência de atividades cuja aplicação mostra-se rápida e eficaz (analogamente à receita de um bolo), conforme abaixo:

- Descreva a visão do produto;
- Priorize os objetivos do produto;

- Descreva os principais usuários, seus perfis e as suas necessidades;
- Entenda as principais funcionalidades;
- Compreenda os níveis de incerteza, tais como o esforço e valor de negócio por funcionalidade;
- Descreva as jornadas mais importantes dos usuários;
- Crie um plano de entrega incremental do produto, impulsionado pelo conceito de MVP;
- Estime o esforço por amostragem;
- Calcule os custos e especifique datas e cronograma de entrega.

A seguir, veremos detalhadamente cada uma destas atividades, com o objetivo de entendê-las completamente para posterior agregação ao modelo sendo proposto.

#### 1.3.3.1 Produto Mínimo Viável

O *Minimum Viable Product* (MVP), já citado anteriormente, mas até então pouco explorado neste estudo, é a versão mais simples de um produto que pode ser disponibilizada para o negócio. O seu foco reside no produto mínimo e na validação de pedaços menores, que obviamente são bem menos elaborados do que uma versão final, mas já úteis para verificar se um direcionamento está no caminho correto (CAROLI, 2015).

A **Figura 19** mostra como o MVP oferece validações pequenas ao longo do tempo, diferentemente da forma tradicional de criação de produtos, que provavelmente só validaria um produto funcionando na sua versão final. No exemplo, o trator cortador de grama (MVP 8) representa o produto final.

### **Figura 19 – Exemplo do Desenvolvimento Incremental com MVPs**



Fonte: CAROLI (2015).

De acordo com CAROLI (2015), considerando um planejamento por MVPs, as hipóteses de negócio são sequenciadas em uma série de outras hipóteses menores, mais simples e, logo, mais fáceis de entender. Elas também são elaboradas de forma mais rápida e disponibilizadas no produto mais cedo, viabilizando a utilização pelo usuário final.

### 1.3.3.2 Concepção Enxuta

A fase de “Concepção Enxuta” (*Lean Inception*, em inglês), ou simplesmente “Concepção”, como chamaremos a partir deste ponto do texto, é definida como um *workshop* presencial e colaborativo realizado para alinhar necessidades de negócio e o entendimento do MVP (CAROLI, 2015). Assim, o sucesso de uma Concepção está diretamente atrelado à capacidade do grupo envolvido em colaborar e participar ativamente das dinâmicas.

Segundo CAROLI (2015), em uma única semana de trabalho colaborativo, os envolvidos na Concepção devem ser capazes de compreender os objetivos do produto e os seus principais usuários. Sendo assim, durante esta cerimônia, deverá ser identificado um escopo funcional de alto nível, que deve ser suficiente para definir uma estratégia de lançamento incremental dos MVPs.

As dinâmicas da cerimônia de Concepção contemplam a descoberta dos objetivos, estratégias e definição do produto, bem como o mapeamento e a priorização das *features* desejáveis para serem entregues gradualmente, formando os MVPs. Desta forma, o principal objetivo do *workshop* é fazer com que a equipe identifique e compreenda coletivamente o que vai ser desenvolvido (CAROLI, 2015), seguindo uma agenda pré-definida (**Figura 20**), que tem sua evolução controlada por um gráfico de “*Burnup da Agenda*” (**Figura 21**).

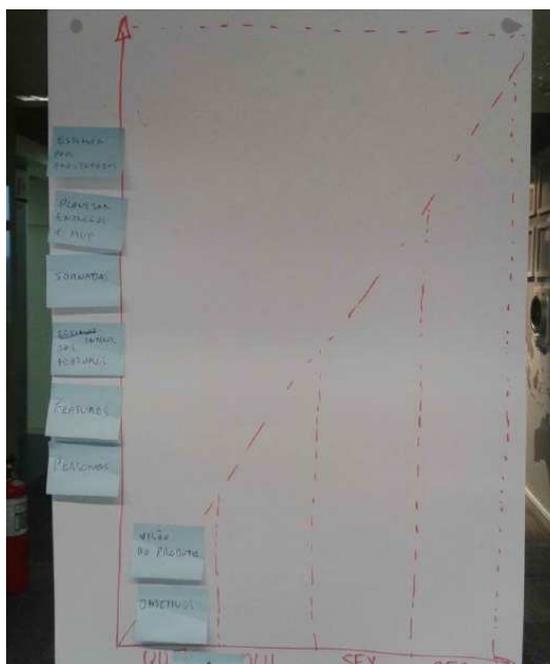
**Figura 20 – Agenda da Concepção Enxuta**



Fonte: CAROLI (2015).

A saber, no gráfico “*Burnup da Agenda*” (**Figura 21**) possui o eixo vertical definido como a quantidade de tópicos ou atividades a serem realizadas e é medido em unidades personalizadas para a agenda específica do workshop. Já o seu eixo horizontal, representa o tempo, que é normalmente medido em horas ou dias.

**Figura 21 – Agenda Burnup**



Fonte: CAROLI (2015).

Conforme CAROLI (2015), a infraestrutura para o *workshop* deve contemplar uma “sala de guerra” (ou *war room* em inglês), preferencialmente exclusiva para o evento, cuja existência visa evitar a perda de tempo com deslocamentos e fomentar a manutenção de todas as informações em um único lugar. Nesta sala, o ideal, segundo o autor, é que estejam disponíveis recursos tais como os que constam na **Figura 22**.

**Figura 22 – Sala de Guerra**



Fonte: CAROLI (2015).

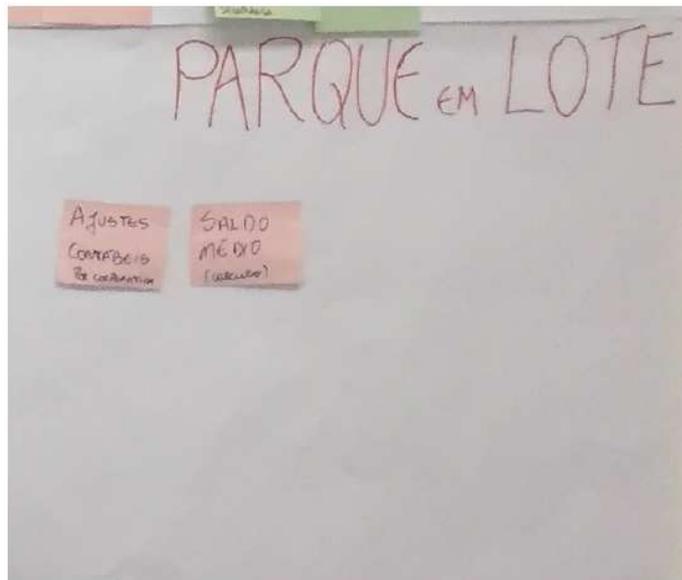
É importante certificar-se também, de que todos os itens necessários para o bom andamento do *workshop* estejam disponíveis, tais como:

- Participantes selecionados e convidados (*stakeholders* e membros ativos);
- Facilitador experiente;
- Reserva de sala (manter a mesma sala durante o período da *Inception*);
- Acessórios, tais como: *Flip chart*, cartões, post-its, papel A4 e canetas para todos;
- *Coffee break*.

Devido ao tempo fixo da Concepção e também pela necessidade de garantir o foco dos envolvidos sem correr o risco de perder *insights* que poderiam ser resgatados em um momento oportuno, CAROLI (2015, pág. 20) propõe ainda a utilização da ferramenta *Parking Lot*.(estacionamento, em português), cuja nomenclatura foi foneticamente adaptada para “Parque em Lote”, para melhor assimilação e manutenção do ambiente divertido do *workshop*.

O “Parque em Lote” ajuda a arquivar momentaneamente quaisquer itens, ideias ou questões que são levantadas durante uma atividade da “Concepção” e que não são úteis para a discussão naquele momento específico em que foram levantadas. É uma ferramenta de grande valia para o facilitador, e por isso ele deve introduzir o conceito logo no início do evento ou assim que a primeira conversa desviar do assunto em foco, pois ela lhe proporciona uma maneira educada e divertida de dizer “sim, eu ouvi você, mas essa conversa fica para depois”.

### **Figura 23 – Parque em Lote (Parking Lot)**



Fonte: CAROLI (2015).

O recurso de “Parque em Lote” pode ser implementado através do uso de um *flip chart*, colocando-o em uma parede na “sala de guerra”. Assim, se o item em discussão ainda não estiver escrito em um *post-it*, deve-se escrevê-lo e colocá-lo no “Parque em Lote” (**Figura 23**). E ao final de cada dia da “Concepção”, pode-se usar cerca de 10 minutos para rever os itens, onde são possíveis duas ações diferentes: esclarecer e remover o item, ou deixá-lo para a próxima revisão.

#### 1.3.3.3 Visão de Produto

A visão de produto, segundo Caroli (2015), deve ser elaborada colaborativamente, refletindo uma mensagem clara aos clientes, podendo ser utilizado para este fim o canvas definido na **Figura 24**, com base nos seguintes procedimentos:

1. Escrever o “*Template da Visão do Produto*” em um quadro branco ou *flip chart*, de forma que ele fique visível para toda a equipe;
2. Dividir a equipe em grupos menores, solicitando que cada membro preencha individualmente uma lacuna (ou mais, dependendo do tamanho da equipe);

3. Reunir o resultado de cada grupo, formando uma única frase.

**Figura 24 – Template da Visão do Produto**

Para [cliente final],  
 cujo [problema que precisa ser resolvido],  
 o [nome do produto]  
 é um [categoria do produto]  
 que [benefício-chave, razão para adquiri-lo].  
 Diferentemente da [alternativa da concorrência],  
 o nosso produto [diferença-chave].<sup>a</sup>

*Fonte: CAROLI (2015, pág. 23).*

A atividade “É - Não é - Faz - Não faz” é outra dinâmica que ajuda a definir e esclarecer um produto, sendo por vezes, é mais fácil descrever o que alguma coisa não é ou não faz. Esta atividade busca indagar cada aspecto específico do produto, seja positivo ou negativo.

Abaixo, há um passo a passo da atividade “É - Não é - Faz - Não faz” (há também um exemplo de aplicação na **Figura 25**):

1. Dividir um quadro branco ou *flipchart* em quatro áreas (É / Não é / Faz / Não faz);
2. Escrever o nome do produto em destaque (acima dos quadrantes);
3. Solicitar que cada participante descreva o produto, utilizando *post-its* e colocando-os nas áreas dos quadrantes;
4. Ler e agrupar as notas afins.

**Figura 25 – É - Não é - Faz - Não faz**



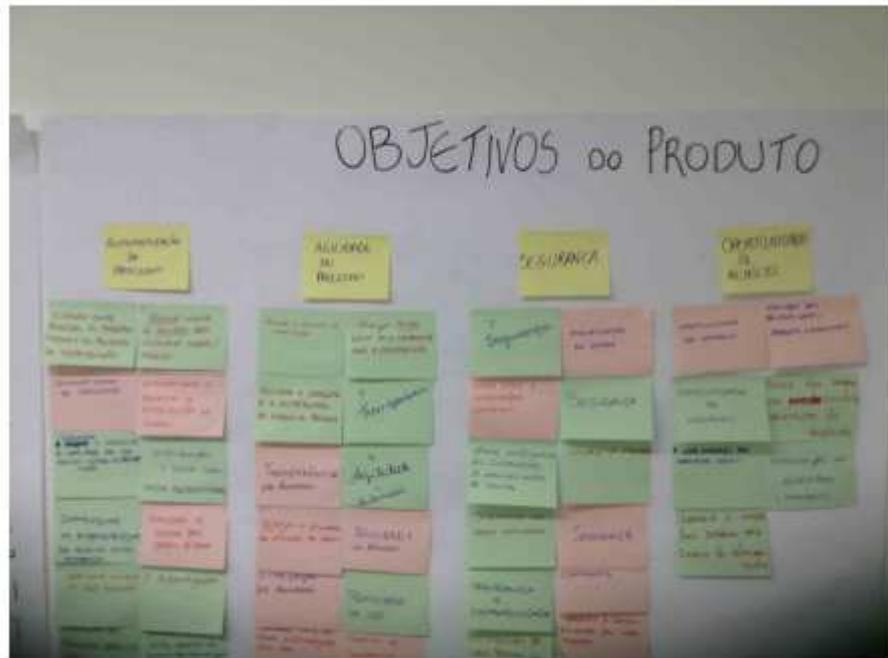
Fonte: CAROLI (2015, pág. 27).

Depois de entender um pouco mais sobre o produto, cada membro da equipe deve compartilhar o que entende como objetivo para o usuário do produto; e na sequência, os vários pontos de vista devem ser discutidos para que se chegue a um consenso sobre eles. CAROLI (2015) propõe as seguintes atividades para cumprir este propósito:

1. Solicitar a cada membro da equipe que escreva, individualmente, três respostas para a seguinte pergunta: “Se você tiver que resumir este produto em três objetivos para seus usuários, quais seriam eles?”;
2. Solicitar aos participantes que compartilhem o que escreveram em um canvas, agrupando os tópicos por similaridade;
3. Solicitar à equipe que reescreva os objetivos, desta vez coletivamente – nesse momento ficará claro que alguns dos elementos listados não são realmente objetivos do produto, devendo o facilitador descartá-los, destacando assim o foco do projeto (**Figura 26**).

### Figura 26 – Objetivos do Produto

Classificação da Informação: **Uso Irrestrito**



Fonte: CAROLI (2015, pág. 29)

CAROLI (2015) julga como uma alternativa interessante para a iniciação da atividade apresentada, a criação de uma enquete entre os participantes do *workshop*. Nesta enquete, cada participante deve elencar, na sua perspectiva, quais são os dois objetivos mais significativos do projeto.

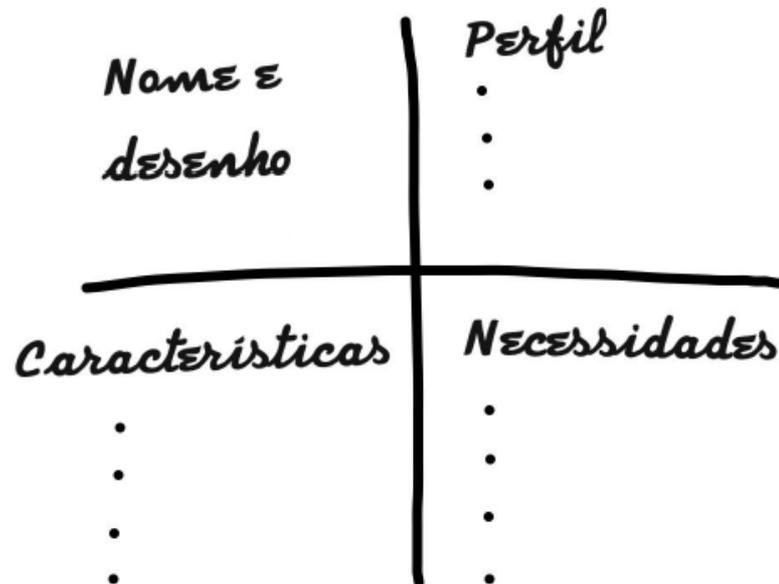
#### 1.3.3.4 Personas

Segundo CAROLI (2015, pág. 32), uma maneira eficaz de identificar funcionalidades do produto de software é utilizando a técnica de “personas”, também muito utilizada no método “*Design Thinking*”, anteriormente apresentado. Em síntese, a técnica visa personificar o usuário, auxiliando o time a descrever funcionalidades do ponto de vista de quem interagirá com o produto. Sendo assim, algumas atividades são necessárias para utilização deste método e são elencadas a seguir:

1. Solicitar ao time que se divida em pares ou trios, entregando um “*Template de Personas*” preenchido (**Figura 27**) para cada grupo;

2. Solicitar a cada grupo que crie uma persona, utilizando o *template* como referência;
3. Solicitar aos participantes que apresentem suas personas a todo o time;
4. Solicitar ao time que mudem os grupos, repetindo os passos de 1 a 3.

**Figura 27 – Template de Personas**



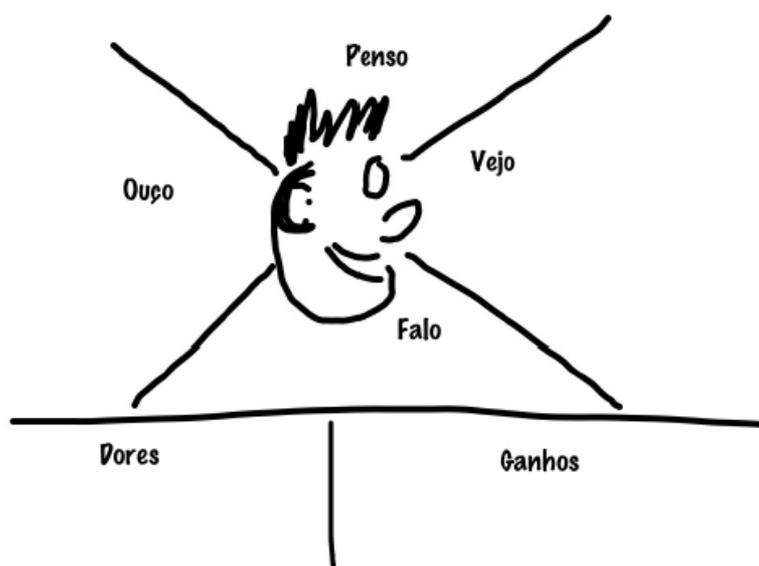
Fonte: CAROLI (2015, pág. 33)

Diversos outros métodos podem ser utilizados para o entendimento do ponto de vista do usuário, como por exemplo, o “Mapa de Empatia”: um *template* visual para identificação e visualização de uma persona. Criado originalmente para análise de segmentos de consumidores, o Mapa de Empatia é uma excelente ferramenta para classificar, explorar e entender os diferentes tipos de personas (CAROLI, 2015, pág. 35). Abaixo, os procedimentos relacionados ao seu uso durante o período de Concepção:

1. Decidir uma persona a ser analisada;
2. Desenhar um *template* do mapa (**Figura 28**), com a persona representada no centro do *template*;
3. Descrever as áreas para tal persona;
4. Repetir os passos 2 e 3 para as próximas personas.

**Figura 28 – Mapa de Empatia**

Classificação da Informação: **Uso Irrestrito**



Fonte: CAROLI (2015, pág. 35)

### 1.3.3.5 Features

Conforme CAROLI (2015, pág. 38), “*feature*” é a descrição de uma ação ou interação de um usuário com o produto, como por exemplo: imprimir nota fiscal, consultar extrato detalhado, e convidar amigos do Facebook. Ela deve ser o mais simples possível e deve compreender as características necessárias ao produto para que uma persona alcance seu objetivo. Os seguintes passos são aplicáveis para execução desta dinâmica:

1. Solicitar que a equipe coloque os objetivos em um canvas comum (**Figura 29**), em ordem de prioridade, da esquerda para direita, como títulos de colunas;
2. Solicitar que a equipe coloque as personas no mesmo canvas, em ordem de prioridade, de cima para baixo, como títulos de linhas;
3. Promover um *brainstorm* de *features*, visto que a discussão deve ser guiada para que se descubram quais *features* são necessárias para atender objetivos e personas (vide exemplo na **Figura 29**).

**Figura 29 – Exemplo de Features, Objetivos e Personas,**



Fonte: CAROLI (2015, pág. 40)

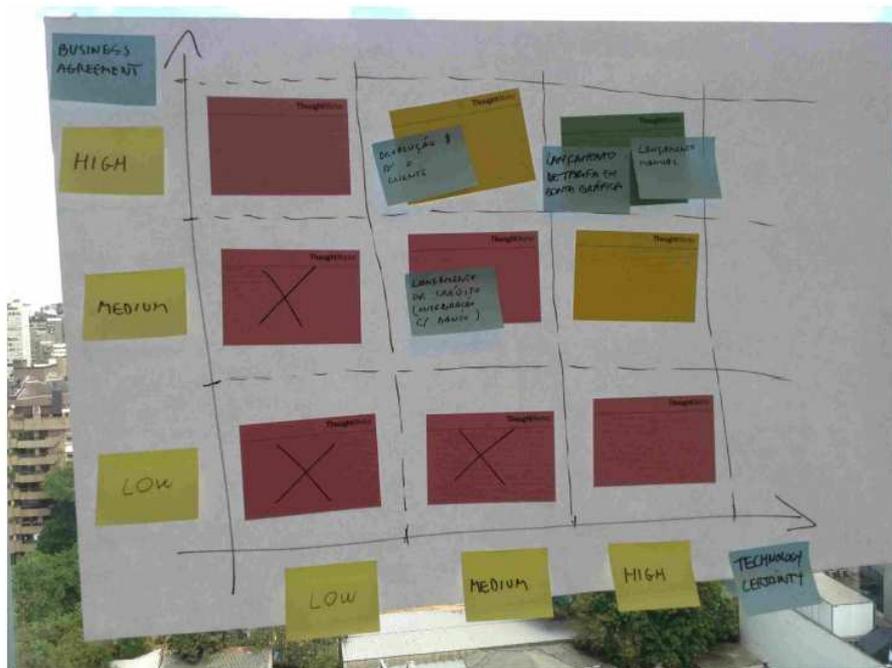
Uma dica do autor para evitar um grande número de *features* nesta atividade é que o número de objetivos e *personas* sejam limitados a depender do contexto, o que implica na escolha de três ou quatro dos principais objetivos e personas, o que pode ser viabilizado com base na seguinte questão a ser levada ao time: “Se nosso orçamento fosse muito curto e pudéssemos trabalhar em apenas um objetivo, qual seria esse objetivo?”.

Depois de mapeadas as *features*, é importante discutir cada uma com a equipe, em relação a como se sentem sobre o seu aspecto entendimento técnico e de negócio do produto, devendo-se atentar que nesta atividade novas notas serão capturadas e que as discordâncias e dúvidas ficarão mais aparentes. Assim, para esta atividade, sugere-se aplicar o seguinte roteiro de atividades:

1. Criar um gráfico em um canvas comum, onde o eixo X representa entendimento técnico (“como fazer”) e o eixo Y representa entendimento sobre o requisito de negócio (“o que fazer”);
2. Solicitar a um membro da equipe que leia uma *feature* em voz alta e a posicione no gráfico de acordo com o seu entendimento sobre ela (entendimento técnico e de negócio);

3. Questionar a equipe, visando verificar se todos compartilham daquela opinião;
4. Se alguém não concordar, a equipe deve discutir os requisitos e a tecnologia envolvidos, de forma que exista um consenso sobre a *feature*. Tudo o que for mencionado e que ajude a alcançar uma melhor compreensão deve ser anotado e anexado à *feature*.
5. Anotar, na *feature*, o nível de entendimento, por exemplo, a **Figura 30** mostra *features* em *post-it* que foram colados em *index cards* verdes, amarelos ou vermelhos, indicando, respectivamente, um nível alto, médio ou baixo de entendimento;
6. Para cada *feature* capturada anteriormente, repetir os passos 2 a 4.

**Figura 30 – Exemplo de Priorização de Features**



Fonte: CAROLI (2015, pág. 40)

Classificadas as *features* de acordo com o entendimento técnico e de negócio, o time da Concepção deverá proceder com uma avaliação de esforço e valor de negócio associado (a cada *feature*). As seguintes atividades são sugeridas por CAROLI (2015):

1. Criar um canvas comum com um gráfico, onde o eixo X representa o esforço e o eixo Y representa o entendimento sobre o valor de negócio;
2. Solicitar que um membro da equipe leia uma *feature* em voz alta e a posicione no gráfico de acordo com o seu entendimento sobre ela (esforço e valor de negócio).
3. Questionar se todos da equipe compartilham daquela opinião – e se alguém não concordar, a equipe deve discutir os requisitos e a tecnologia envolvidos de forma que haja um consenso sobre a *feature*;
4. Anotar, no próprio *card* da *feature*, o nível de esforço e o valor de negócio, assim como demais considerações importantes - por exemplo, a figura abaixo mostra *features* em *index cards*, nas quais foram adicionadas marcações de \$, \$\$ ou \$\$\$, respectivamente, para valor de negócio alto, muito alto e altíssimo, bem como E, EE, ou EEE, respectivamente, para nível de esforço baixo, médio e alto.
5. Para cada *feature* capturada anteriormente, deve-se repetir os passos 2 a 4.

Na **Figura 31**, pode ser visualizado um exemplo de *features* categorizadas em relação ao nível de esforço e o seu valor de negócio. Note que a cor do *index card* representa níveis de entendimento, de acordo com a dinâmica anterior, ou seja, estas atividades são complementares (vide **Figura 32**).

### **Figura 31 – Exemplo de Features com Marcações de Esforço**



Fonte: CAROLI (2015, pág. 52)

**Figura 32 – Entendimento de Negócio x Técnico e Esforço x Retorno**



Fonte: CAROLI (2015, pág. 56)

Segundo CAROLI (2015), depois de passarem pelas atividades de “Entendimento Técnico e Entendimento de Negócio” e de “Esforço e Valor de Negócio”, as *features* devem ser reorganizadas, como apresentado na **Figura 33**.

**Figura 33 – Features depois das atividades**

Classificação da Informação: **Uso Irrestrito**



Fonte: CAROLI (2015, pág. 60)

#### 1.3.3.6 Jornadas de Usuário

Citado também no processo criativo de *Design Thinking*, conforme CAROLI (2015, pág. 61), a “Jornadas do Usuário” é um método simples e eficaz, que descreve o percurso de um usuário (a “persona”) em uma sequência de passos dados para alcançar um objetivo, passos os quais são representados por post-its e configuram diferentes pontos de contato do usuário com o produto. Na **Figura 34** é apresentado um exemplo de jornada mapeada para um usuário.

**Figura 34 – Exemplo de "jornadas do usuário"**



Fonte: CAROLI (2015, pág. 60)

Para identificar a jornada do usuário, Caroli sugere que sejam utilizados *post-its*, papel e caneta, tendo as seguintes questões-chave como um ponto de partida:

- Qual objetivo tal persona quer alcançar?
- Como ela começa o seu dia?
- O que ela faz antes disso?
- O que ela faz depois disso?

### 1.3.3.7 Features nas Jornadas

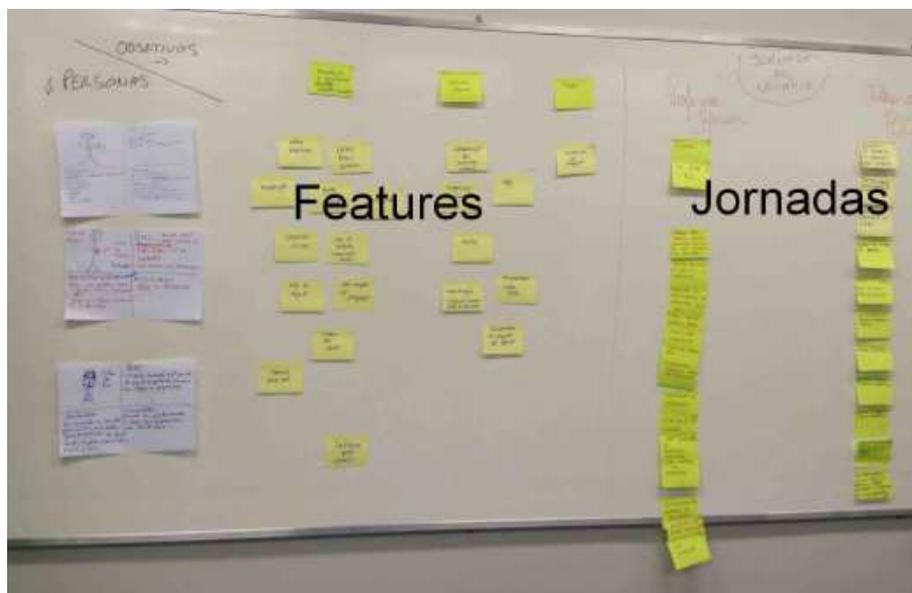
Após a classificação das *features* e após finalizado o mapeamento das jornadas do usuário, estes recursos devem ser relacionados para que se possa avaliar como será a interação do usuário com o produto. Desta forma, a jornada se apresentará como uma sequencia ordenada de passos e as *features* disponíveis como cartões avulsos. Apenas duas atividades são necessárias:

- Colocar as jornadas principais e as *features* visíveis, se possível, lado a lado, conforme ilustrado na **Figura 35**;

Classificação da Informação: **Uso Irrestrito**

- Seguindo a jornada, buscar as *features* necessárias para cada passo (Figura 36).

**Figura 35 – Exemplo de Features x Jornadas**



Fonte: CAROLI (2015, pág. 67)

**Figura 36 – Exemplo de Features vinculadas às Jornadas**



Fonte: CAROLI (2015, pág. 72)

### 1.3.3.8 Planejamento dos MVPs

Ao final da “Concepção” as *features* já possuem alguns parâmetros atribuídos, tais como: valor de negócio (\$, \$\$ ou \$\$\$), esforço (E, EE ou EEE) e marcação de nível de entendimento/ incerteza (cartão de cor verde, amarela ou vermelho, identificando, respectivamente, nível de incerteza baixo, médio ou alto). E estes parâmetros contribuirão no planejamento das *features* e suas prioridades relativas, com foco no MVP.

Neste contexto, para a visão de MVPs, utiliza-se o Canvas MVP, o produto final da Concepção, cujo conceito já foi anteriormente abordado. O Canvas organiza e planeja os lançamentos do produto além do primeiro MVP, considerando uma combinação mínima de *features* que devem ser disponibilizadas para validar um pequeno conjunto de hipóteses sobre o negócio.

Para conceituar o primeiro MVP e seus incrementos subsequentes, deve-se planejar uma sequência de ondas para agrupamento das *features* visando organizar a produção como algo fácil de entender e compreender, uma onda após a outra, em sequência, preferencialmente tendo o suporte de um flipchart ou quadro branco para o desenho do *template*, que receberá os *cards* das *features*, além dos rótulos com a identificação dos MVPs.

Para a adição de *features* a cada onda, CAROLI (2015, pág. 75) defende a existência de seis regras de planejamento:

1. Uma onda pode conter no máximo três *features*;
2. Uma onda não pode conter mais de uma *feature* em cartão vermelho;
3. Uma onda não pode conter três *features* somente em cartões amarelos e vermelhos;
4. A soma de esforço das *features* não pode ultrapassar cinco elementos do tipo “E”;
5. A soma de valor das *features* não pode ser menos de quatro elementos do tipo “\$”;
6. Uma onda tem de conter no mínimo duas *features*.

Após atribuir *features* às ondas, é chegado o momento de entender os incrementos e a criação evolutiva do produto. Isso significa que uma ou mais ondas, onde já foram atribuídas as *features*, serão agrupadas em um ou mais MVPs – como uma versão simples do produto que pode ser disponibilizada ao usuário (vide exemplo na **Figura 37**).

**Figura 37 – Exemplo de Canvas MVP**



Fonte: CAROLI (2015, pág. 72)

#### 1.4 Estratégias de Contratação de Software

Os modelos de desenvolvimento de software, na perspectiva das organizações que contratam serviços de software, podem ser trabalhados através de duas estratégias de *sourcing*: o *insourcing* e o *outsourcing*.

Enquanto a estratégia do *insourcing* compreende a execução de serviços de desenvolvimento de software por um braço especializado da própria organização, ou seja, os colaboradores ou prestadores de serviços estão sob a sua gestão direta, o termo “*outsourcing*” (ou “terceirização”), segundo NETO (2014), refere-se ao ato de transferir a responsabilidade por um determinado serviço, operação ou fase de um processo de produção ou comercialização de uma empresa para outra (ou outras),

denominada “terceira”. Sendo assim o termo em inglês “*outsourcing*” e o termo em português “Terceirização” podem ser utilizados como sinônimos.

#### 1.4.1 Estratégias de Terceirização

Para NETO (2014), a terceirização é um método administrativo que possibilita o estabelecimento de um processo gerenciado de transferência, a terceiros, das atividades acessórias e de apoio das empresas, permitindo-as concentrar-se no seu negócio principal (*core business*).

As organizações que utilizam a estratégia de *outsourcing* de TI, ou seja, àquelas que adquirem software e/ou serviços, assim como produtos relacionados à tecnologia da informação, quando em quantidade significativa, por vezes possuem um processo estruturado e/ou uma estrutura funcional física ou conceitual para tratamento dos assuntos relacionados à gestão de fornecedores e/ou contratos de tecnologia da informação. E este processo tem como responsabilidade suportar e monitorar as necessidades de aquisições no contexto que lhe foi atribuído na organização. Neste contexto, e visto a necessidade expressa pelo estudo em questão, é importante destacar que em uma relação de terceirização gerenciada, são aplicáveis custos de transação (NETO, 2014), os quais consistem em trocas (transações) entre empresas ou mesmo dentro destas, que podem ser classificados da seguinte forma:

- Custo de informação – relacionados à busca de informações a respeito de produtos, preços, insumos e compradores ou fornecedores;
- Custos de negociação – surgem do ato da transação, tal como a negociação e a elaboração de contrato ou o pagamento a um intermediário;
- Custos de monitoramento – são custos que surgem após a troca ter sido negociada, por exemplo: custos relativos ao monitoramento da qualidade de atendimento do fornecedor, para garantir que o acordo contratual está sendo seguido.

Existem ainda, alguns elementos da atividade econômica produtiva pressupostos pela abordagem de custos de transação, que são:

- Racionalidade limitada – a capacidade de decisão das pessoas é limitada e nem sempre racional;
- Oportunismo: em algumas situações, uma empresa ou outra buscará atuar em seu proveito próprio, mas isso não significa que o tempo todo assim o serão;
- Especificidade de ativos – onde uma empresa investiu na operação e a outra tenta apropriar-se do retorno obtido;
- Informação assimétrica – uma das partes possui informação mais qualificada e a utiliza eventualmente de modo oportunista.

Quanto ao “Regime de Execução” via terceirização, o SISP (2014) cita a execução indireta sob qualquer dos seguintes regimes:

- Empreitada por preço global - quando se contrata a execução da obra ou do serviço por preço certo e total;
- Empreitada por preço unitário - quando se contrata a execução da obra ou do serviço por preço certo de unidades determinadas;
- Tarefa - quando se ajusta mão-de-obra para pequenos trabalhos por preço certo, com ou sem fornecimento de materiais;
- Empreitada integral - quando se contrata um empreendimento em sua integralidade, compreendendo todas as etapas.

#### *1.4.1.1 Terceirização com Métodos Ágeis*

Em 2013, o Tribunal de Contas da União (TCU), realizou um estudo amplo sobre a contratação de serviços de software ágil nas principais organizações públicas que naquela época utilizavam tais métodos de trabalho. São elas: Banco Central do Brasil (BACEN), o Tribunal Superior do Trabalho (TST), o Instituto do

Patrimônio Histórico e Artístico Nacional (IPHAN), o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) e o Supremo Tribunal Federal (STF). Estas empresas, no período em que foi realizado o referido estudo, haviam licitado os objetos destacados no **Quadro 2**, onde o TST utilizou a unidade base de contratação em Horas de Serviço Técnico (HST) e os demais órgãos contrataram utilizando pontos de função.

**Quadro 2 – Objetos Contratados em Órgãos Públicos com Métodos Ágeis**

Órgão	Pregão	Tipo Obj.	Contratada
TST	146/2012	FSW	Squadra Tecnologia em Software Ltda.
BACEN	DEMAP 7/2012	FSW	Politec Tecnologia da Informação S.A.
IPHAN	2/2011	Projeto	EGL Engenharia Ltda.
	Não informado	FSW	Não se aplica
INEP	1/2010	FSW	Squadra Tecnologia em Software Ltda.
	14/2012	FSW	Cast Informática S.A.
STF	84/2012	FSW	Basis Tecnologia da Informação S.A.

*Fonte: TCU (2013).*

Sobre o planejamento, gestão de demandas, aceitação de produto e forma de pagamento, é importante destacar alguns resultados da pesquisa executada pelo TCU (2013):

- Uma demanda para construção de um produto é precedida pelo planejamento do produto, que pode ser realizado apenas pela instituição contratante ou em conjunto, entre essa e a empresa contratada;
- As instituições visitadas pelo TCU (2013) emitem uma ordem de serviço por ciclo, iteração ou sprint, ou por release de software, sendo mais comum o primeiro caso;
- A aceitação do produto entregue pela contratada, embora no *framework* Scrum seja preceituado que ocorra na reunião de revisão da sprint, essa prática não é executada nos contratos estudados, algumas instituições apenas verificam se os artefatos exigidos foram entregues, caracterizando o recebimento provisório (devido a questões legais);
- Constatou-se que algumas instituições remuneraram os serviços de planejamento quando realizados, enquanto outras remuneraram apenas os serviços de construção do software;

- Algumas instituições, em um determinado período de tempo, enquanto a empresa contratada executava a construção do software em um ciclo, iteração ou sprint, a contratante preparava os itens do backlog do produto, para serem implementados no próximo ciclo e homologava o produto entregue no ciclo anterior;
- A forma de pagamento não segue um padrão, e cada órgão a trata de uma forma, como exemplo:
  - No IPHAN, são considerados os ciclos de projeto (planejamento, implantação e encerramento), destinando um percentual da quantidade estimada de pontos de função para remuneração do Planejamento fixa em 5%, prevendo uma revisão ao final, uma vez que a quantidade final de pontos de função pode não corresponder aos pontos inicialmente estimados. Para as fases de Execução e Encerramento, respectivamente, são destinados 80 e 15% do que foi efetivamente entregue.
  - No STF, o pagamento das ordens de serviço é calculado com base no tamanho funcional do produto homologado em pontos de função brutos e na efetiva execução do serviço. Ele é feito em duas etapas: a primeira, correspondente a 90%, é realizada após o recebimento definitivo da OS, referente à última sprint de cada produto; a segunda, correspondente a 10%, é realizada somente ao final do período de garantia, estipulado em 180 dias (peça 25, p. 21 e 39).
- Em relação a níveis de serviço, observou-se que quase todas as instituições visitadas instituíram vários níveis de serviço a serem atendidos pela contratada e que o não atendimento a critérios mínimos aceitáveis implicaria a redução do valor a ser pago à contratada, são alguns exemplos de indicadores:
  - No TST foi instituído o Índice de Atraso na Entrega de OS, o qual tem por objetivo garantir a entrega da OS no prazo estimado para a sua conclusão, tolerando atrasos inferiores a 10% do prazo para a entrega da OS como aceitável, o que se afasta dos preceitos do

- framework Scrum, no qual uma sprint deve ser executada no prazo estabelecido;
- O BACEN também instituiu um indicador referente ao Atraso na Entrega da OS de projeto de construção, tolerando, assim como o TST, pequenos atrasos;
  - O TST criou o indicador Desconformidade na OS (DOS) para assegurar qualidade da OS, bem como nas listas de verificação para a avaliação dos artefatos produzidos, e caso alguma desconformidade seja detectada, o produto, e, conseqüentemente, a Ordem de serviço, é rejeitada.
  - O Bacen definiu o Índice de Defeitos por Pontos de Função da OS e o Índice de Defeitos Impeditivos por Pontos de Função da OS, onde para o primeiro indicador há tolerância para a aceitação da OS, enquanto para o segundo a tolerância é zero;
  - No IPHAN foi definido o Índice de Pontos Executados (IPE), um indicador que se destina a avaliar a qualidade dos produtos entregues, medindo a quantidade de pontos de função brutos executados e aceitos da ordem de serviço. Caso o IPE seja inferior ou igual a 80%, multas podem ser aplicadas à contratada – caso algum produto construído no ciclo não seja aprovado, este poderá ser considerado perdido, devendo um novo ciclo ser iniciado para adequação desse produto;
  - No STF, por sua vez, o Índice de Desconformidade do Produto afere a qualidade dos produtos entregues pela contratada.
- Sobre os valores ágeis, constatou-se:
    - A preocupação de todas as instituições com relação à entrega de artefatos de documentação associados ao software produzido a cada iteração, facilitando, por exemplo, futuras manutenções por terceiros alheios ao processo de desenvolvimento;
    - Que a relação contratual prevalece sobre a possível colaboração entre as partes.

- Ao final da análise realizada pelo TCU (2013), foram mapeados os seguintes riscos relacionados às contratações utilizando métodos ágeis no âmbito público, os quais se constituem de pontos de atenção:
  - 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;
  - Alteração da metodologia ágil adotada no instrumento convocatório no decorrer da 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;
  - Falta de comprometimento ou colaboração insatisfatória do responsável indicado pela área de negócios (Product Owner) no desenvolvimento do software;
  - Falta do conhecimento necessário do indicado pela área de negócios (Product Owner) para o desenvolvimento do software;
  - Excessiva dependência da visão do indicado pela área de negócios (Product Owner);
  - 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 (Product Owner);
  - 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 a utilização e avaliação dos reais usuários;
- Forma de pagamento não baseada em resultados.

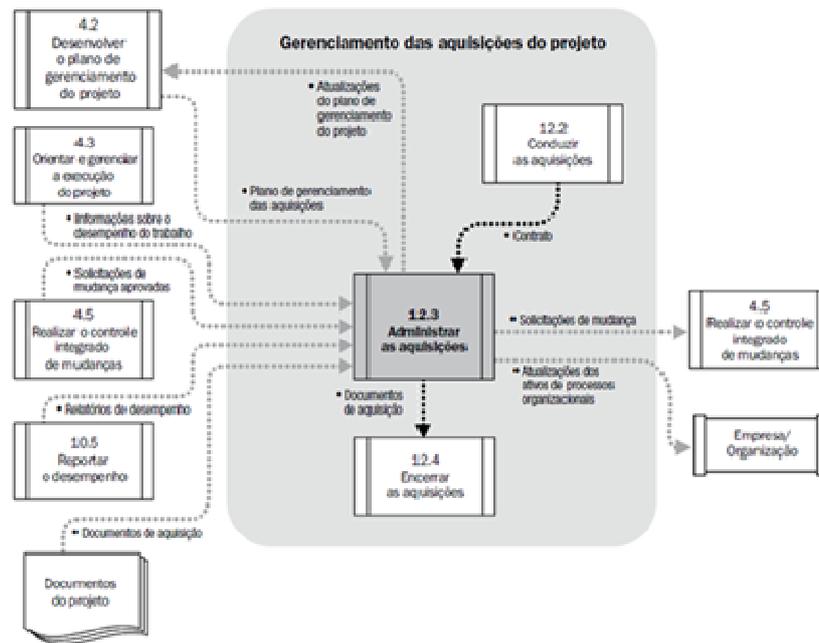
#### *1.4.1.2 Aspectos de Aquisição e de Gestão de Fornecedores*

Segundo o PMBOK quarta edição (PMBOK, 2008), gerenciar aquisições em um projeto é usar os processos sugeridos que se fazem necessários conforme o caso, para a aquisição de um ou mais produtos ou serviços como resultados externos à equipe do projeto. Na publicação, as tarefas relacionadas às contratações foram organizadas nos seguintes quatro processos:

- Planejar as aquisições;
- Conduzir as aquisições;
- Administrar as aquisições (Figura 2);
- Encerrar as aquisições.

Os processos citados, além de se inter-relacionarem, são idealmente executados ao menos uma vez em cada projeto, ou ciclicamente (se for um projeto organizado por fases). Eles também podem ocorrer em outras áreas de conhecimento.

#### **Figura 38 – Contexto de Administração das Aquisições**



Fonte: PMBOK (2008, p. 336).

Ressalta-se ainda, que os seguintes processos do PMBOK podem ser aplicados na administração das aquisições:

- Orientar e gerenciar a execução do projeto: autorizando o trabalho do fornecedor na ocasião apropriada;
- Reportar o desempenho: monitorando o escopo do contrato, os custos, o cronograma e o desempenho técnico;
- Realizar o controle da qualidade: inspecionando e verificando a adequação do produto do fornecedor;
- Realizar o controle integrado de mudanças: garantindo que as mudanças sejam aprovadas de forma adequada e certificando que todas as pessoas envolvidas estejam cientes dessas mudanças;
- Monitorar e controlar os riscos: garantindo a mitigação dos riscos.

Em complemento, segundo o MPS-BR (SOFTEX, 2013, p. 22 e 23), que também apresenta processos para gestão de contratos e fornecedores, é possível quantificar a evolução do projeto, visualizando se um produto está em conformidade ou como evoluiu em termos de:

- Atendimento aos requisitos;

- Custos e prazos;
- Riscos envolvidos;
- Nível de problemas que estão sendo enfrentados;
- Aderência ao processo que foi contratado;
- Níveis de retrabalho encontrado.

Sendo assim, relacionando os itens estudados aos processos e métricas que permitam tal controle, identificam-se componentes fundamentais para a tomada de decisão, como:

- Revisão de prazos e requisitos;
- Alocação de recursos;
- Descarte de recursos;
- Interrupção de atividades;
- Aceitação de artefatos;
- Rejeição de artefatos;
- Aplicação de penalidades;
- Aplicação de prêmios;
- Solicitação do envolvimento de interessados (stakeholders);
- Até mesmo a interrupção do contrato.

Salienta-se, contudo, que este é um tópico bastante abrangente e abordado apenas para efeito de introdução ao conteúdo, visto que possui forte relação com o contexto geral do trabalho.

#### 1.4.2 Planejamento, Execução e Gestão de Contratos

A seguir, serão abordados assuntos relacionados ao planejamento das contratações de software, conforme sua afinidade com o tema deste trabalho.

Conforme citado anteriormente, os contratos de desenvolvimento e manutenção de software são peculiares em termos das suas características, formas e normas. E visando apresentar alternativas para melhor operá-los e geri-los,

veremos a seguir algumas regras e práticas constatadas em guias e contratos mantidos por órgãos do Governo do Brasil, os quais disponibilizam publicamente e de forma integral os artefatos utilizados, facilitando o acesso a um conjunto rico de informações, que em outras organizações, seriam muito provavelmente consideradas confidenciais.

#### *1.4.2.1 Objetos de Contratação*

De acordo com o SISP (2014, pág. 56), um guia aplicado a contratações públicas em um conjunto específico de órgãos, no seu contexto de aplicação, a definição do objeto da contratação deve ser precisa, suficiente e clara, sendo vedadas especificações excessivas, irrelevantes ou desnecessárias, que limitem ou frustrem a competição ou o fornecimento dos serviços.

Sendo assim, na visão do SISP (2014), um objeto de contratação deve contemplar aspectos tais como os seguintes:

- Não tecnológicos (SISP, 2014, pág. 56), como:
  - Requisitos de negócio;
  - Capacitação;
  - Requisitos legais;
  - Requisitos de manutenção;
  - Requisitos temporais;
  - Requisitos de segurança;
  - Requisitos sociais;
  - Requisitos ambientais;
  - Requisitos culturais.
- De contratação (SISP, 2014, pág. 57 e 58), dentre os quais:
  - Arquitetura tecnológica, composta de hardware, software, padrões de interoperabilidade, linguagens de programação, interfaces, dentre outros;

- Projeto e implementação, que estabelecem o processo de desenvolvimento de software, técnicas, métodos, forma de gestão, de documentação, dentre outros;
- Implantação, que definem o processo de disponibilização da solução em ambiente de produção, dentre outros;
- Garantia e manutenção, que definem a forma como será conduzida a manutenção e a comunicação entre as partes envolvidas;
- Capacitação, que definem o ambiente tecnológico dos treinamentos a serem ministrados, os perfis dos instrutores, dentre outros;
- Experiência profissional da equipe que projetará, implementará e implantará a Solução de Tecnologia da Informação, que definem a natureza da experiência profissional exigida e as respectivas formas de comprovação dessa experiência, dentre outros;
- Formação da equipe que projetará, implementará e implantará a Solução de Tecnologia da Informação, que definem cursos acadêmicos e técnicos, formas de comprovação dessa formação, dentre outros;
- Metodologia de trabalho;
- Segurança da informação; e
- Requisitos aplicáveis.

O SISP (2014) orienta ainda, que devem ser avaliadas contratações separadas para os itens que possam ser divididos, em tantas parcelas quantas se comprovarem técnica e economicamente viáveis, de forma a obter o melhor aproveitamento dos recursos disponíveis no mercado. Prevê-se, ainda, que cada serviço ou produto do lote, deverá ser discriminado em itens separados nas propostas de preços, de modo a permitir a identificação do seu preço individual.

#### 1.4.2.2 Habilitação e Homologação de Fornecedores

O SISP (2014, pág. 111) orienta os órgãos conveniados, a descrever os critérios técnicos que serão utilizados para habilitação dos concorrentes, seguidos das respectivas justificativas que levaram à sua exigência, considerando o objetivo da contratação e a legislação pertinente. Recomenda ainda, que para cada papel a ser desempenhado pela contratada, exista uma indicação dos requisitos de capacitação necessários para execução do contrato.

Sendo assim, a habilitação deverá ser realizada conforme o conjunto de critérios técnicos (pontuáveis), no caso de uma concorrência do tipo “técnica e preço”, utilizando parâmetros como os que seguem:

- Critério Técnico Pontuável: Descrição do critério pontuável a ser contabilizado;
- Pontuação: Descrição da pontuação relacionada ao critério escolhido;
- Percentual (%): Definição do percentual que o mesmo representa em relação à pontuação total;
- Justificativa: Justificativa motivada para a escolha do critério relacionado.

No modelo citado, deverá ser considerada também a descrição dos critérios de julgamento, conforme os requisitos habilitatórios definidos, os quais não devem contrariar as normas e regras previamente estabelecidas.

O SISP (2015b) dispõe que não é suficiente ter a contratante preparada, com pessoas capacitadas e processos definidos para gestão e fiscalização de projetos de software com práticas de métodos ágeis, pois quando se utiliza terceirização do desenvolvimento de software, devem ser definidos também, critérios de seleção de fornecedor que levem à seleção de empresa capacitada e com preço exequível para a prestação desses serviços de desenvolvimento.

### 1.4.2.3 Execução da Contratação

Segundo o SISP (2014, pág. 60), o modelo de execução de uma contratação de software deve ser elaborado de forma a contemplar as condições para fornecimento da solução a ser contratada. E durante a sua elaboração, deve-se atentar para a fixação de rotinas de execução como: prazos, horários de fornecimento ou prestação de serviço, local de entrega, documentação mínima exigida para os padrões de qualidade. Também devem ser previstas a quantificação e estimativa prévia de volume de serviços, definindo mecanismos formais de comunicação para troca de informação, forma de pagamento, entre outros.

Neste modelo, durante a execução contratual, a contratante encaminha Ordens de Serviço (OS), sendo que este instrumento tem por atribuição, formalizar a execução dos serviços.

Segundo o SISP (2014), uma Ordem de Serviço (OS) deve conter, idealmente:

- Definição e a especificação dos serviços a serem realizados ou bens a serem fornecidos;
- Volume de serviços a serem realizados ou a quantidade de bens a serem fornecidos, segundo as métricas definidas;
- Cronograma de realização dos serviços ou entrega dos bens, incluídas todas as tarefas significativas e seus respectivos prazos; e
- Identificação dos responsáveis pela solicitação do bem ou serviço na área requisitante da solução.

O SISP (2014) orienta que depois de recebidos todos os itens constantes da OS, deve-se elaborar um “Termo de Recebimento Provisório”, um instrumento que deve ser entregue ao Preposto da Contratada, informando que os itens constantes da Ordem de Serviço foram entregues para a contratante, para sua avaliação.

O requisitante então avaliará a qualidade dos itens recebidos, de acordo com os critérios de aceitação definidos no contrato, identificando, sendo o caso, as não

conformidades. E caso não sejam identificadas inconformidades, o mesmo deverá emitir o “Termo de Recebimento Definitivo”.

#### *1.4.2.4 Monitoramento e Gestão da Contratação*

Após definir o modelo de execução do contrato, ainda segundo o SISP (2014), deverá ser elaborado o modelo de gestão do contrato, que contemplará as condições de gestão e fiscalização do contrato de fornecimento da solução de TI. Nesse Modelo devem ser planejados os seguintes aspectos:

- Fixação dos “Critérios de Aceitação” dos serviços,
- Procedimentos de teste e inspeção que servirão de base para os termos de “Recebimento Provisório” e “Definitivo”;
- Fixação de valores e procedimentos para retenção ou glosa de pagamento;
- Definição clara e detalhada das sanções administrativas;
- Procedimentos para emissão de nota fiscal e pagamento, descontados os valores oriundos da aplicação de eventuais glosas.

#### 1.4.3 Instrumentos

Este tópico apresentará os instrumentos pré-contratuais e contratuais para contratações de fornecedores de software conforme o modelo de contratações do SISP (2014), utilizado como principal referência ao assunto em questão:

#### 1.4.3.1 *Termo de Referência*

O SISP (2014) orienta que um “Termo de Referência” deve conter no mínimo as seguintes informações sobre o contexto de soluções de tecnologia de informação que poderão se contratadas:

- Definição do objeto;
- Justificativa da contratação;
- Descrição da Solução de TI;
- Especificação técnica;
- Deveres e responsabilidades da contratante;
- Deveres e responsabilidades do órgão gerenciados do registro de preços;
- Deveres e responsabilidades da contratada;
- Modelo de execução do contrato;
- Modelo de gestão do contrato;
- Estimativa de preço;
- Adequação orçamentária;
- Critérios de seleção do fornecedor.

O Termo de Referência deve descrever um ou mais objetos que podem ser cotados e/ou contratados no mercado, sendo este artefato muito utilizado em licitações públicas, com o propósito de definir e delimitar o contexto e os processos de software da organização, os quais são complementares aos aspectos padrões considerados no instrumento principal (o edital ou RFP, a depender da organização ser um órgão público ou uma entidade privada).

#### 1.4.3.2 *Termo de Ciência (TA)*

Artefato utilizado para obter o comprometimento formal dos empregados da contratada diretamente envolvidos no projeto sobre o conhecimento da declaração de manutenção de sigilo e das normas de segurança vigentes na Instituição. O

Classificação da Informação: **Uso Irrestrito**

documento é assinado por todos os funcionários da contratada envolvidos no projeto com acesso as suas informações, mesmo que o ingresso nas suas dependências não seja necessário para a execução (SISP, 2014).

Alguns dados sugeridos para este artefato são os que seguem:

- Contrato – número do contrato;
- Objeto do Contrato – descrição do objeto;
- Contratante – identificação da contratante;
- Gestor do Contrato – nome do gestor do contrato;
- Matrícula – matrícula do gestor do contrato;
- Contratada – identificação da empresa contratada para prover a solução;
- CNPJ – CNPJ da empresa contratada;
- Preposto – nome do preposto da contratada;
- CPF – CPF do preposto da contratada.

#### 1.4.3.3 Termo de Compromisso (TM)

Compreende a assinatura e entrega pela contratada, de um artefato, cujo template foi previamente estabelecido pela contratante com vistas a demonstrar as condições de manutenção de sigilo e ciência das normas de segurança vigentes na organização. Este documento deve ser assinado por um representante legal da organização contratada.

#### 1.4.3.4 Ordem de Serviço (OS)

O artefato de Ordem de Serviço é um documento utilizado para solicitar formalmente à contratada a prestação de serviço ou fornecimento de bens relativos ao objeto do contrato, sendo que a requisição encaminhada deve conter (SISP, 2014):

- A definição e a especificação dos serviços a serem realizados ou bens a serem fornecidos;
- O volume de serviços a serem realizados ou a quantidade de bens a serem fornecidos, segundo as métricas definidas;
- O cronograma de realização dos serviços ou entrega dos bens, incluídas todas as tarefas significativas e seus respectivos prazos; e
- A identificação dos responsáveis pela solicitação do bem ou serviço na Área Requisitante da Solução.

#### *1.4.3.5 Termo de Recebimento Provisório (TP)*

O Termo de Recebimento Provisório (TP) deve ser entregue ao preposto da contratada e visa garantir que os itens constantes da Ordem de Serviço (OS) foram entregues à Contratante para avaliação (SISP, 2014).

Após receber o objeto contratado e emitir o Termo de Recebimento Provisório, a contratante deve avaliar a qualidade dos itens recebidos com apoio de listas de verificação (checklists) e de acordo com os critérios de aceitação definidos no Contrato, identificando, caso ocorram, as não conformidades.

#### *1.4.3.6 Termo de Recebimento Definitivo (TD)*

O Termo de Recebimento Definitivo (TD) tem como objetivo declarar formalmente para a contratada, de que os serviços prestados ou os bens fornecidos foram devidamente avaliados e atenderam aos requisitos estabelecidos em contrato (SISP, 2014).

Abaixo, alguns dos dados sugeridos para este artefato:

- Nº do Contrato – número do contrato a que o termo se refere;

- Nº da OS – identificação da Ordem de Serviço (OS) que formalizou a execução dos serviços ou a entrega dos bens objetos do termo;
- Objeto – relação sucinta do objeto do termo, ou seja, dos serviços ou bens entregues para verificação;
- Contratante – identificação da Instituição que recebe os serviços ou bens;
- Contratada – identificação da empresa responsável pela execução da OS.

#### 1.4.3.7 Termo de Encerramento do Contrato (TE)

O instrumento Termo de Encerramento do Contrato (TE) deve ser utilizado para encerrar formalmente o pacto contratual entre a contratante e a contratada (SISP, 2014). Os seguintes campos podem ser contemplados:

- Nº Contrato – número do contrato a que o termo se refere;
- Objeto – descrição sucinta do objeto contratual;
- Contratante – identificação da organização contratante;
- Contratada – identificação da empresa contratada;
- Motivo do Encerramento – descrição do motivo do encerramento.

## 1.5 Estimativas e Medição de Software

Alan Albrecht, funcionário da IBM ao final da década de 1970, visando minimizar problemas relacionados à previsão do esforço de desenvolvimento do projeto de software, criou um método de estimativas que considerava a visão do usuário como principal entrada do processo de medição de tamanho, em detrimento aos métodos direcionados à medição física, e, portanto, voltados à forma de implementação (ALBRECHT, 1979). Tal foi batizado como “Análise de Pontos de Função” (APF) e se tornou muito conhecido a partir da década de 1980, através da

sua comunidade de usuários, o IFPUG<sup>2</sup> (2010), cujo propósito era (e ainda é) garantir a continuidade e manutenção do método - que também possui um manual, o CPM (*Counting Practices Manual*).

A APF se tornou na década de 1990, um padrão de *facto*, em conformidade com a ISO/IEC 14.143-1 (que trata do processo de medição de tamanho funcional) e instaurou um padrão próprio (a ISO/IEC 20926). Neste contexto, a saber, a ISO/IEC 14143-1 distingue dois subconjuntos de requisitos do usuário (RU): Requisitos Funcionais do Usuário (RFU) e Requisitos Não-Funcionais (RNF), definindo a medição funcional, atuante sobre o primeiro grupo de requisitos (IFPUG, 2010).

Nos últimos anos também foram criados outros métodos de medição de tamanho, onde se destacam à luz da norma ISO/IEC 14143-1: o SNAP (2015) e o método COSMIC (2015):

- O SNAP (Processo de Avaliação Não Funcional) foi criado recentemente pelo IFPUG e cobre a medição de requisitos não funcionais de software – foi desenvolvido para ser utilizado em junto com a APF;
- O COSMIC apresenta-se com uma nova geração de métricas, com outra interpretação sobre os requisitos funcionais do usuário, acabando por cobrir também os aspectos de desenvolvimento de software em geral, de forma bastante flexível, com certa equivalência ao escopo da APF e SNAP – é mantido pela organização de mesmo nome e tem recebido um bom retorno sobre a sua utilização e aplicabilidade, tanto na sua comunidade, como dentre os adeptos de métodos do IFPUG.

A seguir, serão apresentados os métodos na abordagem dos principais métodos atualmente utilizados, representando, portanto, a visão do IFPUG e do COSMIC sobre este assunto e também vinculando estudos complementares relacionados.

---

<sup>2</sup> *International Function Points Users Group*

### 1.5.1 Abordagem do IFPUG

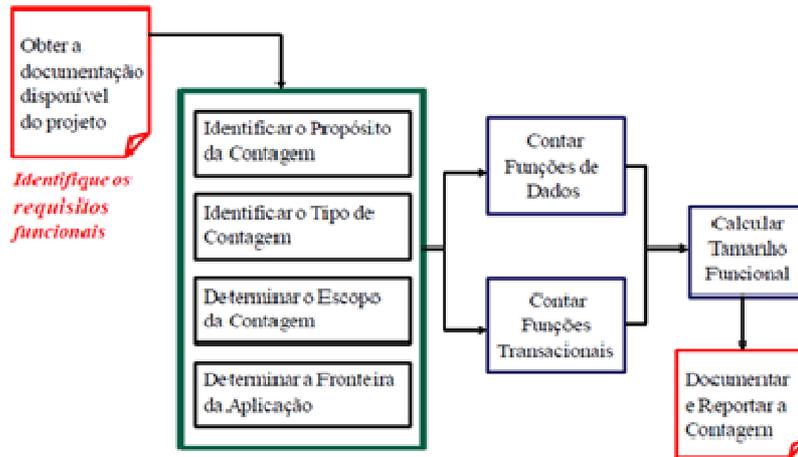
Neste trabalho, foi denominado “Abordagem do IFPUG”, o conjunto de métodos e práticas de contagem de pontos de função mantidos pelo órgão IFPUG e também as técnicas e conceitos associados. Sendo assim, foram considerados na perspectiva do estudo: os métodos de estimativa gerados com base na Análise de Pontos de Função (APF) e no Processo de Avaliação Não Funcional (SNAP); e as estimativas derivadas a outras dimensões, inclusive com contribuições de outros autores senão o próprio IFPUG.

#### *1.5.1.1 Medições de Tamanho Funcional com APF*

A Análise de Pontos de Função (APF) é o método padrão de medição funcional do IFPUG, utilizado para prover a quantificação do tamanho funcional a partir da visão do usuário e com base nos “Requisitos Funcionais do Usuário” (RFU). Desta forma, ele é aplicado independente da forma de implementação ou estratégia de projeto adotada.

A APF possibilita a realização de uma estimativa ou medição de tamanho funcional, onde o processo de medição necessita de um maior detalhamento dos requisitos para ser executado, mas é mais confiável, pois provê a quantificação exata dos itens do escopo de um projeto, de acordo com sua complexidade. Ressalta-se, portanto, que neste contexto, uma estimativa funcional, na visão do IFPUG é apenas uma aproximação do tamanho funcional utilizando-se da APF.

### **Figura 39 – Processo de Medição Funcional**



Fonte: IFPUG (2010).

O já referido método, que possui como unidade base o “Ponto de Função”, possibilita quantificar a funcionalidade solicitada e entregue sob o ponto de vista do usuário em termos de tarefas e serviços, ou ainda de suas estruturas de dados, que são classificadas em funções de transação ou funções de dados, abrangendo um total de cinco tipos de função possíveis (**Quadro 3**) e três níveis de complexidade que podem ser associados (**Quadro 4**): alto, médio e baixo.

**Quadro 3 – Tipos de Função na APF**

Funções	Tipos de Funções		
	Sigla	Nome	Conceito
De Dados	ALI	Arquivo Lógico Interno	Estrutura formada por grupos de dados relacionados e que armazena informações internas ao sistema.
	AIE	Arquivo de Interface Externa	Estrutura formada por grupos de dados relacionados e que armazena as informações externas ao sistema (mantida por outro sistema e apenas referenciadas pelo sistema em foco).
De Transação	EE	Entrada Externa	Processo elementar que primariamente mantém um ALI ou modifica o comportamento da aplicação.
	CE	Consulta Externa	Processo elementar que primariamente consulta dados de um ou mais ALI ou AIE e apresenta-os a um usuário.
	SE	Saída Externa	Processo elementar que primariamente consulta dados de um ou mais ALI ou AIE e apresenta-os a um usuário, mas adicionalmente executa cálculos, fórmulas, mantém um ALI e/ou altera o comportamento do sistema.

Fonte: elaborado pelo autor, com base em IFPUG (2010).

Em relação às funções de dados, conforme o IFPUG (2010), a partir da sua classificação como um ALI ou um AIE, identifica-se uma complexidade correspondente com base em critérios objetivos (**Quadro 4**). Isso ocorre a partir da quantificação dos dados de negócio e de referência mantidos ou referenciados pela aplicação na função específica sendo contada (estes dados são chamados Dados

Classificação da Informação: **Uso Irrestrito**

Elementares Referenciados, ou simplesmente “DER”) e do quantitativo de grupos lógicos de dados desta função específica (chamados RLR, ou Registros Lógicos Referenciados).

**Quadro 4 – Complexidade em Funções de Dados**

Qtde RLR	Qtde DER		
	De 1 a 19	De 20 a 50	51 ou mais
Apenas 1	Baixa	Baixa	Média
De 2 a 5	Baixa	Média	Alta
6 ou mais	Média	Alta	Alta

*Fonte: elaborado pelo autor, com base em IFPUG (2010).*

Se a contagem for realizada sobre uma função de transação, ela será classificada como uma EE, CE ou SE. Em seguida, deverão ser identificados e quantificados os dados utilizados na interface entre os usuários, separando-os do que é interno ao sistema sendo contado (chamamos “DER” também para as funções de transação). Depois, avalia-se o total de Arquivos Lógicos Referenciados (chamados ALRs e que compreendem os arquivos lógicos mantidos ou referenciados pela função), e pode-se identificar a complexidade da função de transação, conforme os critérios que constam no **Quadro 5** e no **Quadro 6** abaixo (IFPUG, 2010).

**Quadro 5 – Complexidade do Tipo de Função EE na APF**

Qtde ALR	Qtde DER		
	De 1 a 4	De 5 a 15	16 ou mais
0 ou 1	Baixa	Baixa	Média
2	Baixa	Média	Alta
3 ou mais	Média	Alta	Alta

*Fonte: elaborado pelo autor, com base em IFPUG (2010).*

**Quadro 6 – Complexidade dos Tipos de Função CE e SE na APF**

Qtde ALR	Qtde DER		
	De 1 a 5	De 6 a 19	20 ou mais
0* ou 1	Baixa	Baixa	Média
2 ou 3	Baixa	Média	Alta
4 ou mais	Média	Alta	Alta

*Fonte: elaborado pelo autor, com base em IFPUG (2010).*

Para cada tipo de função e complexidade identificada, há ainda uma ponderação a ser considerada, chamada “contribuição”, sendo que a relação de complexidade *versus* contribuição pode ser visualizada na **Tabela 1**.

**Tabela 1 – Complexidade versus Contribuição (IFPUG)**

Tipo de Função	Complexidade		
	Alta	Média	Baixa
ALI	15	10	7
AIE	10	7	5
EE	6	4	3
CE	6	4	3
SE	7	5	4

*Fonte: elaborado pelo autor, com base em IFPUG (2010).*

Destaca-se, que é muito comum, principalmente no Brasil, a realização de contratações utilizando pontos de função como unidade de referência (SISP, 2015). Neste caso, é importante, embora não essencial, estabelecer convenções locais e deflatores de produtividade para cada tipo de manutenção, tais como redutores para representar o impacto em funções alteradas, excluídas, ou apenas testadas, por exemplo, visando estabelecer uma melhor correlação entre as medidas de tamanho e as estimativas de esforço e/ou custo (SISP, 2015).

Neste sentido, e visando padronizar a aplicação do método, o Governo do Brasil, através do Ministério do Planejamento, publicou um guia com diversas necessidades consideradas comuns aos órgãos públicos na utilização de métricas, o Roteiro de Métricas do SISP (2015). Ele é aderente à Instrução Normativa 04/2010, que orienta que os contratos de software públicos sejam mantidos com base em critérios objetivos, com um maior controle sobre o que está sendo contratado através de um processo de mensuração maduro, auditável e repetível – já que os mesmos resultados podem ser obtidos por diferentes profissionais, com base em regras estabelecidas.

Além do método padrão do IFPUG, existem ainda outras métricas para estimativa e medição de tamanho funcional que possibilitam uma aplicação conjunta, tais como as técnicas NESMA Indicativa, Estimativa e de Melhoria, entre outras. Destaca-se neste contexto, o método de Estimativa NESMA (2015), que é mantido pelo órgão com este mesmo nome e foi concebido com base na APF.

### 1.5.1.2 Estimativas de Tamanho Funcional com APF

O método de Estimativa NESMA (2015) é uma alternativa às estimativas de tamanho com base na aproximação do tamanho. Ele atribui a complexidade baixa oriunda da APF para todas as funções de dados (repositórios, definidos como ALI ou AIE) e a complexidade média para todas as funções de transação (definidas como EE, CE ou SE) resultando na contribuição obtida pela análise da **Tabela 2**. Ele é normalmente utilizado em fases iniciais do ciclo de vida do desenvolvimento de software, especialmente quando ainda não é possível visualizar o detalhamento das funcionalidades do escopo, embora seja possível identificar a maioria delas.

**Tabela 2 – Tabela de Contribuição Estimada NESMA**

Tipo de Função	Complexidade
ALI	7
AIE	5
EE	4
CE	4
SE	5

*Fonte: elaborado pelo autor, com base em IFPUG (2010).*

### 1.5.1.3 Tamanho Não Funcional com SNAP

Conforme citado anteriormente, os Requisitos Não-Funcionais (RNF) são assim classificados conforme a norma ISO/IEC 14143-1: 2007, mas eles são categorizados através da norma ISO/IEC 25010: 2011, que trata o modelo de qualidade de produto e as propriedades da qualidade de acordo com oito características: adequação funcional, confiabilidade, eficiência de desempenho, usabilidade, segurança, compatibilidade, manutenibilidade; e portabilidade (IFPUG, 2015).

Para cada categoria, existe também um conjunto de subcategorias associadas, conforme a relação abaixo:

- Adequação Funcional
  - Completude Funcional
  - Correção Funcional
  - Adequação Funcional
- Eficiência de Desempenho
  - Tempo de Resposta
  - Utilização de Recursos
  - Capacidade
- Compatibilidade
  - Coexistência;
  - Interoperabilidade.
- Usabilidade
  - Reconhecimento da Adequabilidade
  - Aprendizabilidade
  - Operabilidade
  - Proteção a erros do usuário
  - Estética da interface com o usuário
  - Acessibilidade
- Confiabilidade
  - Maturidade;
  - Disponibilidade;
  - Tolerância a falhas;
  - Recuperabilidade.
- Segurança
  - Confidencialidade
  - Integridade
  - Não repúdio
  - Responsabilidade (*'accountability'*)
  - Autenticidade
- Manutenibilidade
  - Modularidade
  - Reusabilidade

- Analisabilidade
- Modificabilidade
- Testabilidade
- Portabilidade
  - Adaptabilidade
  - Instalabilidade
  - Substituibilidade

Conforme o IFPUG (2015), o tamanho não funcional de um software, assim como o tamanho funcional, pode ser medido, e isso pode ser realizado através do Processo de Avaliação Não Funcional (SNAP). Este método tem como unidade de medida o Ponto SNAP, e avalia tanto requisitos técnicos como os de qualidade. Desta forma, as organizações podem construir seus repositórios de dados históricos, com vistas a auxiliar na tomada de decisão sobre aspectos técnicos e/ou de qualidade das suas aplicações.

O objetivo principal do projeto que definiu o SNAP foi garantir que um framework não funcional pudesse ser utilizado para estabelecer uma ligação entre o tamanho não funcional e o esforço necessário para prover os requisitos não funcionais. Foram então consideradas com este propósito, as seguintes características:

- O *framework* como um todo é uma avaliação do tamanho dos requisitos não funcionais;
- O *framework* é composto de categorias e subcategorias de avaliação;
- As subcategorias são avaliadas utilizando critérios específicos;
- A avaliação utiliza ambos os critérios de avaliação e/ou de medição.

É importante ressaltar também, que o SNAP foi especialmente projetado para obter os seguintes resultados ao realizar uma avaliação não funcional:

- Pode ser utilizado em conjunto com o tamanho funcional e irá auxiliar na explicação da variação do esforço de desenvolvimento e da produtividade;
- Juntamente com o tamanho funcional, pode ser utilizado como entrada para modelos de estimativas;

- É determinado a partir de uma visão não funcional do usuário, porém entendidos e acordados entre o usuário e as equipes de desenvolvimento.

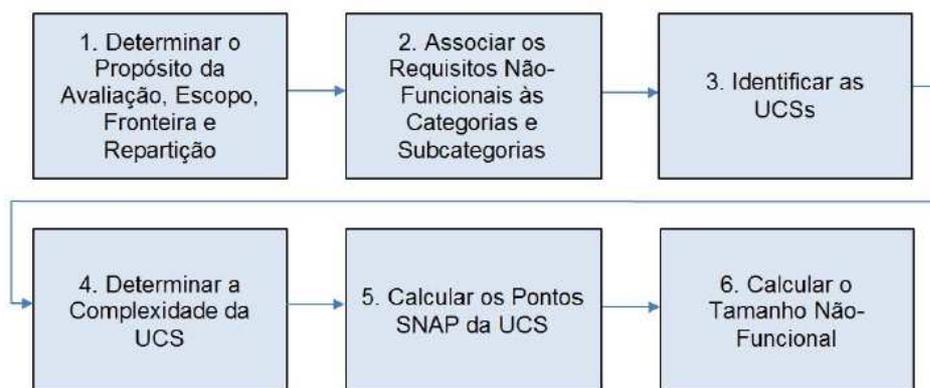
Uma avaliação não funcional, assim como uma avaliação funcional que é realizada sobre os requisitos funcionais, auxilia as organizações de tecnologia da informação de diversas maneiras, provendo informações sobre projetos e aplicações para ajudar na estimativa e itens com impacto na sua qualidade e produtividade, tornando viável aos profissionais de software (IFPUG, 2015):

- Planejar e estimar melhor os projetos;
- Identificar áreas de melhoria de processos;
- Auxiliar na determinação de estratégias técnicas futuras;
- Quantificar os impactos das estratégias técnicas atuais;
- Prover dados específicos sobre questões não funcionais na comunicação com diversos públicos.

No SNAP (IFPUG, 2015), uma avaliação realizada utiliza uma série de questões agrupadas por categorias, para medir o tamanho dos requisitos não funcionais para o desenvolvimento e a entrega do produto de software, sendo suas premissas as que seguem:

- Categorias e subcategorias focarão nos requisitos não funcionais que afetam o tamanho do produto, onde:
  - Uma categoria é um grupo de componentes, processos ou atividades que são usados com a finalidade de satisfazer o requisito não funcional, sendo que as mesmas agrupam as subcategorias com base em um mesmo nível de operações e/ou tipos similares de atividades executadas pelo processo de avaliação não funcional.
  - Uma subcategoria deriva de uma categoria e é definida como um componente, um processo ou uma atividade executada no projeto para satisfazer o requisito não funcional.
- O processo (**Figura 40**) possibilitará a medição dos requisitos não funcionais utilizando uma série de questões e medidas;
- O processo pode ser usado para Projetos de Desenvolvimento, Projetos de Melhoria, Atividades de Manutenção e Aplicações.

**Figura 40 – Processo de Avaliação Não Funcional**



Fonte: IFPUG (2015).

Desta forma, visando complementar a visão geral do método, o conjunto das categorias e subcategorias avaliadas pelo SNAP é apresentado abaixo (IFPUG, 2015):

- 1. Operações de Dados
  - 1.1. Validações na Entrada de Dados
  - 1.2. Operações Lógicas e Matemáticas
  - 1.3. Formatação de Dados
  - 1.4. Movimentações de Dados Internos
  - 1.5. Entregar valor agregado aos usuários por configuração de dados
- 2. Projeto de Interface
  - 2.1. Interfaces do Usuário
  - 2.2. Métodos de Ajuda
  - 2.3. Múltiplos Métodos de Entrada
  - 2.4. Múltiplos Métodos de Saída
- 3. Ambiente Técnico
  - 3.1. Múltiplas Plataformas
  - 3.2. Tecnologia de Banco de Dados
  - 3.3. Processos Batch
- 4. Arquitetura
  - 4.1. Software baseado em componentes
  - 4.2. Múltiplas Interfaces de Entrada / Saída

### 1.5.1.4 Estimativas Derivadas

As estimativas de esforço, prazo, custo e ainda outras, podem ser derivadas de uma medição ou estimativa de tamanho em pontos de função, sendo muito úteis ao planejamento de projetos de manutenção e desenvolvimento de software (SISP, 2015).

Em relação ao esforço, o “Modelo Simplificado de Estimativas de Esforço” é um dos métodos que possibilitam realizar cálculos de estimativas de esforço. Ele consiste na obtenção de um quantitativo em horas a partir de um índice de produtividade pré-definido e o tamanho de software correspondente ao escopo do projeto em questão (SISP, 2015). Assim, o esforço resultante do cálculo, é representado pelo produto do tamanho em Pontos de Função do projeto e do Índice de Produtividade (IP) identificado, conforme fórmula abaixo (**Fórmula 1**), proposta por VAZQUEZ (2010):

#### **Fórmula 1 – Modelo Simplificado de Estimativas de Esforço**

$$EP = QPF * IP$$

Onde:

*EP = Esforço (horas)*

*QPF = Tamanho (PF)*

*IP = Índice de Produtividade (HH/PF)*

Contudo, é necessário observar que nem todas as organizações controlam ou sequer desejam compor e utilizar estimativas de esforço, situação evidente nos contratos firmados entre as organizações públicas brasileiras e as fábricas de software que lhe prestam serviços, onde são geradas estimativas de custos de forma direta, através do valor unitário do ponto de função - obtidas pelo produto da taxa horária e produtividade em horas para cada unidade de ponto de função (SISP, 2015). Esta produtividade, também chamada “Índice de Produtividade” (ou simplesmente, IP) depende de diversas características técnicas e ambientais para ser aferida, tais como:

- Plataforma tecnológica;
- Complexidade do domínio organizacional;
- Aspectos de segurança, desempenho e usabilidade;

Classificação da Informação: **Uso Irrestrito**

- Tamanho do projeto;
- Tipo de manutenção executada;
- Desenvolvimento para o reuso;
- Processo e artefatos gerados, entre outros.

Além de estimativas de esforço, dependendo das características da operação, pode ser necessário avaliar o tempo de desenvolvimento de um produto ou pedaço de software, variável esta denominada “duração” ou “prazo”, e que pode ser obtida através de uma estimativa de prazo. Segundo JONES (2007), as estimativas de prazo não são lineares com o tamanho do projeto, sendo que o melhor tempo de desenvolvimento, no qual há uma melhor relação de custo/ benefício de alocação de recursos e o menor prazo de desenvolvimento, pode ser estimado a partir da **Fórmula 2** (que utiliza o tamanho funcional do projeto como uma entrada), com base no **Quadro 7**.

### Fórmula 2 – Estimativas de Prazo

$$Td = V t$$

Onde:

**Td:** prazo de desenvolvimento

**V:** tamanho do projeto em pontos de função

**t:** o expoente *t* é definido de acordo com a Tabela 8

### Quadro 7 – Expoente t por tipo de Projeto

Tipo de Sistema	Expoente t
Sistema Comum – Mainframe (desenvolvimento de sistema com alto grau de reuso ou manutenção evolutiva).	0,32 a 0,33
Sistema Comum – WEB ou Cliente Servidor.	0,34 a 0,35
Sistema OO (se o projeto OO não for novidade para equipe, não tiver o desenvolvimento de componentes reusáveis considerar sistema comum)	0,36
Sistema Cliente/Servidor (com alta complexidade arquitetural e integração com outros sistemas).	0,37
Sistemas Gerenciais complexos com muitas integrações, Datawarehousing, Geoprocessamento, Workflow.	0,39
Software Básico, Frameworks, Sistemas Comerciais.	0,40

Fonte: SISP (2010).

Segundo o Roteiro de Métricas do SISP 2.0, a fórmula supracitada só funciona para projetos maiores do que 100 pontos de função, e caso seja menor,

deve ser definido o prazo máximo por faixa de tamanho funcional, com base em dados históricos da organização, como no modelo apresentado no **Quadro 8**. No entanto, os valores relacionados na publicação citada podem ser utilizados como referência na falta de dados históricos.

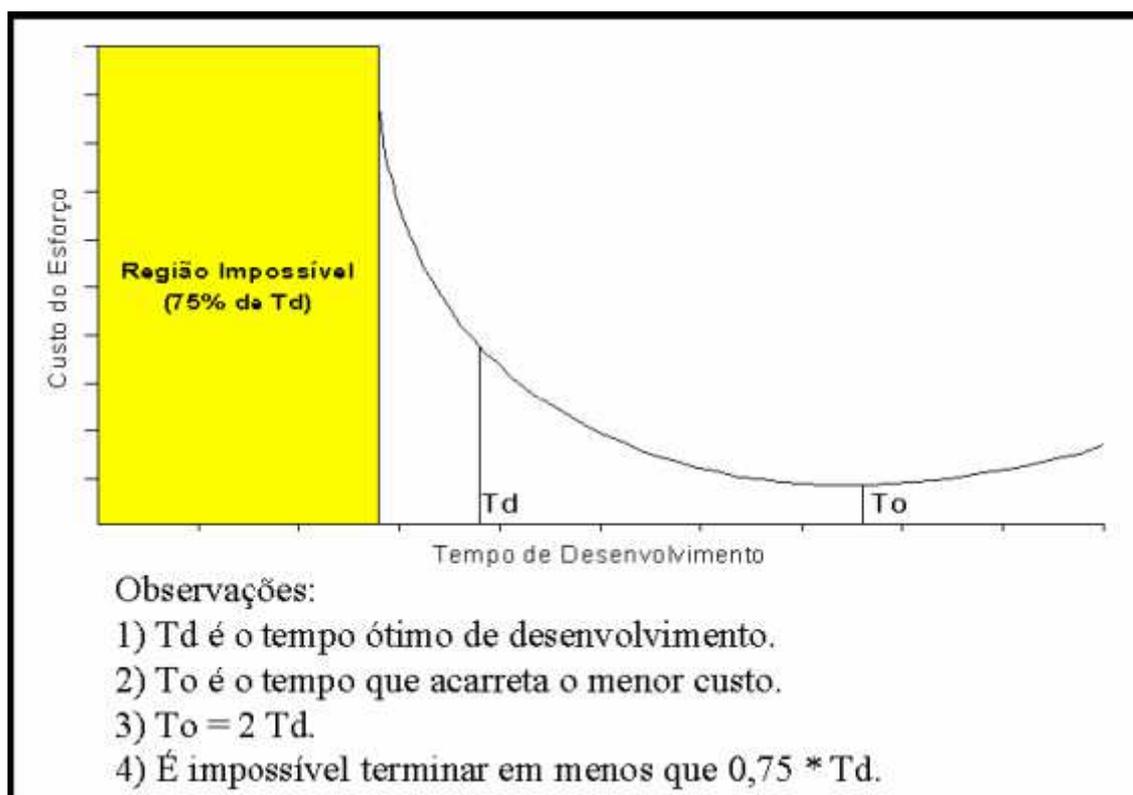
**Quadro 8 – Estimativas de Prazo de Projetos menores que 100 PF**

<b>Tamanho do Projeto</b>	<b>Prazo máximo (em dias úteis)</b>
Até 10 PF	10 dias
De 11 PF a 20 PF	20 dias
De 21 PF a 30 PF	30 dias
De 31 PF a 40 PF	40 dias
De 41PF a 50 PF	50 dias
De 51 PF a 60 PF	60 dias
De 61 PF a 70 PF	70 dias
De 71 PF a 85 PF	88 dias
De 86 PF a 99 PF	104 dias

*Fonte: SISP (2010)*

O Roteiro de Métricas do SISP (2010) reforça ainda, que o prazo calculado pelo método de estimativas considera todo o ciclo de vida do projeto, desde a fase de requisitos até a implantação, além da relação entre esforço e custos, com base na “região impossível” (**Figura 41**). É um modelo orientado ao método cascata (*waterfall*), mas que é utilizado no roteiro de forma mais ampla.

**Figura 41 – Região Impossível e a Relação de Custo versus Esforço**



Fonte: SISP (2010)

A estimativa de custos do projeto, citada anteriormente, considera então, o custo unitário de um ponto de função, que deve abranger uma ponderação do custo/hora dos profissionais envolvidos no processo de ciclo de desenvolvimento da solução de software. Ela é realizada através do cálculo que consta na **Fórmula 2**.

### Fórmula 2 – Estimativas de Custo com APF

$$CP = QPF \times CPF$$

Onde:

**CP** = Custo do projeto

**QPF** = Tamanho do projeto em PF

**CPF** = Custo para implementar um ponto de função

Nos contratos onde se utiliza o Índice de Produtividade (IP) para o cálculo de esforço, as estimativas de custo de um projeto podem também ser obtidas a partir da estimativa de esforço (**Fórmula 3**).

### Fórmula 3 – Estimativas de Custo com Base no Esforço

$$CP = QPF * EP * CTH$$

Onde:

**CP** = Custo do projeto

*QPF = Tamanho do projeto em pontos de função*

*EP = Esforço estimado do projeto*

*CTH = Custo da taxa hora de fábrica de software*

Quando for utilizado também o método SNAP para análise dos requisitos não funcionais, a apuração dos custos dependerá também da base utilizada para derivação (**Fórmula 2** ou **Fórmula 4**).

#### **Fórmula 4 – Estimativas de Custo com APF/SNAP**

$$\mathbf{CP = QPF * CPF + QPS * CPS}$$

*Onde:*

*CP = Custo do projeto*

*QPF = Tamanho do projeto em pontos de função*

*CPF = Custo unitário do Ponto de Função*

*QPS = Tamanho do projeto em pontos SNAP*

*CPS = Custo unitário do Ponto SNAP*

##### *1.5.1.5 Aplicabilidade no Contexto Ágil*

O Guia de Projetos Ágeis, mantido pelo SISP (2015b), refere-se a uma atividade específica para medição de software. Nessa atividade deverão ser realizados os serviços de medição de tamanho ou esforço do software para o escopo de cada Ordem de Serviço (OS), sendo que o seu faturamento deve ser realizado com base nas medições realizadas.

Conforme o SISP (2015b), a remuneração da contratada de desenvolvimento de software depende da adoção de uma métrica de faturamento que seja objetiva e justa, dentro da legalidade e que não onere contratante nem contratada. Contudo, para projetos com métodos ágeis, existem alguns desafios, como por exemplo, a remuneração de refinamento de requisitos entre iterações.

Conforme DEKKERS (2015), que também decorre sobre este tema, ao aplicar pontos de função em um contexto de utilização da metodologia ágil devem ser avaliadas algumas questões, tal como se o propósito é obter uma medição para comparar a produtividade de desenvolvimento ou contratar um fornecedor em um modelo de *outsourcing*. DEKKERS (2015) identifica um dilema referente à

consistência de aplicação da técnica entre os modelos tradicional e ágil, visto que dependendo da abordagem utilizada, o tamanho funcional pode ser distorcido, e, portanto diferente ao compará-los, o que não é adequado em relação ao que prega o IFPUG (2010), uma vez que a APF é independente da forma de implementação, o que inclui a estratégia e metodologia de projeto, e não apenas a tecnologia de construção.

Em sua proposta, DEKKERS (2015), ressalta o cuidado necessário ao considerar a definição do que seria uma funcionalidade completa e defende que as funcionalidades já tratadas devem ser consideradas alteradas apenas quando existirem releases diferentes do software sendo entregues, ou seja, num cenário onde a mudança ocorre sobre algo já desenvolvido. Ela ressalta ainda, que a funcionalidade deve, ou ao menos deveria ser sempre medida de forma independente de como foi desenvolvida.

#### 1.5.1.6 *Benchmarking e o IFPUG*

Como já citado anteriormente, dentre os propósitos de uma contagem de pontos função, estão: torná-lo mensurável e comparável em relação a outros projetos da organização e até mesmo entre organizações ou contextos. Isso pode ser feito não somente através de ordens de grandeza como um padrão aplicável às estimativas de tamanho, esforço, prazo e custo dos projetos, mas também através dos aspectos que tangem à qualidade apurada (ISBSG, 2013).

Na década de 1990, foi fundada a organização ISBSG (*International Software Benchmarking Standards Group*), uma empresa sem fins lucrativos e que hoje consolida informações de cerca de 6.000 projetos de software oriundos de diferentes países e organizações; e considerando tanto projetos desenvolvidos através de métodos tradicionais de desenvolvimento de software, como àqueles concebidos através de práticas ágeis. Atualmente, o seu principal produto é o relatório de *benchmarking* para produtividade de software, chamado “*Reifer Consultants LLC and ISBSG Joint Report: Software Productivity Benchmarks*” (ISBSG, 2013),

composto por análises estatísticas vinculadas a cerca de 6.000 projetos de software executados por diversas organizações ao redor do mundo. Como todos estes projetos foram medidos utilizando Pontos de Função do IFPUG, os seus dados são muito úteis como fonte de *benchmarking* às empresas que querem comparar seus principais indicadores operacionais (tais como, os de produtividade e qualidade).

No relatório citado, existem projetos executados tanto na metodologia tradicional como na metodologia ágil, assim como existem informações direcionadas para setores específicos da indústria ou famílias de tecnologias (agrupadas por geração de linguagem ou plataformas tecnológicas). Os setores da indústria analisados no relatório são:

- Finanças;
- Bancos;
- Manufatura;
- Eletrônicos/Computação;
- Governo;
- Seguros;
- Serviços.

Em relação às tecnologias, segundo o ISBSG (2013, pág. 51 e 57) há diferenças de produtividade entre plataformas e também entre gerações de linguagem analisadas (exemplo: terceira, quarta e quinta gerações – principalmente esta última em relação às demais).

### 1.5.2 Abordagem do COSMIC

A “abordagem do COSMIC”, no contexto deste trabalho, remete a utilização do método de medição funcional mantido pela organização COSMIC (2015) e contempla também estimativas derivadas.

### 1.5.2.1 Medições de Tamanho Funcional com COSMIC

O método COSMIC (2015) de medição de tamanho funcional é mantido pela organização COSMIC (*Common Software Measurement International Consortium*). Ele é um padrão internacional de domínio público e foi projetado para atender tanto as necessidades de aferição de aplicações de negócio, quanto de software *real-time* e híbrido.

A aplicação do método pode ocorrer através de uma medição ou aproximação de tamanho funcional, à luz dos requisitos funcionais do usuário, sendo o resultado da avaliação é expresso em unidades de Pontos de Função COSMIC (PFC).

Basicamente, o processo de medição é executado em três fases:

- Fase de Definição da Estratégia de Medição: consiste em avaliar os objetivos da medição em termos do modelo de contexto de software, a partir da definição do propósito da medição de cada pedaço do escopo a ser medido, incluindo as fronteiras impactadas;
- Fase de Mapeamento: permite avaliar os requisitos funcionais do usuário do software em questão conforme o modelo geral de software. Nesta fase são organizados os requisitos para que na fase de medição, eles sejam convertidos no tamanho funcional, em unidades de pontos de função;
- Fase de Medição: possibilita associar movimentos de dados entre as fronteiras definidas nas fases anteriores, compondo uma valoração em pontos de função COSMIC.

A seguir será apresentado o detalhamento para cada fase citada, de modo que os principais conceitos (os mais usuais) sejam apresentados, proporcionando um melhor entendimento pelos leitores.

#### **Fase de Estratégia**

Nesta fase, avalia-se primeiramente o propósito da medição, sendo alguns dos possíveis propósitos, os citados abaixo:

- Prover uma estimativa de tamanho do projeto de software;

- Realizar análise de produtividade de projetos de software.

Em uma medição COSMIC (2015), o propósito da contagem determina um escopo geral a ser analisado, com foco em um projeto ou sistema específico. Este escopo geral pode ser dividido em várias partes, tais como pedaços de software ou componentes da arquitetura de software. Partes as quais, por sua vez, interagem com os usuários funcionais da(s) aplicação(ões) envolvida(s).

Nesta fase, o Estimador COSMIC deverá listar todos os tipos de usuários identificados para cada parte do escopo em análise.

### **Fase de Mapeamento**

Usuários funcionais sensibilizam o acionamento de eventos, que iniciam seus processos funcionais. E os processos funcionais, em sua lógica de processamento, utilizam grupos de dados com o objetivo de consultá-los (leitura) ou mantê-los (escrita), o que ocorre através de movimentos de entrada e saída de informações do sistema, os quais constituem objetos de interesse do usuário.

O Estimador COSMIC terá, portanto, ao final da Fase de Mapeamento, uma lista de processos funcionais e seus respectivos objetos de interesse e grupos de dados.

### **Fase de Medição**

Após a Fase de Mapeamento, o Estimador COSMIC deve avaliar, para cada processo funcional identificado, e com base no propósito, escopo, fronteiras e usuários previamente definidos, os quatro tipos de movimentos existentes, cujas regras encontram-se no manual do método: Entradas (“E”), Saídas (“X”), Leituras (“R”) e Escritas (“W”).

Um processo funcional COSMIC contará com no mínimo duas unidades de PFC, sendo uma para cada tipo de movimento identificado naquele processo funcional específico sendo medido. Contudo, não há um limite superior de tamanho definido pelo método, característica que o distingue da maioria dos demais padrões de medição funcional e o torna mais versátil.

É importante destacar, contudo, que ao realizar uma medição COSMIC, deve-se sempre que possível, identificar os possíveis defeitos visualizados na análise e premissas para tamanhos que por ventura tenham sido assumidos, tais como aqueles associados às estimativas antecipadas de tamanho e que impactam no planejamento de projetos de software.

### *1.5.2.2 Estimativas de Tamanho Funcional com COSMIC*

O método COSMIC (2015) é comumente utilizado para medição do tamanho funcional, mas por vezes é comum que o Estimador COSMIC precise realizar apenas uma aproximação, como por exemplo, nas situações citadas abaixo:

- Durante as fases iniciais do ciclo de vida de um projeto, onde os requisitos funcionais do usuário ainda não possuem o nível de detalhe suficiente para a realização de uma contagem;
- Quando é necessário obter uma ordem de grandeza de software, porém o tempo disponível para a sua avaliação é reduzido;
- Quando a qualidade dos insumos disponíveis para a contagem não é suficiente para garantir o nível de precisão esperado.

Diferentemente da APF (IFPUG, 2010), onde um processo funcional pode ter no máximo três níveis de contribuição, atribuídos através de uma aferição de complexidade, no COSMIC o tamanho funcional de cada processo não possui um limite superior de pontos de função possíveis. Logo, abordagens como a da NESMA (2015), onde se elege uma contribuição generalista (através da atribuição da complexidade “baixa” para as funções de dados; e “média” para as funções de transação), não são obtidas de forma tão simples. Como resultado, há para o COSMIC, um conjunto de diferentes abordagens de estimativa antecipada, que possibilitam calcular ou arbitrar sob um quantitativo de Pontos de Função COSMIC (PFC) e que devem ser analisadas conforme o cenário e necessidade identificada pelo Estimador COSMIC.

Veremos a seguir, as principais abordagens de estimativa antecipada, as quais utilizam basicamente métodos de escala e/ou por classificação, ressaltando seus prós e contras, assim como considerações para sua aplicabilidade. Em tempo: fatores de escala consistem na atribuição de um valor a uma escala, normalmente com base em dados calibrados, e classificações são categorias para as quais são atribuídas fatores de escala.

É importante destacar, que a definição de fatores de escala realmente deve ser calibrada com base em dados locais, sendo altamente recomendável criar critérios de classificação objetivos ou exemplos contundentes para isto, visto que estas ações permitem que o método COSMIC seja aplicado consistentemente nas diferentes contagens realizadas.

Antes de selecionar uma abordagem, é necessário, portanto:

- Entender o nível de granularidade, a completude e a qualidade dos requisitos;
- Obter o maior nível de detalhe possível;
- Atentar para os aspectos locais.

Todavia, ressalta-se que até o momento, não há uma proposta padrão do COSMIC para tratar de forma diferente as funções incluídas, alteradas e excluídas, assim como aquelas apenas testadas, mas é sugerido ter médias diferentes para a calibração de cada cenário (explicitamente o COSMIC trata cenários de inclusão e alteração com esta orientação, mas pode ser desejável também criar uma base histórica para os cenários de exclusão e testes).

### *1.5.2.3 Estimativas Derivadas*

Embora não tenham sido identificadas referências próprias para o cálculo de estimativas de prazo com base em um tamanho COSMIC, existem modelos que utilizam o tamanho funcional como referência e talvez possam ser utilizados com este intuito (a exemplo: o COCOMO II). Ainda assim, algumas organizações

preferem avaliar o comportamento histórico e desempenho do fornecedor em termos de velocidade, ou seja, o quanto ele consegue entregar em pontos de função por mês, o que parece razoável para alguns cenários organizacionais.

O custo de um PFC pode ser cotado diretamente com os fornecedores para cada projeto ou para ser utilizado numa operação (COSMIC, 2015), assim como um PF do IFPUG o é em diversas organizações públicas brasileiras (SISP, 2015). Desta forma, com base no valor e no tamanho estimado/medido para o software sendo contratado, o valor total da contratação do desenvolvimento do pedaço de software em questão pode ser calculado.

Por configurar uma unidade similar, outra forma de contratação que pode ser utilizada em projetos medidos com COSMIC, também no mesmo modelo construído inicialmente sob a perspectiva de PF do IFPUG, é o estabelecimento de uma produtividade média e uma taxa horária para o serviço, sendo que o produto destes resulta no preço do ponto de função, levando ao mesmo cálculo final demonstrado na primeira abordagem.

#### *1.5.2.4 Aplicabilidade no contexto ágil*

Segundo TRUDEL e BUGLIONE (2010), em um mundo ideal, os times ágeis possam estimar e se comprometer com as suas próprias estimativas durante uma iteração, mas na prática, as organizações precisam estimar o projeto de forma mais ampla (desde o seu início) e consistente para que possam realizar suas projeções de budget. Os autores sustentam ainda, que as estimativas ágeis não suportam adequadamente e a necessidade de comparações e avaliações de desempenho interno e externo dos projetos (benchmarks). Informações estas, que são importantes principalmente para os gestores, mas também aos próprios membros do time, que podem querer saber o desempenho da metodologia ágil em relação aos processos tradicionais da organização.

Ainda conforme TRUDEL e BUGLIONE (2010), uma comparação só passa a ser possível utilizando-se de uma mesma unidade base, que deve ser oriunda de um

padrão de medição para que o processo de estimativa seja consistente, como aquele fornecido pelo Functional Size Measurement (FSM) e que é mantido e publicado pela ISO. O método COSMIC constitui uma medida padronizada e aderente ao FMS e pode ser utilizado com este fim, possibilitando comparar a taxa de entrega, produtividade e custo unitário resultantes de atividades de projetos ágeis ou não.

Ressalta-se, que uma comparação pode ser realizada entre projetos de uma mesma organização ou entre projetos ou operações de diferentes organizações, idealmente considerando um domínio adequado. Em complemento, o custo unitário de desenvolvimento de software poderia ser definido por projeto, por tecnologia ou por domínio de negócio, por exemplo.

Em síntese, as organizações utilizam parâmetros de custo, esforço, taxa de entrega, qualidade e escopo, não apenas para comparar seus dados atuais contra o orçamento ou valores planejados para um projeto, mas para criar e acompanhar novos indicadores que podem monitorar a eficiência do processo de desenvolvimento.

De acordo com TRUDEL e BUGLIONE (2010, pág. 9), as organizações que enviam ou consultam dados para benchmarking têm interesse em comparar o custo unitário de projetos similares na indústria, independente da metodologia utilizada. Sendo assim, os autores entendem que poderia ser adequado considerar uma iteração ou *release* como um projeto (desde que o incremento efetivamente implantado em produção), o que é justificado como uma analogia aos métodos tradicionais de desenvolvimento de software. Nesta proposta, portanto, podem ser vislumbrados os seguintes pontos de estimativa/medição:

- Estimativa Inicial do Projeto – realizada no início do projeto (por aproximação), com base em uma lista de US ou requisitos em outro formato, desde que possuam um nível de detalhe suficiente para a identificação das funcionalidades requeridas;
- Estimativa Inicial da Iteração – opcionalmente, no início ou durante de cada iteração, pode ser realizada uma estimativa, para fins de planejamento da iteração;

- Medição Final da Iteração – ao final de cada iteração e considerando o total de movimentos de dados incluídos, alterados e excluídos durante a mesma realiza-se a medição do software entregue ao usuário, considerando, portanto, a evolução e o retrabalho associado a cada funcionalidade pronta em uma iteração anterior e novamente envolvida no processo de construção;
- Medição Final do Projeto – considera o tamanho total do software efetivamente entregue ao usuário ao final do projeto, eliminando o retrabalho associado.

#### 1.5.2.5 Benchmarking e o COSMIC

Conforme publicado no *website* da organização COSMIC (2016), existe uma iniciativa em andamento junto ao ISBSG para realização de uma análise conjunta sobre projetos de software da base do ISBSG para *benchmarking*. Há resultados parciais já publicados e disponibilizados na forma de um relatório técnico pelo próprio ISBSG.

O referido relatório, cuja leitura é sugerida para membros de organizações que estão iniciando com o método COSMIC, contempla uma análise detalhada sobre como a produtividade (tamanho/esforço) e a velocidade (tamanho/duração) dos projetos software variam, dependendo do domínio, tipo de projeto e linguagem de programação. Este relatório também contempla:

- Uma análise de como o esforço é distribuído sobre as atividades de projeto mais significativas, como um percentual;
- Uma análise das características das medições de tamanho funcional como COSMIC, como por exemplo, a proporção de cada tipo de movimentação e como ela varia de acordo com o domínio do software;
- Exemplos de como dados de benchmarking existentes de projetos medidos com a abordagem do IFPUG podem ser comparados com os resultados obtidos com dados de projetos medidos com o COSMIC.

## 1.6 Escalando o Desempenho Organizacional

Neste tópico serão apresentados conceitos que permitem escalar a utilização de métodos ágeis no contexto enxuto, viabilizando sua aplicação integral à necessidade de software identificada. Primeiramente, será abordada a definição de camadas de ritmo, que provê uma classificação para software conforme o seu nível de inovação, e em seguida o processo de Portfólio.

Ao final, serão apresentados meios de medir o desempenho de projetos, contratos e operações com foco no desenvolvimento de software.

### 1.6.1 Inovação em Camadas de Ritmo

AMARAL (2011, págs. 6 e 8) identifica que os diferentes tipos de projetos que envolvem aspectos de inovação, seja de produto, processo, modelo de negócio ou serviço, possuem em comum como principais características: a incerteza e a complexidade; além de necessidades específicas, tais como: o envolvimento de clientes e usuários no desenvolvimento; e a composição de equipes multidisciplinares. Ele reconhece que estes desafios impactam diretamente sobre o gerenciamento dos projetos de natureza inovadora, tanto em termos dos recursos necessários e seus prazos e riscos, como em todas as suas demais dimensões (tal como o esforço e custos relacionados).

Em relação aos níveis de inovação para produtos de software e sua classificação estratégica nas organizações, o GARTNER (2012) definiu uma proposta de segregação de sistemas em camadas de ritmo, com o propósito de fornecer uma resposta mais rápida e um melhor ROI às aplicações sem sacrificar a integração, integridade e / ou governança: o *Pace Layered*.

Segundo o GARTNER (2012), no passado, muitas empresas utilizavam uma estratégia única para seleção, implementação e gerenciamento de suas aplicações; e mesmo que algumas delas tenham utilizado métodos para classificar os aplicativos

por valor ou viabilidade tecnológica, não avaliavam como eles eram utilizados na organização. Contudo, a abordagem do GARTNER (2012) propõe justamente isso ao considerar camadas de ritmo que reconhecem processos e requisitos de integridade de dados que podem ser diferentes dentro de cada uma das camadas, além de definir um conjunto de padrões arquitetônicos, também em cada nível, para acelerar a capacidade de uma organização em se adaptar.

O método do *Pace-Layered* distingue três categorias de aplicativos, ou “camadas”:

- Sistemas de Inovação – categoria (ou segmento) que engloba as aplicações cuja incerteza do direcionamento para a construção ou evolução de funcionalidades é muito grande, sendo assim, normalmente não se sabe exatamente o que tem de ser feito para atender os requisitos de negócio;
- Sistemas de Diferenciação – são aplicações que possuem funcionalidades cujas características são conhecidas com base em produtos similares da concorrência, porém que precisam ser diferentes para manter o negócio competitivo;
- Sistemas de Registro – são aplicações que apresentam recursos conhecidos e normalmente sua manutenção e desenvolvimento é executada com maior facilidade e previsibilidade.

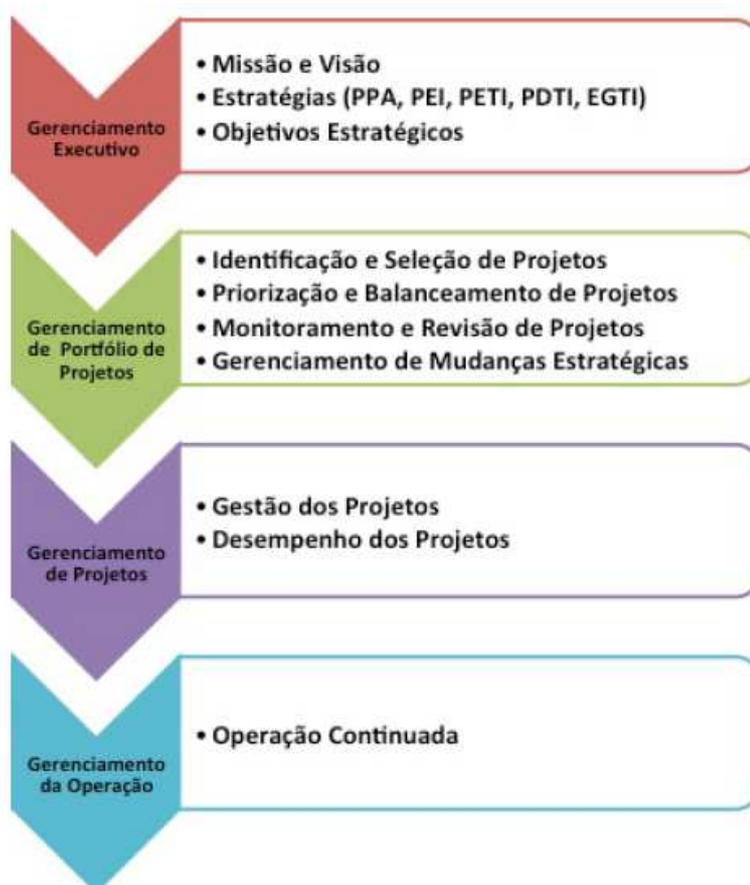
Considerando as categorias elencadas, o GARTNER (2012) defende que um mesmo aplicativo possa ser classificado de formas distintas em organização diferentes (devido a sua visão sobre os aspectos do produto) e que este produto possa ser reclassificado conforme a evolução do nível de familiaridade da organização com as suas características. Propõe ainda, que as organizações estabeleçam uma nova estratégia para aplicativos de negócios e que estas respondam ao desejo de utilizar a tecnologia para estabelecer diferenciação sustentável e conduzir novos processos inovadores, o que fomenta um ambiente seguro e sustentável para apoiar os seus principais processos de negócio.

### 1.6.2 Portfólios de Projetos (para o SISP)

Segundo o SISP (2013), um portfólio é uma coleção de projetos, programas e outros trabalhos, em andamento ou planejados (estando eles relacionados de alguma forma ou não), que estão agrupados com o propósito de facilitar o gerenciamento efetivo das ações para atender aos objetivos estratégicos organizacionais (**Figura 42**).

Deve-se destacar, que enquanto projetos e programas são temporários, os portfólios são contínuos. Sendo assim, uma organização pode possuir mais de um portfólio, cada um tratando de áreas (departamentos) ou objetivos específicos (em última instância, deve haver um portfólio abrangente para toda a organização).

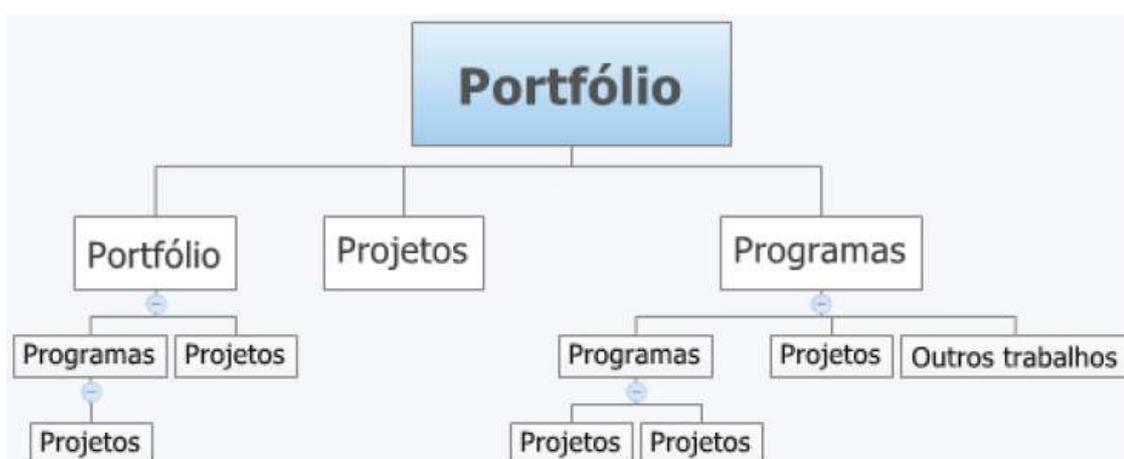
**Figura 42 – Portfólio de Projetos e os Demais Processos Organizacionais**



Fonte: SISP (2011).

O gerenciamento de portfólios de projetos refere-se ainda, à gestão centralizada de um ou mais portfólios (**Figura 43**), o que inclui a identificação, priorização, autorização, monitoramento e controle de projetos, programas e outros trabalhos relacionados. O gerenciamento de portfólios se concentra em garantir que os projetos e programas sejam analisados a fim de priorizar a alocação de recursos e que sejam consistentes e integrados às estratégias organizacionais (SISP, 2011, pág. 20).

**Figura 43 – Relação entre Projetos, Programa e Portfólio**



*Fonte: Metodologia de Gerenciamento de Portfólio de Projetos do SISP (SISP, 2011).*

Em um portfólio (SISP, 2013), todos os componentes exibem certas características comuns:

- Representam investimentos feitos, em curso ou planejados;
- Estão alinhados com as metas e objetivos estratégicos;
- Têm algumas características que, tipicamente, os distinguem, o que permite agrupá-los para o gerenciamento efetivo;
- São quantificáveis e, portanto, podem ser medidos, classificados e priorizados.

O SISP (2013) define dois tipos de portfólio, sob duas perspectivas diferentes, conforme abaixo:

- Portfólio de Projetos Departamental - inclui projetos ou programas de um determinado departamento ou unidade organizacional, em uma visão de projetos limitada ou restrita;
- Portfólio de Projetos Corporativo - são projetos ou programas que englobam toda a organização, sendo que a alta administração é patrocinadora desses projetos e possui a visão do todo (secretarias, departamentos, coordenações).

É importante ressaltar que o gerenciamento de portfólios de projetos provê um gerenciamento coordenado dos componentes do portfólio, o qual tende a auxiliar no alcance de objetivos maiores na instituição. Dessa forma, oferece benefícios para a tomada de decisão baseada em prioridades e informações estratégicas. Existem ainda, inúmeras vantagens na adoção dessas práticas, como por exemplo:

- Alinhar os projetos ou programas com as estratégias organizacionais;
- Prover a gestão das prioridades, mantendo o foco em projetos de maior valor para a organização;
- Permitir a alocação eficiente de recursos e investimentos;
- Possibilitar a redução do número de projetos redundantes ou sem alinhamento estratégico;
- Permitir uma visão clara das interdependências entre os projetos;
- Favorecer o envolvimento da alta administração com a gestão dos projetos.

Ressalta-se também, que o gerenciamento de portfólio tende a assegurar que a organização está desempenhando um trabalho alinhado às necessidades, evitando os problemas abaixo listados:

- Quantidade excessiva de projetos desenvolvidos simultaneamente;
- Recursos preciosos utilizados em projetos de baixa prioridade;
- Projetos ou programas sem relação com os planos estratégicos;
- Seleção de projetos que trazem pouco valor à organização ou com grandes riscos e poucos benefícios.

Em geral, um projeto nasce na área de negócio e passa pela gestão estratégica e de portfólio de projetos, onde sua viabilidade é avaliada. Posteriormente são realizados os critérios de priorização e balanceamento, permitindo que projetos de maior valor para a organização sejam aprovados. Depois de aprovado, o projeto entra na etapa de execução, ou seja, inicia-se o seu desenvolvimento (SISP, 2013). A seguir, são elencadas as principais atividades inerentes ao processo de Gestão de Portfólio:

- Identificar Projetos – identificar projetos de TI em andamento e os novos projetos a serem executados, conforme o planejamento estratégico vigente;
- Selecionar Projetos – selecionar e categorizar os projetos e programas que comporão o portfólio, de acordo com suas características e relevância para a organização;
- Priorizar Projetos – priorizar os projetos de acordo com os critérios pré-estabelecidos (vide exemplo na **Figura 44**), de forma a possibilitar distinguir àqueles com maior potencial de entrega de valor à organização;
- Balancear Portfólio – definir quais projetos serão executados (emitindo-se o Termo de Abertura do Projeto) e quais aguardarão na fila para execução (exemplo na **Figura 45**);
- Monitorar Portfólio – monitorar e controlar as atividades de modo a assegurar a realização do gerenciamento de portfólio de projetos de TI de forma alinhada aos objetivos estratégicos da organização;
- Gerir Portfólio – realizar a gestão do portfólio de projetos, afim de garantir que os recursos estão sendo investidos conforme a sua importância estratégica para a organização;
- Gerenciar Mudanças e Estratégias – acompanhar a evolução das estratégias e acompanhar mudanças que causam impacto no portfólio de projetos.

#### **Figura 44 – Critérios de Priorização de Portfólio (exemplo)**

Número	Critério	Avaliação	Valor	Projeto
1	Importância estratégica	Alta	2000	
		Média	1000	
		Baixa	500	
2	Abrangência dos Resultados do Projeto	Traz mudança para muitas áreas	1000	
		Traz mudança para poucas áreas	250	
		Traz mudanças para uma área somente	125	
3	Urgência do Projeto	Alta	1000	
		Média	500	
		Baixa	250	
4	Tempo Estimado do Projeto	Curto	500	
		Médio	250	
		Longo	125	
5	Conhecimento da Equipe	Profundo	500	
		Razoável	250	
		Pouco	125	
6	Fator Político	Alto	1000	
		Médio	500	
		Baixo	250	
7	Tempo Aguardando a Execução	Menos de 6 Meses	125	
		De 6 meses a 1 Ano e 6 meses	250	
		Mais de 1 Ano e 6 meses	500	
			<b>TOTAL</b>	

Fonte: Metodologia de Gerenciamento de Portfólio de Projetos do SISP (SISP, 2011).

**Figura 45 – Exemplo de Planilha de Acompanhamento de Portfólio**

ID	Nome do Projeto	Área de Negócio	Patrocinador	Solicitante	Categoria	Riscos	Prioridade	Balancamento	Status do Projeto	Status Gráfico	Observação
1									Em execução		
2									Adiado/Suspensão		
3									Em execução		
4									Priorizado		
5											
6											
7											
8											
9											
10											

Fonte: Metodologia de Gerenciamento de Portfólio de Projetos do SISP (SISP, 2011).

São resultados esperados de um Gerenciamento de Portfólio de Projetos, conforme especificação do Nível F do modelo de qualidade MPS-BR (SOFTEX, 2013):

- As oportunidades de negócio, as necessidades e os investimentos são identificados, qualificados, priorizados e selecionados em relação aos objetivos estratégicos da organização por meio de critérios objetivos;
- Os recursos e orçamentos para cada projeto são identificados e alocados;

- A responsabilidade e autoridade pelo gerenciamento dos projetos são estabelecidas;
- O portfólio é monitorado em relação aos critérios que foram utilizados para a priorização;
- Ações para corrigir desvios no portfólio e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;
- Os conflitos sobre recursos entre projetos são tratados e resolvidos, de acordo com os critérios utilizados para a priorização;
- Projetos que atendem aos acordos e requisitos que levaram à sua aprovação são mantidos, e os que não atendem são redirecionados ou cancelados;
- A situação do portfólio de projetos é comunicada para as partes interessadas, com periodicidade definida ou quando o portfólio for alterado.

#### 1.6.2.1 Programas e Portfólios Ágeis e Escaláveis

Segundo GOMES (2013) quando um time ágil não é suficiente ou há muitos produtos ou ainda produtos grandes demais para serem suportados por uma única equipe, podem ser constituídos vários times com atuação paralela em um mesmo ciclo de *releases*, cenário trabalhado normalmente com uma data e qualidade fixos e um escopo variável. Este nível de desenvolvimento é chamado nível de Programa, sendo comum a definição de um *Product Manager* ou uma comunidade prática de *Product Owners* para coordenar, alinhar e lidar com requisitos de alto nível, organizados em um *Program Backlog*.

Quando existem muitos programas, trabalha-se com portfólios, que tratam os requisitos de nível mais alto, chamados “épicos”, a granularidade em que são tratados os investimentos da organização, que são realizados através dos programas e projetos (GOMES, 2013).

### 1.6.3 Avaliação de Desempenho

Segundo PRESSMAN (2006), um indicador de desempenho é formado por uma métrica ou um conjunto de métricas, possibilitando obter uma visão sistemática do que se está medindo.

No desenvolvimento de software ágil, conforme GOMES (2013, pág. 103), as referidas métricas ajudam o time a compreender onde ele está e aonde ele quer chegar, sendo pertinentes desde que sejam explícitas e visíveis aos seus integrantes.

Especificamente sobre os indicadores de projetos e práticas ágeis, COHN (2005) sugere que as métricas devam ter uma série de critérios, tais como: seguir tendências e não números, reforçar os princípios ágeis, ser de fácil coleta, fornecer *feedback*, etc.

Tanto COHN (2005) como GOMES (2013) recomendam o uso de dois tipos de indicadores, conforme abaixo:

- Indicadores indutores:
- Indicadores de resultado:

GOMES (2013, 107), no entanto, sugere medir times e não indivíduos, e defende a ótica de que se as pessoas acreditam que serão afetadas pelos resultados e métricas sobre seu trabalhos individuais, elas rapidamente encontram formas de corromper tais métricas. O autor alerta ainda para o risco de ter prejudicada a colaboração entre os integrantes do time ao utilizar uma perspectiva individual, destacando que as metas coletivas tem o efeito contrário.

#### 1.6.3.1 Indicadores para Projetos

Conforme TERRIBILI FILHO (2010, pág. 26), indicadores de projetos são instrumentos de avaliação que permitem comprovar a progressão de uma ou mais

dimensões de um projeto diante de metas estabelecidas, empiricamente (com base na experiência e observação) e com objetividade. Desta forma, um indicador deve atender basicamente dois requisitos:

- Permitir comparações históricas, para avaliar variações ocorridas;
- Permitir estabelecer prognósticos (projeções).

TERRIBILI FILHO (2010, pág. 27), destaca também que existem quatro tipos de indicadores de projetos:

- De impacto – medem o objetivo geral do projeto com resultados no longo prazo, assim como a sua contribuição para a organização ou sociedade;
- De efetividade – medem os resultados dos objetivos em um determinado período de tempo, após a disponibilização dos resultados do projeto;
- De desempenho – evidenciam se os resultados planejados foram alcançados; e
- Operacionais – medidos durante o ciclo de vida de um projeto, como uma análise de tendência sobre suas atividades e recursos.

Os indicadores operacionais e de desempenho representam a realidade imediata do projeto, seja durante o seu desenvolvimento ou ao seu final. Desta forma, embora todos os tipos de indicadores sejam importantes, estes são considerados, respectivamente, a bússola para o gerenciamento e o instrumento efetivo de para o acompanhamento e tomada de decisões (TERRIBILI FILHO, 2010, pág. 28 e 29).

Os indicadores operacionais podem representar perspectivas de várias dimensões de um projeto, como: satisfação do patrocinador, custos, tempo, riscos. O conjunto desses indicadores é nomeado *Cockpit* e é utilizado como um norteador do projeto. Sua estrutura básica deve conter, no mínimo, os seguintes indicadores: custo, prazo, qualidade e satisfação do patrocinador (TERRIBILI FILHO, 2010, pág. 28 e 29).

### 1.6.3.2 Indicadores Gerais da Operação

Segundo o SISP (2013), é aconselhável definir indicadores para o acompanhamento da operação, mais especificamente do portfólio de projetos, visando medir a capacidade dos recursos empregados e o desempenho geral da operação de TI.

São exemplos de indicadores que podem constar em um Relatório de Desempenho de Portfólio:

- Orçamento gasto versus orçamento planejado;
- Percentual de conclusão do cronograma em relação às metas estabelecidas;
- Índice de satisfação do cliente;
- Índice de aderência à metodologia de gerenciamento de projetos;
- Quantidade de projetos em fila;
- Quantidade de projetos em andamento;
- Valor médio dos projetos no portfólio.

### 3 MÉTODOS E PROCEDIMENTOS

Conforme VERGARA (2010), um método é um caminho, uma forma e uma lógica de pensamento, podendo ser entendido como uma intervenção realizada pelo pesquisador. Já a pesquisa, é um processo sistêmico, que tem por finalidade prover respostas aos problemas apresentados através de métodos, sendo necessária quando não há informações suficientes para responder um problema em foco.

Desta forma, este capítulo apresenta os procedimentos e métodos utilizados durante a realização da pesquisa, tais como: unidade de pesquisa, técnicas de coleta de dados e limites do estudo.

#### 3.1 Delineamento da Pesquisa

Esta pesquisa foi realizada utilizando a técnica de estudo de caso qualitativo, onde o pesquisador teve a oportunidade de realizar observações e análises com vistas à obtenção de dados para posterior tratamento. Segundo YIN (2001, p. 21), este método permite investigar com a finalidade de preservar características holísticas e significativas de eventos da vida real.

Para GODOY (1995) a pesquisa qualitativa não tem por objetivo enumerar ou medir os eventos estudados, e tão pouco, procura utilizar-se de ferramentas estatísticas para análise dos dados estudados. Segundo o autor, na pesquisa qualitativa deve ser valorizado o contato direto e prolongado do pesquisador com o objeto de pesquisa, com foco no ambiente e situação em estudo. Além disso, ela tem por base uma corrente filosófica denominada “fenomenologia”, a qual indica que o pesquisador deve tirar suas conclusões diante das situações que as pessoas atribuem a certos fatos. (JONES *apud* ROESCH, 1999).

Segundo YIN (2005), o estudo de caso é um método de pesquisa que busca contribuir com o conhecimento sobre fatos organizacionais, sociais e políticos que ocorrem dentro da organização. Ele ressalta também, que o estudo de caso utiliza fontes de evidências, tais como documentos, registros em arquivos, entrevistas, observação direta, observação participante.

Para YIN (2005) existem três estratégias utilizadas para nortear as análises das evidências encontradas em um estudo de caso:

Classificação da Informação: **Uso Irrestrito**

- Proposições relativas ao projeto original, visto que esta é baseada nos objetivos da pesquisa;
- Definições e testes de explicações correntes, muito úteis ao realizar avaliações de um estudo de caso;
- Descrições de caso, que têm por objetivo apresentar uma estrutura descritiva para organizar o estudo de caso, especialmente utilizada em situações de dificuldades para utilizar proposições teóricas ou explicações concorrentes.

Diante do exposto, neste trabalho o autor utilizou o estudo de caso com a estratégia de pesquisa descritiva, possibilitando que a pesquisa fosse aprofundada na empresa FINA S/A.

A opção pelo método de pesquisas descritivas justifica-se, uma vez que o estudo tem como objetivo primordial: a descrição das características de determinada população ou fenômeno; ou, então, o estabelecimento de relações entre variáveis, conforme argumentação de GIL (2002, p.42). O autor afirma ainda, que pode haver situações em que as pesquisas descritivas podem ir além do objetivo de identificar a existência de relações entre as variáveis, com o propósito de definir qual é a natureza dessa relação.

### **3.2 Unidade-Caso e Sujeitos da Pesquisa**

A pesquisa em questão foi realizada na empresa FINA S/A, empresa nacional do ramo bancário/ financeiro, de grande porte, e com atuação destacada no âmbito de financiamentos e empréstimos a produtores rurais.

A organização ultimamente tem comercializado novos produtos e apresentado uma miscigenação do seu público alvo, principalmente pelo avanço nas relações com clientes de áreas urbanas, tendo sido diagnosticado que os seus sistemas de informação deveriam inicialmente equalizados com seus concorrentes do nicho comercial, o que por sua vez, impactou em mudanças significativas no modelo de trabalho para processo de desenvolvimento de software, e conseqüentemente no formato das práticas até então vigentes para gestão de projetos, além dos critérios e agilidade nas contratações relacionadas.

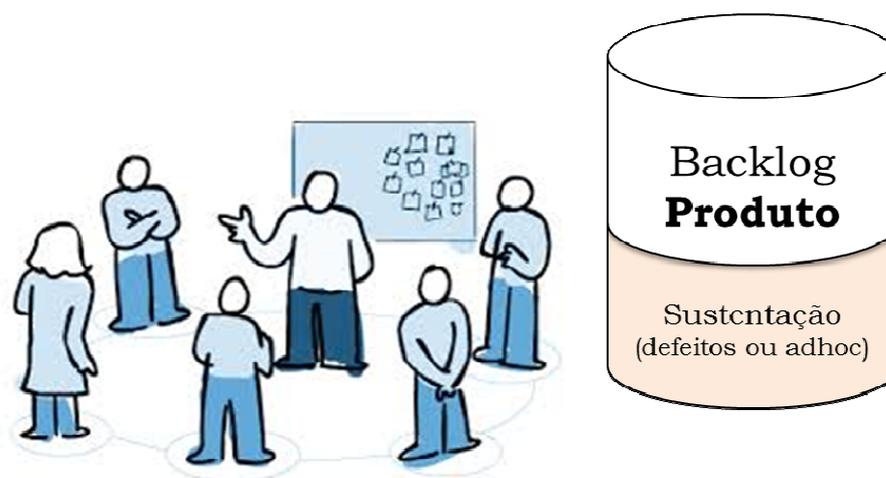
Com o propósito de oferecer a tecnologia da informação adequada e aderente ao negócio da empresa, proporcionando principalmente um avanço no tempo de mercado esperado para o lançamento de uma solução, e embasado em uma política de eliminação de desperdícios, o processo de desenvolvimento foi estruturado sob a perspectiva ágil/ enxuta, ou seja, utilizando-se dos preceitos oriundos das definições de métodos ágeis e da filosofia *Lean*, no entanto, optou-se, ao menos para o momento, até pelas características tecnológicas do seu parque de aplicações, que esta nova metodologia seria utilizada para um conjunto específico de produtos, que trabalham com um nível maior de incerteza, incorporando aspectos de inovação.

Ressalta-se que a empresa FINA S/A, possui uma estrutura fortemente funcional, o que é uma característica comum em organizações do mesmo ramo de atividade, mas que por vezes mostra-se como burocrática e fonte de conflitos internos. Esta é também uma característica do Departamento de TI, que é organizado por superintendências, que também possuem gerências (áreas) e coordenações (por vezes, subáreas). É importante ressaltar, que nos últimos anos ocorreram mudanças significativas nas estruturas funcionais e suas responsabilidades, inclusive com rearranjo de funções.

Contudo, atualmente, a organização conta com um Escritório de Projetos no nível corporativo, sendo que alguns dos projetos são gerenciados pelo Departamento de TI (conforme suas características), especificamente pela Superintendência de Projetos e Sistemas, a qual reporta periodicamente seus indicadores para acompanhamento da sua evolução.

A Superintendência de Projetos e Sistemas é subdivida em dois grandes grupos de gerências: Projetos e Sistemas. Enquanto as gerências de Projetos fazem a gestão dos projetos de sistemas de informação, as gerências de Sistemas preocupam-se com a viabilização das entregas específicas de um projeto.

É importante salientar, que quando da execução de demandas de software utilizando a perspectiva tradicional de desenvolvimento, há também uma estrutura de Sustentação, responsável por atuar sobre incidentes e problemas dos sistemas legado. Sendo que no modelo ágil de desenvolvimento, no entanto, como há a utilização de times perenes, há um *backlog* único, segregado por tipos de itens distintos a serem tratados, porém idealmente mantidos pelo mesmo time que desenvolveu a solução (conforme ilustrado na **Figura 46**).

**Figura 46 – Backlog do Produto e Sustentação**

*Fonte: elaborado pelo autor.*

O desenvolvimento de software é completamente executado por fornecedores de software da organização, os quais têm seus dados e resultados mantidos e gerenciados por uma gerência específica com foco de controle, ligada à Superintendência de Qualidade e Segurança. O processo é também suportado pelo direcionamento definido por gerências de Estratégia de TI e Arquitetura Corporativa e de TI (Superintendência de Estratégia e Infraestrutura de TI), especialmente no que tange aos processos definidos pela subárea de Metodologia e Processos de TI, que define “o quê” e “como” deve ser a atuação de cada papel durante a execução de todas as atividades/fluxos inerentes ao desenvolvimento de software (o próprio processo e os relacionados).

Destaca-se também, que internamente existem diferentes papéis que compõem a estrutura do Departamento de TI, tais como: diretores, superintendentes, gerentes, coordenadores, especialistas, analistas de sistemas de negócio, analistas de qualidade, analistas de testes, arquitetos, engenheiros, analistas de segurança da informação, líderes de projeto, facilitadores, entre outros, os quais estão distribuídos em diferentes estruturas funcionais. As fábricas de software que suportam o departamento possuem também diversos papéis, a depender do modelo de desenvolvimento adotado para cada projeto, onde se destacam principalmente os seguintes: gerente de relacionamento/ preposto, gerente de fábrica, líderes técnicos, analista de requisitos, analista de sistemas,

analista de qualidade, arquiteto, engenheiro, desenvolvedor, analista de testes, testador, engenheiro de testes e outros.

Historicamente, desde a definição pela utilização da estratégia de *outsourcing*, no ano de 2010, o desenvolvimento de projetos de software passou a ser integralmente realizado pelas fábricas de software, contemplando *baselines* de horas/mês contratadas com cada fornecedor, e eventualmente através de demandas pontuais de desenvolvimento, o que contempla um complexo mecanismo de controle das contratações e dificuldade em realizar *benchmarking* destas.

Devido a recente implantação de um processo contemplando a utilização de métricas de software, no entanto, o controle passou a ser implícito ao modelo e o esforço de gestão tem aparentemente diminuído significativamente, além de dar visibilidade para comparações com dados obtidos junto ao mercado.

Os resultados deste trabalho poderão ser utilizados como insumo para o diagnóstico e evolução da operação da empresa FINA S/A, pois tendem a estar alinhados com os objetivos estratégicos e princípios da eficiência organizacional.

### **3.3 Coleta de Dados**

Para YIN (2005), a coleta de dados exige que o investigador possua habilidades e conhecimentos metodológicos para proceder à mesma. E neste contexto, as evidências utilizadas podem ser obtidas através de seis fontes distintas, a saber: documentação, registros em arquivo, observação direta, entrevistas, observação participante e artefatos físicos.

YIN (2005, p, 199) define que além da atenção que se dá a essas fontes em particular, existem alguns princípios que são importantes para o trabalho de coleta de dados na realização de estudos de caso, tais como: utilizar várias fontes de evidência; criar banco de dados para o estudo de caso; e, manter o encadeamento das evidências. O autor reforça ainda estes princípios para a coleta de dados são importantes para todas as seis fontes de evidências citadas anteriormente, podendo auxiliar o pesquisador e fazer frente ao problema de estabelecer a validade do constructo e confiabilidade de um estudo de caso. (YIN, 2005, p. 124).

Elaborado de acordo com o método de estudo de caso, este trabalho utilizará as seguintes técnicas para coleta de dados:

1) Documentação

- a. Foram utilizadas como principal insumo para entendimento dos processos de TI da organização, as suas descrições, conforme elas constam no Portal de Processo de TI da organização, uma aplicação *web* responsável por apresentar as suas atividades, papéis, responsabilidades e regras vinculadas;

2) Observação participante

- a. De acordo com GIL (1994), a técnica observação participante ou observação ativa é utilizada quando observador está inserido dentro do contexto que está sendo analisado, onde o mesmo assume um papel de membro do grupo. Conforme citado em tópicos anteriores, o pesquisador deste trabalho atua na área de TI da empresa estudada e participa da gestão dos projetos.

3) Entrevista parcialmente estruturada;

- a. Foram convidados sete profissionais para a entrevista, atuantes num contexto multifuncional da organização;
- b. O perfil dos entrevistados compreende analistas de três diferentes superintendências, sendo que estes profissionais possuem experiência e visibilidade suficiente dos processos em análise para entender e criticá-los;
- c. Optou-se pela técnica de entrevista parcialmente estruturada para fornecer certo grau de liberdade ao entrevistado, conforme ressalta GIL (2002), que defende que a entrevista parcialmente estruturada é orientada pelos entrevistados com vistas a atingir seus pontos de interesse – para a realização da entrevista foi utilizado o questionário apresentado no **Quadro 9**.

**Quadro 9 – Questões da Entrevista**

Constructo	Variável	Base Teórica	Pergunta de Pesquisa
Processos	Processo de Portfólio de Projetos de TI	Gerenciamento de Portfólio do SISP	Como é ou como deveria ser o processo de Portfólio de Processos?
	Processo de Gerenciamento de Projetos de TI	Gerenciamento de Projetos Ágeis	Como é ou como deveria ser o processo de Gerenciamento de Projetos de TI?

Classificação da Informação: **Uso Irrestrito**

	Processo de Desenvolvimento de Software	Desenvolvimento de Software	Como é ou como deveria ser o processo de Gerenciamento de Projetos de TI?
	Processo de Homologação		Como é ou como deveria ser o processo de Homologação de Software?
	Processo de Estimativas de Software	Abordagens do IFPUG e do COSMIC	Como é ou como deveria ser o processo de Estimativas de Software?
	Processo de Contratação de Fornecedores	Contratos	Como é ou como deveria ser o processo de Contratação de Fornecedores?
	Processo de Contratação de Serviços	Operação	Como é ou como deveria ser o processo de Contratação de Software?
	Processo de Gestão de Fornecedores	Gestão de Fornecedores	Como é ou como deveria ser o processo de Gestão de Fornecedores?
Aspectos Gerais	Pessoas e Cultura	Cultura	Qual a sua opinião sobre o assunto "Pessoas e Cultura"?
	Processos e Boas Práticas	Todos	Qual a sua opinião sobre o assunto "Processos/ Boas Práticas"?
	Arquitetura e Tecnologia	-	Qual a sua opinião sobre o assunto "Arquitetura e Tecnologias"?
	Ferramentas Internas	-	Qual a sua opinião sobre o assunto "Ferramentas Internas"?

Fonte: Elaborado pelo autor.

Para identificação de um público capacitado e realmente relacionado com a proposta deste trabalho, inicialmente foi enviado um questionário aos participantes, com o auxílio da ferramenta SurveyMonkey. Para estas questões, foram elencadas as seguintes premissas:

- Definir uma escala de avaliação – que possibilite um passível entendimento e preenchimento, e que possa ser dimensionar um fato relevante quantitativamente;
- Elaborar questões autoexplicativas – poucas pessoas leem instruções que são apresentadas, que por vezes se mostram confusas ou redundantes;
- Limitar o número de perguntas – o questionário não precisa ser extenso, para que não se torne cansativo ao respondente;
- Apresentar o questionário com boa qualidade visual – o questionário deve ser bem apresentado, sem erros de idioma, para que instigue o público definido a responder a pesquisa.

#### Quadro 10 – Questões da Enquete (Survey).

Constructo	Variável	Base Teórica	Pergunta de Pesquisa
Qualificação e Contextualização	Cocriação, filosofia enxuta e métodos ágeis.	Métodos Ágeis	Você conhece o Manifesto Ágil ou já utilizou métodos de desenvolvimento ágil? (pouco → muito)
		Design Thinking	Você conhece algum método de cocriação, tal

Classificação da Informação: **Uso Irrestrito**

			como o Design Thinking? (pouco → muito)
		Lean e Lean Startup	Você conhece métodos tais como Lean, Lean Startup ou o Lean MVP? (pouco → muito)
		Concepção Enxuta (Direto ao Ponto)	Você conhece a Concepção Enxuta utilizando o método "Direto ao Ponto"? (pouco → muito)
	Métricas de Software	APF, COSMIC	Você entende que a utilização de métricas de software é positiva para a organização? (pouco → muito)
			Você conhece métricas de software tais como pontos de função do IFPUG (APF) ou COSMIC? (pouco → muito, p/ a abordagem que mais conhece)
Contratos	Contratos	Você conhece algo sobre contratos de software? (pouco → muito)	
Geral	Todos	<p>Em que nível você acha que o modelo de trabalho da organização proporciona... (em relação a outros modelos que você conhece)</p> <ul style="list-style-type: none"> <li>• Eficácia (1 a 5)</li> <li>• Eficiência (1 a 5)</li> <li>• Transparência (1 a 5)</li> <li>• Flexibilidade (1 a 5) – é suficientemente genérico, para utilização em diferentes tipos de organizações.</li> <li>• Inovação no modelo de negócios (1 a 5)</li> <li>• Eliminação de desperdícios (1 a 5)</li> <li>• Controle dentro do que você considera como necessário (1 a 5)</li> <li>• Economia para o contratante (1 a 5)</li> <li>• Relação ganha-ganha (1 a 5)</li> <li>• Comparabilidade entre projetos e entre fornecedores (1 a 5)</li> </ul>	
Processos	Processo de Portfólio de Projetos de TI	Gerenciamento de Portfólio do SISP	Você entende que o processo de portfólio é adequado para o gerenciamento de projetos ágeis? Justifique.
			Seria adequado se a organização utilizasse um backlog de Épicos para a metodologia ágil?
	Processo de Gerenciamento de Projetos de TI	Gerenciamento de Projetos Ágeis	Você acredita que o processo de gerenciamento de projetos ainda deve sofrer adequações para contemplar a metodologia ágil?
	Processo de Desenvolvimento de Software	Desenvolvimento de Software	O processo de desenvolvimento de software é adequado à organização?
			Você acha que a organização poderia utilizar apenas a metodologia ágil?
			Você enxerga algum problema na proposta ágil proposta pela organização?
			O que você acha que poderia melhorar neste processo?
			Você acha que o processo ágil realmente tem características de um modelo enxuto?
			Você entende que o modelo de classificação de projetos através do Pace Layered é adequado?
			O que mais lhe chama a atenção no modelo ágil definido?
Qual a sua percepção sobre a Concepção Enxuta?			
Qual a sua percepção sobre o Planejamento no ciclo de desenvolvimento com métodos ágeis?			
Qual a sua percepção sobre a fase ágil de Execução?			

	Processo de Homologação		Você entende que a metodologia ágil deveria seguir o processo tradicional de homologação?
			Você entende que a proposta vigente para homologação em métodos ágeis é adequada?
	Processo de Estimativas de Software	Abordagens do IFPUG e do COSMIC	Você entende que o modelo utilizando APF é adequado para uma relação comercial?
			Você entende que é necessário poder comparar de forma objetiva o processo ágil e o tradicional?
			Você entende que tanto APF/SNAP como o COSMIC podem ser utilizados para contratações ágeis?
	Processo de Contratação de Fornecedores	Contratos	Você entende que o modelo de contratação ágil oportuniza a redução de custos de negociação?
	Processo de Contratação de Serviços	Operação	Você entende que o melhor modelo para a organização é a contratação de baselines ou via demanda pontual? Justifique.
			Na sua visão, qual a modalidade de contratação mais se adequa a necessidade dos requisitantes? Por quê?
			O que você acha que poderia melhorar neste processo?
		Processo de Gestão de Fornecedores	Gestão de Fornecedores
			Você entende que o modelo ágil oportuniza a redução de custos de monitoramento do fornecedor?
Aspectos Gerais	Pessoas e Cultura	Cultura	Você identifica problemas relacionados às pessoas e/ou cultura organizacional?
			Você visualiza que as pessoas entendem o que significa realmente eliminar desperdícios?
			Você identifica que em geral as pessoas são qualificadas para as suas funções e tem experiência de mercado suficiente para contribuir com a organização?
			Você acha que a estrutura funcional da organização é uma barreira para o sucesso da utilização de métodos ágeis?
	Processos e Boas Práticas	Todos	Você identifica falta de aderência às boas práticas? Ou ainda, identifica que as pessoas tendem a não seguir os processos?
Arquitetura e Tecnologia	-		Você entende que a arquitetura define padrões de tecnologia que são realmente seguidos?
Ferramentas Internas	-		As ferramentas internas apoiam o trabalho dos envolvidos nos processos citados?

Fonte: Elaborado pelo autor.

### 3.4 Análise de Dados

Visando analisar os dados coletados do processo de desenvolvimento de *software*, utilizou-se natureza predominante qualitativa, conforme previsto por GIL (2002, pág. 141). Isso ocorre, pois os resultados das pesquisas qualitativas podem

fornecer informações valiosas para diagnósticos do Departamento de TI da organização.

Como forma de análise, o autor realizou uma observação participante, onde realizou acompanhamento presencial do processo de desenvolvimento de *software* e pôde realizar avaliações do modelo de trabalho. Ressalta-se, que a técnica de observação é apresentada por Gil (2002) como uma forma para se construir hipóteses.

Neste contexto e conforme a execução das etapas anteriormente apresentadas, o pesquisador buscou identificar oportunidades para adequação e evolução dos processos utilizados pelo Departamento de TI da organização, em especial aqueles que remetem ao gerenciamento, contratação e desenvolvimento de *software*, visto que para Gil (2002), a forma de pesquisa sem um formato exato definido, ou seja, assistemático, para estabelecer relações entre as ocorrências do dia-a-dia pode fornecer indícios na busca de uma solução para problemas propostos pela ciência.

### **3.4 Análise de Documentos**

Segundo ROESCH (1999) ao final da coleta de dados, o pesquisador depara-se com uma quantidade de dados por vezes significativa, tendo, portanto, que organizá-los e interpretá-los posteriormente. Sendo assim, neste estudo foram contempladas múltiplas fontes de evidência.

Com base no cenário exposto, a análise de dados balizou-se na técnica de triangulação, sobre a qual VERGARA (2010) afirma que esta propõe uma estratégia para o alcance da validade do estudo, viabilizando uma alternativa para a obtenção de novas perspectivas e conhecimentos sobre um tema. Em complemento, YIN (2005) reforça que a triangulação dos dados possibilita várias análises de um mesmo tema, considerando as múltiplas fontes de evidência.

Desta forma, utilizando-se do método de triangulação dos dados, foi possível avaliar as informações existentes nos documentos obtidos, nas entrevistas com os colaboradores e também as informações que estão presentes no referencial teórico.

### 3.5 Limitações do Método

Devido à realização da pesquisa em um formato de estudo de caso, a análise realizada delimita os resultados encontrados ao campo de estudo referente à organização específica em questão. Assim, eles não contribuem com um propósito de aplicação em contexto de outras organizações. No entanto, as comunidades de engenharia de software que tratem os temas envolvidos podem basear-se nas práticas e nos problemas relatados, os quais podem ser comuns a outros contextos organizacionais.

Outro aspecto importante, que tange à temporalidade, é que visto que o caso em questão foi aplicado em um determinado espaço de tempo que pode logo já não representar mais a realidade da organização, sua utilização em um momento futuro poderá impactará em refazer parte, ou toda a pesquisa novamente, principalmente quando identificadas mudanças nas práticas ou estruturas em uso.

Ainda, ressalta-se que as questões elaboradas no protocolo de pesquisa parcialmente estruturada podem ter compreensões diferentes pelos diferentes entrevistados, o que pode induzir a erros de questionamento e pode implicar em eventuais divergências de entendimento.

## **4 DESENVOLVIMENTO E RESULTADOS**

Utilizando-se da técnica de triangulação de dados, este capítulo contemplará uma apresentação da organização em foco (FINA S/A), assim como um detalhamento das áreas que formam o Departamento de Tecnologia da Informação. Serão apresentados também, os resultados das entrevistas realizadas (em grupo e individuais), reproduzindo-se suas respectivas respostas obtidas.

### **4.1 Perfil da Organização**

A empresa FINA S/A forma um grupo econômico atualmente composto por noventa e cinco organizações de base comercial e onze empresas que estruturam o sistema. Em relação ao seu ramo econômico, a empresa situa-se no mercado financeiro/ bancário e atualmente possui cerca de três milhões de clientes, distribuídos em onze estados brasileiros.

O portfólio da organização é composto por mais de 100 produtos e serviços, com destaque para: conta corrente, débito em conta, crédito, poupança, cartões de crédito e débito, investimentos, seguros, consórcios, câmbio, cobrança e previdência. Além dos produtos básicos, há também canais e facilidades, que atuam como caixas eletrônicos e serviços pela internet.

Devido às características do negócio, cada vez mais dependente da tecnologia, a empresa investiu, nos últimos anos, na ampliação da infraestrutura de TI, sendo esta uma responsabilidade do seu Departamento de TI.

#### **4.1.1 Departamento de Negócio**

As áreas de negócio de todas as empresas vinculadas ao FINA S/A são representadas por uma empresa centralizadora do grupo, portanto, um grupo de áreas que integram o Departamento de Negócio.

#### *4.1.1.1 Áreas de Negócio*

As “Áreas de Negócio” constituem-se de unidades funcionais da empresa agregadora, que definem regras de negócio comuns ao grupo. Elas são responsáveis por cadastrar as “Demandas” que posteriormente poderão ser agregadas como um ou mais projetos da organização.

#### *4.1.1.2 Escritório de Projetos Corporativo*

Os projetos, por sua vez, são definidos e monitorados por uma estrutura funcional corporativa, chamada “Escritório de Projetos”. Sendo assim, tanto os projetos de negócio como os projetos especificamente da TI, são tratados pela mesma área. Contudo, diferentemente das demais áreas, que são vinculadas às diretorias, esta é gerida por um Superintendente, que se reporta diretamente à Presidência da organização.

#### *4.1.2 Departamento de TI*

O Departamento de Tecnologia da Informação constitui-se de Diretoria de Tecnologia da Informação, representada pelo CIO da organização, diretamente vinculado à Presidência. A missão deste departamento é fornecer serviços de TI adequados à necessidade do negócio das empresas agregadoras e também das demais empresas vinculadas ao mesmo grupo.

Atualmente, a TI da empresa possui algumas centenas de colaboradores, cuja estrutura hierárquica funcional é formada por um diretor, diversos superintendentes, gerentes e coordenadores, além de especialistas, analistas e líderes de projetos responsáveis pelo apoio à gestão.

#### *4.1.2.1 Áreas de Projetos de Software*

As áreas de Projetos são segmentadas por assuntos sob a sua responsabilidade, tais como Cartões, Investimentos ou Convênios. Elas são formadas essencialmente por gerentes, especialistas e líderes de projetos.

Os líderes de projeto realizam um conjunto de atividades de gestão, demandando as áreas de Sistemas, que se responsabilizam pela coleta e entrega dos requisitos. Observa-se que ambas as estruturas são vinculadas à Superintendência de Projetos e Sistemas.

Destaca-se ainda, que a situação de cada projeto é reportada ao Escritório de Projetos, para que se assegure um acompanhamento corporativo das iniciativas estratégicas planejadas e em execução.

#### *4.1.2.2 Áreas de Sistemas*

As áreas de Sistemas são normalmente acionadas pelas áreas de Projetos, atuando no suporte e acompanhamento da execução dos projetos de desenvolvimento e evolução de software. O seu foco é a entrega de funcionalidade ao usuário, no tempo adequado (prazo acordado).

#### *4.1.2.3 Sustentação de TI*

Assim como as áreas de Projetos e de Sistemas, a Sustentação atua na gestão e execução das demandas de software, porém tem foco em pequenas unidades/ pacotes, caracterizadas principalmente como manutenções corretivas e de performance e identificadas através de um incidente ou problema surgido no ambiente de produção.

#### *4.1.2.4 Qualidade de TI*

Os artefatos (inclusive o código fonte), os quais são gerados pelos fornecedores através da sua atuação no processo de desenvolvimento de software (independente da área responsável pelas demandas vinculadas), são avaliados de forma qualitativa (código fonte apenas) e quantitativa (por amostragem) pela área de Qualidade de TI.

Embora a maioria dos artefatos seja avaliada através de uma análise humana, também são providas ferramentas que avaliam o código gerado a partir de visualizações fornecidas por uma ferramenta de código aberto chamada “Sonar”.

Salienta-se que esta mesma área é também responsável pela “Gerência de Configuração”, “Homologação de software” e “Mudanças e Liberações” (cada uma, uma subárea), auxiliando assim, no controle do ambiente técnico da organização.

#### *4.1.2.5 Estratégia de TI e Arquitetura*

As definições sobre a arquitetura de TI são vinculadas à arquitetura corporativa da empresa e configuram assuntos complexos e necessários para o direcionamento da TI, impactando diretamente no modelo de desenvolvimento de sistemas e da infraestrutura idealizada como solução padrão.

Da mesma forma, atuando como direcionadores de um caminho a ser traçado pelos demais, a estratégia e a metodologia definida para cada contexto na organização é considerado um fator crítico para o sucesso das implementações de software, de acordo com cada contexto vislumbrado.

#### *4.1.2.6 Infraestrutura e Serviços de TI*

Tão importante quanto à definição dos objetos a serem contratados ou o modo como eles serão desenvolvidos, a organização em foco entende que um conjunto da infraestrutura e serviços é necessário, e deve suportar o crescimento do ecossistema de maneira escalável. Sendo assim, esta é uma responsabilidade das áreas de Infraestrutura e Serviços de TI. Estas áreas, a saber, fornecem as condições e a flexibilidade necessária para que as demais consigam desempenhar seu papel estratégico no que tange ao desenvolvimento de software.

#### *4.1.2.7 Outsourcing de TI*

As contratações de fornecedores de software, devido à necessidade de padronização e controle destas, ocorrem de forma centralizada, em uma área cuja

responsabilidade abrange a formalização e contratação de demandas relacionadas à projetos de Tecnologia de Informação e Comunicação, envolvendo portanto, as contratações de software e hardware.

Especificamente sobre as contratações de software, embora a estratégia de *sourcing* seja direcionada pela área de Estratégia de TI, a área responsável por criar, manter e executar as regras relacionadas ao *outsourcing* é o *Outsourcing de TI*.

A área de *Outsourcing de TI* realiza interfaces frequentes com as demais áreas do seu departamento, principalmente as de Projetos e de Sistemas, fornecendo suporte especializado para as atividades de contratação, orçamentação, capacidade e nível de serviço dos fornecedores sob sua gestão. Destaca-se no último ano, a implantação do Escritório de Métricas da organização como parte integrante desta área, instaurando assim, um processo formal de estimativas de software, o qual se constitui de um conjunto de padrões internacionais.

## **4.2 Estudo de Caso: Apresentação e Análise**

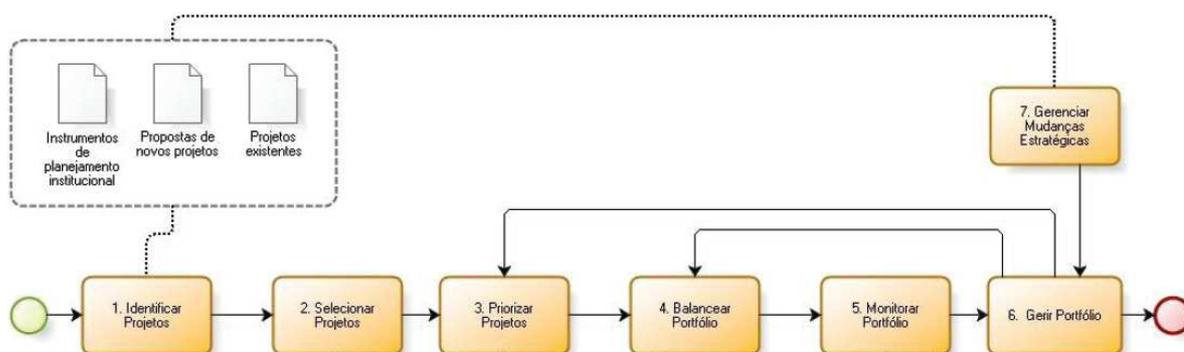
Conforme exposto anteriormente, este trabalho contempla técnicas de coleta dos dados que compreendem a realização de entrevistas individuais e em grupo, as quais foram parcialmente estruturadas, de forma a contemplar o potencial de desenvolvimento de software e os processos relacionados a este assunto dentro da organização.

### **4.2.1 Processo de Portfólio de Projetos**

O processo de portfólio de projetos está atualmente em reestruturação para contemplar uma perspectiva ágil e direcionada para prover melhor planejamento e controle a curto e médio prazo, com maior visibilidade do escopo a ser trabalhado ou do progresso sobre os itens em execução. As entrevistas expuseram que haverá uma composição gradual do Portfólio de Projetos, por ciclos, já em 2016, onde será criado também um fluxo específico para Projetos maiores, de forma segregada dos projetos menores, que serão chamados Melhorias.

O formato de trabalho atual, em termos das atividades executadas, mostrou-se muito similar à abordagem sugerida pelo SISP (2011) para tratamento de Portfólio de Projetos, conforme apresentado na **Figura 47**. Deste modelo, diferenciam-se os agrupamentos das atividades em etapas (presente no modelo da organização em questão, mas ausentes na abordagem do SISP) e os artefatos gerados, assim como os procedimentos que são executados internamente a cada atividade pelos envolvidos no processo. Ou seja, as atividades executadas e o sequenciamento, são idênticos e bem representados como sequencia lógica na **Figura 47**.

**Figura 47 – Processo de Portfólio de Projetos de TI**



Fonte: SISP (2011).

**Quadro 11 – Respostas: Gerenciamento de Portfólio de Projetos**

Pergunta de Pesquisa	Na sua visão: como é ou como deveria ser o processo de Portfólio de Processos?
1	Não sei.
2	Deveríamos pensar nos três modelos e talvez definir algo novo para o portfólio.
3	Não sei.
4	Não sei.
5	Portfólio com Épicos seria viável?
6	Visão de produto ao invés da visão de projeto.
7	Não sei.

Fonte: Elaborado pelo autor.

## 4.2.2 Processo de Gerenciamento de Projetos

Em relação aos projetos de desenvolvimento tradicionais, conforme informações disponíveis no Portal de Processos do Escritório de Projetos, foi identificado que o Processo de Gerenciamento de Projetos (**Figura 48**), contempla o papel do Líder de Projeto, assim como um conjunto de fases fundamentadas no PMBOK.

**Figura 48 – Fases de Gerenciamento de Projetos**



*Fonte: elaborado conforme Portal de Processos do EP.*

A fase de “Iniciação” compreende autorizar o projeto ou fase, sendo caracterizada pela emissão de um Termo de Abertura, que remete à aprovação de um comunicado relacionado pelo solicitante e patrocinador do projeto (no caso de ausência de um destes, um superior poderá substituí-lo).

No “Planejamento” pode-se definir e refinar os objetivos, bem como planejar as estratégias e o Plano de Trabalho para se alcançar tais objetivos. São artefatos obrigatórios desta fase: o Checklist de Planejamento, o Cronograma do Projeto e a Matriz de Custos; São opcionais: o Blue Print, a Matriz de Riscos e um Plano de Gestão de Mudança.

A fase de Execução compreende coordenar pessoas e outros recursos (humanos, tecnológicos, materiais, mercadológicos e diversos) para realizar o planejamento realizado. Pode contemplar uma ou mais ocorrências da Solicitação de Alteração de Escopo, caso exista incidência de change request do projeto, que

deve ser aprovada com um comunicado relacionado pelo próprio solicitante e patrocinador do projeto (no caso de ausência de um destes, um superior poderá substituí-lo).

O Monitoramento contempla a realização de Status Report e Atas de Reunião. Seu objetivo é verificar o alcance dos objetivos, controlar o desempenho e identificar desvios no projeto, bem como as suas ações corretivas.

Já na fase de Encerramento, é necessário formalizar a aceitação do projeto ou fase, através do Termo de Encerramento, com aprovação do comunicado que o contem, devendo isso ser realizado pelo solicitante e patrocinador do projeto (no caso de ausência de um destes, um superior poderá substituí-lo). Há também a necessidade de emissão do Relatório de Aceitação.

No entanto, o processo de gerenciamento de projetos atualmente não possui um padrão claramente definido para a metodologia ágil, sendo que ao questionar os usuários do processo, constatou-se que não há uma orientação específica para o direcionamento que deve ser utilizado. Na prática, contudo, conforme pontuaram os entrevistados, existem procedimentos diferenciados sendo executados.

Foi diagnosticado que alguns líderes de projeto adaptam o modelo de gestão para uma abordagem como o a do Scrum, até mesmo para projetos cujas demandas foram categorizadas no método tradicional. Desta forma, sabe-se que existem diversos *frameworks* e metodologias que podem ser utilizados com vistas ao gerenciamento de projetos, sejam tradicionais ou ágeis, mas na falta de um padrão formal e no contexto deste estudo, foi identificada uma abordagem mista, mas essencialmente tradicional na organização, no entanto, com forte propensão a contemplar mudanças.

Conforme um dos entrevistados pontuou, o vínculo das fases de gerenciamento de projetos com as fases de desenvolvimento de software vislumbradas para o modelo ágil pode ser ilustrado na **Figura 49**. Nesta ilustração, a Iniciação do projeto foi chamada de Target, e embora este passo não esteja detalhado no Portal de Processos da organização, é a primeira visão de escopo do projeto, que se é realizado em um momento separado da Concepção do projeto.

**Figura 49 – Fases de Projetos e Etapas de Desenvolvimento**



Fonte: elaborado pelo autor.

**Quadro 12 – Respostas: Gerenciamento de Projetos**

Pergunta de Pesquisa	Na sua visão: como é ou como deveria ser o processo de Gerenciamento de Projetos de TI?
1	Deveríamos ter um fluxo previsto para o ágil.
2	Deveríamos ter uma metodologia que utilizasse mais os recursos dos modelos ágeis.
3	Tenho dúvidas para métodos ágeis.
4	Poderíamos melhorar.
5	Atende.
6	Não sei.
7	Não sei.

Fonte: Elaborado pelo autor.

#### 4.2.3 Processo de Desenvolvimento de Software

Oficialmente, existem três metodologias de desenvolvimento utilizadas pela empresa, que configuram, portanto, em três modelos de trabalho distintos, a saber:

- Modelo Tradicional, aplicável aos sistemas das demandas de software cuja camada de ritmo (*Pace Layered*) seja do tipo “Registro”;
- Modelo Híbrido, quando a camada de ritmo do sistema é definida como “Diferenciação”;

- Modelo Ágil, quando a camada de ritmo do sistema for do tipo “inovação”.

O modelo tradicional de desenvolvimento de software utiliza o conjunto de fases demonstrado na **Figura 50**, onde são destacados os pontos de responsabilidade do cliente (em verde), do fornecedor (em cinza) e de ambos (em verde claro). Observa-se que, conforme informação obtida nas entrevistas realizadas, os artefatos/atividades do modelo tradicional são considerados obsoletos por grande parte dos usuários do processo, tanto atualmente a área responsável pela Metodologia está trabalhando em uma forma de torná-lo mais enxuto, ou seja, mas aderente à filosofia *Lean*.

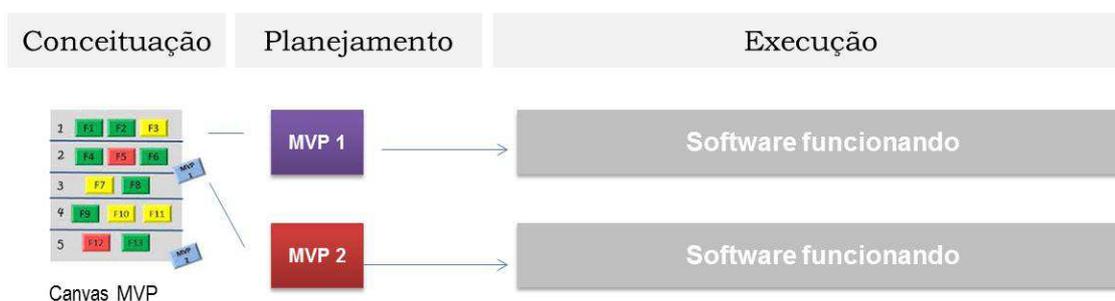
**Figura 50 – Etapas do Método Tradicional**



*Fonte: elaborado pelo autor.*

Em relação à metodologia ágil, foi utilizada integralmente a proposta de Concepção Enxuta proposta por Caroli (2015b), conhecida como “Direto ao Ponto” (derivada principalmente do *Lean Startup* e do *Design Thinking*). Sendo que, neste caso, o Planejamento e a Execução dos projetos seguem uma abordagem mista, orientada pelos métodos *Lean Startup*, *Scrum* e *Extreme Programming*, além de práticas *DEVOPS*. Este modelo de trabalho é representado pela **Figura 51**.

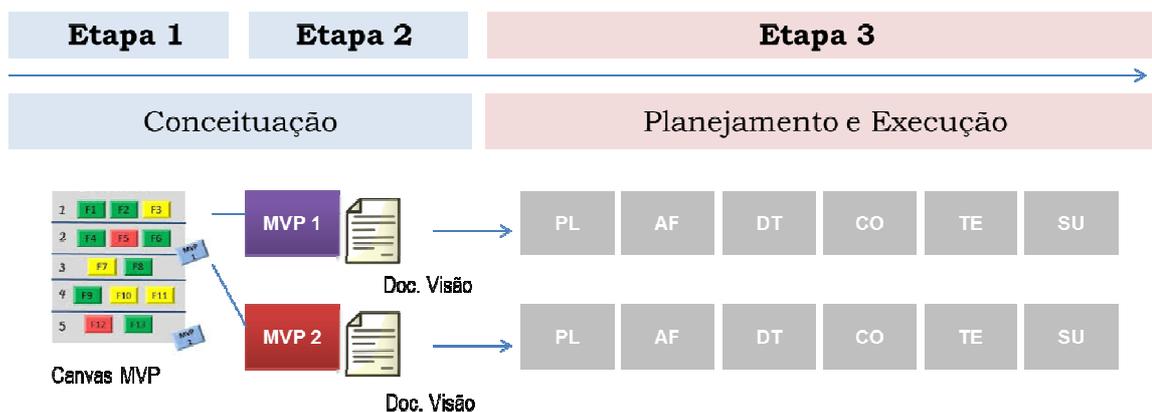
**Figura 51 – Etapas do Método Ágil**



*Fonte: elaborado pelo autor.*

Há também uma abordagem utilizando o “Direto ao Ponto” para o método tradicional, onde a camada de ritmo é a “Diferenciação”. Este modelo de trabalho é ilustrado na **Figura 52**.

**Figura 52 – Etapas do Método Híbrido**



*Fonte: elaborado pelo autor.*

Cabe ressaltar, no entanto, que o processo está formalizado, porém foi identificado que ele nem sempre é seguido, conforme relatos obtidos nas entrevistas realizadas. Foi reportado também, que a depender do direcionamento dado pelas áreas de Projetos e/ou Sistemas para o fornecedor contratado, por vezes dependendo também da tecnologia ou tipo de aplicação em questão, existem situações e roteiros de atividades/ entregáveis executados bastante distintos, porém não formalizados e/ou documentados no “Portal de Processos de TI”.

**Quadro 13 – Respostas: Processo de Desenvolvimento**

Pergunta de Pesquisa	Na sua visão: como é ou como deveria ser o processo de Desenvolvimento de Software?
1	Teríamos que ter roteiros diferentes por tipo de aplicação, tecnologia, tipo de negócio.
2	O processo atual é obsoleto.
3	Não seguimos fielmente.
4	Deveríamos ter inspeção.
5	Não tenho opinião.
6	Não atende.
7	Muito velho.

Fonte: Elaborado pelo autor.

#### 4.2.6 Processo de Homologação de Software

No que diz respeito ao modelo de homologação de software estabelecido, a pesquisa evidenciou que os entrevistados, de uma forma geral, procuraram comunicar que existe um processo que cumpre o seu papel, porém o contrato, formalizado com o fornecedor foi firmado em base horária e prevê um consumo de horas mensal, em um volume significativo em relação ao desenvolvimento executado (pelo fornecedor contratado para este tipo de atividade). Da mesma forma, contratualmente também não há uma regra objetiva para as estimativas realizadas para homologação.

Outra questão levantada, é a incerteza sobre a eficiência do processo de homologação do modelo ágil, que poderia ter uma segregação na fase de Execução, para contemplar a migração para produção, tal como no modelo ágil sugerido pelo SISP, que implementa uma fase específica para a Transição.

### Quadro 14 – Respostas: Homologação de Software

Pergunta de Pesquisa	Na sua visão: como é ou como deveria ser o processo de Homologação de Software?
1	Não sei.
2	Para o modelo ágil deveríamos ter algo diferente do que foi previsto.
3	Não sei ainda. Precisamos rodar no ágil primeiro.
4	Deveríamos ter mais flexibilidade.
5	Deveríamos usar PF para pagar o fornecedor.
6	Não sei.
7	Não sei.

Fonte: Elaborado pelo autor.

Foi citado também, que na metodologia ágil não há efetivamente um processo de homologação, mas uma fase de testes funcionais acompanhados pelo mesmo fornecedor que executa a homologação no modelo de desenvolvimento Tradicional (também aplicável ao método Híbrido), ou seja, é uma substituição de atividades e não propriamente uma homologação sendo realizada.

#### 4.2.7 Processo de Estimativas de Software

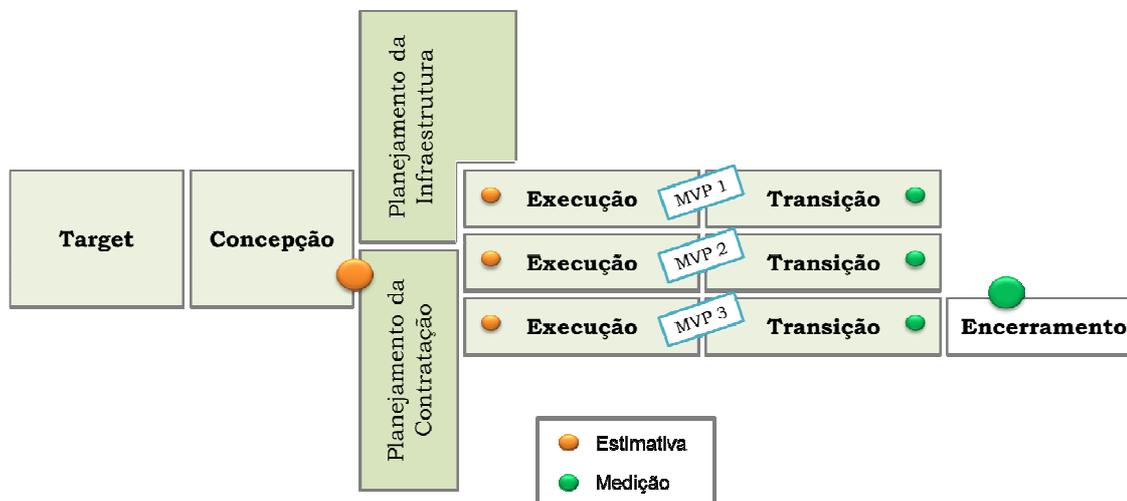
No modelo tradicional as estimativas são realizadas utilizando Pontos de Função e SNAP, do IFPUG, em todos os cenários – efetuando-se pagamentos conforme a aferição realizada –, além do COSMIC, para formação de base histórica (cenários de BI - *Business Intelligence* e barramentos desenvolvidos nas tecnologias OSB - *Oracle Service Bus* e BPEL - *Business Process Execution Language*). No processo tradicional, as estimativas são realizadas no início do ciclo de vida, na etapa de Planejamento, já como uma contagem detalhada executada sobre o Documento de Visão trabalhado pelo fornecedor (por aproximação), o que exige um alto grau de detalhamento inicial da solução. Em caso de ocorrência de solicitações de mudança de escopo, conforme disparado nos processos de Gerenciamento de Projetos e Desenvolvimento de Software, há contagem que revisa a estimativa inicial realizada. Ressalta-se um processo bastante similar para o modelo híbrido de

desenvolvimento, onde também se remunera com base numa estimativa de pontos de função. Isso se deve principalmente à existência de contratos com *baseline* em horas firmado, com desembolso mensal previsto.

As estimativas para contratações de software ágil, por sua vez, são obtidas através de contagens de pontos de função, realizadas também com base nas abordagens do IFPUG e COSMIC, seguindo conforme o modelo ilustrado na **Figura 53**. O diferencial deste modelo, é que uma estimativa é realizada durante a etapa de Planejamento, gerando um adiantamento, e cujo valor demonstra-se corrigido na entrega final, a cada MVP entregue.

Assim, neste modelo de métricas, o pagamento dá-se efetivamente contra as entregas realizadas e a última contagem (medição final) acaba por não ser necessária ao utilizar a abordagem do IFPUG em conjunto com as *Diretrizes Ágeis de Contagem na Abordagem IFPUG*, uma transcrição dos principais trechos que constam no Guia de Diretrizes da organização em relação à metodologia ágil.

**Figura 53 – Pontos de Estimativa ou Medição**



*Fonte: elaborado pelo autor.*

Foi destacado um ponto de atenção significativo em relação à estrutura funcional onde o Escritório de Métricas situa-se, visto que a mesma área que define o direcionamento técnico das contratações também formaliza e negocia os contratos com os fornecedores.

### Quadro 15 – Respostas: Estimativas de Software

Pergunta de Pesquisa	Na sua visão: como é ou como deveria ser o processo de Estimativas de Software?
1	O processo parece adequado para o modelo de contratação que temos.
2	Não sei.
3	Creio que esteja aderente à proposta.
4	Não sei.
5	Não sei dizer.
6	Tentamos coletar requisitos muito detalhados no início das demandas. Isso é prejudicial ao processo.
7	Não poderíamos medir na Análise Funcional?

*Fonte: Elaborado pelo autor.*

#### 4.2.7.1 Diretrizes Ágeis de Contagem na Abordagem IFPUG

A contagem será realizada considerando as funcionalidades dos MVPs de maneira incremental, sendo que todas as funções referentes aos MVPs anteriores permanecerão na mesma planilha de contagem (portanto, será atualizada a contagem a cada MVP), conforme os exemplos a seguir.

#### Exemplo MVP1

O MVP1 do projeto/ demanda 9999 envolve o seguinte escopo mapeado (e respectiva estimativa):

- Cadastro de cliente (Inclusão);
- Incluir cliente (Inclusão);
- Excluir cliente (Inclusão);
- Relatório de clientes (Inclusão).

#### Contagem do MVP1 (demanda 9999)

Função	Manutenção	Tamanho (PF)
Cadastro de cliente	I	10
Incluir cliente	I	4
Excluir cliente	I	3

Emitir relatório de clientes	I	5
<b>Total:</b>		<b>22</b>

### Exemplo MVP2

O MVP2 deste projeto/ demanda envolve o seguinte escopo mapeado (e respectiva estimativa):

- Alterar cliente (Inclusão);
- Consultar cliente (Inclusão).

#### Contagem do MVP2 (demanda 9999)

Função	Manutenção	Tamanho (PF)
Cadastro de cliente	I	10
Incluir cliente	I	4
Excluir cliente	I	3
Emitir relatório de clientes	I	5
<b>Alterar cliente</b>	<b>I</b>	<b>4</b>
<b>Consultar cliente</b>	<b>I</b>	<b>4</b>
<b>Total:</b>		<b>30</b>

### Exemplo MVP3

O MVP3 do mesmo projeto/ demanda envolve o seguinte escopo mapeado (e respectiva estimativa):

- Incluir cliente (alteração da funcionalidade para inclusão de novas regras de negócio, respondendo a uma mudança de escopo solicitada através de artefato específico de *change request*, sendo que esta necessidade foi identificada somente no MVP3).

#### Contagem do MVP3 (demanda 9999)

Função	Manutenção	Tamanho (PF)
Cadastro de cliente	I	10
Incluir cliente	I	4
Excluir cliente	I	3
Emitir relatório de clientes	I	5

Alterar cliente	I	4
Consultar cliente	I	4
<b>Incluir cliente</b>	<b>A</b>	<b>2</b>
<b>Total:</b>		<b>32</b>

#### 4.2.6 Processo de Contratação de Fornecedores

O modo como são realizadas as contratações de fornecedores software foi considerado burocrático e lento pelos entrevistados, além de eficaz em alguns aspectos, mas não necessariamente eficiente.

A área de *Outsourcing de TI*, no entanto, tem demonstrado interesse em formalizar um modelo de homologação de fornecedores considerando critérios objetivos e com um contrato do tipo “guarda-chuva” firmado, do qual se originam uma ou mais Ordens de Serviço relacionadas, contribuindo para redução de etapas burocráticas do processo, que impacta diretamente no fluxo de entregas das áreas de Sustentação, Projetos e Sistemas.

Outro ponto ressaltado, é que as áreas requisitantes e as áreas de controle deveriam interagir mais, quebrando barreiras culturais e silos criados pela própria natureza corporativa de uma instituição financeira.

#### Quadro 16 – Respostas: Contratação de Fornecedores

Pergunta de Pesquisa	Na sua visão: como é ou como deveria ser o processo de Contratação de Fornecedores?
1	Mais dinâmico e mais rápido.
2	Modelo padrão de contratação por categoria de contrato.
3	Deveríamos pensar em reduzir o tempo de contratação. Muito longo!
4	Há muitos repiques atualmente (entre as áreas).
5	SLAs muito longos.
6	Deveríamos homologar os fornecedores com modelos de RFPs.
7	Deveríamos ter critérios e homologar o fornecedor previamente.

*Fonte: Elaborado pelo autor.*

#### 4.2.7 Processo de Contratações de Software

Este processo contempla o controle de ciclo de vida das Ordens de Serviço geradas através de um contrato do tipo “guarda-chuva”, além do controle do ciclo de desenvolvimento sobre itens desenvolvidos pelos fornecedores.

Segundo os entrevistados, as contratações de software são realizadas através de uma ferramenta de gestão de demandas, que atualmente não supre todas as necessidades de cadastro e controle. Foi identificado que o nível de controle não é adequado e que há grande volume de trabalho operacional. São utilizadas planilhas para formalização e controle de demandas, desde o seu registo até a finalização do processo de prestação de contas, o que foi avaliado como desperdício, em contra razão ao modelo *Lean* de operação.

**Quadro 17 – Respostas: Contratações de Software**

Pergunta de Pesquisa	Na sua visão: como é ou como deveria ser o processo de Contratação de Software?
1	Poderíamos ter um modelo baseado em instrumentos do contrato principal (como ordens de serviço e termos associados no modelo do SISP).
2	Poderia ser mais enxuto.
3	Não há ferramentas de controle, o processo é muito operacional.
4	Não sei.
5	Há muita coisa manual.
6	Não sei.
7	Não sei dizer.

*Fonte: Elaborado pelo autor.*

#### 4.2.8 Processo de Gestão de Fornecedores

O processo de contratação, manutenção e gestão dos fornecedores foi analisado em termos de sua efetividade.

A gestão dos serviços prestados pelos fornecedores contratados contempla, portanto, as principais atividades de suporte e gestão dos serviços terceirizados de desenvolvimento de software, tais como:

- Validação e Reporte de Serviços Prestados – compreende os serviços a declaração de serviços prestados pelo fornecedor no mês de referência de prestação de serviços;
- Validação e Reporte de Nível de Serviço – compreende a apuração e validação do nível de serviço relacionado à prestação dos serviços conforme declaração apresentada pelo fornecedor;
- Reporte de Prestação de Contas e Aprovações – compreende o reporte dos serviços prestados considerando multas e bonificações, obtendo-se as devidas autorizações para a realização do pagamento ao fornecedor na data estabelecida.

Foi mapeado, que a cada período, após a apuração do nível de serviços dos fornecedores, é confeccionado pela área responsável pelo Outsourcing de TI um *dashboard* executivo sobre as contratações realizadas, reportando as áreas requisitantes dos serviços (Projetos e Sistemas) e servindo para o acompanhamento executivo pelo CIO e superintendentes impactados. As atualizações e reportes, no entanto, são realizados de forma manual, sem uma ferramenta própria para esta finalidade.

### Quadro 18 – Respostas: Gestão de Fornecedores

Pergunta de Pesquisa	Na sua visão: como é ou como deveria ser o processo de Gestão de Fornecedores?
1	Muito burocrático na questão de aprovações.
2	Muito manual.
3	Não é efetivo se os indicadores não tiverem um propósito maior associado.
4	Deveríamos saber por que medimos os indicadores.
5	As pessoas deveriam ser envolvidas em relação ao que se quer medir realmente (o que faz diferença para a entrega e para o negócio).
6	Não sei.
7	Não sei opinar.

*Fonte: Elaborado pelo autor.*

#### 4.2.9 Aspectos Gerais da Organização

Em um contexto geral, foi identificado que alguns aspectos precisam ser analisados e registrados para aprofundamento da análise, ou ainda, visualizados como um ponto de atenção, conforme será abordado a seguir.

##### 4.2.9.1 Pessoas e Cultura

Na pesquisa, foi citada a preocupação com o grande número de colaboradores com muito tempo de empresa e pouca experiência prévia no mercado.

Também foi citado, que há um elevado número de profissionais oriundos de áreas de formação não diretamente relacionadas, o que especificamente em relação a TI pode ser particularmente prejudicial à qualidade e eficiência dos serviços.

Especificamente para o contexto ágil, onde a organização pretende ter times de trabalho aparentes, foi sugerido atentar para o perfil solicitado.

### Quadro 19 – Respostas: Pessoas e Cultura

Pergunta de Pesquisa	Na sua visão: qual a sua opinião sobre o assunto “Pessoas e Cultura”?
1	Existe muita discussão entre as áreas,
2	Pessoas discutem por motivos tolos.
3	Discussões intermináveis na TI prejudicando o negócio.
4	Qualificação dos profissionais é questionável (em algumas áreas).
5	Estrutura muito funcional prejudica as relações humanas.
6	Assunto delicado.
7	Não sei opinar.

*Fonte: Elaborado pelo autor.*

#### 4.2.9.2 Processos e Boas Práticas

Há processos que se mostraram burocráticos e alguns outros que muitas vezes são considerados incipientes em sua proposta. Contudo, o maior problema identificado é que os processos são definidos, mas nem sempre seguidos ou reavaliados contra uma proposição de valor.

Foi identificado durante as entrevistas, que há um movimento pela revisão do ciclo de desenvolvimento e demais processos relacionados, constituindo um movimento bastante significativo envolvendo e diversas áreas e pessoas chave da TI da organização. Neste movimento, o principal direcionador será a visão enxuta dos processos, o que vem de encontra a proposta deste trabalho.

### Quadro 20 – Respostas: Boas Práticas

Pergunta de Pesquisa	Na sua visão: qual a sua opinião sobre o assunto “Processos/ Boas Práticas”?
1	Deveríamos ter um fórum de processos efetivo.
2	Os processos não são seguidos.
3	O processo tradicional de desenvolvimento está ultrapassado.
4	Deveríamos ter uma central de ideias, para serem votadas.
5	Fazemos as coisas e não nos perguntamos por quê.
6	O processo de desenvolvimento tradicional é pesado.
7	Tudo pra ontem e as melhorias acabam ficando pra depois.

*Fonte: Elaborado pelo autor.*

#### 4.2.9.3 Arquitetura e Tecnologias

As tecnologias utilizadas e o direcionamento definido pela Arquitetura de TI mostraram-se robustos, porém nem sempre são efetivamente padronizados a ponto de refletirem um processo realmente instaurado.

Foi citado por dois dos entrevistados que o Parecer de Arquitetura, um dos artefatos do ciclo de desenvolvimento, poderia ser eliminado na maioria das situações se houvesse um direcionador claro para alguns cenários padrões.

### Quadro 21 – Respostas: Arquitetura e Tecnologias

Pergunta de Pesquisa	Na sua visão: qual a sua opinião sobre o assunto “Arquitetura e Tecnologias”?
1	O parecer poderia ser um fluxo de exceção? É um caminho crítico nos projetos.
2	Não sei opinar.
3	Os padrões deveriam ser mais divulgados.
4	O fornecedor deveria estar comprometido em seguir os padrões.
5	Não sei opinar.
6	As pessoas deveriam ser conscientizadas.
7	Não sei opinar.

*Fonte: Elaborado pelo autor.*

#### 4.2.9.4 Ferramentas Internas

As ferramentas internas atualmente utilizadas na organização para controle do ciclo de desenvolvimento (ALM) e para execução de fluxos operacionais mostraram-se incapazes de lidar com as necessidades de seus usuários. Os entrevistados demonstraram preocupação com o trabalho redundante, falta de suporte para atividades básicas e para com a usabilidade dos sistemas utilizados.

Também foi citado durante uma das entrevistas, que a falta de uma infraestrutura confiável de certas aplicações internas, e até mesmo falta de definição sobre um responsável pelos produtos em questão, são fatores críticos que tem impactos significativos no esforço consumido nas ações da TI.

Ações de adoção de práticas *DEVOPS*, desde automação de testes e integração contínua até automação de ambientes, ações que estão no roadmap da organização, foram lembrados como fatores que poderão prover ganhos escaláveis a curto e médio prazo, reduzindo o desperdício associado ao desenvolvimento de software e aumentando a confiabilidade das soluções adotadas.

**Quadro 22 – Respostas: Ferramentas Internas**

<b>Pergunta de Pesquisa</b>	<b>Na sua visão: qual a sua opinião sobre o assunto “Ferramentas Internas”?</b>
1	Usabilidade das ferramentas geralmente é deficiente.
2	Não atende a necessidade.
3	Não atende.
4	Faço muito trabalho operacional.
5	Eficazes e em partes.
6	Não fornece controles.
7	Falta muito para realmente ajudar.

*Fonte: Elaborado pelo autor.*

## 5 CONSIDERAÇÕES FINAIS

O objetivo principal deste trabalho foi analisar os requisitos e procedimentos necessários para melhorar aspectos dos processos de gerenciamento, contratações e desenvolvimento de software na TI da empresa FINA S/A, de forma aderente ao direcionamento estratégico que privilegia a entrega de software com maior frequência com vistas aos critérios de economicidade oriundos da adoção do Lean, pelo princípio da eliminação de desperdícios.

A situação atual da organização, representada pelo diagnóstico provido pela utilização do método de estudo de caso, com base nos portais de processos, documentos e entrevistas realizadas, fornece visibilidade sobre o modelo de trabalho e critérios para execução das demandas de desenvolvimento de software, desde a solicitação pelas áreas de negócio até o atendimento das solicitações de desenvolvimento vinculadas.

Durante a etapa do estudo que compreendeu o diagnóstico, foi possível constatar que alguns fornecedores estratégicos, no passado eram preferencialmente contratados com base em bolsões de horas, chamados baselines, prevendo o consumo mensal de um quantitativo pré- definido (em horas). E que as novas contratações, no entanto, têm um direcionamento mais “pontual”, devido a uma série de questões, principalmente às que remetem ao prazo de entrega, cada vez mais um fator determinante no mercado.

O processo de gerenciamento de gerenciamento de projetos de software, assim como o de portfólio de projetos, para o contexto de métodos ágeis, ainda está em elaboração, e no momento o modelo ágil de desenvolvimento é vinculado ao gerencialmente tradicional com pequenas adaptações.

Contudo, mesmo em relação ao processo de desenvolvimento de software, foi identificado que existem oportunidades de melhoria com significativa importância, visto que em algumas questões da entrevista os respondentes apresentaram divergências em suas respostas e que foi possível visualizar que há necessidade de encaixar um modelo genérico de contratações para múltiplos fornecedores, contrastando com a visibilidade de times perenes atuando neste contexto, sendo tais

times mantidos pelo fornecedor de software contratado, os quais são remunerados através da métrica de pontos de função. Além disso, há de se atentar para a quantidade e conseqüente custo dos recursos utilizados no processo de Concepção definido, de forma a mitigar o risco de desperdício, um dos principais direcionadores da filosofia enxuta.

No que diz respeito a indicadores de desempenho das Fábricas de Software identificou-se que são necessárias algumas iniciativas pontuais, e avaliações dos mesmos contra os objetivos da apuração ou uma revisão para um novo Cockpit de Indicadores.

Foi possível evidenciar também, que devido à recente implantação de métodos ágeis na organização, e falhas de comunicação entre as áreas, há diferenças de entendimento que fazem com que existam alguns empecilhos em relação aos objetivos mapeados para a utilização do método. Em algumas respostas ao questionário realizado, inclusive fica evidente a questão cultural, remetendo a aspectos tais como a relação política existente entre áreas correlacionadas, a falta de autonomia dos técnicos entre outros aspectos impactantes e que precisam ser melhorados para atender plenamente a necessidade da organização.

Em síntese, embora existam diversos pontos de melhoria nos processos analisados com vistas à agilidade e cultura enxuta, foi possível evidenciar também que algumas iniciativas que seriam indicadas já estão ocorrendo que atendem, em partes, ao propósito de prover uma melhor integração entre as equipes, além de tornar alguns procedimentos e fluxos dos processos abordados aderentes às boas práticas relacionadas.

## **5.1 Oportunidades de Melhoria**

Verificou-se que o modelo estudado neste trabalho, está parcialmente aderente às boas práticas de desenvolvimento de software e que apresenta uma proposta diferente, mas aderente às práticas utilizadas por outras organizações para resolver problemas similares. Desta forma, é possível sugerir algumas melhorias no processo de desenvolvimento de software para, primeiramente, assim como nos demais processos que tangem ao estudo realizado.

Salienta-se, que cada melhoria foi mapeada como uma pergunta de pesquisa, justamente para que se possa dar continuidade futura ao trabalho em questão.

**Quadro 23 – Propostas de Melhorias**

ID	Questão de Pesquisa	Pergunta de Pesquisa	Melhoria Proposta
1	Processo de Portfólio de Projetos	Como deveria ser o processo de portfólio da TI para melhor conviver com os modelos tradicional, híbrido e ágil?	Durante a pesquisa percebeu-se que existe um processo de gerenciamento de portfólio da TI que pode não cobrir bem os diferentes tipos de modelos de desenvolvimento da organização.
2	Processo de Gerenciamento de Projetos	Quais são as práticas que melhor se ajustam à necessidade de gerenciamento de projetos ágeis?	Durante a pesquisa, foi citado pelos entrevistados que o processo de gerenciamento de projetos deveria ter uma perspectiva ágil.
3	Processo de Desenvolvimento de Software	Quais artefatos do ciclo de desenvolvimento de software realmente agregam valor no processo?	Foi explicitado durante as entrevistas, que alguns artefatos do ciclo de desenvolvimento tradicional (híbrido também, por consequência) não agregam valor para os seus usuários. Exemplo: parecer de Segurança da Informação quando o cenário é conhecido e existem instruções divulgadas no portal de padrões da área em questão.
4		Quais roteiros de desenvolvimento (conjunto de artefatos conforme uma categoria) deveriam ser criados para atender a necessidade da organização?	Percebeu-se que dependendo da tecnologia, tipo de aplicação ou ainda tipo de negócio associado ao desenvolvimento, há um conjunto de atividades/entregáveis necessário por vezes diferente.
5	Processo de Homologação de Software	Qual a melhor abordagem de homologação para o modelo ágil de desenvolvimento de software?	Foi constatado que o modelo ágil possui um processo implícito de homologação, em substituição a testes. Há a percepção por alguns entrevistados, de que este modelo não é adequado e não é na realidade uma homologação.
6	Processo de Estimativas de Software	Como alterar os contratos atuais da organização nos modelos tradicional e híbrido para o modelo de pagamento contra a entrega com PF/PS/PFC?	Constatou-se a necessidade de adiantar o detalhamento do escopo em uma fase do ciclo de desenvolvimento em que isso não é adequado, para viabilizar o pagamento contra a estimativa, o que também não é adequado em relação aos interesses da organização (busca-se sempre o pagamento contra a entrega).
7	Processo de Contratação de	Como reduzir o tempo de contratação de	Alguns respondentes relataram que o tempo de contratação de um novo fornecedor ou de um fornecedor atual com um escopo diferente

	Fornecedores	fornecedores?	é demasiado, com muitos repiques.
8		Como definir uma política de homologação de fornecedores?	Citou-se que é uma iniciativa já em andamento, mas como é fundamental que este processo seja o mais maduro possível, sugere-se definir uma política de homologação com critérios objetivos para novos fornecedores.
9	Processo de Contratações de Software	Como estabelecer um modelo de contratação pontual utilizando instrumentos enxutos?	Com base nos instrumentos do SISP citados, propor um modelo enxuto de contratação.
10	Processo de Gestão de Fornecedores	Como reduzir o esforço operacional e procedimentos burocráticos para a gestão de fornecedores mais eficiente?	Foi citado que o processo em questão é bastante burocrático e pouco eficiente.
11		Qual seria o Cockpit de indicadores padrão ideal para avaliar fornecedores de software da organização?	Foi citado que nem sempre os indicadores parecem ter um propósito alinhado com os objetivos de negócio e TI.
12	Pessoas e Cultura	Como fomentar uma cultura organizacional na TI que privilegie a cocriação?	Foi citado que existe resistência para um trabalho efetivo entre áreas.
13	Processos e Boas Práticas	Como fomentar um fórum de melhoria contínua de processos com base em ideias dos usuários?	Foi sugerido criar uma base de ideias, para serem votadas pelos usuários. Seria uma proposta de fórum com base em uma ferramenta de apoio.
14	Arquitetura e Tecnologias	Como reduzir a dependência da formalização do parecer técnico da Arquitetura de TI no ciclo de desenvolvimento?	O parecer em questão foi citado como estando no caminho crítico dos projetos, e com significativo SLAs, porém os padrões da Arquitetura atualmente são aparentemente bem definidos. Sugeriu-se publicar e divulgar melhor tais padrões, solicitando o parecer em um fluxo exceção.
15	Ferramentas Internas	Quais ferramentas poderiam auxiliar os usuários da TI a melhorar sua eficiência?	Foi citado que as ferramentas internas não atuam com base no princípio da eficiência e que há muito trabalho operacional/ manual.

*Fonte: Elaborado pelo Autor.*

## **5.2 Limitações do Estudo**

Além de referências, documentos e informações de portais ou ainda observações realizadas, foram entrevistadas sete pessoas da área de TI da empresa FINA S/A; logo, a percepção se limita aos envolvidos e não considera pontos que por ventura teriam sido levantados pelos demais colaboradores da empresa. Desta forma, poderá haver diferentes percepções caso este estudo seja aplicado a este público adicional.

## **5.3 Desdobramentos Futuros**

A continuidade deste estudo se dará com a apresentação do resultado do mesmo aos envolvidos no estudo, assim como para os demais colaboradores da organização que possam querer utilizar este trabalho como fonte para análises próprias que beneficiem a evolução do desempenho da empresa.

Espera-se também que as oportunidades de melhorias mapeadas a partir dos resultados obtidos junto aos entrevistados possam ser utilizadas por outros colaboradores com o intuito de dar andamento a pesquisas estruturadas, fornecendo importantes insumos para a melhoria de processos na organização.

## REFERÊNCIAS

- ABRAN, Alain. *Software Metrics and Software Metrology*. IEEE, 2010.
- ALBRECHT, Allan J. “Medindo a Produtividade do Desenvolvimento de Aplicativos.” *FattoCS*. 1979. <http://www.fattocs.com.br/artigos/MedindoProdutividadedoDesenvolvimentodeAplicativos.pdf> (acesso em 13 de Dez de 2015).
- AMARAL, Daniel Capado et al. *Gerenciamento ágil de projetos: aplicação em produtos inovadores*. São Paulo: Saraiva, 2011.
- ARCHIBALD, R. D. *Managing High-Technology Programs and Projects, 3ª Edição*. New York: John Wiley & Sons, 2003.
- ASTELS, D. et. al. *eXtreme Programming – Guia Prático*. Rio de Janeiro: Campus, 2002.
- AUDY, Jorge L. N., e Rafael PRIKLANDNICKI. *Desenvolvimento Distribuído de Software*. Rio de Janeiro: Elsevier, 2007.
- AUGUSTINE, S. *Managing agile Project*. Virgínia: Prentice Hall PTR, 2005.
- BECK, Kent et al. *Manifesto for agile software development*. 2001. <http://www.agilemanifesto.org/> (acesso em 08 de Nov de 2015).
- BECK, Kent et. al. *Planning Extreme Programming. 1. ed*. Boston: Addison-Wesley, 2001.
- BECK, Kent. *Extreme Programming explained: embrace change. 1. ed*. MA: Addison-Wesley, 2000.
- CAROLI, Paulo. *Direto ao Ponto: criando produtos de forma enxuta*. São Paulo: Casa do Código, 2015.
- COHN, M. *Agile Estimating and Planning*. Massachusetts: Pearson Education, 2005.
- COSMIC. *Benchmark: measure of performance*. s.d. <http://cosmic-sizing.org/cosmic-fsm/benchmarking/> (acesso em 03 de Janeiro de 2016).
- . *The COSMIC Functional Size Measurement Method, Version 4.0.1*. The Common Software Measurement International Consortium, 2015.
- CRUZ, Fábio. *Scrum e PMBOK: unidos no gerenciamento de projetos*. Rio de Janeiro: Brasport, 2013.
- DEKKERS, Carol. “Counting Function Points for Agile / Iterative Software Development.” *IFPUG*. s.d. <http://www.ifpug.org/Articles/Dekkers-CountingAgileProjects.pdf> (acesso em 02 de Janeiro de 2016).

- FRANCO, E. F. *Um modelo de gerenciamento de projetos baseado nas metodologias ágeis de desenvolvimento de software e nos princípios da produção enxuta*. Dissertação de Mestrado, São Paulo: Escola Politécnica da Universidade de São Paulo, 2007.
- GARTNER. *Gartner Says Adopting a Pace-Layered Application Strategy Can Accelerate Innovation*. 2012. <http://www.gartner.com/newsroom/id/1923014/> (acesso em 10 de Nov de 2015).
- GIL, Antonio Carlos. *Como Elaborar Projetos de Pesquisa*. 5ª ed. São Paulo: Atlas, 2010.
- GIL, Antonio Carlos. *Como Elaborar Projetos de Pesquisa*. 4. ed. – São Paulo: Atlas, 2002
- GODOY, A. S. Introdução à pesquisa qualitativa e suas possibilidades. *Revista de Administração de Empresas*, v. 35, n. 2, p. 57-63, 1995.
- GOMES, Andre F. *Desenvolvimento de software com entregas freqüentes e foco no valor de negócio*. São Paulo: Casa do Código, 2013.
- HIGHSMITH, J. *Agile Project Management: creating innovative products*. Boston: Addison - Wesley, 2004.
- . *Agile Software Development Ecosystems*. Boston: MA: Addison, 2002.
- IFPUG. *Function Point Counting Practices Manual.(CPM)”. Release 4.3*. New Jersey: International Function Point users Group, 2010.
- . *Processo de Avaliação Não Funcional (SNAP)”. Release 2.2*. New Jersey: International Function Point Users Group, 2015.
- ISBSG. *Reifer Consultants LLC and ISBSG Joint Report: Software Productivity Benchmark*. Técnico, Victoria (AUS): ISBSG, 2013.
- JONES, C. *Estimating Software Costs: Bringing Realism to Estimating, 2ª Edition*. New York: McGraw-Hill, 2007.
- LEVINE, H.A. *Project Portfolio Management, Jossey-Bass*. San Francisco (USA), 2005.
- PMI. *A Guide to the Project Management Body of Knowledge – PMBOK® Guide 2000 Edition*. Pennsylvania-USA: Project Management Institute (PMI), 2000.
- POPPENDIECK, Mary et al. *Implementando o Desenvolvimento Lean de Software: do conceito ao dinheiro*. Porto Alegre: Bookman, 2011.
- PRESSMAN, Roger S. *Engenharia de Software*. 6. ed. Porto Alegre: McGraw-Hill, 2006.
- RIES, Eric. *A Startup Enxuta*. São Paulo: Lua de Papel, 2012.
- . *The Lean Startup*. USA: Crown Business, 2011.

- ROESCH, Sylvia Maria Azevedo. *Projetos de Estágio e de Pesquisa em Administração: guia para estágios, trabalhos de conclusão, dissertações e estudos de caso*. 2 ed. São Paulo: Atlas, 1999.
- SABBAGH, Rafael. *Scrum: gestão ágil para projetos de sucesso*. São Paulo: Casa do Código, 2013.
- SATO, Danilo. *DevOps – na prática: entrega de software confiável e automatizada*. São Paulo: Casa do Código, 2013.
- SCHWABER, Ken. *Guia do Scrum*. Scrum Alliance, 2009.
- SILVA, L.V., L. MACHADO, A. SACCOL, e D. AZEVEDO. *Metodologia de Pesquisa em Administração: uma abordagem prática - Coleção EAD*. São Leopoldo: Editora Unisinos, 2012.
- SISP. *Guia de Boas Práticas em Contratação de Soluções de Tecnologia da informação, V 2.0*. Brasília: Ministério do Planejamento, Secretaria de Logística e TI, 2014.
- . *Metodologia de Gerenciamento de Portfólio de Projetos do SISP*. Brasília: SLTI, 2013.
- . *Metodologia de Gerenciamento de Projetos do SISP, versão 1.0*. Brasília: MP-SISP, 2011.
- . *Roteiro de Métricas de Software do SISP: versão 2.1*. Brasília: Ministério do Planejamento, Orçamento e Gestão, 2015.
- SOFTEX. *MPS-BR - Melhoria de Processo do Software Brasileiro: Guia de Aquisição*. SOFTEX, 2013.
- TCU. *TC 010.663/2013-4: conhecimento acerca da utilização de métodos ágeis nas contratações...* Levantamento de Auditoria, Brasília: Tribunal de Contas da União, 2013.
- TERRIBILI FILHO, Armando. *Indicadores de Gerenciamento de Projetos: monitoração contínua*. São Paulo: M. Books do Brasil Ltda, 2010.
- VAZQUEZ, C. E., G. S. SIMÕES, e R. M. ALBERT. *“Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software. 9ª Edição*. São Paulo: Editora Érica, 2010.
- VERGARA, S. C. *Métodos de pesquisa em administração*. São Paulo: Atlas, 2010.
- VIANNA, Maurício et al. *Design thinking: inovação em negócios*. Rio de Janeiro: MJV Press, 2012.
- YIN, Robert K. *Estudo de Caso: Planejamento e Métodos*. Tradução Daniel Grassi. 2 ed. Porto Alegre Bookman, 2001

YIN, Robert K. Estudo de Caso: Planejamento e Métodos. Tradução Daniel Grassi. 3 ed. Porto Alegre Bookman, 2005

WOMACK, J., Jones, D. *A mentalidade enxuta nas empresas (Lean Thinking)*, 1ª Edição. São Paulo: Editora Campus, 2004.