

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

**Adoção de Métodos Ágeis e do Pensamento
Lean na Gestão de Contratos de Fornecedores
de Desenvolvimento de *Software* em
Organizações Públicas Brasileiras: Um Estudo
de Caso**

Autor: Aline Gonçalves dos Santos
Orientador: Msc. Hilmer Rodrigues Neri

Brasília, DF
2014



Aline Gonçalves dos Santos

**Adoção de Métodos Ágeis e do Pensamento Lean na
Gestão de Contratos de Fornecedores de
Desenvolvimento de *Software* em Organizações Públicas
Brasileiras: Um Estudo de Caso**

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Msc. Hilmer Rodrigues Neri

Brasília, DF

2014

Aline Gonçalves dos Santos

Adoção de Métodos Ágeis e do Pensamento Lean na Gestão de Contratos de Fornecedores de Desenvolvimento de *Software* em Organizações Públicas Brasileiras: Um Estudo de Caso/ Aline Gonçalves dos Santos. – Brasília, DF, 2014-143 p. : il. (algumas color.) ; 30 cm.

Orientador: Msc. Hilmer Rodrigues Neri

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2014.

1. contratos. 2. ágeis. I. Msc. Hilmer Rodrigues Neri. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Adoção de Métodos Ágeis e do Pensamento Lean na Gestão de Contratos de Fornecedores de Desenvolvimento de *Software* em Organizações Públicas Brasileiras: Um Estudo de Caso

CDU 02:141:005.6

Aline Gonçalves dos Santos

**Adoção de Métodos Ágeis e do Pensamento Lean na
Gestão de Contratos de Fornecedores de
Desenvolvimento de *Software* em Organizações Públicas
Brasileiras: Um Estudo de Caso**

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 25 de junho de 2014:

Msc. Hilmer Rodrigues Neri
Orientador

Dr. Eduardo Martins Guerra
Convidado 1

Dra. Simone Borges Simão
Convidado 2

Brasília, DF
2014

Agradecimentos

Gostaria de agradecer aos meus pais (Eliane e Wilson) por me apoiarem e me proporcionarem ótimas oportunidades de estudos que foram determinantes para alcançar os meus objetivos, agradeço por dedicarem todos os seus esforços para que eu pudesse ter uma boa qualidade de vida. Serei eternamente grata.

A toda a minha família, e principalmente a minha irmã, Amanda, e a minhas avós, Maria Ana e Margarida, que sempre me colocaram em suas orações.

Ao professor Hilmer Rodrigues Neri pelas oportunidades oferecidas, pela confiança em mim depositada, pelos ensinamentos, pelo apoio nos momentos conflituosos e pela orientação neste trabalho.

A todos meus amigos de graduação pela parceria em inúmeros trabalhos.

Aos meus amigos Marília, Izabela, Leandro e Vítor que estiveram ao meu lado durante toda essa jornada e por compreenderem minha ausência durante os últimos meses. Em especial a Marília por todo apoio, força e verdadeira amizade em todos os momentos.

E a Deus, que sempre me iluminou e esteve comigo nos momentos de dificuldade, sempre me guiou a tomar as melhores escolhas, me deu força para superar todos os obstáculos e desafios da minha vida e colocou estas pessoas maravilhosas em minha vida.

*É importante saber valorizar. Valorizar as pessoas,
o trabalho, o amor, as conquistas e a vida.
Valorizar os outros e a si mesmo.
Todo o esforço e dedicação serão recompensados.*

Resumo

No Brasil, o cenário das contratações públicas de fornecedores de desenvolvimento de *software* vem passando por mudanças. Algumas organizações públicas tem compactuado entendimento de que os instrumentos contratuais, hoje em vigor, não priorizam a entrega de *software*, tampouco sua qualidade interna e valor de negócio. Isso contribui para que projetos terminem sem sucesso, o que onera os cofres públicos. Atualmente, algumas organizações da Administração Pública Federal iniciam investimentos para adotar contratações de serviços de desenvolvimento de *software* utilizando métodos ágeis. Contudo, cada uma dessas organizações têm vivenciado diferentes experiências e vêm compartilhando dificuldades comuns no que diz respeito às limitações impostas pelo normativo de contratação de *software*. O objetivo deste trabalho foi realizar um estudo de caso para análise da influência do uso de métodos ágeis e do pensamento *lean* na gestão de contratos públicos de terceirização de software. A pesquisa foi descritiva, com apoio da técnica de estudo de caso e a estratégia utilizada foi um estudo de caso exploratório sobre a solução de gestão de contratos ágeis definida por um órgão público brasileiro a fim de analisar a influência da mesma sobre o resultado final do contrato no que diz respeito aos efeitos sobre a entrega de ordens de serviço, sobre a satisfação do cliente e sobre a qualidade interna do código fonte. Como trabalhos futuros, objetiva-se a validação e refinamentos do protocolo deste estudo de caso, e em seguida, a ampliação deste estudo para outras organizações públicas brasileiras.

Palavras-chaves: Contratações. *Software*. Organizações. *Lean*. Métodos Ágeis. Gestão. Ordens de Serviço. Satisfação do Cliente. Qualidade Interna.

Abstract

In Brazil, the situation of public contracting software development suppliers has changed over time. Some public organizations have agreed on the understanding that the contractual instruments in force today, do not prioritize the delivery of software, nor its internal quality and business value. This contributes to finish projects without success. Currently, some organizations of the Federal Public Administration begin investments to adopt contracting software development services using agile methods. However, each of these organizations have experienced different experiences and have shared common difficulties with regard to the limitations imposed by the normative of software contracting. The purpose of this study was to conduct a case study to analyze the influence of the use of agile methods and lean thinking in the management of public demands of software development. The research was descriptive, with technical support from the case study and the strategy used was an exploratory case study on the management solution for agile contracts defined by a Brazilian government organization in order to analyze the influence of the same on the result of contract with regard to the effects on the delivery of service orders, on customer satisfaction and the quality of the internal source. As future work, the objective is the validation and refinement of the protocol of this case study, and then the expansion of this study to other Brazilian public organizations.

Key-words: Contracting. Software. Organizations. Lean. Agile methods. Management. Service Orders. Customer Satisfaction. Internal quality.

Lista de ilustrações

Figura 1 – Seleção de Metodologia de Pesquisa	19
Figura 2 – Fases do Estudo de Caso	20
Figura 3 – Modelo de Contratações de Soluções de TI (MPOG, 2013)	27
Figura 4 – Gerenciamento de Contratações de Soluções de TI (MPOG, 2013)	30
Figura 5 – <i>Ranking</i> de Vendas da Indústria Automobilística (GUEDES, 2012)	34
Figura 6 – Pirâmide de Princípios Lean	35
Figura 7 – Quadro Kanban (JENSES, 2009)	42
Figura 8 – <i>Work in Progress</i> (KLIPP, 2011)	43
Figura 9 – Gráfico de Medições Kanban (KLIPP, 2011)	43
Figura 10 – Framework Scrum	48
Figura 11 – Comparação entre métodos segundo suas características prescritivas (KNIBERG; SKARIN, 2009)	49
Figura 12 – Quadro Scrum x Quadro Kanban (KNIBERG; SKARIN, 2009)	50
Figura 13 – Escopo do Estudo de Caso	59
Figura 14 – Estrutura do Estudo de Caso	60
Figura 15 – Quadro Kanban	77
Figura 16 – Transição para a raia Preparado	78
Figura 17 – Transição entre raias	78
Figura 18 – Limitação de WIP	79
Figura 19 – Parelização de Atividades	80
Figura 20 – Nível de Visibilidade do Processo do Projeto SICG	95
Figura 21 – Nível de Satisfação do Produto Entregue do Projeto SICG	95
Figura 22 – Qualidade da Métrica LOC nas <i>Sprints</i> do Projeto SICG	101
Figura 23 – Qualidade da Métrica ACCM nas <i>Sprints</i> do Projeto SICG	101
Figura 24 – Qualidade da Métrica AMLOC nas <i>Sprints</i> do Projeto SICG	102
Figura 25 – Qualidade da Métrica ACC nas <i>Sprints</i> do Projeto SICG	102
Figura 26 – Qualidade da Métrica ANPM nas <i>Sprints</i> do Projeto SICG	103
Figura 27 – Qualidade da Métrica CBO nas <i>Sprints</i> do Projeto SICG	103
Figura 28 – Qualidade da Métrica DIT nas <i>Sprints</i> do Projeto SICG	104
Figura 29 – Qualidade da Métrica LCOM4 nas <i>Sprints</i> do Projeto SICG	104
Figura 30 – Qualidade da Métrica NOC nas <i>Sprints</i> do Projeto SICG	105
Figura 31 – Qualidade da Métrica NOM nas <i>Sprints</i> do Projeto SICG	105
Figura 32 – Qualidade da Métrica NPA nas <i>Sprints</i> do Projeto SICG	106

Figura 33 –Qualidade da Métrica RFC nas <i>Sprints</i> do Projeto SICG	106
Figura 34 –Qualidade Geral do Código Fonte	107
Figura 35 –Nível de Percepção da Qualidade do Código Fonte	108
Figura 36 –Matriz de Gerenciamento de Tempo (BELL; ORZEN, 2011)	132
Figura 37 –Quadro Andon, com o mesmo objetivo do Kanban, na fábrica da Toyota (WILLIAMS, 2009)	139
Figura 38 –Processo MIDAS (IPHAN, 2013)	141
Figura 39 –Subprocesso Realizar Ateste Técnico (IPHAN, 2013)	142
Figura 40 –Subprocesso Sprint (IPHAN, 2013)	143

Lista de tabelas

Tabela 1 – Fases do MCTI	27
Tabela 2 – Valores do Manifesto Ágil x Análise segundo Acórdão no 2314/2013	56
Tabela 3 – Perfil da Equipe do IPHAN	68
Tabela 4 – Fatores motivacionais	69
Tabela 5 – Perfil da Equipe da EGL - Engenharia Ltda	70
Tabela 6 – Práticas Ágeis adotadas pela EGL - Engenharia Ltda	71
Tabela 7 – Visão geral da situação do IPHAN	73
Tabela 8 – Práticas Ágeis adotadas pelo IPHAN	76
Tabela 9 – Modelo de Remuneração do projeto SICG	80
Tabela 10 – Papéis e Responsabilidades	84
Tabela 11 – Gerenciamento na IN 04/2010 x Gerenciamento no IPHAN	85
Tabela 12 – Riscos Acórdão 2314/2013 x Solução IPHAN	89
Tabela 13 – Valores do Manifesto Ágil e Princípios da APF x Solução do IPHAN	90
Tabela 14 – Resultados Métrica de QE1. a QE9.	91
Tabela 15 – Porcentagem de Requisitos Atendidos por Ordem de Serviço	92
Tabela 16 – Custo por Ordem de Serviço	93
Tabela 17 – Nome dos Intervalos de Frequência e Qualitativos	96
Tabela 18 – Intervalos de Qualidade Java	98
Tabela 19 – Valores das Métricas do Projeto SICG da <i>Sprint 2</i> a <i>Sprint 8</i>	99
Tabela 20 – Valores das Métricas do Projeto SICG da <i>Sprint 9</i> a <i>Sprint 13</i>	99
Tabela 21 – Valores das Métricas do Projeto SICG da <i>Sprint 14</i> a <i>Sprint 20</i>	100
Tabela 22 – Valores das Métricas do Projeto SICG da <i>Sprint 21</i> a <i>Sprint 24</i>	100
Tabela 23 – Papéis MIDAS x Papéis IN 04	141

Lista de abreviaturas e siglas

APF	Administração Pública Federal
ATI	Analista de Tecnologia de Informação
BACEN	Banco Central
DEPAM	Departamento de Patrimônio Material e Fiscalização
EGTI	Estratégia Geral de Tecnologia da Informação
GM	General Motors
GCTI	Gerenciamento do Contrato
IPHAN	Instituto do Patrimônio Histórico e Artístico Nacional
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
JIT	Just in Time
MCTI	Modelo de Contratações de Soluções de TI
MGDS	Metodologia de Desenvolvimento de Sistemas
MIDAS	Metodologia IPHAN de Desenvolvimento Ágil de Software
MPOG	Ministério do Planejamento, Orçamento e Gestão
MP/SLTI	Ministério do Planejamento, Orçamento e Gestão/Secretaria de Logística e Tecnologia da Informação
OS	Ordem de Serviço
PCTI	Plano de Contratação
PDCA	Plan Do Check Act
PDS-AGIL	Processo de Desenvolvimento de Software Ágil
PDTI	Plano Diretor de Tecnologia da Informação
SEFTI	Secretaria de Fiscalização de Tecnologia da Informação
SFTI	Seleção do Fornecedor
SICG	Sistema Integrado de Conhecimento e Gestão

SISP	Sistema de Administração dos Recursos de Informação e Informática
STF	Supremo Tribunal Federal
TCU	Tribunal de Contas da União
TI	Tecnologia da Informação
TST	Tribunal Superior do Trabalho
XP	Extreme Programming
WIP	Work in Progress

Sumário

1	Introdução	16
1.1	Contexto	16
1.2	Justificativa	17
1.3	Problema	17
1.4	Objetivos	18
1.5	Metodologia de Pesquisa	19
1.6	Organização do Trabalho	20
2	Contratações de Fornecedores de Desenvolvimento de Software	22
2.1	Importância da Contratação de Fornecedores de Desenvolvimento de Software	22
2.2	Normas, Processos e Legislação Pertinentes à Contratação	23
2.2.1	Lei nº 8.666/93	23
2.2.2	Instrução Normativa nº 04	25
2.2.2.1	Disposições Gerais	26
2.2.2.2	Processo de Contratação	27
2.3	Considerações Finais do Capítulo	30
3	Pensamento Lean	32
3.1	Lean na Manufatura	32
3.1.1	Histórico	32
3.1.2	Princípios	34
3.2	Lean no Desenvolvimento de <i>Software</i>	35
3.2.1	Abordagem	35
3.2.2	Princípios	35
3.2.2.1	Eliminar o Desperdício	35
3.2.2.2	Integrar Qualidade	36
3.2.2.3	Criar Conhecimento	37
3.2.2.4	Adiar Comprometimentos	37
3.2.2.5	Entregar Rápido	37
3.2.2.6	Respeitar as Pessoas	38
3.2.2.7	Otimizar o Todo	38
3.2.3	Práticas	38
3.2.3.1	Gerenciamento de Código Fonte e <i>Scripts</i> de <i>Builds</i>	38
3.2.3.2	Teste Automatizado	39
3.2.3.3	Integração Contínua	39
3.2.3.4	Menos Código	40

3.2.3.5	Iterações Curtas	40
3.2.3.6	Participação do Cliente	41
3.2.3.7	Kanban	41
3.3	Considerações Finais do Capítulo	44
4	Adoção de Métodos Ágeis na Gestão de Contratos de Desenvolvimento de <i>Software</i> em Organizações Públicas Brasileiras	45
4.1	Metodologias Ágeis de Desenvolvimento de <i>Software</i>	45
4.1.1	Scrum	47
4.2	Kanban, Scrum e o Pensamento Lean	48
4.2.1	Cadência	50
4.2.2	Limitação do WIP	50
4.2.3	Equipe de Trabalho	51
4.3	Contratação Ágil nas Organizações Públicas Brasileiras	51
4.3.1	Iniciativas de Adoção de Métodos Ágeis no Governo	51
4.3.2	Desafios	57
4.4	Considerações Finais do Capítulo	57
5	Projeto do Estudo de Caso	59
5.1	Definição	59
5.2	Background	63
5.3	Design	63
5.4	Seleção	64
5.5	Fonte e Método Coleta de Dados	64
5.6	Validade	65
5.7	Trabalho de Campo	66
5.8	Análise	66
5.9	Considerações Finais do Capítulo	66
6	O Estudo de Caso	68
6.1	A Organização	68
6.2	A Empresa Contratada	70
6.3	Caracterização do Projeto Contrato	71
6.3.1	Visão Geral	71
6.3.2	Objetivos da Contratação	73
6.3.3	Metodologia de Trabalho	74
6.4	Caracterização da Solução	75
6.4.1	Análise da Adequação da Solução ao Pensamento Lean e ao Scrum	81
6.4.2	Análise da Adequação da Solução aos Valores Ágeis e a Legislação	83
6.5	Análise dos Dados	91

6.5.1	Efeitos sobre a entrega de ordens de serviço	91
6.5.2	Efeitos sobre a satisfação do cliente	94
6.5.3	Efeitos sobre a qualidade do código	96
7	Conclusão	110
	Referências	113
	Apêndices	117
	APÊNDICE A Questionário Gestão do Contrato	119
	APÊNDICE B Métricas de Código Fonte	126
	Anexos	129
	ANEXO A Princípios e Práticas do Lean na Manufatura	130
A.1	Princípios	130
A.1.1	Constância de Propósitos	130
A.1.2	Respeito às Pessoas	130
A.1.3	Melhoria Contínua e Perfeição	131
A.1.4	Comportamento Proativo	131
A.1.5	Voz do Cliente	132
A.1.6	Qualidade na Raíz	133
A.1.7	Pensamento Sistêmico	133
A.1.8	Fluxo Contínuo, Produção Puxada e <i>Just in Time</i>	134
A.1.9	Cultura	135
A.2	Conceitos de Valor e Desperdício	135
A.2.1	Valor	136
A.2.2	Cadeia de Valor	136
A.2.3	Os Três Ms	136
A.3	Práticas e Ferramentas	136
A.3.1	A3 Thinking	137
A.3.2	Mapeamento da Cadeia de Valor	137
A.3.3	Kaizen	137
A.3.4	Metodologia 5S	138
A.3.5	Kanban	139
	ANEXO B Metodologia IPHAN de Gestão de Demandas de Desenvolvimento Ágil de Software	140

1 Introdução

1.1 Contexto

Nos últimos anos, tem surgido iniciativas isoladas de algumas entidades da Administração Pública, em adotar métodos ágeis em suas equipes de desenvolvimento, com destaque para o Scrum e o *Extreme Programming-XP* (TCU, 2013) (MELO et al., 2012). Ainda mais recente, houve a iniciativa de mesclar o uso de métodos ágeis com a filosofia de gestão da produção, conhecida como *Lean*, com o foco em gerenciar o contrato dos fornecedores de desenvolvimento de *software*.

O *lean* no desenvolvimento de *software* é uma abordagem que busca aplicar os princípios do pensamento *lean* no desenvolvimento do *software*. Dentre os princípios do pensamento *lean* se destacam: a eliminação de desperdícios; o respeito às pessoas envolvidas no processo; a qualidade; a simplicidade; a otimização do todo e entregas rápidas (POPPENDIECK; POPPENDIECK, 2011).

É importante ressaltar que apesar de sugerir diversas ferramentas, como o Kanban, também presente em métodos ágeis, o *Lean* está mais relacionado à forma de pensar, exige muito mais uma mudança cultural de cada organização do que a aplicação e utilização de ferramentas.

O pensamento *lean* está relacionado com métodos ágeis, não só pela semelhança dos seus princípios e práticas, mas também porque ambos valorizam as pessoas em detrimento de ferramentas e buscam agregar valor de negócio ao sistema que está sendo desenvolvido. Do ponto de vista teórico os métodos ágeis e o pensamento *lean*, se baseiam em diferentes teorias, como: teoria geral dos sistemas (VIEIRA et al., 2005), teoria da complexidade (LIMA; SILVA, 2003), teoria das restrições (KATAYAMA, 2010) e teoria de gestão da escola de relações humanas (SALDANHA, 2005).

Essas teorias representam um contraponto a metodologias mais prescritivas e preditivas, também conhecidas como tradicionais, que possuem seu amparo teórico principalmente na visão da teoria da administração científica, que tem como um dos percursores Frederick W. Taylor e pressupõe que todo saber e tomadas de decisões são funções específicas da gerência e que os trabalhadores devem executar suas tarefas por meio de métodos e procedimentos pré-definidos (SALDANHA, 2005).

O *scrum* é uma metodologia ágil desenvolvida para a gestão do processo de desenvolvimento de *software*. É uma abordagem que aplica idéias de controle do fluxo de trabalho, oriundo da indústria de manufatura, ao desenvolvimento de *software*, resultando assim, numa abordagem que reintroduz a ideia de flexibilidade, adaptabilidade e

produtividade (PEREIRA; SOUSA, 2011). O scrum surgiu a partir do “Manifesto Ágil”, publicado em 2001, e como método ágil tem como valores: indivíduos e interações mais do que processos e ferramentas; *software* funcionando para o cliente em vez de documentação; colaboração com o cliente mais do que negociação de contratos e resposta rápida às mudanças mais do que seguir planos (CUNNINGHAM, 2001).

A ideia principal é que no desenvolvimento de *software* existem diversas variáveis, quer sejam de natureza ambiental ou técnica, que provavelmente mudarão ao longo da execução do processo. Essa característica torna o processo de desenvolvimento pouco previsível e complexo, requerendo flexibilidade e personalização para ser capaz de responder às mudanças. Outro sim, o contato do cliente deve ser ativo, haja visto que, ele é determinante para a criação do *software*, e por consequência, afeta o próprio processo do serviço, no caso as atividades do desenvolvimento.

A terceirização de serviços em organizações públicas no contexto de contratação de empresas de desenvolvimento de *software* é crescente. A gestão do processo de desenvolvimento de *software* é um grande desafio para essas organizações, pois a maioria delas não são responsáveis diretamente pelo desenvolvimento do *software* e ao mesmo tempo elas precisam, como contratantes, gerenciar o andamento do processo de desenvolvimento do *software* de suas contratadas.

A legislação brasileira de contratação de serviço de desenvolvimento vigente, se apoia na Lei 8.666/93, IN 04/2010 SLTI/MPOG (para Poder Executivo) e acórdãos do TCU, que serão detalhados no Capítulo 2 deste trabalho.

1.2 Justificativa

A utilização de metodologias ágeis e semelhantes em contratações de serviços de tecnologia da informação está ganhando espaço nas organizações públicas brasileiras e gerando questões importantes de estudo para a academia. Assim, recentemente, iniciativas de inserção de tais metodologias na gestão de contratos de fornecedores de desenvolvimento de *software* em organizações públicas estão sendo feitas e, portanto, torna-se necessário estudos que evidenciem os resultados advindos do uso de métodos ágeis neste contexto. Uma das principais contribuições deste trabalho será evidenciar para os gestores de contratos e para os demais envolvidos no processo de gestão e desenvolvimento de *software* terceirizado, os resultados do uso de métodos ágeis na gestão de contratos.

1.3 Problema

Durante muitos anos as organizações privadas e públicas utilizaram metodologias tradicionais no desenvolvimento de *software* ou na gestão de contratos da terceirização

do desenvolvimento das soluções de TI. Nos últimos anos, com o surgimento das metodologias ágeis, esse cenário começou a mudar no contexto das instituições privadas a fim de aumentar sua produtividade, eliminar desperdícios e aumentar o valor de negócio produzido para o cliente. Nas instituições públicas essa mudança teve início recentemente. Tais organizações perceberam que uma grande quantidade de documentos estava sendo produzida e pouco do *software* requisitado estava sendo entregue no final do contrato (TCU, 2013). Assim, o problema é que alguns contratos de desenvolvimento de *software* das organizações públicas brasileiras não resultaram na entrega do software requisitado ao final do contrato. A partir desse problema, foi formulada a questão de pesquisa geral deste trabalho que é:

Como o uso de métodos ágeis e do pensamento lean na gestão de contratos de fornecedores de desenvolvimento de software influenciaram no resultado final do contrato do ponto de vista do gestor de contrato e do fiscal técnico do contrato, que juntos gerenciam o contrato?

1.4 Objetivos

O objetivo geral deste trabalho é realizar um estudo de caso onde será analisado a influência do uso de métodos ágeis e do pensamento *lean* em um contrato de uma organização pública brasileira a partir dos dados coletados da documentação dos processos da organização, de questionário e de entrevistas com os principais envolvidos no contrato, além do código-fonte do objeto do contrato. Dentre os objetivos específicos deste trabalho estão:

- Construir uma fundamentação teórica com a caracterização do processo de contratações de soluções de TI nas organizações públicas brasileiras com base na Lei 8.666/93, na IN 04/2010 SLTI/MPOG e no Acórdão 2314/2013, a caracterização do pensamento *lean* no desenvolvimento de *software* e de metodologias ágeis e a caracterização e relacionamento do *Kanban*, com o *Lean* e o *Scrum*;
- Definir, projetar e caracterizar o estudo de caso;
- Coletar dados do contrato executado e gerenciado com métodos ágeis e pensamento *lean* da organização selecionada;
- Realizar análise dos dados coletados;
- Relatar os resultados obtidos.

1.5 Metodologia de Pesquisa

Nessa seção apresenta-se a metodologia de pesquisa adotada neste trabalho. Para isso, foram definidos: a natureza da pesquisa; o tipo de metodologia de pesquisa; o tipo de abordagem de pesquisa; os métodos de procedimentos de pesquisa e os tipos de técnicas de coletas de dados.

O procedimento de pesquisa escolhido foi o estudo de caso. As técnicas de coleta de dados selecionadas foram documentos, questionários e entrevistas informais. A seleção metodológica é apresentada na Fig. 1.



Figura 1 – Seleção de Metodologia de Pesquisa

Conforme apresentado por Yin (2010), um estudo de caso pode ser composto por 6 etapas: Plano, Projeto, Preparação, Coleta, Análise e Compartilhamento. Nesta pesquisa, o esquema adotado compreende as fases: Planejamento; Coleta; Análise, e Compartilhamento. Portanto, as etapas Plano, Projeto e Preparação definidas por Yin foram agrupadas na fase Planejamento nesse estudo de caso, conforme ilustrado na Fig. 2.

O Planejamento consiste na determinação da questão de pesquisa, a escolha da metodologia de pesquisa, a definição das fases da pesquisa, a definição dos procedimentos de pesquisa e das técnicas de coleta de dados, a construção do referencial teórico e a proposta do trabalho final.

Na Coleta são executados os procedimentos de pesquisa e as técnicas de coletas de dados a seguir:

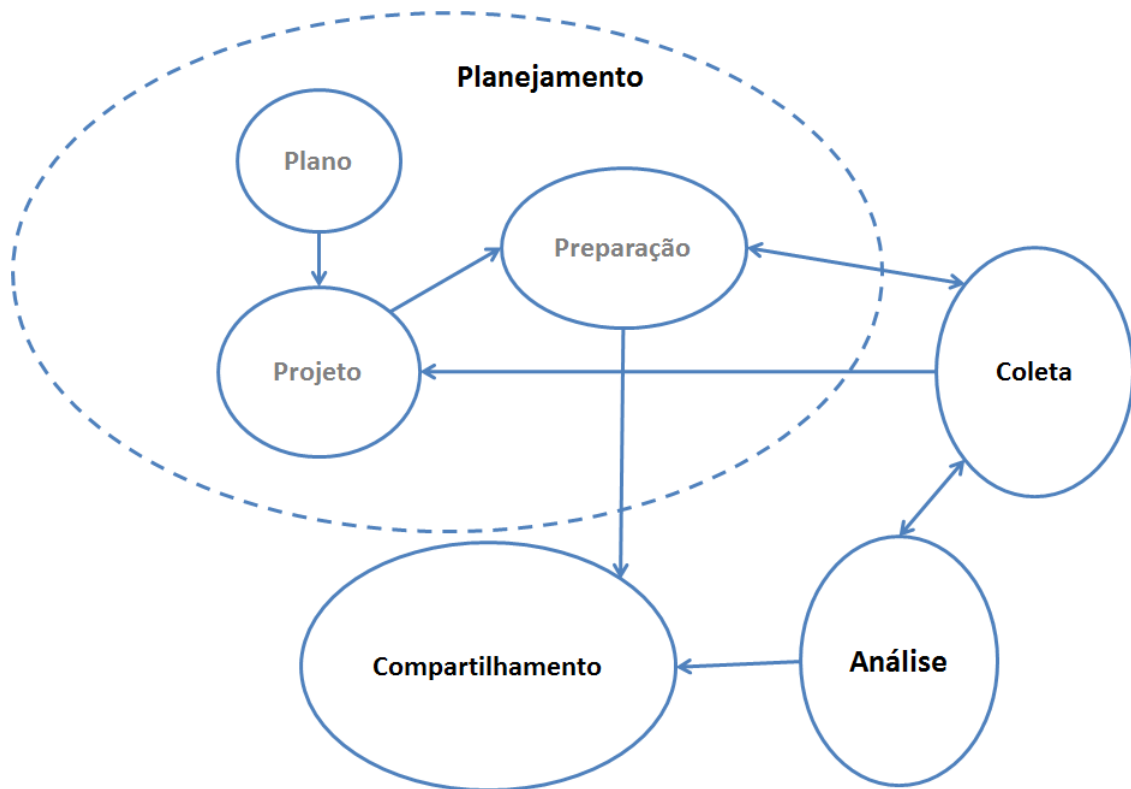


Figura 2 – Fases do Estudo de Caso

- Revisão Bibliográfica do aporte teórico sobre *lean* e métodos ágeis a partir de livros, dissertações e trabalhos relacionados à área de pesquisa;
- Estudo de Caso: utilizar um estudo de caso real de uma organização pública brasileira;
- Entrevistas: dados serão coletados por meio de entrevistas informais, além de questionário, para incremento do estudo de caso;
- Documentos: coleta de dados dos documentos dos processos fornecidos pelo órgão público do estudo de caso será realizada para coleta de dados para análise.

A Análise diz respeito a fase em que os dados coletados serão analisados e interpretados. A análise compreende tanto a análise quantitativa quanto a análise qualitativa.

O Compartilhamento diz respeito a redação dos resultados de forma adequado para o leitor alvo.

1.6 Organização do Trabalho

Este trabalho está organizado em sete capítulos. Neste Capítulo 1 encontra-se a introdução do trabalho que consiste em: contexto do trabalho, justificativa, problema, objetivos e metodologia de pesquisa adotada.

No Capítulo 2 - Contratações de Fornecedores de Desenvolvimento de *Software* - apresenta-se as principais informações referentes à Contratação de Fornecedores de Desenvolvimento de *Software*. Para isso, o capítulo é iniciado com uma visão geral sobre a importância das contratações e suas principais características. Posteriormente, é apresentado um resumo das legislações relacionadas à Contratação de Soluções de Tecnologia da Informação.

No Capítulo 3 - Pensamento *Lean* - são apresentados os conceitos referentes ao *lean* na Manufatura e ao *lean* no desenvolvimento de *software*. Com este fim, é apresentado um breve histórico sobre o surgimento do *lean* na manufatura e os seus principais princípios e práticas e, posteriormente, é apresentado como o *lean* manufatura foi adaptado para o desenvolvimento de *software* e para isso é abordado os princípios e práticas do *lean* no desenvolvimento de *software*.

No Capítulo 4 - Adoção de Métodos Ágeis na Gestão de Demandas de Desenvolvimento de *Software* em Organizações Públicas Brasileiras - são apresentados os princípios e valores das Metodologias Ágeis e caracterizado o Framework Scrum, posteriormente, é feito o relacionamento entre o Scrum, o *Leane* o Kanban. Ainda são apresentados as iniciativas de adoção de métodos ágeis no Governo e os desafios inerentes a adoção de métodos ágeis em organizações públicas brasileiras.

No Capítulo 5 - Projeto do Estudo de Caso - é apresentado o projeto do estudo de caso resultante da fase de planejamento. Este capítulo contém a definição do estudo de caso, o background, o design, o critério de seleção, a fonte e método de coleta de dados, as considerações de validade e as definições de como o trabalho de campo e análise foram feitas.

No Capítulo 6 - O Estudo de Caso - é apresentado o estudo de caso deste trabalho. Neste capítulo é onde os resultados dos dados coletados e análise qualitativa e quantitativa é realizada.

2 Contratações de Fornecedores de Desenvolvimento de Software

O descumprimento da legislação de licitações e contratos gera riscos para a contratação de tecnologia da informação e, portanto, devem ser conhecidos e usados como base de qualquer processo de contratação de fornecedores de desenvolvimento de *software*. Assim, neste capítulo será apresentada uma visão sobre a importância da contratação de serviços de TI e a caracterização de contratação de tecnologia da informação pelas organizações públicas brasileiras, segundo a legislação pertinente, apresentando-se, de forma resumida, os conceitos de contratação de serviços de TI presentes em normas, modelos, guias e processos de contratação de soluções de TI.

2.1 Importância da Contratação de Fornecedores de Desenvolvimento de Software

Um dos atos administrativos principais, mais complexos e mais frequentemente utilizados é a contratação. Contratar é fazer contrato, que é um acordo ou convenção entre duas ou mais pessoas para a execução de alguma coisa, sob determinadas condições. O contrato é, portanto, o documento em que se registra esse acordo ou convenção (MPOG, 2011).

Há vários anos, os órgãos da Administração Pública Federal (APF) têm adotado a prática da execução indireta de muitos serviços que dão suporte às suas áreas-fim, conhecida comumente como “terceirização de serviços”. O Decreto-Lei 200/1967 traz, no Art.10 § 7º, a diretriz para que a Administração Pública Federal se desobrigue da realização de tarefas executivas (execução de tarefas operacionais), recorrendo, sempre que possível, à execução indireta, desde que a iniciativa privada esteja suficientemente desenvolvida na área, bem como não haja comprometimento da segurança nacional (Art.10 §8º). De acordo com o Decreto-Lei 200/1967, Art.10, §7º, as razões para se partir para execução indireta são: possibilitar que a APF execute melhor as tarefas de planejamento, coordenação, supervisão e controle, tarefas que hoje podem ser traduzidas como gestão e governança; impedir o crescimento desmesurado da máquina administrativa, para que o Estado não alcance dimensão indevida em função da incorporação de tarefas de caráter operacional (TCU, 2012).

Um levantamento do Tribunal de Contas da União (TCU) mostrou que o orçamento de gastos em TI da Administração Pública Federal de 2010 era de pelo menos 12,5

bilhões de reais, sendo grande parte desse valor destinado a contratações de serviços relacionados a *software*. No mesmo ano, também era previsto um orçamento de cerca de 1,8 trilhão de reais da União, sendo que a maior parte seria gasto em TI. Com isso, os serviços de TI requerem bastante atenção e tem suma importância para administração pública brasileira, tanto no que diz respeito a contratações ou desenvolvimento de produtos de TI.

A definição e institucionalização de processos de contratação de serviços de TI, especialmente aqueles relacionados a *software*, envolvem ações complexas, principalmente no que diz respeito à identificação dos requisitos necessários, a garantia da qualidade dos resultados esperados, aos critérios de aceitação, a gestão de mudanças, as transferências de conhecimentos, a legislação pertinente, entre outros. Envolvem também questões de relacionamento entre clientes e fornecedores, o que implica em competências administrativas e jurídicas. Essas complexidades apresentam riscos para partes envolvidas e, como consequência, é comum a ocorrência de conflitos (CRUZ; ANDRADE; FIGUEIREDO, 2011).

Assim, contratações envolvem riscos e incertezas oriundos de diversos fatores inerentes do objeto do contrato. Com isso, é preciso observar as características do objeto e do contexto em que ele será inserido, com especial atenção à conformidade legal em contratações de serviços de TI em organizações públicas, para que seja realizado o devido controle.

2.2 Normas, Processos e Legislação Pertinentes à Contratação

2.2.1 Lei nº 8.666/93

A Lei nº 8.666 (BRASIL, 1993) estabelece normas gerais sobre licitações e contratos administrativos. A licitação é tida como antecedente ao contrato administrativo e pode ser considerado o procedimento no qual a Administração Pública escolhe a proposta mais vantajosa para contrato do seu interesse e, ao mesmo tempo, dá igual oportunidade aos que desejam fazer parte do contrato.

Uma licitação supõe concorrência entre ofertantes, portanto, os objetos propensos à licitação são aqueles que podem ser fornecidos por mais uma pessoa ou entidade. Os princípios da licitação de acordo com o Art. 3º são:

- Igualdade: é impedida a existência de cláusulas no edital que favorecem um em detrimento de outros, mas não é considerado atentado ao princípio da igualdade estabelecer requisitos mínimos de participação;
- Legalidade: a licitação é um procedimento completamente vinculado à lei;

- Impessoalidade: a Administração deve considerar critérios claros e objetivos para a escolha do licitante, todos devem ser tratados igualmente em termos de direito e obrigações;
- Moralidade e Probidade Administrativa: é passível de punição qualquer comportamento que vá contra a moral, bons costumes, as regras de boa administração e aos princípios de justiça e equidade;
- Publicidade: diz respeito tanto à divulgação da licitação para todos os interessados como também a todos os atos praticados pela Administração durante as fases, para que sejam abertos aos interessados;
- Vinculação ao Instrumento Convocatório: o edital deve estar vinculado à licitação e sua alteração só é permitida quando for falho ou inadequado ao interesse público;
- Julgamento Objetivo: diz respeito a utilizar o critério indicado no ato convocatório para julgamento das propostas, seja ela técnica ou de preço;
- Fiscalização da Licitação: qualquer cidadão poder controlar a licitação contra irregularidades na aplicação da lei;
- Competitividade: a lei proíbe a existência de cláusulas que vão contra o caráter competitivo da licitação, salvo os casos de só haver um interessado ou um concorrente após a fase de classificação;
- Padronização: sempre que possível, deve ser adotada a padronização.

A licitação pode ser classificada quanto a modalidade, que está relacionada ao valor estimado do contrato, ou ao tipo, que está relacionado ao julgamento. A Lei nº 8.666 prevê cinco modalidades de licitação no Art. 22: concorrência, tomada de preços, convite, concurso e leilão. As modalidades sem finalidade específica entram no grupo formado pela concorrência, pela tomada de preços e pelo convite. O grupo das modalidades com finalidades específicas é formado pelo concurso e pelo leilão. Os quatro tipos de licitação presentes no Art.45 são:

- Menor preço: aquela em que o fator de decisão é apenas o menor preço e as demais características, como qualidade e produtividade, não são consideradas;
- Melhor técnica: não considera só o preço, mas as melhores tecnologias, aquelas mais modernas ou que satisfaçam da melhor forma as necessidades da Administração licitante e que estejam dentro dos recursos financeiros disponíveis;
- Técnica e preço: aquela em que é feita uma ponderação para técnica e preço a fim de considerar ambos;

- Maior lance ou oferta: é escolhida a proposta que faz a maior oferta, este tipo é especialmente usado em vendas de bens ou permissões de uso de bens ou serviços públicos.

Uma licitação é constituída por fases, sendo elas: interna (destinada a firmar a intenção da entidade licitante e obter informações necessárias para a consolidação da licitação); e a externa (destinada a selecionar a melhor proposta). A fase interna é composta por determinar o objeto de licitação, as condições, a estimativa de despesas e a decisão pela modalidade mais adequada, além da verificação da existência de recursos e a obtenção da autorização de abertura do instrumento convocatório.

Já a fase externa é dividida de forma geral em: abertura, habilitação, classificação e julgamento. A abertura é o momento em que o instrumento convocatório, edital ou carta-convite, é tornado conhecido publicamente; a habilitação é onde a comissão de licitação confirma os licitantes aptos a participarem da licitação de acordo com o edital; a classificação é o ato no qual a comissão de licitação reúne as propostas apresentadas formalmente e, de acordo com o edital ou carta-convite, que são aptas ou não para classificação; o julgamento costuma ocorrer logo após a classificação das propostas e é considerado apenas o que foi permitido no instrumento convocatório, feito de acordo com um dos quatro tipos de licitação, previsto no edital.

Após o julgamento, é realizado o processo de homologação e adjudicação. Na homologação, uma autoridade competente, indicada por lei, promove o controle de todo processo licitatório no que diz respeito à legalidade, e na adjudicação, é atribuído ao vencedor o objeto da licitação pela mesma autoridade competente. Outros detalhes da Lei nº 8.666 não foram considerados pertinentes para o escopo deste trabalho.

2.2.2 Instrução Normativa nº 04

A Instrução Normativa nº 04 (BRASIL, 2010), disciplina sobre o processo de Contratação de Soluções de Tecnologia da Informação pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática do Poder Executivo Federal, é a consolidação de um conjunto de boas práticas para Contratação de Solução de TI pela Administração Pública Federal. Este conjunto de boas práticas é chamado de Modelo de Contratações de Soluções de TI (MCTI). Esta Instrução Normativa está dividida em três capítulos: o primeiro diz respeito às Disposições Gerais; o segundo, fala do Processo de Contratação; e o terceiro nos apresenta as Disposições Finais - este não será detalhado, pois não faz parte do escopo deste trabalho.

2.2.2.1 Disposições Gerais

De forma geral, no capítulo das Disposições Gerais é mencionado os atores, os artefatos e o que é vedado no processo de contratações. Estes, de acordo com Art. 2º, são:

- Área requisitante da solução: entidade ou órgão que demanda a contratação de uma Solução de Tecnologia da Informação;
- Área de tecnologia da informação: unidade setorial do SISP responsável por gerir a Tecnologia da Informação do órgão ou entidade;
- Equipe de planejamento da contratação: constituída pelo integrante técnico (servidor da área de TI), integrante administrativo (servidor da área administrativa) e integrante requisitante (servidor da área requisitante);
- Gestor do contrato: servidor com atribuições gerenciais, técnicas e operacionais responsável pela gestão do contrato;
- Fiscal técnico do contrato: servidor da área de TI;
- Fiscal administrativo do contrato: servidor da área administrativa;
- Fiscal requisitante do contrato: servidor da área requisitante;
- Contratada: entidade provedora da Solução de Tecnologia da Informação;
- Preposto: funcionário representante da contratada, responsável por acompanhar a execução do contrato e atuar como interlocutor principal junto à contratante.

Os artefatos que podem ser produzidos e fornecidos para a contratada ou recebidos pelo contratante são: a solução de tecnologia da informação; os requisitos; o documento de oficialização da demanda; a análise de viabilidade da contratação; o plano de sustentação; a estratégia da contratação; a análise de riscos; o plano de inserção; a ordem de serviço ou de fornecimento de bens; o termo de recebimento provisório; o termo de recebimento definitivo; os critérios de aceitação; a gestão e o plano diretor de tecnologia da informação.

Nas Disposições Gerais, é ressaltado, no Art.5º, que não poderão ser objeto de contratação mais de uma solução de TI em um único contrato e a gestão de processos de TI. No Art. 6º é dito que a contratada que provê a solução de TI não poderá ser a mesma que realiza medições, avaliação ou fiscalização e é vedado, segundo o Art. 7º, prever a remuneração dos funcionários da contratada já no edital, reembolsar despesas que são de exclusiva responsabilidade da contratada, exigir qualquer tipo de capacitação ou certificação no edital, demandar aos funcionários da contratada tarefas fora do escopo do objetivo da contratação e indicar pessoas para compor o quadro funcional da contratada.

Está presente, ainda nas Disposições Gerais, a Estratégia Geral de Tecnologia da Informação (EGTI), que contém orientações gerais para as áreas de TI e dos órgãos e entidades da Administração Pública, Federação e a formulação de um Plano Diretor de Tecnologia da Informação (PDTI) por parte de cada entidade integrante do Sistema de Administração dos Recursos de Informação e Informática (SISP) do Poder Executivo Federal. De forma geral, neste documento, são apresentados a avaliação e o diagnóstico dos recursos de TI, as necessidades da entidade e o planejamento de investimentos e recursos.

2.2.2.2 Processo de Contratação

O Processo de Contratação é, de fato, o MCTI (Fig. 3), que divide as contratações de soluções de TI em três fases: Planejamento da Contratação (PCTI); Seleção do Fornecedor (SFTI) e Gerenciamento do Contrato (GCTI).

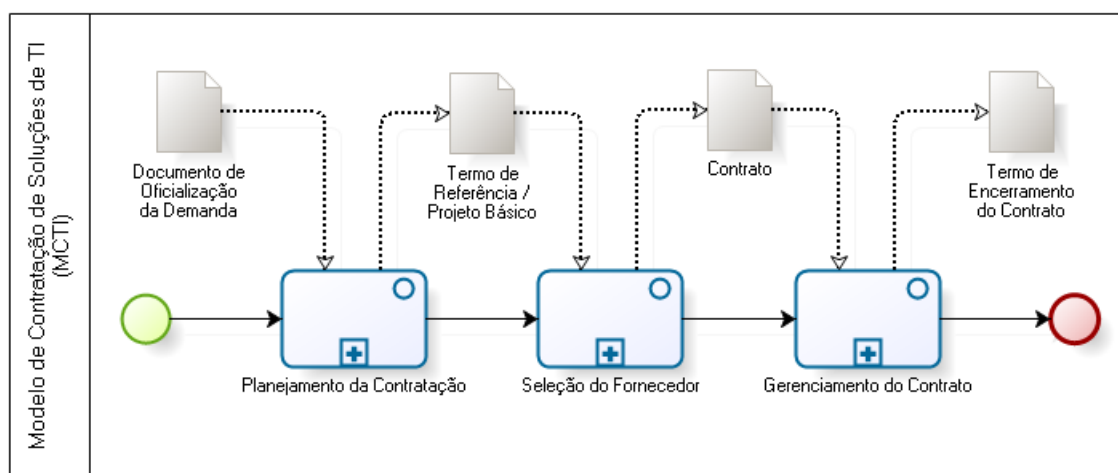


Figura 3 – Modelo de Contratações de Soluções de TI (MPOG, 2013)

Cada fase é constituída de processos/etapas, atividades, artefatos e atores, conforme mostrado na Tab. (1).

Fases	Etapas	Atividades	Artefatos	Atores
PCTI	5	41	8	7
SFTI	3	7	1	4
GCTI	5	19	4	5

Tabela 1 – Fases do MCTI

Planejamento da Contratação

No Planejamento da Contratação é necessário ter, no mínimo, o documento mostrando quais são as necessidades corporativas da instituição e seus objetivos estratégicos, motivação, resultados esperados, fonte de recursos e a indicação do integrante requisitante que fará parte da equipe de planejamento. Após o recebimento desse documento,

a área de TI indicará o integrante técnico que também fará parte da equipe de planejamento e, então, finalmente, o documento será encaminhado para a área administrativa, que indicará o integrante administrativo que fará parte da equipe de planejamento e dará prosseguimento para a contratação. Assim, a equipe estará completa para acompanhar e apoiar todas as atividades presentes nas fases de planejamento de contratação e seleção do fornecedor.

A fase de Planejamento da Contratação é obrigatória. Independentemente do tipo de contratação, deve ser elaborada em harmonia com o PDTI e contém cinco etapas:

- Análise de viabilidade;
- Plano de sustentação;
- Estratégia da contratação;
- Análise de riscos;
- Termo de referência ou projeto básico.

A análise de viabilidade da contratação compreende a definição e a especificação de requisitos, levando em conta que compete ao integrante técnico especificar os requisitos tecnológicos, e ao integrante requisitante os demais requisitos, a identificação de possíveis soluções, análise e comparação de custos totais dessas soluções, a escolha da solução de TI e a justificativa da solução escolhida e avaliação das necessidades de adequação para viabilização da execução contratual.

O plano de sustentação deve conter quais os recursos materiais e humanos que serão necessários, atividades de transição contratual e encerramento do contrato, assim como a continuidade do fornecimento da solução de TI em caso de interrupção contratual e a estratégia de independência do contratante com relação à contratada.

A estratégia da contratação será elaborada a partir das duas etapas anteriores e conterá a indicação da solução de TI a ser contratada. A definição das responsabilidades da contratada, além das estabelecidas no contrato, é: a indicação dos termos contratuais, observando os elementos contidos na Lei nº 8.666, de 1993, a elaboração do orçamento detalhado, da estimativa do impacto econômico-financeiro no orçamento do órgão requisitante, a elaboração dos termos de compromisso e sigilo e a definição dos critérios técnicos de julgamento das propostas para a fase de seleção de fornecedor.

A análise de riscos deverá conter identificação dos principais riscos que podem comprometer o sucesso dos processos de contratação e de gestão contratual ou que possam fazer com que a solução de TI não atenda às necessidades esperadas, a mensuração das probabilidades de ocorrência e dos dados potenciais relacionados a cada risco, a definição

das ações de contingência em caso de ocorrência dos riscos e a definição dos responsáveis pelas ações de prevenção e de contingência dos riscos.

O termo de referência ou projeto básico deverá conter, no mínimo, a definição do objetivo, a fundamentação da contratação, a descrição da solução de TI, os requisitos da solução, o modelo de prestação de serviços ou de fornecimento de bens, os elementos para gestão do contrato, a estimativa de preços, a adequação orçamentária, as definições dos critérios de sanções e os critérios de seleção do fornecedor.

Seleção do Fornecedor

Na fase de Seleção de Fornecedor, são observadas as normas pertinentes, e tem como recomendação a utilização da modalidade de Pregão na forma eletrônica devido à padronização existente no mercado de TI. A área de licitações conduzirá esta fase e cabe à área de TI analisar as sugestões feitas, apoiar tecnicamente o pregoeiro na reposta a questionamentos e na análise e julgamento das propostas e dos recursos apresentados pelos licitantes. No encerramento desta fase, além do contrato, terá a nomeação do gestor, do fiscal técnico, do fiscal requisitante e do fiscal administrativo.

Gerenciamento do Contrato

A fase de Gerenciamento do Contrato (Fig. 4) visa acompanhar e garantir a adequada prestação do serviço e o fornecimento de bens que compõem a solução de TI. Compreende as seguintes etapas:

- Início do contrato;
- Encaminhamento formal de ordem de serviço ou fornecimento de bens;
- Monitoramento da execução;
- Transição contratual e/ou encerramento do contrato.

O início do contrato abrange a elaboração do plano de inserção da contratada, que contempla o repasse de conhecimento e a disponibilização de infraestrutura, e uma reunião inicial, que tem como objetivo a entrega do termo de compromisso e do termo de ciência e de sigilo, assim como possíveis esclarecimentos.

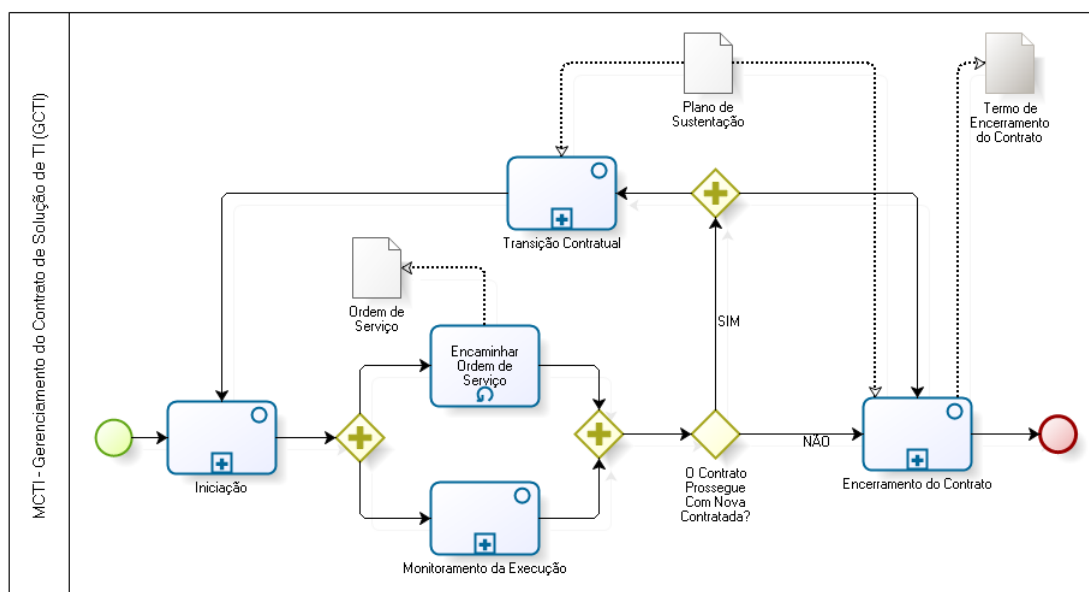


Figura 4 – Gerenciamento de Contratações de Soluções de TI (MPOG, 2013)

O encaminhamento formal de ordem de serviço ou fornecimento de bens pelo gestor do contrato para o preposto da contratada deve conter a definição e especificação dos serviços prestados ou bens fornecidos, o volume dos serviços ou quantidade de bens, segundo métricas, o cronograma e a identificação dos responsáveis pela solicitação da solução de TI.

O monitoramento da execução consiste: na confecção e assinatura do termo de recebimento provisório, na avaliação da qualidade do serviço, na identificação de não conformidade, na verificação de aderência aos termos contratuais, na verificação da manutenção das condições classificatórias, no encaminhamento das demandas de correção à contratada ou de indicação de sanções, na confecção e assinatura do termo de recebimento definitivo, na autorização para emissão de nota fiscal, na verificação das regularidades fiscais, trabalhistas e previdenciárias, na verificação de manutenção de necessidade, economicidade e oportunidade da contratação, no encaminhamento de pedidos de modificação contratual e na manutenção do histórico de gerenciamento de contrato.

E por fim, a etapa de transição contratual ou encerramento do contrato que deverá observar o Plano de Sustentação. Vale ressaltar que, para cada contrato, deverá haver pelo menos uma Ordem de Serviço ou de Fornecimento de Bens e pode haver quantas forem necessárias para execução do objeto contratado.

2.3 Considerações Finais do Capítulo

Foi apresentado neste capítulo as principais referências legais no que diz respeito a contratação de fornecedores de desenvolvimento de *software*. Este trabalho terá como

foco a fase de Gerenciamento do Contrato do Modelo de Contratações de Soluções de TI. O gestor do contrato é o facilitador entre a parte negocial e a parte contratada. A forma ou metodologia de gestão de um contrato é um dos principais fatores que influenciam no sucesso do mesmo. O gerenciamento adequado dos contratos pode diminuir os riscos, promover mais rapidez, garantir conformidade com as necessidades negociais e garantir o controle e transparência sobre o que está sendo desenvolvido. Neste trabalho será analisada uma solução utilizada na Gestão de Contrato de uma organização pública brasileira que foi baseada no Pensamento Lean e em Metodologias Ágeis. O próximo capítulo irá caracterizar o Pensamento *Lean*.

3 Pensamento Lean

Neste capítulo serão apresentados o contexto históricos e a caracterização do Pensamento *Lean*.

3.1 Lean na Manufatura

3.1.1 Histórico

O *Lean* é um modo de pensar a respeito de como entregar valor ao cliente mais rapidamente por meio da eliminação de desperdícios, que impedem a qualidade e a produtividade. O Pensamento *Lean* teve origem no sistema de produção da Toyota (TPS – Toyota Production System) como *Lean Manufacturing* e surgiu com o objetivo de reduzir desperdícios na produção. É uma filosofia de gestão que promove formas de especificar valor para o cliente, melhorar a sequência de fluxos de processos, tornar o desempenho mais eficiente e eliminar desperdícios na produção.

Para entendermos porque e como surgiu o pensamento *Lean*, é preciso saber o que ele substituiu: a produção em massa, que foi popularizada por Henry Ford. Ela tem como objetivo produzir em larga escala a baixo custo, pois divide o processo de manufatura em pequenos passos que possam ser desempenhados por trabalhadores com poucas habilidades e, para isso, é usado um maquinário de alta precisão e um trabalho padronizado. A produção em massa tem uma característica de inflexibilidade, pois alterações na linha de produção podem acarretar alto custo. Sendo assim, é apenas econômico produzir grandes quantidades da mesma coisa e da mesma forma (padronização) (HIBBS; JEWETT; SULLIVAN, 2009).

Em 1945, no Japão pós-guerra, o presidente da *Toyota Motor Company*, Kiichiro Toyoda, desafiou sua companhia a se igualar às companhias da América, caso contrário, a indústria automobilística japonesa não sobreviveria. E ficou claro que isso não poderia ser feito adotando o modelo de produção em massa norte-americano, pois no Japão os materiais eram escassos, as encomendas eram inconstantes e havia demanda por variedade, o que impediria o sucesso de produção em larga escala de produtos idênticos.

Taiichi Ohno, chefe de produção da Toyota, notou essas características que inviabilizavam a indústria automobilística japonesa de adotar o modelo de produção em massa. Com isso, experimentou muitas ideias e técnicas que foram inseridas aos poucos no que veio a ser conhecido como Sistema Toyota de Produção. Ele estudou o modelo de produção de Henry Ford e ampliou sua visão de como os supermercados norte-americanos controlavam seus estoques, além de acrescentar seus conhecimentos de fiação e tecelagem

e as percepções dos trabalhadores que ele supervisionava. Taiichi Ohno descreveu o ato como “um sistema para absoluta eliminação de desperdícios” e explicou que o sistema se mantém sobre dois pilares: *Just-in-Time* (JIT) e Jidoka (autonomação).

O fluxo *Just-in-Time* é o único modelo industrial para gerenciar com eficiência a complexidade inerente do custo de se produzir com variedade. Durante muitos anos, o Sistema Toyota de Produção foi ignorado, até mesmo no Japão, porém, após a crise econômica do petróleo nos anos 70, o sistema foi estudado e adotado por outras empresas japonesas, devido ao fato de a Toyota ter emergido da crise rapidamente. Após uma década, o Japão passou a ser um forte concorrente dos Estados Unidos e da Europa.

O Jidoka surgiu como o mecanismo utilizado pela máquina de tear automática G-Type, criada por Kiichiro. Ele consiste na paragem automática da máquina quando um fio quebrar, evitando que ocorressem desperdícios de material ao se produzir produtos defeituosos. Ao eliminar esse problema e diminuir os desperdícios no processo, a máquina se tornou um sucesso absoluto, melhorando tanto a produtividade quanto a eficiência do trabalho (KATAYAMA, 2010).

Com isso, o conceito que ficou com o Jidoka foi: organizar o trabalho de modo que ele seja imediatamente interrompido quando a menor anomalia for detectada, sendo necessário encontrar a causa e a resolução do problema antes de recomeçar a atividade. No português, “Jidoka” significa “autonomação”, que diz respeito ao fato de responder aos eventos de forma instantânea e correta, sem ter que ir ao cérebro receber instruções, como no caso da máquina, que para assim que surgir um problema (POPPENDIECK; POPPENDIECK, 2011).

O sistema de produção sofreu alterações ao longo das décadas. Nos anos 90, o Pensamento Lean passou a ser o termo usado para descrever este processo desenvolvido e utilizado pela Toyota. Assim, o Sistema Toyota de Produção, baseado no que hoje é chamado de Pensamento Lean, começou a ser formado nos anos 50 e, aos poucos, os resultados puderam ser notados na indústria automobilística mundial.

Como é apresentado na Fig. (5), nos anos 50 a Toyota nem sequer aparecia no *ranking* de vendas liderado pela GM e pela Ford. Já nos anos 70, o cenário começou a mudar, já que a Toyota passou a aparecer na sexta posição do *ranking* e teve uma ascendência até 2007, quando se tornou líder do *ranking* de vendas.

Ranking de vendas				
	1950	1970	2005	2007
1	GM	GM	GM	Toyota
2	Ford	Ford	Toyota	GM
3	Chrysler	Chrysler	Ford	VW
4	Studebaker	VW	Renault	Ford
5	Nash	Fiat	VW	Hyundai
6	Kaiser - Fra.	Toyota	DaimlerC	PSA
7	Morris	Nissan	Honda	Honda
8	Hudson	Renault	Hyundai	Nissan
9	Austin	BL	PSA	Fiat
10	Renault	Peugeot	Fiat	Suzuki

Figura 5 – *Ranking* de Vendas da Indústria Automobilística (GUEDES, 2012)

3.1.2 Princípios

O Pensamento Lean tem como objetivo fornecer o que o cliente deseja sem haver desperdícios, mas para atingi-lo, existem alguns princípios. Muitas companhias e indivíduos que querem implementar o Pensamento Lean cometem o erro de ficarem preocupados e focados em especificar ferramentas e práticas. Quando aplicadas de forma correta, as ferramentas podem gerar bons resultados de desempenho, mas são os princípios inseridos na cultura da organização que resultarão na mudança de comportamento em longo prazo.

Sem uma clara compreensão dos princípios que regem o Pensamento Lean, as empresas conseguirão resultados a curto prazo, mas a longo prazo não seria possível manter esse bom resultado e a melhoria contínua que daria estabilidade ao negócio e à satisfação do cliente. Assim, é preciso enfatizar os princípios acima das ferramentas.



Figura 6 – Pirâmide de Princípios Lean

Os princípios do Pensamento Lean na Manufatura são apresentados na Fig. (6). Eles são usados como base para a definição dos princípios e práticas do Lean no Desenvolvimento de *Software*, que serão apresentados na seção a seguir. Os detalhes a respeito do que consiste cada um dos princípios, práticas e ferramentas do Lean na Manufatura podem ser encontrados no Anexo A - Princípios e Práticas do Lean na Manufatura.

3.2 Lean no Desenvolvimento de *Software*

3.2.1 Abordagem

Mary e Tom Poppendieck (POPPENDIECK; POPPENDIECK, 2011) fizeram um mapeamento dos princípios Lean na Manufatura em sete passos no desenvolvimento de *software*. Tais princípios serão detalhados nas seções a seguir.

3.2.2 Princípios

3.2.2.1 Eliminar o Desperdício

Eliminar desperdícios no Sistema Lean de Produção (manufatura) funciona da seguinte forma: olhar a linha do tempo desde a concepção do produto até a entrega ao cliente, e remover aquilo que não acrescenta valor (desperdícios). No Lean de Desenvolvimento de *Software*, o objetivo de eliminar desperdícios é o mesmo, porém, o início e fim

da linha do tempo podem ser alterados, pois o desenvolvimento de *software* tem início no momento do pedido e vai até quando ele é entregue. Reduzir os desperdícios na linha do tempo significa reduzir essa própria linha, ou seja, entregar o que foi pedido o mais rápido possível e com qualidade (POPPENDIECK; POPPENDIECK, 2011).

Para conseguir eliminar desperdícios é preciso, primeiramente, identificá-lo. Como ele se caracteriza por tudo aquilo que não agrega valor, é preciso ter conhecimento do que realmente é o valor. Na área de desenvolvimento de *software*, identificar o que é valor para o cliente é algo mais complexo, pois, dificilmente, no início do processo, o cliente sabe realmente o que quer, já que as suas necessidades e os seus desejos mudam ao longo do desenvolvimento, alterando a definição do que agrega valor. No entanto, grandes organizações têm conseguido ter um conhecimento profundo do que é valor para o cliente e, conseqüentemente, conseguem satisfazer suas necessidades. Assim, o objetivo de todas as organizações é conseguir alcançar esse entendimento profundo de valor.

A partir do momento que as pessoas conseguem identificar o que é valor, é possível começar a desenvolver a capacidade de identificar o que é desperdício. Qualquer atividade não importante, que impeça o cliente de receber o que ele deseja e quando ele deseja, é desperdício. Semelhante ao estoque no sistema produtivo, trabalhos inacabados no desenvolvimento de *software* gera prejuízos, pois um tempo e um esforço foram perdidos e o trabalho passa a estar tecnologicamente atrasado, a esconder problemas de qualidade e a estagnar dinheiro.

Especificamente no desenvolvimento de *software*, alguns desperdícios acontecem de forma constante: todos os requisitos especificados no início, que obviamente sofrerão mudanças; testes realizados muito perto do fim, que claramente vão gerar muito retrabalho; e, principalmente, funcionalidades desenvolvidas que nunca chegam a ser utilizadas. De fato, estima-se que cerca de apenas 20% das funcionalidades em um sistema personalizado são regularmente usadas, sendo que dois terços delas são raramente usadas, o que gera custos desnecessários.

3.2.2.2 Integrar Qualidade

A meta do Lean no Desenvolvimento de *Software* é construir *software* com qualidade desde o início, não deixando para inserir testes só no final do desenvolvimento. A organização precisa ser bastante disciplinada. Para desenvolver um *software* com qualidade, é preciso controlar as condições desde o começo, de forma a não permitir defeitos. Como não é possível prevenir todos os defeitos, é indicado inspecionar o produto após cada pequena funcionalidade ser implementada, para que possa encontrar o defeito imediatamente após o seu surgimento (POPPENDIECK; POPPENDIECK, 2011).

É importante ressaltar que o slogan “faça certo na primeira vez”, muito comumente usado, não se refere a construir o código e jamais modificá-lo, mas sim ao fato de se usar as

técnicas *Test Driven Development* (TDD), integração contínua e refatoração para garantir que um código simples e limpo agregue valor ao cliente.

3.2.2.3 Criar Conhecimento

O desenvolvimento de *software* é um processo de criação do conhecimento, e com isso, esperará que o projeto evolua durante a codificação e não desperdiçará muito tempo fazendo todos os requisitos e arquitetura prematuramente.

As empresas que já mostraram uma excelência a longo prazo, em desenvolvimento de produtos, compartilham um traço em comum: elas geram novo conhecimento por meio de uma experimentação disciplinada e codificam este conhecimento concisamente para torná-lo acessível ao restante da organização. Essas empresas não somente capturam dados explícitos, mas encontram maneiras de tornar visível o conhecimento e fazê-lo parte da base de conhecimento organizacional (NONAKA; TAKEUCHI, 1995).

É importante ter um processo de desenvolvimento que encoraje o aprendizado sistemático durante todo o ciclo e também é preciso melhorar os processos de desenvolvimento continuamente. Às vezes, na constante busca por padrões ou modelos de processos, as empresas prendem-se a uma documentação que torna mais difícil para as equipes de desenvolvimento reservar um tempo para melhorar diariamente seus próprios processos. Em Lean, devem-se melhorar continuamente os processos, pois em ambientes complexos sempre haverá problemas. Assim, para cada problema é preciso acionar uma busca pela causa raiz, construir soluções possíveis e acionar uma mudança no processo para impedir que o problema ressurgir (POPPENDIECK; POPPENDIECK, 2011).

3.2.2.4 Adiar Comprometimentos

A ideia central deste princípio é deixar as decisões irreversíveis para o último momento possível, ou seja, a última oportunidade de tomar a decisão antes que seja tarde demais, afinal de contas, uma tomada de decisão é um compromisso que é firmado. Porém, isto não quer dizer que todas as decisões devam ser adiadas. Em primeiro lugar, é preciso tentar tornar a maioria das decisões reversíveis, ou seja, que possam ser mudadas ao decorrer do desenvolvimento sem prejuízos significativos. Um sistema de *software* não precisa ser completamente flexível, mas precisa que opções sejam preservadas em pontos onde as mudanças invariavelmente ocorram. É preciso experiência para saber quando devem ser mantidas essas opções (POPPENDIECK; POPPENDIECK, 2011).

3.2.2.5 Entregar Rápido

A ideia deste princípio parte do pressuposto de que é necessário entregar o *software* tão rápido que os clientes não tenham tempo para mudar de ideia. As empresas que competem com base no tempo frequentemente têm uma vantagem significativa de custo

sobre seus concorrentes, e, para isso ser possível, elas eliminaram uma grande quantidade de desperdício - e isso tem influência diretamente no prazo e no custo. Além disso, velocidade repetível e confiável não é possível sem desenvolver com grande qualidade. Para ter sucesso, tais empresas criaram um profundo conhecimento sobre o que é valor para o cliente. Com a rapidez, é possível testar novas ideias e aprender o que funciona ou não (POPPENDIECK; POPPENDIECK, 2011).

3.2.2.6 Respeitar as Pessoas

O princípio de respeitar as pessoas diz respeito a não impedir que elas façam seus trabalhos só porque outra pessoa acredita que não é a melhor forma de fazê-lo. Todos devem respeitar-se e trabalharem juntos para a melhor solução. No processo de desenvolvimento, também deve haver respeito, pois todos devem fazer parte da melhoria contínua dos processos. Existem três bases que sustentam o respeito às pessoas: (POPPENDIECK; POPPENDIECK, 2011):

- Líder empresarial: uma empresa que respeita as pessoas desenvolve bons líderes e garante que a equipe tenha o tipo de liderança que promove pessoas engajadas e pensantes, concentrando seus esforços na criação de um ótimo produto;
- Mão de obra técnica especializada: empresas sábias garantem que a especialização técnica apropriada seja estimulada dentro da própria empresa e que as equipes estejam abastecidas da especialização necessária para atingir determinado objetivo;
- Responsabilidade baseada em planejamento e controle: respeitar as pessoas significa que as equipes recebem planos genéricos e objetivos claros, e em vez de dizer como e o que fazer, desenvolve uma organização reflexiva onde as pessoas usam suas cabeças e descobrem isso sozinhas.

3.2.2.7 Otimizar o Todo

Uma organização Lean otimiza todo o fluxo de valor, do momento em que recebe o pedido visando uma necessidade do cliente, até o momento em que o *software* seja implantado e a necessidade do cliente seja atendida. Se houver concentração em otimizar apenas uma pequena parte do todo, inevitavelmente, o fluxo de valor completo sofrerá.

3.2.3 Práticas

3.2.3.1 Gerenciamento de Código Fonte e *Scripts* de *Builds*

O Gerenciamento de Código Fonte também é conhecido como Controle de Versão e, basicamente, significa manter todo o código fonte e outros artefatos importantes do projeto em um repositório que mantém um histórico de versões de todos os itens de

configuração, que são artefatos que precisam ser mantidos sobre controle, pois sofrem mudanças e evoluções. Com o controle de versão é possível recuperar uma configuração em um momento desejado do tempo e sempre é possível retornar a uma versão estável (*baseline*) quando erros são identificados (HIBBS; JEWETT; SULLIVAN, 2009). O Gerenciamento de Código Fonte e *Scripts de Builds* não são práticas exclusivas do Lean, mas sim uma prática que deveria ser usada adotando Lean ou não, por isso é considerada como um pré-requisito para as práticas do Lean. Elas ajudam a construir um time disciplinado e um ambiente de desenvolvimento estável.

3.2.3.2 Teste Automatizado

Um sistema à prova de erros é uma das práticas fundamentais do pensamento *lean*. É parte central da produção de um produto com qualidade e da redução de desperdícios. O teste automatizado é um dos principais meios de evitar erros no desenvolvimento de *software*, pois engloba todos os tipos de testes: unitário, de integração, de aceitação, de desempenho, de carga e etc.

O teste automatizado suporta três princípios do Lean: eliminação de desperdícios, qualidade na raiz e criar conhecimento. Ele ajuda a eliminar desperdícios que geram retrabalho e custos quando erros são detectados apenas nas fases finais do ciclo de desenvolvimento de *software*. No que diz respeito a construir com qualidade, o teste automatizado é um dos principais meios. Uma base de código com uma suíte de testes automatizados realiza a validação e checagem de forma automática, o que reduz a possibilidade de que erros não detectados sejam introduzidos no *software*. Além disso, os testes automatizados servem como uma documentação atualizada do que está sendo feito. Ele cria um conhecimento primário e confiável para os desenvolvedores porque garante conformidade com o esperado toda vez que a suíte de teste é executada com sucesso (HIBBS; JEWETT; SULLIVAN, 2009).

3.2.3.3 Integração Contínua

A integração diz respeito ao momento que todos os módulos ou componentes do *software*, que está sendo desenvolvido, sejam executados juntos. Componentes individuais, banco de dados, interfaces de usuário e recursos do sistema são todos reunidos e testados em cima da arquitetura. A integração envolve verificar a comunicação dos componentes e garantir que a mensagem passada entre eles é compatível e completa.

Já a integração contínua é o processo de integrar pequenas mudanças em uma base estável para entrega de uma nova *release* do produto. Integrando pequenas mudanças em um curto intervalo, os desenvolvedores podem evoluir o produto um pouco de cada vez enquanto garantem que cada novo pedaço funcione corretamente com todo o resto do sistema. A integração contínua usa testes unitários para garantir que cada pequena parte

do sistema está implementada corretamente, e usa o gerenciamento do código fonte e os *scripts* de *build* para garantir que, a cada nova pequena integração, nada foi quebrado, ou seja, que o sistema continue funcionando corretamente como um todo (HIBBS; JEWETT; SULLIVAN, 2009).

3.2.3.4 Menos Código

A prática de escrever menos código não diz respeito a escrever menos *software*, mas sim a ter todas as funcionalidades implementadas com poucas linhas de código e de maneira simples. Se trata de eliminar o código desnecessário e fazer com que o necessário se torne mais eficiente, mantendo a base de código pequena, ao menos tempo em que funcionalidades (valor) são entregues ao cliente.

Uma base de código grande acarreta em mais componentes, mais *bugs* e grande curva de aprendizagem. Como consequência, o custo em desenvolvimento e manutenção será maior. A definição de desperdício no desenvolvimento Lean é qualquer coisa que aumente os custos sem produzir valor, então o custo de desenvolvimento e manutenção resultante de uma grande base de código pode ser considerado como desperdício.

Os desenvolvedores precisam ser minimalistas durante todo o processo de desenvolvimento, criando, de maneira simples, cada funcionalidade. Um *design* simplista facilita, ainda, a implementação de mudanças (HIBBS; JEWETT; SULLIVAN, 2009). Os padrões e técnicas podem ser utilizados para auxiliar os desenvolvedores a criar apenas o código que é realmente necessário, por exemplo, utilizar padrões de *design*, reuso e refatoração.

3.2.3.5 Iterações Curtas

O desenvolvimento iterativo entrega *software* funcional ao cliente para avaliação a cada intervalo específico de tempo. Cada iteração adiciona novas funcionalidades ao produto e aumenta o entendimento do cliente sobre como o produto final irá atender às suas necessidades. A efetividade do processo iterativo vem da oportunidade de ter o *feedback* do cliente e incorporá-lo ao desenvolvimento. O retorno do cliente é o que irá direcionar a próxima iteração quanto à adição, remoção ou modificação de requisitos e implementação. Quanto menor a iteração, mais oportunidades de receber o *feedback* do cliente e maior é a possibilidade de atender o que o cliente deseja no produto final.

As iterações curtas suportam três princípios do Lean: elimina desperdício, adiar comprometer e entregar rápido. Existem duas formas de desperdícios advindos de iterações largas: o trabalho parcial, que não é construído de forma completa, seja com o requisito não implementado ou com o código não testado, não adicionando valor ao produto; e o replanejamento, quando o planejamento inicial é feito para um futuro muito distante, as necessidades e desejos do cliente podem mudar, fazendo com que seja preciso realizar o replanejamento, com isso, o trabalho planejado anteriormente é desperdiçado.

As iterações curtas previnem tais situações de desperdício, pois o esforço fica concentrado em desenvolver o que é de alta prioridade para o cliente mais rapidamente e planeja tão longe quanto necessário para manter o desenvolvimento.

O princípio de adiar comprometimento, como já explicado anteriormente, diz respeito a adiar decisões importantes tanto quanto possível. Adiar tais decisões fornece aos desenvolvedores tempo para coletar e entender as informações necessárias para o desenvolvimento, e iterações curtas dão oportunidades de coletar informações por meio do *feedback* do cliente. Os desenvolvedores podem criar protótipos do produto final já nas iterações iniciais, pois recebem o *feedback* e o utilizam para decidir a implementação final nas iterações finais (HIBBS; JEWETT; SULLIVAN, 2009).

As iterações curtas, claramente, suportam o princípio de entrega rápida devido a redução dos intervalos de contato com o cliente e por entregar novas funcionalidades rapidamente para o recebimento de *feedback*. Em vez de esperar até o final do desenvolvimento para ver os efeitos das suas solicitações, o cliente já poderá visualizá-los na próxima iteração.

3.2.3.6 Participação do Cliente

Se o objetivo do desenvolvimento de um produto é criar o que o cliente irá usar, é essencial ter a participação do cliente em todo o processo. Os conceitos de priorizar requisitos e de realizar correções de acordo com o *feedback* do cliente necessitam de participação ativa do mesmo no processo de desenvolvimento.

Os clientes são as melhores fontes de informações sobre o domínio do problema. Eles sabem as atividades que precisam ser feitas, as condições na qual o *software* será executado e os objetivos que desejam atingir com a solução. No entanto, eles raramente sabem sobre a tecnologia que será usada na implementação da solução.

Por outro lado, os desenvolvedores sabem sobre as tecnologias e o caminho mais eficiente para modelar e implementar problemas e soluções. O que os desenvolvedores não sabem é sobre os conhecimentos do negócio no qual a solução será incorporada.

Combinando o conhecimento do negócio do cliente com o conhecimento técnico do desenvolvedor cria-se uma equipe dinâmica que resulta em um desenvolvimento de um produto melhor. Desenvolvedor e clientes trabalhando juntos e inspiram um ao outro a criar algo melhor do que eles poderiam criar individualmente (HIBBS; JEWETT; SULLIVAN, 2009).

3.2.3.7 Kanban

O Kanban no desenvolvimento de *software* surgiu a partir do Kanban do *Lean* na Manufatura, já que ambos usam um mecanismo de controle visual para acompanhar o

trabalho à medida que ele flui através das várias etapas do fluxo de valor.

O Kanban não é um processo ou ciclo de vida de gerenciamento de projetos ou de desenvolvimento de *software*, e sim uma abordagem para introduzir mudanças em um ciclo de desenvolvimento de *software* ou metodologia de gerenciamento. Ele possui três conceitos básicos: visualizar o fluxo de trabalho, limitar o trabalho em progresso (WIP) e medir e melhorar o fluxo (KNIBERG; SKARIN, 2009).

Para visualizar o fluxo de trabalho divida o trabalho em partes, escreva cada item em um cartão - ou *post-its* -, e coloque no quadro Kanban (Fig. 7). Use colunas nomeadas para ilustrar onde cada item está no fluxo de trabalho. Com esse mapeamento, já é possível ter o entendimento do processo atual.

Kanban board

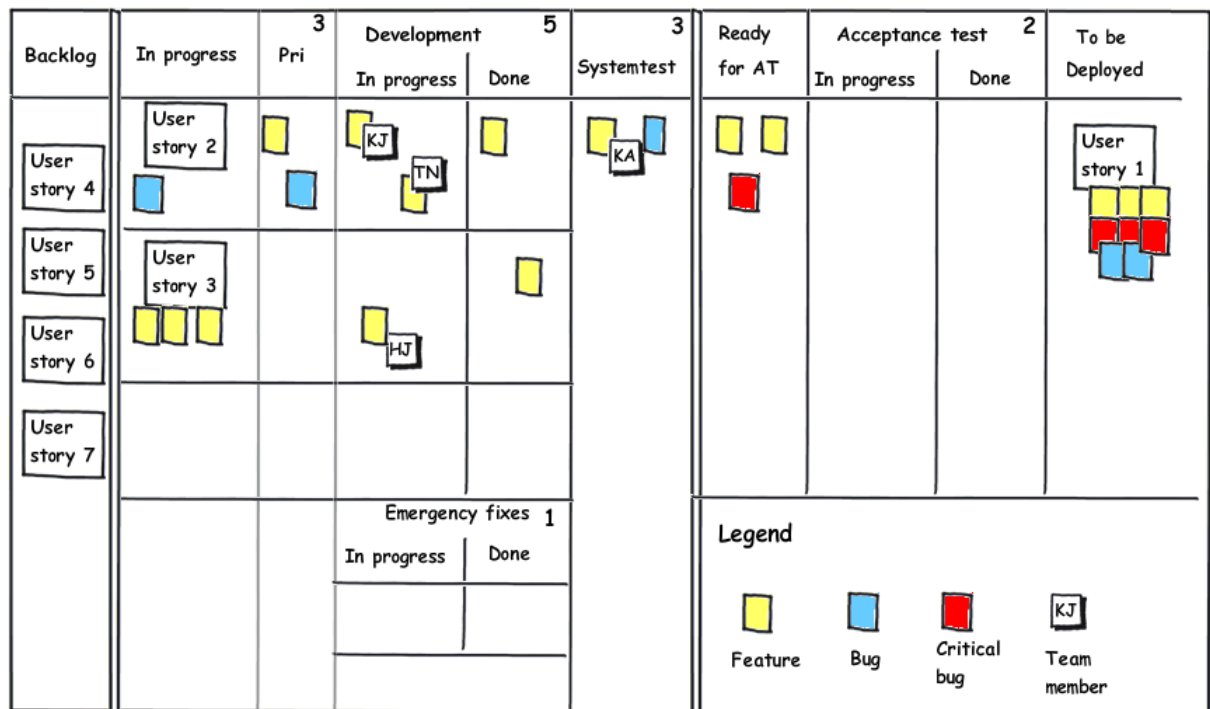


Figura 7 – Quadro Kanban (JENSES, 2009)

Para limitar o WIP é preciso associar limites explícitos para quantos itens podem estar em progresso em cada estado do fluxo de trabalho. Existe um número limite de tarefas que podem ser bem feitas durante um mesmo período de tempo (KLIPP, 2011).

É preciso saber a diferença entre a complexidade das tarefas, já que duas podem ser feitas em uma semana assim como também podem ser feitas em três horas, tudo

dependerá de quanto esforço será gasto em cada uma. As métricas do Kanban ajudam a chegar a um número ótimo.

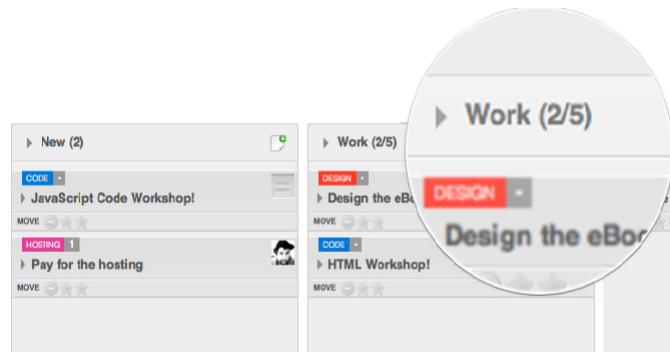


Figura 8 – *Work in Progress* (KLIPP, 2011)

Além disso, com o WIP (Fig. 8) limitado em um sistema Kanban, tudo que fica bloqueado por qualquer motivo tende a parar o sistema. Se certa quantidade de itens de trabalho fica bloqueada, todo o processo irá parar de funcionar. Isso cria a necessidade de concentrar toda a equipe e toda a empresa na solução do problema para desbloquear o item e restaurar o fluxo.

A melhoria deve sempre ser baseada em objetivos mensuráveis (Fig. 9), e no Kanban isto não é diferente. Encontrar e aplicar boas métricas é geralmente um passo difícil, mas algumas métricas simples, automaticamente geradas por uma aplicação, podem dar a informação necessária para otimizar o processo e maximizar a eficiência (KLIPP, 2011).

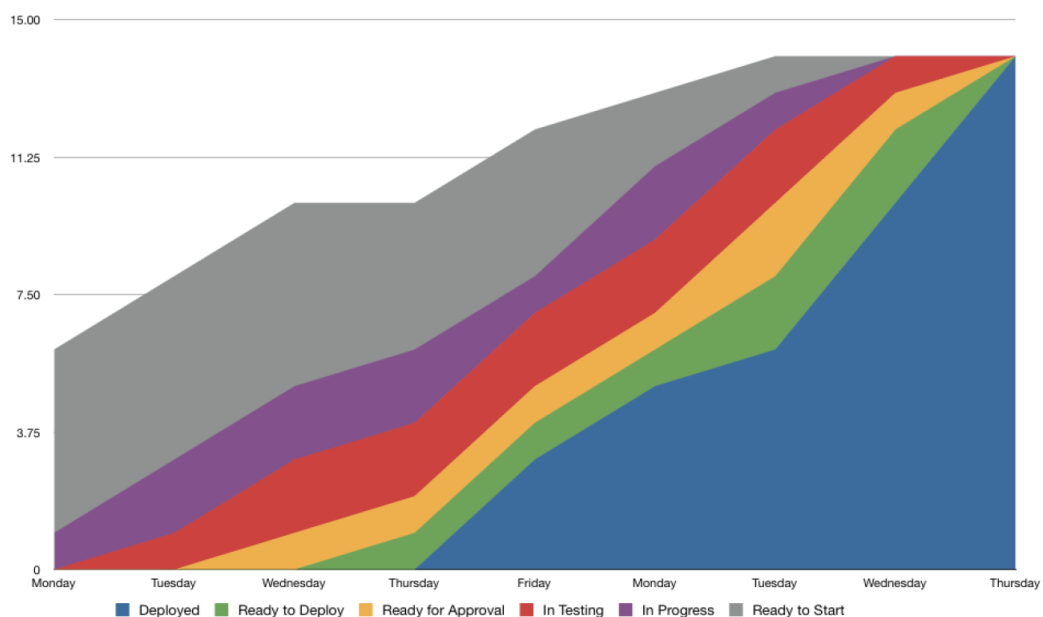


Figura 9 – Gráfico de Medições Kanban (KLIPP, 2011)

O Kanban, no entanto, vai um passo além e dá transparência ao processo e ao seu fluxo. Ele expõe gargalos, filas, variabilidade e desperdício. Ele expõe tudo que impacta o desempenho da organização em termos de quantidade de trabalho, de valor entregue e de tempo de ciclo necessário para entregar determinada funcionalidade. Ele proporciona aos membros da equipe e às partes externas interessadas a visibilidade sobre os efeitos de suas ações (ou falta de ações), e isso incentiva a discussão sobre melhorias que precisam ser feitas nos processos, encorajando a evolução incremental dos procedimentos existentes. Ainda, o Kanban, através da sua natureza de sistema *pull*, encoraja também o comprometimento tardio, tanto em priorização de trabalho novo, quanto na entrega de trabalho existente (KNIBERG; SKARIN, 2009).

3.3 Considerações Finais do Capítulo

O Pensamento *Lean* é, portanto, um modo de pensar e exige uma mudança cultural para que ele seja aplicado com sucesso dentro de uma organização. Ele é baseado em princípios e possui práticas e ferramentas que irão auxiliar na implantação do mesmo. Uma das principais ferramentas oriunda do pensamento *lean* foi o quadro kanban, o qual também é adotado dentro das metodologias ágeis. Essa ferramenta e os princípios do pensamento *lean* são partes fundamentais da solução que será analisada nesse trabalho. O próximo capítulo irá apresentar a caracterização das metodologias ágeis, particularmente o Scrum, e irá apresentar o contexto atual de adoção de métodos ágeis na gestão de contratos de desenvolvimento de *software* em organizações públicas brasileiras.

4 Adoção de Métodos Ágeis na Gestão de Contratos de Desenvolvimento de *Software* em Organizações Públicas Brasileiras

A adoção de métodos ágeis em contratações de desenvolvimento de *software* em organizações públicas brasileiras está se tornando cada vez mais frequente. Assim, neste capítulo serão apresentados os conceitos relativos às metodologias ágeis. Para isso, apresentase o conceito, os valores e os princípios ágeis. Ainda, será caracterizado de forma particular a metodologia ágil *Scrum*. Além disso, será apresentado o contexto atual de utilização de metodologias ágeis em contratações de fornecedores de desenvolvimento de *software* em organização públicas brasileiras segundo o Acórdão 2314/2013.

4.1 Metodologias Ágeis de Desenvolvimento de *Software*

Ao longo da evolução dos processos de Engenharia de *Software*, a indústria se baseou nos métodos tradicionais de desenvolvimento de *software*, que definiram por muitos anos os padrões para criação de *software* nos meios acadêmico e empresarial. Porém, percebendo que a indústria apresentava um grande número de casos de fracasso, alguns líderes experientes adotaram modos de trabalho que se opunham aos principais conceitos das metodologias tradicionais. Aos poucos, foram percebendo que suas formas de trabalho, apesar de não seguirem os padrões no mercado, eram bastante eficientes (FILHO, 2008).

Em 2001, 17 líderes que estavam desenvolvendo projetos de formas diferentes aos padrões até então pregados pela indústria se reuniram em uma estação de ski em Utah para discutir seus trabalhos e experiências, no final da reunião, eles formaram um grupo intitulado de Aliança de Desenvolvimento Ágil e escreveram O Manifesto Ágil, o qual é constituído de princípios e valores que o grupo considerou determinantes para bons resultados no desenvolvido de *software*. Com isso, os métodos de desenvolvimento de *software* ágeis, em contraste com os métodos tradicionais dirigidos a planos, são baseados em quatro valores advindos do Manifesto Ágil, são eles (CUNNINGHAM, 2001):

- Indivíduos e interações acima de processos e ferramentas;
- *Software* operante acima de documentações grandes e completas;
- Colaboração do cliente acima de negociações contratuais;
- Responder à mudanças acima de seguir um planejamento.

As declarações destes valores têm um formato fácil de se identificar. A primeira parte da sentença indica a preferência, enquanto a segunda parte da sentença indica algo que, embora importante, tem prioridade menor. Assim, a Aliança e o Manifesto Ágil, por exemplo, reconhecem a importância da documentação, processos e ferramentas, no entanto, reconhece também que, a interação entre indivíduos capacitados tem ainda maior importância.

Dessa maneira, uma documentação completa e de fácil entendimento não é ruim, porém, o foco primário deve estar na entrega de um *software* operante e na interação entre as pessoas. Nesse contexto, a negociação contratual é uma prática insuficiente, pois, os contratos podem apresentar condições de fronteiras nas quais os envolvidos podem trabalhar, mas somente com a contínua colaboração entre cliente e equipe é que se consegue entender e entregar o que o cliente realmente deseja (HIGHSMITH, 2002). Assim, um plano detalhado, elaborado no início de um projeto de desenvolvimento de *software* pode acarretar riscos, caso ele impeça que mudanças ocorram. Esse ponto pode ser conflitante com a necessidade de planejamento e o princípio da impessoalidade, inerentes às contratações públicas. No entanto, é possível alinhar a utilização desses valores ágeis aos preceitos legais, mediante a atenção a certas cautelas (TCU, 2013).

Os valores ágeis estão relacionados aos doze princípios também advindos do Manifesto Ágil, são eles (CUNNINGHAM, 2001):

- Satisfazer o cliente por meio da entrega rápida e contínua de *software* que traga valor;
- Mudanças nos requisitos são aceitas, mesmo em estágios avançados de desenvolvimento;
- *Software* funcionando é entregue frequentemente em curtos períodos de tempo;
- As pessoas relacionadas ao negócio e os desenvolvedores devem trabalhar em conjunto diariamente;
- Construa projetos formados por indivíduos motivados, forneça o ambiente e suporte necessário e confie que eles realizarão o trabalho;
- O modo mais eficiente e eficaz de transmitir informações dentro e fora do time de desenvolvimento é a comunicação face a face;
- A principal medida de progresso é o *software* funcionando;
- Processos ágeis promovem o desenvolvimento em ritmo sustentável, onde os envolvidos devem ser capazes de manter o ritmo constante;

- Cuidar continuamente da excelência técnica e do bom *design* ajuda a aprimorar a agilidade;
- Simplicidade – a arte de maximizar a quantidade de trabalho não necessário – é essencial;
- Os melhores requisitos, arquiteturas e *design* surgem de equipes auto gerenciadas;
- Em intervalos regulares, a equipe reflete sobre como se tornar mais eficiente, refinando e ajustando seu comportamento apropriadamente.

A partir de então, diversas práticas que já eram aplicadas a alguns projetos se transformaram em métodos de desenvolvimento de *software*, baseadas nos princípios e valores ágeis, como exemplo o Scrum e o *Extreme Programming* (XP). A abordagem ágil compartilha de arcabouço teórico semelhante ao utilizado para definição dos conceitos pensamento *lean*, além de sua adaptação ao contexto do desenvolvimento de *software*. Pode-se observar, por exemplo, que ambas dão ênfase à satisfação do cliente e entrega rápida do produto.

4.1.1 Scrum

O Scrum é uma metodologia que começou a ser desenvolvida na década de 90 (SCHWABER, 1996). Ele se concentra nos aspectos gerenciais do processo de desenvolvimento de *software*. É uma abordagem que favorece o empoderamento de equipes auto organizáveis e integração com outras metodologias que foquem em práticas relacionadas à programação, como o XP.

O Scrum caracteriza-se por ser um processo empírico e adaptativo propondo iterações de curta duração (as chamadas *Sprints*) com reuniões de acompanhamento diárias (chamadas de *Daily Meeting*) realizadas de pé, as quais permitem a identificação antecipada de eventuais problemas e a visibilidade do que está sendo feito por todos. A Figura (10) apresenta o ciclo de vida do processo Scrum.

No processo Scrum são definidos três papéis: Scrum *Master*, aquele que deve ensinar a utilizar o Scrum, acompanhar o desenvolvimento e resolver impedimentos; *Product Owner*, aquele que deve garantir que o produto atenda as necessidades do cliente ou patrocinador do projeto e deve priorizar as funcionalidades que agreguem mais valor ao cliente; e a Equipe de Desenvolvimento (SUTHERLAND'S, 2010).

Os requisitos do produto são expressos por meio das histórias de usuário, que em seu conjunto compõem o artefato conhecido como (*Product Backlog*). No início de cada *Sprint* é feita uma reunião de planejamento (*Sprint Planning*) na qual uma estimativa e a meta da *Sprint* são definidas e algumas histórias de usuário (*User Stories*) que estão no

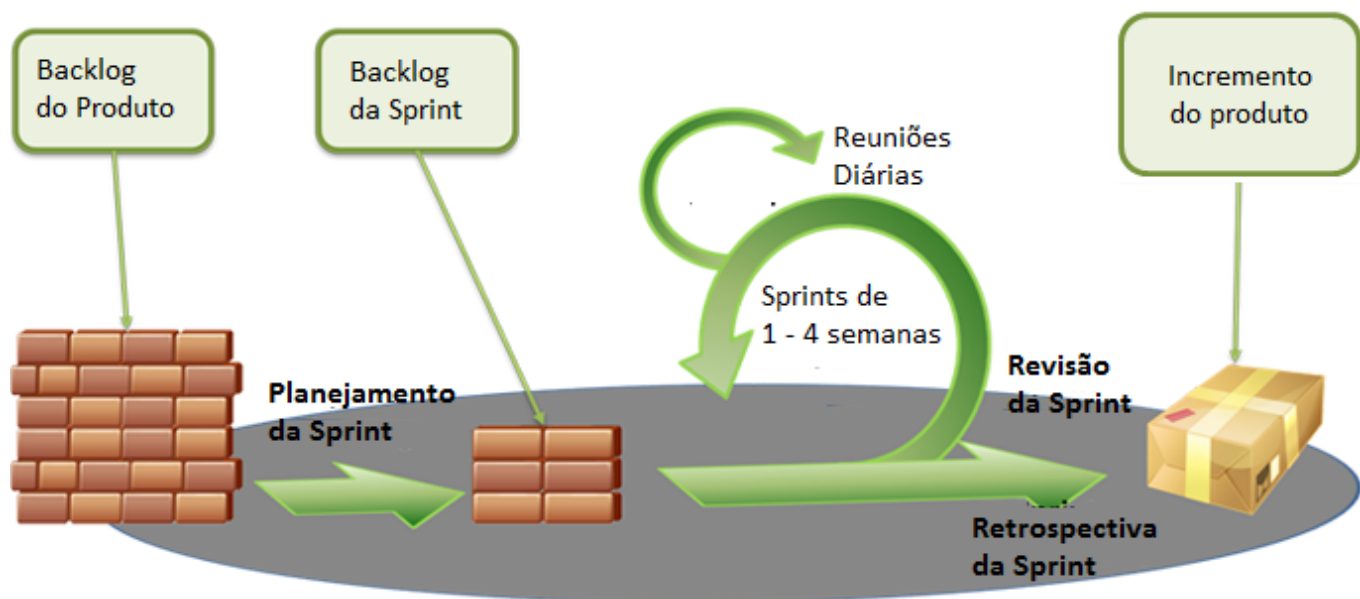


Figura 10 – Framework Scrum

Backlog do Produto são selecionadas e transformadas em tarefas do *Backlog da Sprint*, as quais devem ser desenvolvidas para atingir a meta da *Sprint*. Depois do planejamento, a equipe começa um ciclo de desenvolvimento do produto, que pode durar de uma a quatro semanas. Durante cada *Sprint*, o *Scrum Master* assegura que as histórias de usuário selecionadas não mudarão, permitindo que a equipe fique concentrada em seu objetivo. O *Product Owner* acompanha o desenvolvimento e esclarece eventuais dúvidas. O *Product Owner* só pode interferir no desenvolvimento da *Sprint* se uma mudança de negócio afetar os requisitos do produto que está sendo desenvolvido (SUTHERLAND'S, 2010).

Quando uma *Sprint* é finalizada, a equipe de desenvolvimento apresenta o trabalho feito na reunião de revisão da *Sprint* (*Sprint Review*). O *Product Owner* faz testes e verifica se a meta foi realmente atingida. Ainda, após a entrega, a equipe realiza uma reunião de retrospectiva (*Sprint Retrospective*) para analisar seu trabalho e oportunidades de melhoria para a próximo ciclo de produção. (SUTHERLAND'S, 2010).

4.2 Kanban, Scrum e o Pensamento Lean

Scrum e Kanban são ferramentas de processo que, em certa medida, ajuda toda a equipe a trabalhar de maneira mais eficaz, por meio de instruções do que deve ser feito. Como qualquer ferramenta, Scrum e Kanban não atendem completamente a todas as situações. Eles não dizem absolutamente tudo que precisa ser feito, mas sim oferecem algumas restrições e orientações. Por exemplo, o Scrum restringe um tempo fixo para cada iteração e que as equipes devem ser multifuncionais, enquanto que o Kanban restringe

a utilização de quadros visíveis e a limitar o tamanho da quantidade de trabalho a ser desenvolvido em um mesmo período de tempo. A utilização de ferramentas ajuda a ter sucesso, mas não garante por si só o sucesso. Tanto Scrum quanto Kanban são empíricos no sentido em que se espera que as pessoas experimentem o processo e o personalize segundo às necessidades da organização e do próprio time.

Scrum e Kanban são sistemas puxados (*pull*), que correspondem ao princípio de gestão de inventário de *Just In Time* (JIT) do Lean. Isso significa que a equipe escolhe quando e quanto de trabalho irá se comprometer para então “puxar” o trabalho quando estão prontos para começar, ao invés de ter que empurrar o trabalho de algum lugar. Scrum e Kanban são baseados em otimização empírica e contínua do processo, que corresponde ao princípio de Kaizen do *Lean*.

Um modo de comparar essas duas ferramentas é analisando a quantidade de regras que elas oferecem. Quanto mais prescritivo for um processo, mais regras estão disponíveis para seguir. A idéia passa pelo entendimento de que não seria necessário muito esforço intelectual já que as regras já estão detalhadas. Quanto mais adaptativo um processo menos regras estão definidas para serem seguidas, ou seja, pode-se fazer qualquer coisa. Evidentemente, os dois extremos desta escala não é bom (KNIBERG; SKARIN, 2009).

Os métodos ágeis são por vezes chamados de métodos “leves” por serem menos prescritivos que os métodos tradicionais devido à quantidade de atividades, papéis e artefatos definidos. Scrum e Kanban são ambos adaptativos, mas ao realizar uma comparação entre as duas ferramentas, Scrum pode ser considerado mais prescritivo que o Kanban, pois existem mais restrições. Por exemplo, o Scrum prescreve um quadro de tarefas com as raias fixas, *TO DO*, *DOING* e *DONE*, além do uso de iterações curtas. Já o Kanban não. O segredo está em não se limitar a uma única ferramenta. As ferramentas mais adaptativas podem ser combinadas de forma a trazerem mais benefícios para a organização. Na Figura (11) é apresentada uma escala comparativa em alguns métodos.

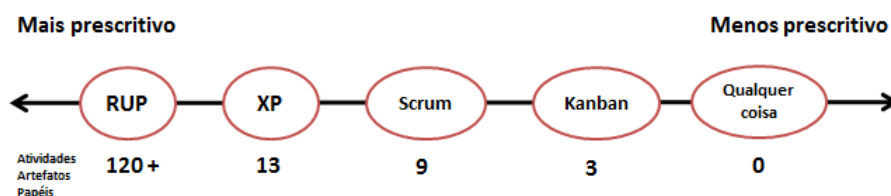


Figura 11 – Comparação entre métodos segundo suas características prescritivas (KNIBERG; SKARIN, 2009)

A despeito das considerações apresentadas e considerando que tanto o Scrum quanto o Kanban promovem um fluxo de trabalho contínuo e rápido. Porém, cabe ressaltar algumas diferenças observadas:

4.2.1 Cadência

No Scrum há uma única cadência de tempo, fixo, que combina quatro principais atividades: o planejamento, o desenvolvimento, a entrega do algo funcional e a melhoria. Uma cadência única diz respeito ao fato de manter as iterações com o mesmo tempo fixo durante um período de tempo realizando as mesmas atividades principais. Já no Kanban, não há esta prescrição de duração fixa de iterações, pode-se escolher uma periodicidade regular para realizar determinadas atividades ou pode-se escolher entregar sempre que tiver algo útil e funcional, ou seja, cadências distintas (KNIBERG; SKARIN, 2009).

4.2.2 Limitação do WIP

No Scrum, o *Sprint Backlog* mostra quais são as tarefas a serem executadas durante a iteração atual. Geralmente, o *Sprint Backlog* é representado no quadro Scrum ou quadro de tarefas. No Kanban, também existe o quadro de tarefas com propósito semelhante ao quadro utilizado no Scrum. Em ambos há o controle das tarefas ao longo do fluxo de trabalho. As colunas de ambos os quadros podem ser determinadas a gosto. A única coisa que torna um quadro diferente do outro é que no Kanban o trabalho em progresso (WIP), na raia “em progresso” ou nas demais colunas, é limitado explicitamente, enquanto no Scrum o limite explícito é apenas no tamanho das iterações (Fig. 12).

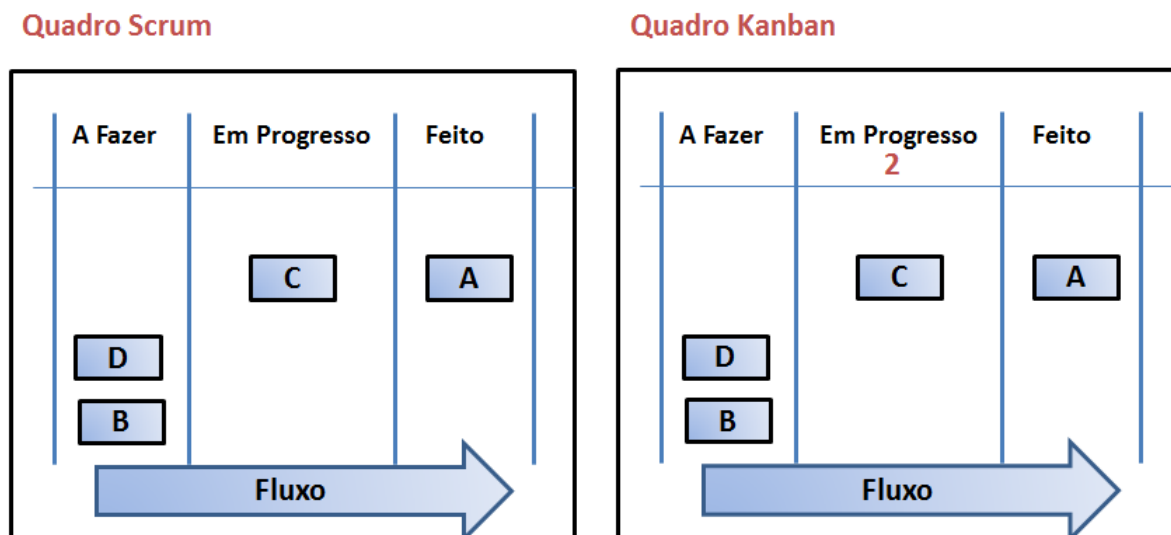


Figura 12 – Quadro Scrum x Quadro Kanban (KNIBERG; SKARIN, 2009)

Por vezes, uma equipe Scrum limita intuitivamente a quantidade de trabalho em progresso, evitando colocar mais atividades na coluna “em progresso” sem antes retirar as que lá estão. Quando uma equipe Scrum decide de fato limitar o WIP, então ela estará utilizando o quadro Kanban. Geralmente, as equipes Scrum medem sua velocidade

(quantos pontos conseguem realizar em um determinado tempo) e com isso conseguem limitar a quantidade de trabalho em progresso.

Outra característica interessante que diferencia os dois quadros é que as tarefas no quadro do Scrum são movidas durante a iteração de forma que ao final espera-se que todas estejam na última coluna e o restante das colunas fique limpo, enquanto no quadro do Kanban sempre que uma tarefa é movida de coluna uma nova entra em seu lugar, sempre tendo tarefas em todas as colunas (KNIBERG; SKARIN, 2009).

4.2.3 Equipe de Trabalho

Um quadro de tarefas no Scrum pertence a exatamente uma equipe Scrum. Uma equipe Scrum é multifuncional e tem papéis definidos, e esta contém todas as habilidades necessárias para completar todos os itens contidos na iteração. Um quadro de Scrum é geralmente visível a qualquer pessoa que esteja interessada, mas somente aqueles que fazem parte da equipe Scrum podem editá-lo – é a sua ferramenta para gerenciar o seu compromisso para esta iteração. Em Kanban, equipes multifuncionais são opcionais, e o quadro de atividades não precisa pertencer exclusivamente a uma única equipe. Um quadro de atividades está relacionado a um fluxo de trabalho, não necessariamente a uma equipe (KNIBERG; SKARIN, 2009).

4.3 Contratação Ágil nas Organizações Públicas Brasileiras

4.3.1 Iniciativas de Adoção de Métodos Ágeis no Governo

As metodologias ágeis começaram a ganhar espaço no início da década de 2000, regidas pelo Manifesto Ágil, e desde então vêm ganhando crescente popularidade. No cenário mundial, elas já são metodologias bastante difundidas entre diversos setores. Porém, no mercado nacional, essas metodologias são mais conhecidas e implementadas em empresas privadas. Atualmente, diversas organizações públicas brasileiras estão iniciando investimentos em adoção de contratações de fornecedores de *software* utilizando métodos ágeis e, portanto, estão começando a difundir tais métodos também no setor público (MELO et al., 2012).

Recentemente, o Tribunal de Contas da União publicou o Acórdão nº 2314/2013 o qual contém um relatório de levantamento elaborado pela Secretaria de Fiscalização de Tecnologia da Informação (SEFTI) cujo objetivo foi conhecer as bases teóricas do processo de desenvolvimento de *software* com metodologia ágil, além de conhecer experiências práticas de contratação realizadas por instituições públicas federais. As instituições analisadas foram Banco Central do Brasil (BACEN), Instituto do Patrimônio Histórico e Artístico Nacional (IPHAN), Instituto Nacional de Estudos e Pesquisas Educacionais

Anísio Texeira (INEP), Tribunal Superior do Trabalho (TST) e Supremo Tribunal Federal (STF). A seguir será relatada de forma resumida a experiência de cada instituição com o uso de metodologias ágeis em contratações de fornecedores de *software* de acordo com o que foi apresentado no Acórdão nº 2314/2013 (TCU, 2013).

O TST publicou seu primeiro edital para contratação de horas de serviço técnico para desenvolvimento e sustentação de sistemas em 2012. Antes disso, o órgão adquiriu uma experiência interna de desenvolvimento de *software* utilizando o Scrum. A principal característica do edital publicado pelo TST foi: cada *sprint* pode agrupar manutenções de sistemas diferentes e a equipe provavelmente terá que interagir com mais de um *Product Owner*, pois este papel poderá ser representado por mais de uma pessoa. De acordo com o Acórdão nº 2314/2013, supracitado, não foi possível analisar a gestão contratual deste órgão, pois no momento da visita não havia ordem de serviço emitida nesse contrato.

O BACEN também publicou seu primeiro edital para contratação de serviço técnico de TI para desenvolvimento, documentação e manutenção de sistemas em 2012. Antes disso, o órgão já tinha uma experiência interna de desenvolvimento de *software* utilizando métodos ágeis. Os serviços de desenvolvimento e manutenção seguem o Processo de Desenvolvimento de *Software* Ágil (PDS-AGIL) definido pelo BACEN. Esse processo é baseado nas metodologias Scrum e XP, porém, por ser um processo de desenvolvimento desenhado para o uso interno, ele não deixa muito claro quais seriam as atividades específicas de contratação.

O IPHAN teve sua primeira experiência com métodos ágeis no seu edital que foi publicado em 2011. Os serviços de contratação devem ser executados de acordo com a Metodologia IPHAN de Gestão de Demandas de Desenvolvimento Ágil de *Software* (MIDAS) definida pelo órgão. A forma de execução dos serviços prevê o desenvolvimento do projeto em ciclos e cerimônias bem definidas baseado no Scrum. Esta instituição foi escolhida para estudo de caso deste trabalho e mais detalhes sobre a metodologia de gestão de demanda utilizada serão apresentados na próxima seção.

O INEP teve seu primeiro edital com métodos ágeis de contratação de fábrica de *software* para prestação de serviços técnicos de TI em 2011, o qual deveria ser executado segundo a orientação da Metodologia de Desenvolvimento de Sistemas (MGDS) definida pelo INEP, que é baseada em XP e Scrum. O órgão já está no seu segundo contrato utilizando métodos ágeis. A primeira contratação fracassou porque a equipe era composta basicamente de analistas juniores e eles não tinham conhecimento em desenvolvimento de sistemas utilizando métodos ágeis. Com isso, houve uma troca de empresas e a segunda contratação, após o período de adaptação, atendeu o esperado.

O STF inicialmente implantou o desenvolvimento de sistemas utilizando metodologia ágil, baseada no Scrum, em suas equipes internas. O seu primeiro edital para contratação de empresa para prestação de serviços de desenvolvimento ágil de soluções de

software foi publicado em 2012. A forma de execução dos serviços também é baseada no Scrum. Mais informações sobre a contratação não foram possíveis, pois no momento da visita da equipe de fiscalização o órgão não havia iniciado ainda a execução do contrato.

De forma geral, o SEFTI analisou o trabalho desenvolvido nas instituições de acordo com três categorias: métrica utilizada, gestão das demandas e execução dos serviços e níveis de serviços estabelecidos. Quanto a métrica utilizada, a maioria das instituições analisadas utiliza como principal métrica o Pontos por Função, com exceção do TST que utiliza a métrica Horas de Serviço Técnico.

Quanto à forma de gestão de demandas, a maior parte do instrumento convocatório das instituições é semelhante. A demanda para construção do produto é precedida pelo planejamento do produto, o qual pode ser feito apenas pela instituição contratante ou em conjunto com a empresa contratada. Além do planejamento do produto em si, algumas instituições também fazem o planejamento das funcionalidades que serão implementadas no próximo ciclo, iteração ou *sprint*, atividade preceituada no Scrum. Estas instituições emitem uma ordem de serviço por ciclo, iteração ou *sprint*, ou por *release* de *software*, sendo mais comum o primeiro caso.

Quanto à aceitação do produto entregue pela contratada, embora no Scrum seja preceituado que ocorra na reunião de revisão da *sprint*, essa prática não foi executada nos contratos estudados. Isso se deveu a força do normativo, como disciplinado no art. 73 da Lei nº 8.666/1993. Nessa ocasião, algumas instituições apenas verificavam se os artefatos exigidos foram entregues, caracterizando o recebimento provisório.

Acerca da forma de pagamento da contratada, algumas instituições remuneraram os serviços de planejamento, quando realizados, enquanto outras remuneraram apenas os serviços de construção do *software*. A entrega adiantada e contínua de *software*, conforme postulado nos princípios dos métodos ágeis, foi observada em algumas das instituições. Para alcançar esse objetivo, elas paralelizam as atividades de preparação, execução e homologação, ou seja, em um dado período de tempo, enquanto a empresa contratada executa a construção do *software* em um ciclo, a contratante prepara os itens do *backlog* do produto (TCU, 2013).

Quanto à execução dos serviços e níveis de serviço estabelecidos observou-se que quase todas as instituições instituíram vários níveis de serviço a serem atendidos pela contratada. Em suma, o não atendimento a critérios mínimos aceitáveis implica a redução do valor a ser pago à contratada. Dos níveis de serviço analisados, verificou-se a preocupação quase comum das contratantes com o prazo para o cumprimento das iterações, ciclos ou *sprints*, e com a qualidade dos produtos entregues (TCU, 2013).

Além disso, foram identificados riscos na contratação de desenvolvimento de *software* com métodos ágeis pelas instituições da APF. Alguns destes não são restritos so-

mente ao uso de métodos ágeis, podendo ocorrer também com metodologias tradicionais de desenvolvimento de software. Os riscos relativos a processos (R1 a R5), pessoas (R6 a R10) e produtos (R11 a R16) são:

- R1 - Contratação de desenvolvimento de software com adaptação de metodologia ágil que desvirtue sua essência;
- R2 - Alteração da metodologia ágil adotada no instrumento convocatório no decorrer da execução contratual;
- R3 - Ausência de definição dos artefatos ou alteração dos artefatos exigidos da contratada no instrumento convocatório durante a execução contratual.
- R4 - Exigência de artefatos desnecessários ou que se tornam obsoletos rapidamente.
- R5 - Utilização de contrato para desenvolvimento de software por metodologias tradicionais para desenvolvimento por métodos ágeis.
- R6 - Falta de comprometimento ou colaboração insatisfatória do Product Owner no desenvolvimento do software;
- R7 - Falta do conhecimento necessário do Product Owner para o desenvolvimento do software;
- R8 - Excessiva dependência da visão do Product Owner;
- R9 - Equipe da empresa contratada não ter expertise em desenvolvimento de software com métodos ágeis;
- R10 - Dificuldade de comunicação entre a equipe de desenvolvimento da contratada com Product Owner;
- R11 - Alteração constante da lista de funcionalidades do produto;
- R12 - Iniciação de novo ciclo sem que os produtos construídos na etapa anterior tenham sido validados;
- R13 - Falta de planejamento adequado do software a ser construído;
- R14 - 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;
- R15 - Não disponibilização do software em ambiente de produção para a utilização e avaliação dos reais usuários;

- R16 - Forma de pagamento não baseada em resultados.

Na Tab. (2) é apresentada a análise subjetiva do TCU (2013) com relação aos valores ágeis e os princípios da APF. Os auditores afirmaram que por ser um estudo inicial na área de adoção de metodologias ágeis por órgãos públicos, a interpretação dos valores ainda requer um maior detalhamento.

Valor do Manifesto Ágil	Análise segundo Acórdão 2314/2013
Indivíduos e interações acima de processos e ferramentas	Esse valor pode entrar em atrito com o princípio da eficiência por possibilitar que os processos da instituição possam ser relegados e pode contribuir para a construção de uma relação de pessoalidade entre os funcionários da contratada e os gestores da instituição contratante.
<i>Software</i> operante acima de documentações grandes e completas	Embora esse valor possa vir ao encontro do princípio da eficiência ao valorizar a entrega do software em funcionamento mais que documentação, pode também ferir este princípio ao menosprezar a adequada documentação do software contratado, podendo ocasionar problemas para a sua manutenibilidade. Para mitigar esse risco, um conjunto mínimo de artefatos deve ser exigido no instrumento convocatório.
Colaboração do cliente acima de negociações contratuais	Em primeira análise, esse valor entra em atrito com o princípio da vinculação ao instrumento convocatório, uma vez que pode fazer com que a contratada execute serviços não cobertos pelo contrato, ocasionando enriquecimento sem causa da Administração. Em contratações públicas, é imperativo que essa prática seja abolida.
Responder à mudanças acima de seguir um plano	A interpretação inicial desse valor contrasta com o princípio fundamental do planejamento por permitir que a tarefa de desenvolvimento de software se afaste das diretrizes e metas inicialmente estipuladas ou, até mesmo, em extremada circunstância, a elaboração do planejamento seja desprezada. Esse valor pode ainda ferir o princípios de economicidade ao acarretar novos desembolsos ao erário devido a retrabalho para ajustes às mudanças. Entretanto, o valor ágil em epígrafe também vai ao encontro do princípio constitucional da eficiência ao permitir, por exemplo, a incorporação de novos requisitos oriundos de necessidades prementes no software em desenvolvimento.

Tabela 2 – Valores do Manifesto Ágil x Análise segundo Acórdão no 2314/2013

4.3.2 Desafios

Algumas peculiaridades da Administração Pública e o seu arcabouço jurídico obrigatório (Lei nº 8.666/90, Instrução Normativa 04/2010 SLTI.MPOG e acórdãos do TCU) dificultam a adesão de instituições pública aos princípios Ágeis, principalmente em relação à contratação/terceirização de empresas prestadoras de serviços de desenvolvimento de *software*.

Ordens de Serviços, Mensuração do Produto em Pontos por Função ou Horas de Serviços Técnicos, Termo de Recebimento Provisório e Termo Definitivo, Homologações Técnicas e Negociais, Atestes e Notas Fiscais são alguns dos documentos mínimos exigidos pela legislação que o Gestor Público e a Empresa Contratada terão que produzir independentemente se a metodologia adotada é ágil ou não. Somam-se ainda todos os documentos exigidos em cláusulas contratuais.

Como isso, o excesso de documentação poderia ser inadequado aos valores ágeis de *software* funcionando mais que documentação abrangente e responder a mudanças mais que seguir um plano. Ao mesmo tempo, a falta das principais documentações exigidas pode ferir o princípio de eficiência da Administração Pública Federal (APF) e, ainda, se não seguir de forma alguma o que foi planejado, pode-se ferir também os princípios de planejamento e economicidade da APF. (RUAS; VAZ, 2013)

Outro ponto relevante é a relação com equipe de desenvolvimento terceirizada. Em uma equipe externa (terceirizada), o Gestor Público não pode, por lei, interferir na conduta das pessoas contratadas, exceto via o Preposto, pois poderia ferir o princípio de impessoalidade da APF.

Na busca pela compatibilização entre o uso de métodos ágeis e adequação a legislação pertinente, algumas alternativas tem sido adotadas por instituições públicas, em suas iniciativas recentes, no uso de métodos ágeis. São elas: entrega de artefatos de documentação associados ao *software* produzido a cada iteração; relação contratual prevalece sobre a possível colaboração entre as partes; escopo fixo dentro das iterações e níveis de serviços vinculados à qualidade do produto (RUAS; VAZ, 2013).

Com essas alternativas, alguns dos riscos inerentes são mitigados e não há o afastamento da legislação aplicável e ao mesmo tempo estão encaminhadas na direção dos valores e princípios ágeis.

4.4 Considerações Finais do Capítulo

A adoção de metodologias ágeis é um tema que vem ganhando destaque nos últimos anos nas organizações públicas brasileiras, e, particularmente, dentro das contratações de fornecedores de desenvolvimento de software. Neste contexto, tem-se observado que é

necessário adaptar os princípios e as práticas definidas nessa metodologia pra que elas se alinhem, também, aos aos princípios e exigências legais da Administração Pública Federal. No próximo capítulo será apresentado o projeto do estudo de caso desse trabalho que tem como objeto de estudo a solução construída por uma organização pública brasileira que alinhou os princípios e as ferramentas ágeis com as exigências legais existentes na gestão de contratos de fornecedores de desenvolvimento de software.

5 Projeto do Estudo de Caso

Neste capítulo será apresentado o projeto do estudo de caso resultante da fase de Planejamento do Estudo de Caso, o qual contém a definição da unidade de análise, a identificação do problema, a definição das questões de pesquisa (geral e específicas) e a definição dos objetivos (geral e específicos), assim como considerações sobre as fontes e métodos de coleta de dados e sobre a validade do estudo de caso.

5.1 Definição

Neste trabalho foi apresentado um arcabouço teórico relacionado à contratação de serviços de TI em organizações públicas brasileiras. Também foram apresentados os principais conceitos *Lean* na Manufatura, *Lean* no Desenvolvimento de *Software* e Metodologias Ágeis.

Com isso, o escopo do estudo de caso está ilustrado na Fig. 13.



Figura 13 – Escopo do Estudo de Caso

Assim, a proposta deste trabalho consiste na investigação, coleta, análise e discussão dos resultados de dados da gestão de contrato de um fornecedor de desenvolvimento de *software* para uma organização pública brasileira. O foco será a análise dos efeitos sobre a entrega de ordens de serviço, a satisfação do cliente e a qualidade interna do código da solução desenvolvida pela organização, a qual é alinhada com os métodos ágeis, com o pensamento *lean* e com a fase de Gerenciamento do Contrato da Instrução Normativa nº 04.

Para tanto, foi estruturado neste trabalho problema, questões de pesquisa e objetivos conforme ilustrado na Fig. 14, os quais guiaram o pesquisador durante a fase de coleta de dados.

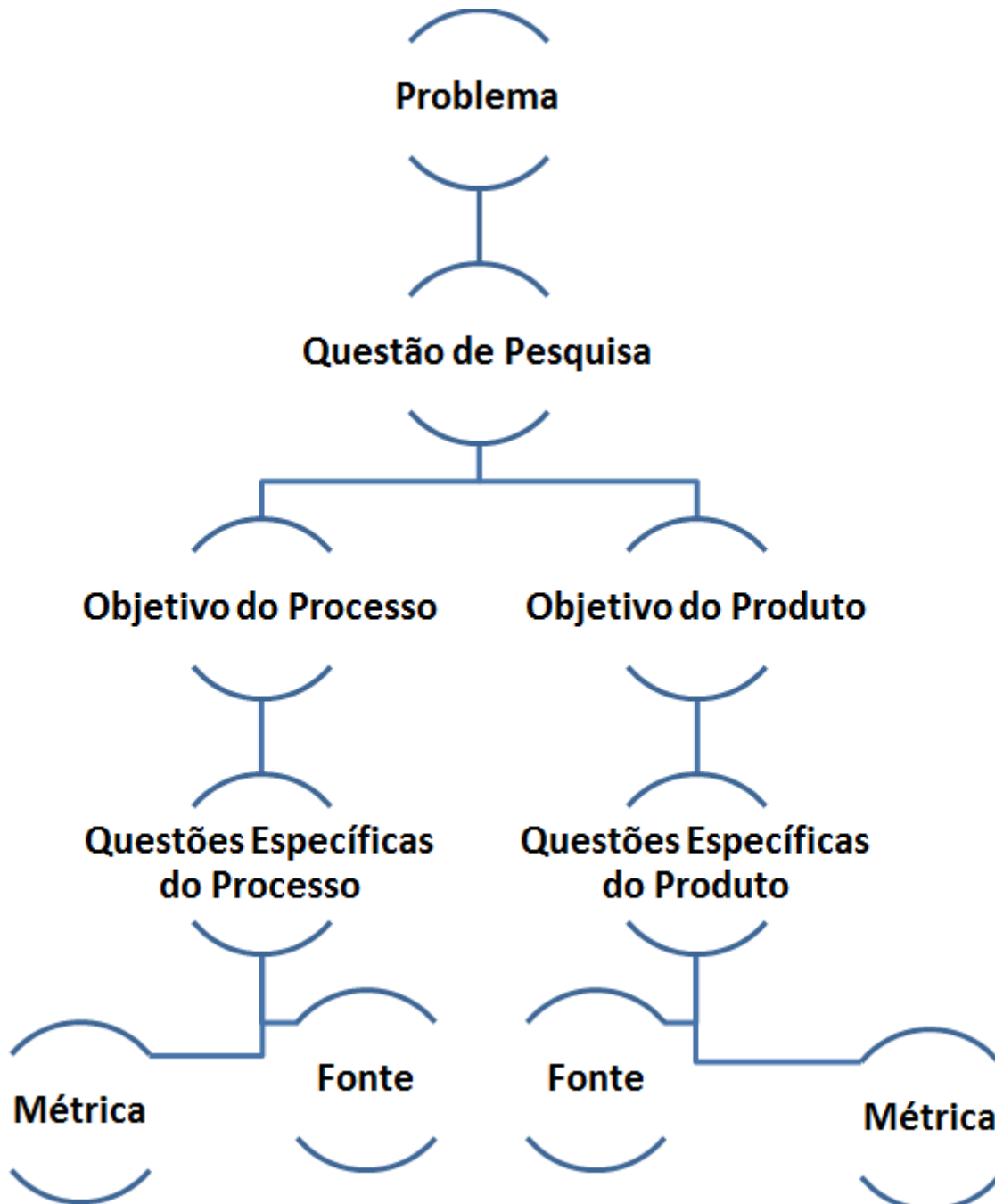


Figura 14 – Estrutura do Estudo de Caso

O problema refere-se ao problema de pesquisa identificado. A questão de pesquisa refere-se a questão de pesquisa que foi derivada do problema. Para responder a questão de pesquisa foram construídos dois GQM's (Goal, Question, Metric) (BASILI; ROMBACH, 1994). Cada GQM possui um objetivo e questões específicas para coleta de métricas a

partir de determinada fonte, a fim de atingir o objetivo e responder a questão de pesquisa do trabalho. Ao todo foram definidas doze questões de pesquisa específicas que serão analisadas no estudo de caso. A definição dessa estrutura está a seguir.

Problema: Alguns contratos de desenvolvimento de *software* da organização não resultaram na entrega do *software* requisitado ao final do contrato.

Questão de Pesquisa: Como o uso de métodos ágeis e do pensamento *lean* na gestão de contratos de fornecedores de desenvolvimento de *software* influenciaram no resultado final do contrato do ponto de vista do gestor de contrato e do fiscal técnico do contrato, que juntos gerenciam o contrato?

OE1. Objetivo do Processo: Analisar a influência do uso de métodos ágeis e do pensamento *lean* na gestão de contrato do contrato do Sistema Integrado de Conhecimento e Gestão (SICG) com a empresa EGL - Engenharia do ponto de vista do gestor de contrato.

Questões Específicas do Processo:

QE1. Qual a quantidade total de ordens de serviço?

Fonte: Documentação

Métrica: quantidade total de ordens de serviço.

QE2. Qual a quantidade de ordens de serviço que tiveram entrega de *software* funcional?

Fonte: Documentação

Métrica: quantidade de ordens de serviço de *software*.

QE3. Qual a quantidade de ordens de serviço que tiveram entrega apenas de documentação?

Fonte: Documentação

Métrica: quantidade de ordens de serviço de documentação.

QE4. Qual a proporção de entrega de *software* funcional?

Fonte: Documentação

Métrica: quantidade de ordens de serviço de *software* que tiveram entrega de *software* funcional / quantidade total de ordens de serviço.

QE5. Qual a duração média de entrega de *software* funcional?

Fonte: Documentação, scrum master, dono do produto, gestor do contrato, fiscal

técnico do contrato, coordenador do projeto.

Métrica: duração total das sprints/duração total das sprints que tiveram entrega de *software* funcional.

QE6. Qual a quantidade de ordens de serviço que não teve entrega de *software* funcional e de documentação?

Fonte: Documentação

Métrica: quantidade de ordens de serviço sem entrega de *software* e de documentação.

QE7. Qual a porcentagem de requisitos atendidos em cada ordem de serviço?

Fonte: Documentação, dono do produto, gestor do contrato, fiscal técnico do contrato, coordenador do projeto.

Métrica: (requisitos atendidos/requisitos pedidos) * 100.

QE8. Quantas multas foram aplicadas no contrato?

Fonte: Documentação, scrum master, dono do produto, gestor do contrato, fiscal técnico do contrato, coordenador do projeto.

Métrica: quantidade de multas.

QE9. Qual o custo de cada sprint e ordem de serviço do projeto?

Fonte: Documentação

Métrica: custo/sprint.

QE10. O quanto de visibilidade do que estava sendo feito o gestor do contrato teve durante o contrato?

Fonte: Dono do produto, gestor do contrato, fiscal técnico do contrato, coordenador do projeto.

Métrica: alto, médio ou baixo.

QE11. Qual o nível de satisfação com o *software* entregue ao final do contrato?

Fonte: Dono do produto, gestor do contrato, fiscal técnico do contrato, coordenador do projeto.

Métrica: muito satisfeito, satisfeito, neutro, insatisfeito, muito insatisfeito.

OE2. Objetivo do Produto: Analisar a qualidade do código fonte com o uso de métodos ágeis e do pensamento *lean* na gestão do contrato do Sistema Integrado de Conhecimento e Gestão (SICG) com a empresa EGL - Engenharia do ponto de vista do e do fiscal técnico do contrato.

Questão Específicas do Produto:

QE12. Qual a qualidade interna do produto entregue até o momento ?

Fonte: Código, scrum master, desenvolvedores, dono do produto, gestor do contrato, fiscal técnico do contrato, coordenador do projeto.

Métrica: bom, excelente, regular e ruim.

5.2 Background

À luz do levantamento bibliográfico realizado, não foram encontrados estudos que buscassem levantar e analisar os efeitos advindos do uso de metodologias ágeis no contexto da gestão de contratos de fornecedores de desenvolvimento de *software* para organizações públicas brasileiras. O Acórdão nº 2314/2013, relatado no Capítulo 4, traz apenas um levantamento do que foi feito de uso de métodos ágeis pelos órgãos analisados. No entanto, não houve uma coleta e análise de dados, tal qual realizada neste trabalho.

5.3 Design

Uma das formas de classificar um estudo de caso é com base na identificação de seus propósitos gerais: exploratórios, descritivos, explicativos e avaliativos. Este estudo de caso é classificado como exploratório, pois não esperamos obter uma resposta definitiva para o problema proposto. A ideia é obter uma visão mais acurada do problema para posteriormente realizar uma pesquisa mais aprofundada. Tal escolha foi feita porque o tema escolhido foi pouco explorado até o momento, constituindo apenas a primeira etapa de uma investigação mais ampla e sistemática.

Uma segunda forma de classificação é relacionada a quantidade de casos: caso único ou casos múltiplos. Considerando as restrições de tempo para realização deste estudo de caso, foi definido o estudo de um caso único (uma organização) e holístico, com uma unidade de análise (um contrato), pois, como mencionado anteriormente, este é um estudo de caso exploratório. Posteriormente, pretende-se ampliar este estudo para múltiplos casos.

5.4 Seleção

A organização selecionada para este estudo de caso foi o Instituto do Patrimônio Histórico e Cultural (IPHAN). Dentre as organizações analisadas no relatório do Acórdão nº 2314/2013, o IPHAN foi a primeira a adotar o uso de métodos ágeis e pensamento *lean* na gestão de seus contratos. O BACEN, foi o primeiro a buscar a implantação de métodos ágeis. No entanto, foi no contexto de desenvolvimento interno de *software* do órgão. Assim, a seleção justifica-se porque o órgão possui o primeiro contrato gerenciado com métodos ágeis dentre os contratos das organizações públicas localizadas em Brasília.

5.5 Fonte e Método de Coleta de Dados

Os dados foram coletados por meio de entrevistas informais, observações, questionários, análise de documentos dos processos da organização e da base de código fonte de um contrato disponibilizado pelo órgão: o contrato do Sistema Integrado de Conhecimento e Gestão (SICG) com a empresa EGL - Engenharia.

Os questionários tinham o objetivo de coletar dados qualitativos e quantitativos a respeito da organização contratante, do gestor de negócio (cliente) e da empresa contratada no que diz respeito a estrutura organizacional, experiência prévia, satisfação, opiniões, percepções e etc. O dados de observação e entrevistas complementaram os questionários sob o ponto de vista qualitativo.

Os dados quantitativos sobre a execução do processo (solução) foram coletados das 25 ordens de serviço documentadas do projeto. Já os dados da qualidade do código fonte foram coletados de 19 sprints do projeto.

Documentos

- Processo nº 01450.011592/2010-30, cujo assunto é Contratação de Serviços de Desenvolvimento de *Software* para o Sistema Integrado de Conhecimento e Gestão (SICG).
- Processo nº 01450.000845/2012-10, cujo assunto é Gestão de Contrato IPHAN Nº 28/2011 - Desenvolvimento de *Software* para o Sistema Integrado de Conhecimento e Gestão (SICG).
- Código Fonte do Sistema Integrado de Conhecimento e Gestão (SICG).
- Backlog do Produto do Sistema Integrado de Conhecimento e Gestão (SICG).

Questionários

- Objetivo: coletar informações dos envolvidos no projeto por parte do IPHAN, fiscal técnico do contrato, coordenador do projeto e do gestor do contrato, que também representa o papel de dono do produto, e dos envolvidos no projeto por parte da empresa contratada, a EGL - Engenharia, scrum master e desenvolvedores. O questionário aplicado aos envolvidos no projeto pode ser encontrado no Apêndice A.
- Amostra: foi aplicado o questionário para todos da área de sistemas do IPHAN (3 pessoas) e para 3 integrantes da equipe da EGL-Engenharia.

5.6 Validade

As principais ameaças aplicáveis aos estudos de caso são mencionadas por Yin (2010). Dentre elas, estão a validade do constructo, validade interna, validade externa e confiabilidade.

A validade de constructo diz respeito ao uso de múltiplas fontes de evidência e seleção de informantes chave. Ela ocorre na fase de coleta de dados e está relacionada ao desenvolvimento de um conjunto de métricas para que se saiba exatamente o que medir e quais dados são relevantes para o estudo, de forma a responder as questões de pesquisa. Como apresentado anteriormente, neste trabalho foram utilizadas múltiplas fontes de evidências e os informantes que responderam ao questionário e às entrevistas foram os principais envolvidos no projeto, tanto do órgão contratante quanto da empresa contratada. Além disso, a utilização da técnica GQM direcionou a definição de métricas, procurando-se então a reforçar a validade do constructo.

A validade interna pode ser atingida por meio de diversas estratégias, dentre elas a construção da explanação. Se as conclusões apresentadas pelo estudo de caso correspondem autenticamente a alguma realidade reconhecida pelos próprios envolvidos no projeto, há uma validade interna. Para Yin (2010), o uso de várias fontes de dados e métodos de coleta permite a triangulação, uma técnica para confirmar se os resultados de diversas fontes e de diversos métodos convergem. Dessa forma é possível aumentar a validade interna do estudo e aumentar a força das conclusões. Nesta pesquisa houve triangulação de dados e de metodologia. A triangulação de dados se deu pelo uso de base de documentos e código organizacionais, questionários e entrevistas para coletar dados. A triangulação de métodos ocorreu pelo uso de métodos de coleta quantitativos e qualitativos.

A validade externa diz respeito a estabelecer o quanto as descobertas podem ser generalizadas. Pode-se testar a coerência entre os resultados do estudo e resultados de outras investigações similares. Para isso, Yin (2010) recomenda a replicação do estudo em múltiplos casos. Como o escopo deste trabalho é o estudo de um único caso, a validade externa não poderá ser atingida.

E, por fim, a confiabilidade diz respeito a mostrar que se o estudo for repetido utilizando as mesmas fontes de dados, poderá obter-se os mesmos resultados. Para isso, a definição do protocolo do estudo de caso e o desenvolvimento de um banco de dados do estudo são táticas para garantir a confiabilidade. Nesse trabalho, o protocolo de estudo de caso é evidenciado nas seções deste capítulo e no questionário, que pode ser encontrado no Apêndice A.

5.7 Trabalho de Campo

O trabalho de campo é parte fundamental do estudo de caso. Por meio dele é que os dados para análise serão coletados. Neste trabalho, as entrevistas informais e a triagem dos processos foram realizadas em campo. Os processos disponibilizados pelo órgão eram extensos, mais de 4000 páginas. Por isso, a triagem sobre esses processos foi realizada para solicitação de cópias das partes que mais interessavam para o contexto e objetivos de pesquisa deste estudo.

5.8 Análise

O processo de análise de dados foi composto por três etapas:

- **Redução:** refere-se ao processo de seleção, categorização, simplificação e transformação dos dados obtidos, reduzindo-os ao essencial para proporcionar sua análise e interpretação. Os dados qualitativos foram categorizados em: organização, empresa contratada, projeto e solução. Os dados quantitativos foram categorizados de acordo com o tipo de questão específica, e as categorias são: entrega de ordens de serviço, satisfação do cliente e qualidade interna do código fonte.
- **Exibição:** consiste na organização dos dados selecionados de forma a possibilitar a análise sistemática das semelhanças e diferenças e seu inter-relacionamento. Os dados foram exibidos, principalmente, por meio de histogramas, tabelas e texto.
- **Conclusão e verificação:** consiste em identificar o significado dos dados, suas regularidades, padrões e explicações. A medida que os resultados foram exibidos, uma discussão e interpretação inicial foi realizada. Posteriormente, na conclusão final foram agrupados os resultados mais relevantes.

5.9 Considerações Finais do Capítulo

Este capítulo consistiu em apresentar o projeto de estudo de caso utilizado como base para coleta de dados e análise. No próximo capítulo serão detalhados a organização,

a empresa contratada, o objeto do contrato e a solução analisada no estudo de caso e serão apresentados os resultados obtidos para as questões específicas apresentadas neste capítulo assim como a interpretação dos mesmos.

6 O Estudo de Caso

Neste capítulo será apresentado o estudo de caso desse trabalho. Na primeira seção será apresentado o contexto e a caracterização da organização do estudo de caso. Na segunda seção será apresentada a empresa contratada para o desenvolvimento do projeto. Na terceira seção será apresentada a caracterização do contrato e seu objeto, ou seja, a caracterização do projeto do qual os dados foram coletados. Na quarta seção será apresentada a solução aplicada na gestão do contrato e será analisada sua conformidade com o pensamento *lean*, métodos ágeis e a legislação atual. E, finalmente, na quinta seção serão apresentadas a análise dos dados quantitativos e a discussão dos resultados.

6.1 A Organização

O organização escolhida, IPHAN, uma autarquia pública federal, possui uma força de trabalho atuante na área de TI de apenas 8 servidores, dos quais apenas 4 trabalham diretamente com sistemas. O perfil dessa equipe é apresentado na Tab. (3). Devido ao número reduzido de servidores disponíveis na área de TI do órgão, frequentemente uma mesma pessoa acaba desempenhando diferentes papéis requeridos pela IN 04/2010.

Área	Perfil	Quantidade
TI Geral	Coordenador de Tecnologia da Informação	1
Infraestrutura	Analista de Tecnologia da Informação do MP	2
Sistemas	Analista de Tecnologia da Informação do MP	2
Apoio	Analista de Tecnologia da Informação do MP	1
	Servidor do Ministério da Ciência, Tecnologia e Inovação	1
	Servidor do IPHAN	1

Tabela 3 – Perfil da Equipe do IPHAN

O contexto atual do órgão foi identificado por meio da aplicação da técnica de entrevista informal, de questionário e da documentação disponível do órgão. O questionário aplicado pode ser encontrado no Apêndice A. Vale ressaltar que esse questionário foi aplicado não só para caracterização da organização como também para coleta de dados quantitativos necessários para o estudo de caso deste trabalho e ele foi projetado de forma generalizada para que tanto o IPHAN quanto a empresa contratada, EGL - Engenharia Ltda, pudessem respondê-lo.

Os fatores mais significantes que são gerenciados pela área de TI do órgão são:

- Atender as demandas para desenvolvimento de sistemas (sistema novo, manutenção, documentação).
- Controlar a qualidade dos sistemas.
- Possuir medições de sistemas.

Por meio do questionário foram identificados fatores que motivaram o uso de métodos ágeis na gestão do contrato do órgão. A Tab. (4) mostra a porcentagem dos fatores motivacionais citados pelos servidores da área de sistemas do órgão.

Fator Motivacional	Porcentagem
Documentação desnecessária estava sendo produzida e entregue	100%
Entrega de <i>software</i> funcional era pouco frequente nos contratos anteriores	66%
Visibilidade do processo era baixa	66%
O fiscal técnico do contrato e o gestor do contrato não estavam exercendo os seus papéis como deveriam	66%
A qualidade do produto era preocupante	66%
Os requisitos de <i>software</i> não eram atendidos	66%

Tabela 4 – Fatores motivacionais

As metas norteadoras para a elaboração do processo de gestão de contrato com métodos ágeis foram:

- Ser aderente à legislação pertinente;
- Entregar *software* mais rapidamente;
- Focar na gestão do contrato e na definição de uma metodologia de gestão de demandas;
- Não focar em dizer como a empresa deveria desenvolver o *software*, ou seja, não definir metodologia de desenvolvimento de *software*;
- Satisfazer as necessidades do cliente.

Com isso, o IPHAN optou por definir uma metodologia ágil de gestão de contrato para o controle do órgão sobre as demandas e entregas de ordens de serviço. O IPHAN não definiu uma metodologia de desenvolvimento de *software*, ou seja, o órgão não determina que a empresa siga uma metodologia ágil de desenvolvimento de *software*, é necessário apenas que a empresa se adeque as restrições impostas independente de qual metodologia de desenvolvimento ela irá utilizar.

Outros instrumentos contratuais que foram modificados com a adoção do processo de gestão de contrato com métodos ágeis foram a forma de pagamento e a aplicação de multas. Em relação a esta, diferentemente das outras formas de gestão de contrato utilizadas anteriormente, onde as multas eram progressivamente aplicadas, por exemplo, sobre a não entrega de documentação, detinha caráter meramente punitivo. Com o novo processo, passou-se a considerar o maturidade e crescimento da empresa no contrato e se não houvesse entrega de *software*, a empresa contratada não recebia o pagamento estimado para aquela ordem de serviço. O faturamento das ordens de serviço era executado a cada entrega, final de *sprint*, o que mantinha o fluxo de caixa da empresa contratada sempre ativo.

6.2 A Empresa Contratada

A empresa contratada para desenvolvimento do projeto foi a empresa EGL - Engenharia Ltda. Formada por uma equipe multidisciplinar, a EGL Engenharia, desenvolve sistemas de informação geográfica e de apoio a gestão administrativa jurídica, privada ou pública. É capacitada a criar sistemas customizados para atender as necessidades específicas, além de fornecer manutenção corretiva e evolutiva destes.

A EGL - Engenharia Ltda possui entre 7 e 9 envolvidos no desenvolvimento do projeto. O perfil dessa equipe é apresentado na Tab. (5). É importante ressaltar que os perfis são apenas os papéis que cada integrante da equipe podia assumir, alguns integrantes eram mais especialistas em determinadas áreas que outro, no entanto, toda a equipe participou do processo de desenvolvimento do sistema. Portanto, a equipe não é uma equipe especialista, mais sim generalista, característica comum de equipes ágeis.

Perfil	Quantidade
Scrum Master	1
Desenvolvedor	2
Analista de Requisitos	1
Analista de Sistemas	1
Arquiteto de Sistemas	1
Analista de Georreferenciamento	1
Preposto	1

Tabela 5 – Perfil da Equipe da EGL - Engenharia Ltda

No desenvolvimento do projeto a empresa utilizou a metodologia *Scrum* e a linguagem de programação Java. No que diz respeito a experiência da equipe com essa metodologia, houve uma grande variação nas respostas do questionário, existindo cerca de 33% dos integrantes com pouca experiência, cerca de 33% dos integrantes com experiência regular e cerca de 33% integrantes com alta experiência. No que diz respeito a

experiência da equipe com a linguagem de programação, 100% dos desenvolvedores tinham uma alta experiência.

A lista de práticas ágeis utilizadas para o desenvolvimento do projeto são apresentadas na Tab. (6).

Padrões de Codificação.	Revisão de Código.
Refatoração de Código.	Testes de Fumaça (Smoke Testing), de Integração e de Aceitação.
Programação em Par.	Histórias de Usuário.
Integração Contínua.	Planning Poker.
Controle de Versão.	Planejamento das Iterações.
Entregas Frequentes.	Backlog do Produto e da <i>Sprint</i> .
Código Limpo.	Quadro Kanban.
Burndown Charts.	Retrospectivas e Reunião Diária.
Equipes Auto-organizadas.	Times Pequenos.

Tabela 6 – Práticas Ágeis adotadas pela EGL - Engenharia Ltda

6.3 Caracterização do Projeto do Contrato

6.3.1 Visão Geral

O IPHAN é responsável pela gestão de diversos processos de preservação do patrimônio cultural, como por exemplo, ações para sua identificação, proteção, gestão e fomento. Decorrente de suas atribuições, o órgão produz uma grande quantidade de informações fragmentadas em termos territoriais e temáticos. Nos quatro anos anteriores ao contrato do SICG, o IPHAN elaborou uma metodologia para definir os processos de cadastro, inventário e gestão do patrimônio cultural material. Essa metodologia tinha por objetivo geral abordar o Patrimônio Cultural de forma integrada, sistêmica e estratégica, conforme detalhado a seguir:

- Integrada: cobrindo todas as categoriais do patrimônio material;
- Sistêmica: estabelecendo moldes a serem utilizados nas diversas etapas de ações de preservação, possibilitando o "diálogo" e troca de informações entre áreas e etapas de trabalho;
- Estratégica: considerando o mapeamento, a organização e a disponibilização de informações sobre o patrimônio como base para a construção de políticas públicas integradas - com outros parceiros - e de planos de preservação e desenvolvimento das regiões onde se inserem os bens.

Em termos específicos, a metodologia buscou, em primeiro lugar, mapear os procedimentos necessários para a execução das ações de cadastramento, proteção, normatização

e fiscalização de bens culturais de natureza material, indicando adicionalmente os dados a serem coletados. Este mapeamento contou com a participação de representantes das Superintendências e Escritórios Técnicos, que, por meio de Grupos de Trabalho, analisaram, de forma crítica, as metodologias até então existentes.

A revisão dos processos levou à formulação da nova metodologia que, por sua vez, permitiu a otimização das atividades de cadastramento de sítios históricos e de bens tombados isoladamente e gerou a normalização das ações de fiscalização. O resultado desse trabalho produziu um conjunto de fichas e procedimentos específicos com demandas para cadastramento de dados textuais, geográficos e imagens.

Havia um entendimento no IPHAN de que era necessária a formação de uma rede de proteção fomentada pelo Sistema Nacional do Patrimônio Cultural (SNPC) que consolidasse o grande volume de informações atualmente produzido por suas unidades administrativas, composto de 27 Superintendências, 30 escritórios técnicos, 4 Unidades Especiais e 2 Parques Históricos Nacionais. Entretanto, a natureza das informações, em grande maioria armazenadas em planilhas e em banco de dados isolados, dificultava o processo de consolidação das informações. Esse fato frustrava os esforços para a construção da rede de proteção baseada nos recursos e tecnologias atualmente adotados, pois demandava o aporte considerável de recursos financeiros e humanos sem ganhos no processo. O órgão se mantinha refém da demora na produção de informações decorrente do intervalo entre ação e recepção das respostas, ou seja, entre a percepção do problema e sua solução.

Apesar do fato de os processos de cadastramento, normatização de sítios urbanos tombados e fiscalização de bens imóveis de metodologia já fazerem parte da realidade das Superintendências e Escritórios Técnicos, o processo manual de suporte e gestão dos dados tornava a execução precária e morosa.

A Tabela 7 apresenta a visão geral da situação vivida pela IPHAN, antes da solução informatizada, a qual subsidiava a contratação da solução informatizada.

O que o IPHAN tinha?	O que o IPHAN não tinha?
Processos de Cadastramento, normatização, tombamento e fiscalização revisados.	Processos automatizados.
Organização territorial sobre o conteúdo cultural (necessidade de vinculação entre o bem cultural e o território e localização física).	Solução informatizada com inteligência geográfica.
Cultura gerencial sobre os processos de trabalho.	Suporte informatizado para tomada de decisão.
Grande volume de dados produzidos ao longo dos últimos 3 anos de implantação da nova metodologia.	Ferramentas adequadas para gestão e disponibilização dos dados existentes.
Grande volume de dados produzidos por outras metodologias de cadastramento, normatização de sítios urbanos e fiscalização.	Ferramentas e procedimentos automatizados adequados para a migração dos dados existentes.

Tabela 7 – Visão geral da situação do IPHAN

Portanto, a solução informatizada era apontada como um caminho adequado para endereçar as dificuldades citadas, pois, possibilitaria a execução de vários processos do IPHAN (cadastramento, normatização, fiscalização e planejamento) de maneira integrada, garantindo agilidade na produção da informação.

Com isso, o sistema que foi proposto, o SICG, foi formado por 7 módulos:

- Módulo 1 - Conhecimento;
- Módulo 2 - Análise e Gestão;
- Módulo 3 - Cadastro;
- Módulo 4 - Administração de Usuários;
- Módulo 5 - Fiscalização;
- Módulo 6 - Cadastro Auxiliares;
- Módulo 7 - Relatórios Adicionais.

6.3.2 Objetivos da Contratação

Os objetivos da contratação eram:

- Automatizar o processo de trabalho decorrente da metodologia de inventário, cadastramento, análise e gestão do patrimônio material, denominado Sistema Integrado de Conhecimento e Gestão (SICG);
- Centralizar as diversas bases de informações utilizadas pelo Departamento de Patrimônio Material e Fiscalização (DEPAM) em uma base de dados única, possibilitando integração futura com bases de dados de outros departamentos do IPHAN, o problema atual de falta de integração, da dispersão e da redundância dos dados;
- Possibilitar a descontinuidade gradativa de 100% das bases de dados isoladas, de caráter local ou descentralizado existentes no DEPAM;
- Mapear as informações produzidas pelas diversas coordenações do DEPAM, pelas Superintendências e Escritórios Técnicos, pelos parceiros do Sistema Nacional de Patrimônio Cultural (SNPC), e as informações presentes no Arquivo Central do IPHAN, por meio de dados geográficos convergentes;
- Possibilitar a realização de análises territoriais e de uma visão territorial das ações do DEPAM, provendo atributos de tamanho, proximidade, área, localização, topologia e outros, por meio de dados georreferenciados;
- Consolidar informações originadas nas diversas coordenações do DEPAM, pelas Superintendências e Escritórios Técnicos, pelos parceiros do SNPC e uma base de dados geográfica de alta escalabilidade e que contenha 100% das informações georreferenciadas;
- Garantir celeridade no processo decisório, por meio da entrega de informações consistentes, precisos e que sejam entregues em tempo hábil, evitando ônus para a administração pública;
- Possibilitar a redução do tempo e esforço necessário para a compilação das informações do DEPAM, advindas da recuperação e do cruzamento de produção das unidades do IPHAN de todo o país. Espera-se que o tempo médio de resposta para compilação das informações seja reduzido de 30 dias para resposta em tempo real.

6.3.3 Metodologia de Trabalho

Forma de Encaminhamento das Ordens de Serviço

Toda ordem de serviço era repassada pessoalmente ao preposto da contratada.

Ciclos

Todo o sistema foi desenvolvido de forma iterativa e incremental, por meio de ciclos, ao longo de toda a vigência contratual, até sua conclusão total. O desenvolvimento

e, por consequência, o repasse de conhecimento à contratada será feito por ciclos de planejamento e reuniões de levantamento de requisitos e aprendizado. Cada ciclo tem duração de 2 a 4 semanas.

O Ciclo de Planejamento (*Sprint 0*) tinha como objetivo o planejamento de todo o projeto.

Os Ciclo de Desenvolvimento (*Sprints* de 1 a 24) tinha como objetivo o desenvolvimento do sistema.

O Ciclo de Implantação (Penúltima *Sprint*) tinha como objetivo a implantação do sistema.

O Ciclo de Encerramento (Última *Sprint*) tinha como objetivo o treinamento dos seus usuários e o encerramento do projeto.

Reuniões

Cada ciclo teve uma Reunião de Planejamento no início do mesmo e uma Reunião de Encerramento ao final do mesmo.

Na Reunião de Planejamento, a Contratada apresentou à Contratante uma proposta de Ordem de Serviço para o ciclo em planejamento. A Contratante emitiu parecer sobre a proposta. Aprovada a proposta, a Contratante, por meio de Ordem de Serviço, autorizou a Contratada a executar o ciclo planejado.

Na Reunião de Encerramento, a Contratada entregou e apresentou à Contratante o conjunto de produtos resultantes da execução do respectivo ciclo.

Os produtos entregues pela Contratada tinham um prazo de no máximo um ciclo para serem homologados pela Contratante.

6.4 Caracterização da Solução

Para desenvolver uma metodologia de gestão de contratos *software* baseada no Scrum e no Pensamento *Lean*, o IPHAN definiu alguns procedimentos que deveriam ser feitos com o foco na minimização dos riscos da execução contratual e na obtenção do sucesso no contrato de terceirização. O *framework* utilizado não foi considerado o mais relevante, mas sim os valores e princípios do Manifesto Ágil, além do atendimento à legislação vigente.

As metodologias ágeis foram utilizadas como o meio para atingir o sucesso ou para identificar de forma rápida os riscos iminentes. O sucesso contratual pode ser entendido como aquele contrato que atende às necessidades do órgão, com sistemas, sem comprometer o erário (tesouro público). Assim, para atingir sucesso em um contrato é preciso que pelos menos esses três procedimentos sejam realizados:

- Definir premissas nos artefatos desde o planejamento da contratação;
- Alinhar diretrizes e condições com a Direção de TI;
- Convalidar com a Alta Administração, ou seja, validar e sustentar essas diretrizes durante o contrato.

A lista de práticas ágeis utilizadas no projeto SICG para a gestão do contrato estão na Tab. (8).

Definição de Pronto	Histórias de Usuário.
Planejamento das Iterações.	Backlog do Produto.
Backlog da <i>Sprint</i> .	Quadro Kanban.
Burndown Charts.	Retrospectivas.
Tempo de Ciclo (Lead time).	Revisão de Código.

Tabela 8 – Práticas Ágeis adotadas pelo IPHAN

Com isso, foram definidas algumas premissas que orientaram o planejamento e execução do contrato. A saber:

- O IPHAN não deveria definir, ou exigir, o uso de Metodologia Ágil da entidade contratada. Não defina Metodologia de Desenvolvimento de *Software* (MDS), mas sim a forma de gerenciar as demandas (ordens de serviço), os produtos que devem ser entregues e seus critérios de aceitação.
- A recontagem de Pontos por Função nos moldes do roteiro do SISP com metodologia ágil que pode mudar constantemente é um risco. É preciso alterar o percentual definido para a alteração, manutenção ou refatoração de uma funcionalidade, definir corretamente o conceito de manutenção evolutiva, refatoração e alteração de requisito e evidenciar no processo o custo de uma alteração e fazer com que o gestor negocial, que pediu a alteração, assine a ordem de serviço e ateste a nota fiscal;
- Só abra uma Ordem de Serviço (OS) por vez e por projeto. Pode-se ter várias OS abertas com a mesma Contratada, porém, será uma OS para cada projeto e uma OS por *Sprint*. Além disso, nunca comece oficialmente a próxima demanda sem receber ou finalizar a demanda anterior. Caso uma OS não estiver atendendo o que foi solicitado, ou por uma mudança negocial essa OS não for necessária, cancele-a e abra outra ordem de serviço que atenda a nova exigência do gestor contratual;
- Não gerencie atrasos ou defeitos. No fim da *Sprint*, receba o que estiver pronto, mesmo que não seja tudo que foi solicitado. Se nada foi entregue é uma ausência de entrega, não existe atraso, a *Sprint* é considerada perdida. O produto não entregue

ou com defeito dever voltar para fila de demandas e entrará na próxima OS ou *Sprint* se o gestor negocial a quiser novamente, nunca aceite que corrijam um produto com defeito dentro da mesma OS;

- Entenda a demanda antes de executá-la. É preciso planejar, pelo menos, com quantas ordens de serviço o projeto será validado, qual o processo de negócio que será desenvolvido, como será feita a gestão de demandas e qual será a demanda da próxima *Sprint* ou OS;
- Não aceite documentos sem sistemas. É importante ter em mente que não deve-se aceitar entregas apenas de documentação sem um produto funcional;
- Acredite na evolução da empresa. No começo, a empresa contratada poderá não conseguir entregar o que foi solicitado ou entregar um produto funcional, no entanto, progressivamente ela irá se adequar ao processo e evoluir.

A partir dessas premissas definidas, a autarquia construiu um Kanban para auxiliar no processo de gestão de contratos.

O Kanban definido pelo IPHAN possui quatro colunas ou raias e está ilustrado na Fig. (15)¹.

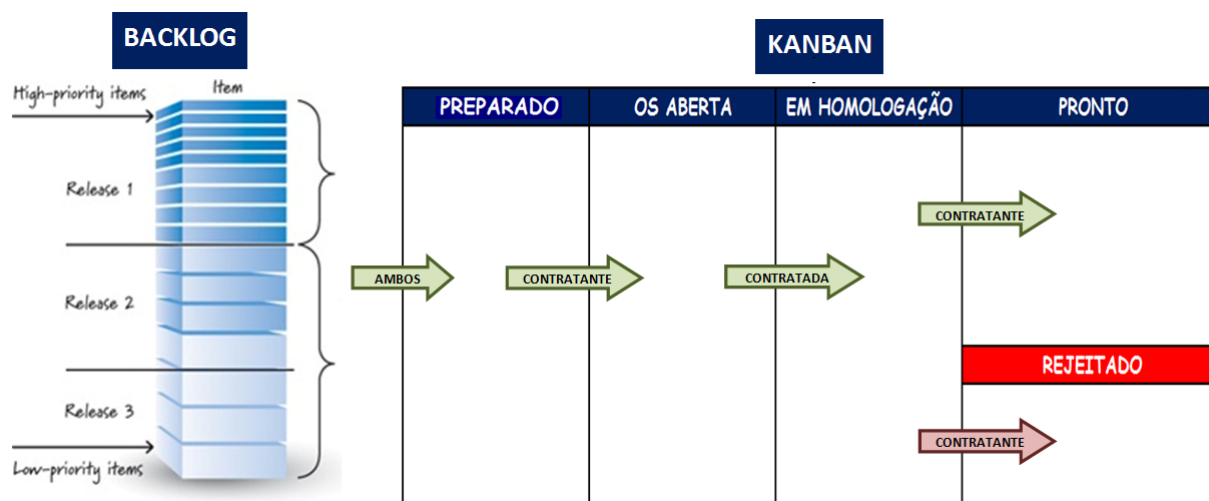


Figura 15 – Quadro Kanban

A primeira raia do Kanban diz respeito aos itens que estão no estado “Preparado”. A condição de transição para esta raia pode ser feita da forma que o órgão quiser, Fig.

¹ Imagem disponível em <http://www.slideshare.net/herbertparente/contratao-de-fabrica-de-software-metodologia-gil>

(16)². Por exemplo, os itens com mais prioridade podem ser os primeiros a irem para esta coluna. É importante que a definição de “Preparado” e a definição de “Pronto” estejam bem claras para todos os envolvidos.



Figura 16 – Transição para a raia Preparado

A transição de um item da raia “Preparado” para a raia “OS Aberta” ocorreu na abertura de uma ordem de serviço (Fig. (17))³. Esta transição ocorreu após o planejamento da *sprint*, onde uma ordem de serviço de desenvolvimento era aberta com os itens que devem ser desenvolvidos para aquela *sprint* e o desenvolvimento é iniciado.

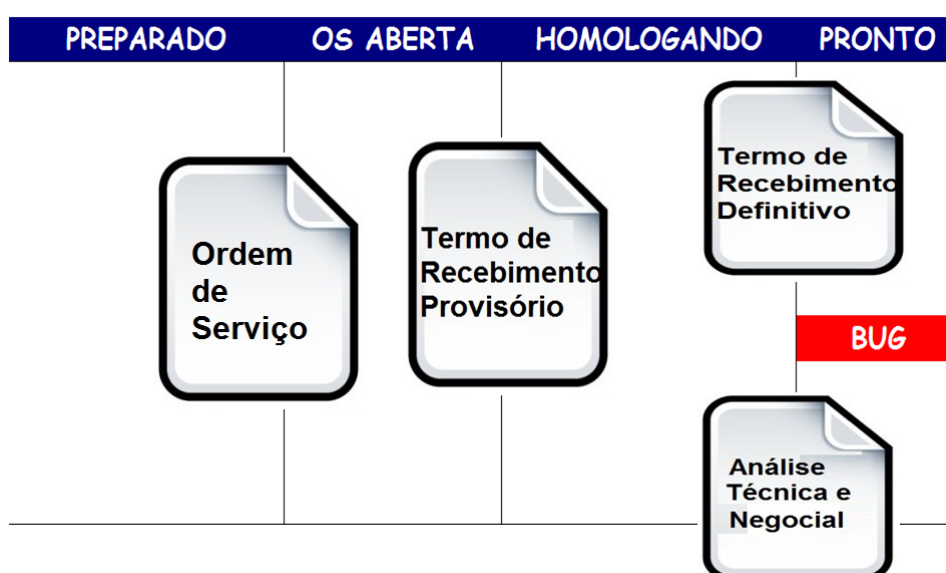


Figura 17 – Transição entre raias

² Imagem disponível em <http://www.slideshare.net/herbertparente/contratao-de-fbrica-de-software-metodologia-gil>

³ Imagem disponível em <http://www.slideshare.net/herbertparente/contratao-de-fbrica-de-software-metodologia-gil>

A transição da raia “OS Aberta” para a raia “Homologando” ocorreu quando o Termo de Recebimento Provisório era emitido (Fig. (17)). Esta transição ocorreu ao se realizar o recebimento dos produtos e teve como saída o termo de recebimento provisório. Com a emissão do termo de recebimento provisório, os produtos recebidos entravam no processo de homologação.

A transição da raia “Homologando” para a raia “Pronto” ocorreu quando o Termo de Recebimento Definitivo era emitido, ou seja, quando todos os produtos que foram anteriormente entregues eram verificados e aprovados (Fig. (17)). Para tanto eram aferidos a aderência aos padrões técnicos e aos requisitos a partir de uma análise técnica e negocial, realizadas conjuntamente pelo Fiscal do Contrato e o Gestor do Contrato. Se fossem detectados defeitos nos produtos entregues ou se eles fossem rejeitados ou tivessem necessidade de refatoração, eles retornavam para a fila de demandas, iniciando novamente o ciclo.

A sinalização de rejeitado ou *bug* diz respeito a funcionalidade que foi rejeitada por não atender o que foi pedido tanto funcionalmente quanto tecnicamente. A sinalização de refatoração diz respeito a mudança que era pedida em uma funcionalidade depois de ela já ter sido implementada. Para que uma funcionalidade entre nessa sinalização era preciso que o gestor de negócio assumisse a responsabilidade pelos impactos que a mudança causará no custo, tempo e escopo.

Vale ressaltar que é importante que o trabalho em progresso (WIP) fosse limitado conforme o que é conceituado no método Kanban. O IPHAN definiu um limite de 210 Pontos por Função por ciclo de trabalho (Fig. (18) ⁴).

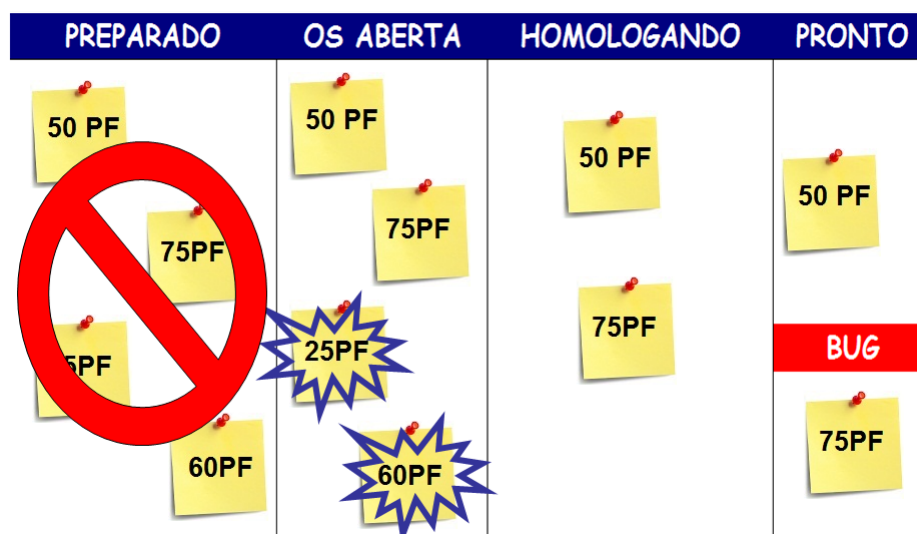


Figura 18 – Limitação de WIP

⁴ Imagem disponível em <http://www.slideshare.net/herbertparente/contratao-de-fabrica-de-software-metodologia-gil>

É importante sempre valorizar a entrega de produto funcional e não pagar por apenas documentação. Assim, o IPHAN dividiu a forma de pagamento da contratada em percentuais, de acordo com a fase, valorizando a fase de execução, como ilustrado na Tab. (9). Na *Sprint* 18 do projeto SICG, ocorreu uma mudança acordada com a empresa contratada de modificar o pagamento da fase de execução para 100%, porque não fazia sentido ter retenção de pagamento pois as fases de planejamento, desenvolvimento e encerramento já estavam sendo executadas pela empresa contratada, valorizando - se ainda mais a fase de execução.

Fase	Percentual de Pagamento
Planejamento (1 vez)	5%
Execução (n vezes)	75%
Implantação (1 vez)	10%
Encerramento (1 vez)	10%

Tabela 9 – Modelo de Remuneração do projeto SICG

Outra técnica importante que foi construída diz respeito a parelização das atividades Fig. (19)⁵. Enquanto uma ordem de serviço estava na etapa de homologação, outra ordem de serviço podia ser preparada, evitando que o fluxo do processo parasse e houvesse desperdício.

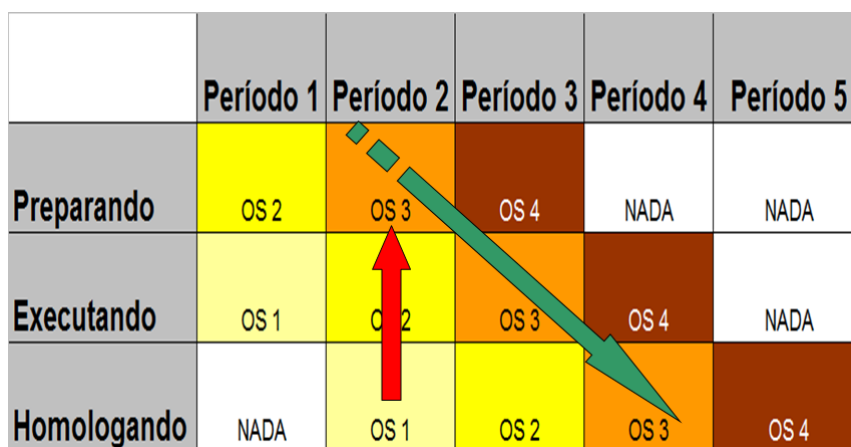


Figura 19 – Parelização de Atividades

Após alguns meses de aplicação dessa solução o órgão deu início a construção da Metodologia IPHAN de Gestão de Demandas de Desenvolvimento Ágil de *Software* (MIDAS). Uma parte do MIDAS, o mapeamento dos principais processos, pode ser visualizada no Anexo B.

⁵ Imagem disponível em <http://www.slideshare.net/herbertparente/contratao-de-fbrica-de-software-metodologia-gil>

6.4.1 Análise da Adequação da Solução ao Pensamento Lean e ao Scrum

Ao analisarmos os princípios do Pensamento *Lean*, apresentados no Capítulo 3, evidenciamos, na solução apresentada, a aplicação de alguns princípios de Desenvolvimento Lean, como: eliminação de desperdício, integração da qualidade, criação de conhecimento, entrega rápida, respeito as pessoas e a constância de propósitos. Ainda, identificamos as práticas gerenciamento de código, iterações curtas, participação do cliente e Kanban.

A parelização das atividades, Fig. (19), procurou reduzir alguns desperdícios de trabalho, ao evitar que o fluxo de trabalho fosse interrompido e, com isso, a produção, no que se refere ao desenvolvimento, fosse interrompida, à espera da finalização de uma determinada atividade.

Considerando o ponto de vista do gestor do contrato e fiscal técnico, foi possível observar que a qualidade era integrada a cada ordem de serviço entregue por meio da aferição técnica. No entanto, essa aferição técnica realizada consistia em verificar se as classes estavam documentadas e se o modelo de dados era válido. Com relação a aferição técnica realizada pelo fiscal técnico, durante as *sprints*, não evidenciamos a realização de utilização da prática de análise estática do código fonte. Esta técnica foi evidenciada apenas ao final do projeto por meio da contratação de uma outra empresa de qualidade, também terceirizada.

A criação do conhecimento pôde, por exemplo, ser evidenciada ao utilizar o aprendizado do processo definido nesta solução para a construção da metodologia de gestão de demandas do órgão, o MIDAS.

O IPHAN também definiu ciclos curtos de entrega de ordens de serviço, fazendo com que a entrega rápida de *software* funcional acontecesse propiciando o *feedback* constante do *Product Owner* fosse possível.

O respeito as pessoas pôde ser observado, por exemplo: i) ao considerar o aprendizado e adaptação da empresa contratada à solução de gestão de contratos definida; ii) a não aplicar multas punitivas; iii) a valorizar a fase de execução como principal fase de remuneração do trabalho.

Observamos a constância de propósitos no que se refere a definição de objetivos claros da entrega de *software* funcional.

O gerenciamento de código foi realizado por meio da ferramenta de controle de versão Apache Subversion (SVN).

Foram definidas ciclos de produção curtos durante todo o projeto, as quais variaram de duas a quatro semanas.

A participação do cliente, desempenhada pelo *Product Owner*, foi constante em todo o projeto. Ele sempre estava disponível para as dúvidas de requisitos.

As raias da ferramenta Kanban foram adaptadas para o contexto de contratações públicas e o trabalho em progresso foi limitado, o que corrobora com a teoria das restrições devido ao foco do IPHAN em controlar o fluxo de trabalho e não a metodologia de desenvolvimento da contratada.

A limitação do trabalho em progresso realizada com a métrica Pontos por Função é resultado do conhecimento da capacidade produtividade da equipe. Neste trabalho não houve a investigação das limitações do Ponto por Função para medição de tamanho e unidade de faturamento das ordens de serviço, uma vez que, este trabalho têm foco primário em ser exploratório.

Não foram encontradas evidências, a partir dos dados coletados, da utilização dos princípios de adiar comprometer, de otimizar o todo, das práticas de teste automatizado e menos código. No que diz respeito a otimização do todo, não foi encontrada uma definição de cadeia de valor, onde a otimização seja feita desde a abertura da ordem de serviço até a implantação do *software* junto ao usuário. Com relação ao teste automatizado, apesar da empresa contratada realizar alguns testes, não houve a entrega de uma base de código com uma suíte completa de testes automatizados. E com relação a prática menos código, segundo o IPHAN, foram encontradas muitas duplicações no código final do projeto, porém, essa métrica (código duplicado) não foi analisada nesse trabalho.

Ao analisarmos a metodologia Scrum, apresentada no Capítulo 4, identificamos que esta solução tem uma característica empírica e adaptativa conforme o Scrum. A utilização de iterações de curta duração e as cerimônias realizadas pelo IPHAN evidenciam a aderência a essa metodologia.

O IPHAN definiu cadências nas iterações ou *sprints* com duração que variaram de duas a quatro semanas no máximo. O órgão juntamente com a empresa contratada realizou reuniões tanto de planejamento de cada *sprint* quanto de retrospectiva de cada *sprint* do projeto.

O papel de *Product Owner* foi desempenhado pelo gestor do contrato e o papel de Scrum Master pelo gestor de projeto da empresa contratada. Também foi utilizado o *Product Backlog*, o qual continha todas as funcionalidades a serem desenvolvidas priorizadas.

Não foram encontradas evidências, de acordo com os dados coletados, de que reuniões diárias e reuniões de revisão da *sprint* eram realizadas. No que diz respeito a revisão da *sprint*, ela era feita apenas pelo fiscal técnico no momento de homologação das funcionalidades entregues.

Em que pese não tenhamos medido especificamente a evidência de cada princípio, prática ou valor, pudemos observar indícios de que as premissas utilizadas como base para o desenvolvimento desta solução corroboram com a maioria dos princípios do Pensamento

Lean e com algumas práticas do *Lean* no Desenvolvimento de *Software* e da metodologia Scrum. No entanto, alguns princípios e práticas fundamentais não foram evidenciados, tais como otimização do todo, testes automatizados e menos código.

Embora não tenhamos planejado no protocolo deste estudo caso a aferição de cada prática e princípio em particular, observamos ao longo deste estudo empírico o indício da aplicação destes. Como trata-se de um estudo exploratório, procuramos registrar o resultado dessas observações, de forma a balizar estudos posteriores.

6.4.2 Análise da Adequação da Solução aos Valores Ágeis e a Legislação

O projeto SICG, do contrato analisado neste estudo de caso, uma vez que é caracterizado como "serviço comum" foi contratado mediante a modalidade Pregão, na forma eletrônica, pelo critério de "menor preço", conforme os termos da Lei nº 8.666/93, apresentada no Capítulo 2.

É importante ressaltar que o Acórdão nº 237/2009 - Plenário do Tribunal de Contas da União orienta claramente sobre o uso da modalidade pregão para bens e serviços de tecnologia da informação considerados comuns e descarta que a complexidade, a criticidade ou a relevância dos bens possam justificar o não uso dessa modalidade. E o Decreto nº 7.174/2010 Art. 9º, Parágrafo 1º declara a recomendação do tipo "menor preço" para a aquisição de bens e serviços de informática e automação considerados comuns. Com isso, no que diz respeito a modalidade e ao tipo da licitação, o IPHAN fez a definição adequada com respaldo da legislação pertinente.

Conforme apresentado no Capítulo 2, a IN 04/2010 apresenta os papéis que estão presentes no processo de contratações e o mapeamento do processo da fase de Gerenciamento de Contrato.

A Tabela 10 apresenta adequação de atores, papéis e responsabilidades do IPHAN aos papéis apresentados na IN 04/2010.

Atores IPHAN	Papéis IN 04/2010	Responsabilidades
CGI- Coordenação de Inventários e Conhecimento. CGCI – Coordenação Geral de Cidades. DEPAM – Direção do Departamento de Patrimônio Material e Fiscalização.	Fiscal Requisitante Gestor do Contrato	<ul style="list-style-type: none"> - Gestão negocial do contrato. - Aprovação do sistema. - Auxílio no processo de definição de regras do sistema. - Solicitações de alterações. - Repasse de informações sobre o sistema. - Homologação dos artefatos sistêmicos. - Operação do sistema nas atividades diárias.
CGLOG – Coordenação Geral de Logística, Convênios e Contratos.	Fiscal Administrativo	<ul style="list-style-type: none"> - Fiscalização administrativa do contrato.
CGTI – Coordenação Geral de Tecnologia da Informação.	Fiscal Técnico Coordenador do Projeto	<ul style="list-style-type: none"> - Fiscalização técnica do contrato. - Coordenação técnica do projeto quanto à solução, à documentação sistêmica, à metodologia empregada e à arquitetura tecnológica.
Usuários nas Superintendências e outras unidades do IPHAN.	Usuário	<ul style="list-style-type: none"> - Auxílio no processo de definição de regras e procedimentos existentes nas Superintendências e outras unidades do IPHAN. - Operação do sistema nas atividades diárias.
EGL - Engenharia	Preposto Gestor do Projeto	<ul style="list-style-type: none"> - Acompanhar a execução do contrato. - Atuar como interlocutor junto a Contratante. - Receber, diligenciar, encaminhar e responder as principais questões técnicas, legais e administrativas referentes ao andamento contratual.

Tabela 10 – Papéis e Responsabilidades

A Tabela 11 apresenta a análise da adequação da solução do IPHAN às etapas definidas na fase de Gerenciamento do Contrato da IN 04/2010, a qual é ilustrada na Fig. 4.

IN 04/2010		Solução IPHAN	
Etapa	Principais Atividades	Fase	Principais Atividades
Iniciação	Elaborar Plano de Inserção	Planejamento	Realizar Reunião Inicial
	Convocar Reunião Inicial		Planejar Projeto
	Realizar Reunião Inicial		Realizar Ateste Técnico
	Enviar Ata para Aprovação		Realizar Aceitação da Fase
Encaminhar	Ordem de Serviço	Desenvolvimento	Planejar <i>Sprint</i>
Monitoramento da Execução	Receber Objeto	Desenvolvimento	Acompanhar <i>Sprint</i>
	Elaborar Termo de Recebimento Provisório		Receber Produtos e emitir Termo de Recebimento Provisório
	Avaliar Qualidade		Realizar Ateste Técnico
	Encaminhar Demandas de Correção		Elaborar Termo de Recebimento Definitivo das Funcionalidades Homologadas
	Verificar Aderência aos Termos Contratuais		Encaminhar Demandas de Correção das Funcionalidades Não Homologadas
	Elaborar Termo de Recebimento Definitivo		Emitir Nota Fiscal
	Autorizar Emissão de Nota Fiscal		Verificar Regularidades Fiscais, Trabalhistas e Previdenciárias
	Verificar Regularidades Fiscais, Trabalhistas e Previdenciárias		Verificar manutenção da necessidade, economicidade e oportunidade da contratação.
	Verificar manutenção da necessidade, economicidade e oportunidade da contratação.		Realizar Aceitação da Fase
Transição Contratual		Contratada deve passar documentos para transição	
Encerramento do Contrato		Encerramento	Encerrar Projeto
			Realizar Ateste Técnico
			Realizar Aceitação da Fase

Tabela 11 – Gerenciamento na IN 04/2010 x Gerenciamento no IPHAN

Conforme apresentado no Capítulo 4, o TCU (2013) identificou alguns riscos que poderiam comprometer a utilização de metodologias ágeis na gestão de contratos de fornecedores de desenvolvimento de *software*.

Assim como mencionamos na seção anterior, não planejamos formalmente a análise desses riscos, no protocolo deste estudo de caso. Portanto, não se trata de um estudo sistemático em investigar a materialização ou não dos riscos apontados por TCU (2013), contidos na seção 4.3.1 do Capítulo 4. Contudo, registramos algumas observações, que foram possíveis de serem feitas com as fontes que analisamos. Não se trata, portanto, de um estudo aprofundado sobre esse fator, porém, reflete algumas percepções que emergiram no decorrer dessa investigação empírica.

A Tab. (12) apresenta como o IPHAN lidou com cada um desses riscos.

Riscos segundo Acórdão 2314/2013	Solução do IPHAN
R1	A solução desenvolvida pelo IPHAN foi pensada para se adequar ao contexto de contratações de <i>software</i> , com isso, algumas práticas e princípios ágeis não foram seguidos em sua plenitude. Porém os valores foram mantidos sem necessariamente desvirtuar da essência de ágeis. Não observamos, na documentação analisada, além do resultado das entrevistas e questionários, conflitos ou problemas de competência com a empresa contratada no que diz respeito a adaptação da metodologia de gestão de contratos definida. Vale ressaltar que a adaptabilidade é um dos pilares de qualquer método científico.
R2	As alterações realizadas na metodologia foram realizadas em comum acordo com a empresa contratada. Como exemplo, inicialmente a iteração de cada ordem de serviço tinha a duração máxima de três semanas, posteriormente, a duração máxima foi alterada para quatro semanas a pedido da empresa contratada em comum acordo com o órgão. É importante frisar nesse contexto, que o IPHAN não se preocupou em controlar e/ou definir uma metodologia de desenvolvimento para a empresa contratada. Preocupou-se em definir uma metodologia de gestão de contrato de desenvolvimento, focada no controle do fluxo de trabalho e no controle das restrições do sistema. Não se procurou em prescrever um método de gestão, materializado por um desenho de processos detalhados.

R3	<p>O IPHAN definiu no termo de referência os artefatos mínimos a serem entregues em cada fase do projeto. Na fase de Planejamento: plano de projeto e 3 propostas de identidade visual. Na fase de Desenvolvimento: plano de ciclo individual, código fonte (documentado), código compilado, modelo de entidade relacionamento, manual do sistema, manual do usuário e ajuda do sistema. Na fase de Implantação: manual do usuário e manual do sistema atualizados. Na fase de Encerramento: plano de treinamento, material de treinamento, relatório de encerramento do projeto com lições aprendidas e capacitação. Não identificamos alterações feitas sobre esse conjunto mínimo de artefatos.</p>
R4	<p>O IPHAN definiu no termo de referência os artefatos mínimos a serem entregues em cada fase do projeto. O pagamento a empresa durante a fase de desenvolvimento era realizado sobre as funcionalidades entregues e homologadas. Artefatos documentais, como o manual de usuário, eram exigidos durante a execução do projeto, porém foram apenas remunerados em sua versão final na fase de implantação ou encerramento. Observa-se o fato do IPHAN estar valorizando mais aspectos relacionados ao produto, do que preocupação demasiada com documentação exigida por um processo.</p>
R5	<p>A definição de metodologia ágil de gestão de contrato foi definida pelo IPHAN já no instrumento convocatório.</p>
R6	<p>O responsável pela área de negócios do IPHAN que desempenhou o papel de <i>Product Owner</i> participou e colaborou constantemente com o desenvolvimento do projeto, exercendo também o papel de Gestor do Contrato. Algumas das atividades realizadas por ele foram: gerenciar funcionalidades, priorizar funcionalidades e realizar a homologação comercial dos artefatos entregues a cada iteração.</p>
R7	<p>O <i>Product Owner</i> tinha pouco conhecimento sobre metodologias ágeis, assim como em fazer a priorização de funcionalidades essenciais para o bom desempenho do <i>software</i>. Este risco materializou-se e foi tratado com a colaboração do fiscal técnico nas reuniões de planejamento da <i>sprint</i>. Além disso, o <i>Product Owner</i> se empenhou em aprender os aspectos técnicos e da metodologia.</p>

R8	No IPHAN, quando o <i>Product Owner</i> afastou-se, em decorrência dos recessos de fim de ano, o desenvolvimento de novas funcionalidades foi suspenso e a empresa utilizou esse período para introduzir melhorias técnicas no produto, melhorando a qualidade do código e não paralisando o fluxo de trabalho do projeto. Isto ocorreu entre a <i>sprint</i> 10 e 11. Quando o <i>Product Owner</i> afastou-se, na <i>sprint</i> 16 em decorrência de férias, ele foi substituído por outro servidor do IPHAN da mesma área negocial sem ocorrer alteração no resultado da entrega.
R9	De acordo com as respostas ao questionário aplicado, apesar do <i>Scrum Master</i> da empresa contratada possuir pouca experiência em metodologias ágeis, os desenvolvedores possuíam uma experiência alta, de forma que não foi reportado pelo IPHAN problemas decorrentes desse risco, em que pese não analisamos esse fator de forma controlada. Uma sugestão seria que a empresa contratada fosse submetida a treinamentos para se adaptar mais rapidamente a metodologia ágil.
R10	Eram realizadas reuniões de planejamento e retrospectiva a cada ciclo com o <i>Product Owner</i> e representantes da empresa contratada. Reuniões esporádicas também foram realizadas quando necessário, além de troca de e-mails, de forma que a comunicação entre contratante e contratada, foi observado pelos entrevistados como um fator positivo na relação.
R11	A mudança de requisitos ocorreu durante as <i>sprints</i> do projeto. Com relação ao impacto sobre o custo do projeto, houve uma diferença de 1,8% entre o custo planejado e o custo real ao final da <i>sprint</i> 24. Com relação ao impacto sobre o prazo, foram planejados as mesmas quantidades de ciclos de desenvolvimento que foram realizados (24 ciclos), sendo que cerca de 78% dos requisitos foram atendidos ao final da <i>sprint</i> 24.
R12	A paralelização de atividades ocorreu no IPHAN. Enquanto um ciclo entregue era homologado outro iniciava-se e era desenvolvido. Caso houvesse funcionalidades não homologadas no ciclo anterior, as mesmas eram passadas para o próximo ciclo. Não encontramos evidência de que esta prática tenha ocasionado diretamente atrasos na entrega do produto.
R13	No IPHAN, o planejamento foi iniciado de forma global por meio da elaboração de um cronograma do projeto e, posteriormente, de forma mais específica nas reuniões de planejamento da <i>sprint</i> . Não realizamos um estudo específico para analisar esse efeito sobre o custo total do projeto.

R14	O valor pago no final de cada ciclo era realizado apenas sobre as funcionalidades que foram homologadas. Com relação a mudanças realizadas sobre funcionalidades já homologadas o IPHAN pagava 50% do valor do ponto de função. Este percentual foi alterado com relação ao percentual constante no roteiro do SISP para minimizar o custo desse risco.
R15	O IPHAN disponibilizou de forma tardia o <i>software</i> desenvolvido em ambiente de produção, apenas ao final do projeto, contradizendo um dos objetivos dos métodos ágeis que é entrega adiantada e contínua de <i>software</i> funcional "nas mãos" dos usuários. Isto causou a detecção tardia de erros. Porém, como havia uma garantia de cerca de doze meses após a finalização do contrato, os erros identificados foram corrigidos pela empresa contratada sem um novo pagamento ser necessário. Não avaliamos os impactos na percepção de uso, por parte dos usuários.
R16	Para mitigar esse risco, o IPHAN valorizou a fase de execução mais do que as outras, fazendo com que ela correspondesse a 75% da remuneração total do projeto. Além disso, o IPHAN deixou aberto para a empresa contratada a possibilidade de utilizar outra métrica desde que a empresa apresentasse uma referência bibliográfica adequada e uma justificativa formal para o uso da mesma.

Tabela 12 – Riscos Acórdão 2314/2013 x Solução IPHAN

Ainda, como apresentado no Capítulo 4, o Acórdão nº 2314/2013 apresentou uma análise subjetiva com relação ao uso dos valores ágeis em contratações, relacionada a possibilidade de ferir os princípios da APF. A Tab. (13) apresenta como o IPHAN lidou com os princípios da APF juntamente com os valores ágeis.

Valor do Manifesto Ágil e Princípio da APF	Solução IPHAN
Indivíduos e interações acima de processos e ferramentas e Princípio da Eficiência e Princípio de Pessoalidade	Os processos da organização não foram desprezados. Segundo relatos dos entrevistados, quando uma solicitação da área requisitante era recebida, procurava-se verificar se eles tinham o processo de negócio daquela requisição mapeado. Caso não tivesse, era recomendado, porém não obrigatório, um mapeamento desses processos antes da demanda passar para o desenvolvimento. Como forma de mitigar a relação de pessoalidade entre os funcionários da contratada e os gestores da instituição contratante, as reuniões de planejamento e retrospectiva foram todas registradas em ata de reunião.
<i>Software</i> operante acima de documentações grandes e completas e Princípio da Eficiência	Conforme o valor apresentado, o IPHAN valorizou a entrega de <i>software</i> funcional acima de documentação abrangente. Ainda sim, houve uma preocupação de se valorizar a documentação do <i>software</i> desenvolvido. Isso pode ser observado em nível de documentação de código, onde foi exigida a documentação por classe, método, arquivo, pacote e entre outros, em aderência a premissa de melhoria da manutenibilidade. Além disso, um conjunto mínimo de artefatos foi exigido no instrumento convocatório. Vale ressaltar que segundo o IPHAN, um documento que poderia ter sido acrescentado era o documento de regras de negócio.
Colaboração do cliente acima de negociações contratuais e Princípio da Vinculação ao Instrumento Convocatório	O instrumento convocatório determinou qual era o processo de gestão de contratos do órgão assim como as alterações previstas. O <i>Product Owner</i> do IPHAN estava frequentemente à disposição da equipe da empresa contratada para solucionar dúvidas com relação aos requisitos.
Responder à mudanças acima de seguir um plano e Princípio do Planejamento e Princípio da Economicidade	Um planejamento global foi realizado no início do contrato e planejamentos específicos realizados a cada ciclo. No momento em que um ciclo (com duração de quatro semanas) era planejado ele não poderia ser alterado, as alterações ficavam para uma iteração posterior, não ferindo o princípio do planejamento. Além disso, as iterações curtas adotadas pelo IPHAN facilitaram as mudanças. As mudanças realizadas por motivos técnicos não eram pagas novamente, não ferindo o princípio de economicidade. No entanto, as mudanças negociais eram tratadas com mais rigor, o <i>Product Owner</i> assumia a responsabilidade sobre o impacto no custo, no tempo e no escopo. Essas mudanças de funcionalidades já homologadas eram remuneradas com 50% do valor do ponto de função.

Tabela 13 – Valores do Manifesto Ágil e Princípios da APF x Solução do IPHAN

Ressaltamos novamente que não se trata de estudo exaustivos sobre os efeitos observados no riscos apontados em TCU (2013), o que ensajaria estudos mais controlados para diminuir os viéses da observação.

6.5 Análise dos Dados

Nesta seção serão apresentados os resultados dos dados coletados, categorizados em três efeitos: entrega de ordens de serviço, satisfação do cliente e qualidade interna do código fonte. É importante destacar que é a partir da análise desses efeitos que responderemos às questões específicas elaboradas para investigar o problema apresentado. As subseções a seguir têm o objetivo de responder diretamente às questões específicas definidas no Capítulo 5.

6.5.1 Efeitos sobre a entrega de ordens de serviço

A partir dos dados coletados da documentação dos Processos do projeto SICG foram calculadas as métricas para responder as questões específicas QE1 a QE9, que caracterizam os efeitos sobre a entrega de ordens de serviço do uso de métodos ágeis na gestão do contrato de fornecedores de desenvolvimento de *software*.

A Tabela 14 apresenta os resultados das métricas para cada uma das questões específicas anteriormente mencionadas.

ID	Questão	Resultado da Métrica
QE1.	Qual a quantidade total de ordens de serviço?	25 ordens de serviço
QE2.	Qual a quantidade de ordens de serviço que tiveram entrega de <i>software</i> funcional?	20 ordens de serviço
QE3.	Qual a quantidade de ordens de serviço que tiveram entrega apenas de documentação?	1 ordem de serviço
QE4.	Qual a proporção de entrega de <i>software</i> funcional?	80%
QE5.	Qual a duração média de entrega de <i>software</i> funcional?	3.7 semanas
QE6.	Qual a quantidade de ordens de serviço que não teve entrega de <i>software</i> funcional e de documentação?	4 ordens de serviço
QE7.	Qual a porcentagem de requisitos atendidos em cada ordem de serviço?	Tabela 15
QE8.	Quantas multas foram aplicadas no contrato?	0 multas
QE9.	Qual o custo de cada <i>sprint</i> e ordem de serviço do projeto?	Tabela 16

Tabela 14 – Resultados Métrica de QE1. a QE9.

A Tabela 15 apresenta a porcentagem de requisitos atendidos em cada ordem de serviço ou *sprint*, respondendo assim a questão específica QE7.

<i>Sprint</i>	OS	Porcentagem de Requisitos Atendidos
0	1	100%
1	2	0%
2	3	94%
3	4	0%
4	5	94%
5	6	100%
6	6.1	0%
7	7	100%
8	8	94%
9	9	100%
10	10	100%
11	11	100%
12	12	100%
13	13	84%
14	14	100%
15	15	100%
16	16	100%
17	17	84%
18	18	0%
19	19	100%
20	20	100%
21	21	100%
22	22	36%
23	23	60%
24	24	100%

Tabela 15 – Porcentagem de Requisitos Atendidos por Ordem de Serviço

A média geral de requisitos atendidos ao longo do projeto de acordo com os dados levantados da documentação foi de 78%, o que coincide com a percepção dos envolvidos no projeto: 75% dos envolvidos responderam ao questionário que a percepção geral de requisitos atendidos durante o projeto estava entre 70% e 90%.

A Tabela 16 apresenta o custo de cada ordem de serviço ou *sprint*, respondendo assim a questão específica QE9. Como pode ser observado, nas ordens de serviço que

tiverem a porcentagem de requisitos atendidos como 0% não houve remuneração para a contratada. O custo estimado do contrato antes da contratação era de R\$ 990,000. Portanto, o custo final foi próximo do custo estimado, ultrapassando apenas 1,8%.

Sprint	OS	Custo
Planejamento Inicial	1	R\$ 49,500.00
1	2	R\$ 0.00
2	3	R\$ 53,316.25
3	4	R\$ 0.00
4	5	R\$ 36,382.50
5	6	R\$ 31,927.50
6	6.1	R\$ 0.00
7	7	R\$ 45,478.13
8	8	R\$ 31,741.88
9	9	R\$ 29,143.13
10	10	R\$ 65,896.88
11	11	R\$ 64,597.50
12	12	R\$ 107,291.25
13	13	R\$ 56,801.25
14	14	R\$ 39,909.38
15	15	R\$ 63,669.38
16	16	R\$ 61,627.50
17	17	R\$ 31,185.00
18	18	R\$ 0.00
19	19	R\$ 53,831.25
20	20	R\$ 50,490.00
21	21	R\$ 63,112.50
22	22	R\$ 13,612.50
23	23	R\$ 22,027.50
24	24	R\$ 36,630.00

Tabela 16 – Custo por Ordem de Serviço

É importante destacar que apesar das *sprints* 1, 3, 6 e 18 não terem sido faturadas, não houve aplicação de multa para a contratada. Pois, conforme a solução caracterizada anteriormente, o órgão optou por modificar o modelo de remuneração e considera o aprendizado e a adaptação da empresa ao tipo de metodologia de gestão de contrato que está sendo aplicado. Além disso, o fato de a empresa não ter um pagamento em seu fluxo

de caixa já é penalização suficiente para que a empresa faça a adequação do seu processo de trabalho.

Outro fato importante é que ao entregar uma ordem de serviço para homologação, uma recontagem de pontos por funções do que foi entregue era realizada, e ao realizar a homologação o fiscal técnico do IPHAN realizava uma nova contagem de pontos por função sobre as funcionalidades homologadas. Apenas o valor de pontos de função das funcionalidades homologadas eram faturados. Cada ponto de função corresponde a R\$ 495,00. Até a *Sprint* 17 era realizado o pagamento de 75% do valor homologado, no entanto, a partir da *Sprint* 18, o valor do pagamento realizado passou a ser 100% do valor homologado.

Ainda, pelos dados coletados neste estudo, não foi evidenciada a utilização de uma técnica de gerenciamento de custos do projeto, como exemplo, o valor agregado. Uma ferramenta que automatize o processo de acompanhamento de custos poderia prover uma informação mais adequada e poderia auxiliar o gestor na tomada de decisão. O gestor do contrato poderia analisar, por exemplo, se o que está sendo pago está de acordo com o que está sendo produzido ou de acordo com o que foi planejado.

6.5.2 Efeitos sobre a satisfação do cliente

O cliente do projeto SICG é representado pelo papel de *Product Owner*, que também atuou como gestor do contrato, e por todos os envolvidos no projeto por parte do IPHAN. Este nunca havia participado anteriormente de um projeto onde a gestão do contrato era realizada com o uso de métodos ágeis, portanto, a experiência dos envolvidos no projeto com esses métodos era pouca. Para coletar a opinião do cliente acerca das questões específicas QE10 e QE11 foi aplicado o questionário encontrado no Apêndice A.

A Figura 20 apresenta a visibilidade do processo de todos os envolvidos no projeto por parte do IPHAN. Cerca de 80% dos envolvidos no projeto considerou a visibilidade do processo como "Alta", dentre eles o gestor do contrato.

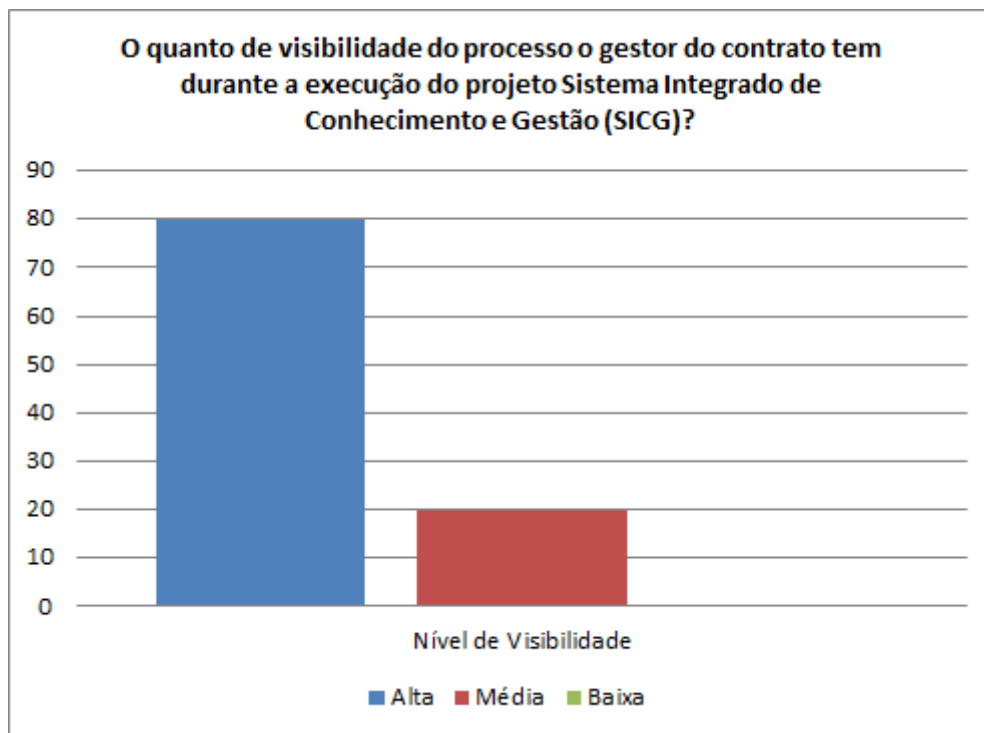


Figura 20 – Nível de Visibilidade do Processo do Projeto SICG

A Figura 21 apresenta o resultado de satisfação do IPHAN com relação ao produto entregue. De forma unânime foi considerado como "Satisfeito" o produto entregue pela empresa contratada.

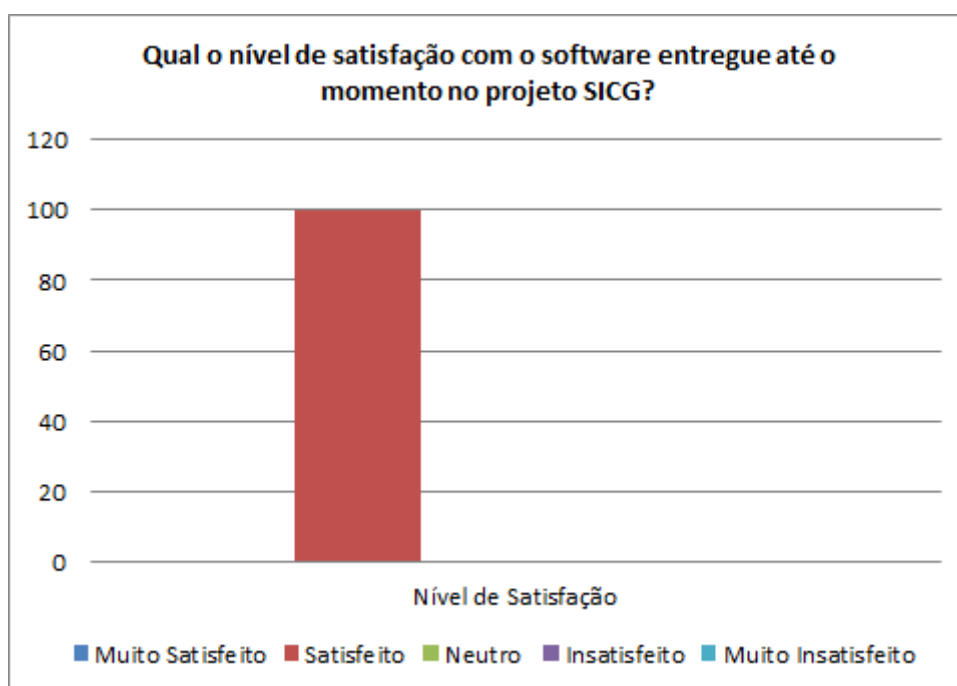


Figura 21 – Nível de Satisfação do Produto Entregue do Projeto SICG

6.5.3 Efeitos sobre a qualidade do código

A qualidade do código fonte do SICG pode ser avaliada por meio de métricas que evidenciam a qualidade externa (ISO/IEC 25023, 2011). Neste trabalho foram selecionadas doze métricas de código fonte levantadas e categorizadas por Meirelles (2013): métricas de tamanho e complexidade e métricas de orientação à objetos. A definição dessas métricas pode ser encontrada no Apêndice B.

Um dos objetivos científicos do estudo de Meirelles (2013) foi a identificação das distribuições estatísticas dos valores das métricas apresentadas anteriormente em 38 projetos de *software* livre, onde foram analisadas 344.872 classes das aplicações mais utilizadas de *software* livre, tais como, Chrome, Firefox, OpenJDK e VLC. A partir de cada uma das distribuições estatísticas Meirelles (2013) classificou as métricas de código fonte de acordo com a frequência dos valores apresentados, com os intervalos: muito frequente, frequente, pouco frequente e não frequente. Para simplificar o entendimento das métricas de código fonte, Moraes interpretou os rótulos dos intervalos de frequência definidos por Meirelles (2013) em em rótulos qualitativos, tal como apresentado na Tab. (17).

Intervalo de Frequência	Rótulo Qualitativos
Muito Frequente	Excelente
Frequente	Bom
Pouco Frequente	Regular
Não Frequente	Ruim

Tabela 17 – Nome dos Intervalos de Frequência e Qualitativos

Posteriormente, é apresentando na Tab. (18), retirada do estudo de Meirelles (2013), com os intervalos encontrados para C++ e Java. Com isso, os intervalos apresentados na Tab. (18) foram os utilizados como os indicadores de qualidade de código fonte na análise deste estudo de caso. É importante ressaltar que quando apresentarmos o resultado da análise sobre o código fonte deste estudo de caso e dissermos que o código fonte tem um rótulo "Excelente" ou "Ruim" para determinada métrica, significa dizer que, esse código fonte é "Excelente" ou "Ruim" em comparação ao melhor projeto de *software* livre selecionado por Meirelles (2013) na Linguagem Java, que foi o Open JDK8.

Métrica	Rótulos Qualitativos	Intervalos Qualitativos
LOC	Excelente	[de 0 a 33]
	Bom	[de 34 a 87]
	Regular	[de 88 a 200]
	Ruim	[acima de 200]
ACCM	Excelente	[de 0 a 2,8]
	Bom	[de 2,9 a 4,4]

	Regular	[de 4,5 a 6,0]
	Ruim	[acima de 6]
AMLOC	Excelente	[de 0 a 8,3]
	Bom	[de 8,4 a 18]
	Regular	[de 19 a 34]
	Ruim	[acima de 34]
ACC	Excelente	[de 0 a 1]
	Bom	[de 1,1 a 5]
	Regular	[de 5,1 a 12]
	Ruim	[acima de 12]]
ANPM	Excelente	[de 0 a 1,5]
	Bom	[de 1,6 a 2,3]]
	Regular	[de 2,4 a 3,0]
	Ruim	[acima de 3]
CBO	Excelente	[de 0 a 3]
	Bom	[de 4 a 6]]
	Regular	[de 7 a 9]
	Ruim	[acima de 9]
DIT	Excelente	[de 0 a 2]
	Bom	[de 3 a 4]]
	Regular	[de 5 a 6]
	Ruim	[acima de 6]
LCOM4	Excelente	[de 0 a 3]
	Bom	[de 4 a 7]
	Regular	[de 8 a 12]
	Ruim	[acima de 12]]
NOC	Excelente	[0]
	Bom	[1 a 2]
	Regular	[3]]
	Ruim	[acima de 3]
NOM	Excelente	[de 0 a 8]
	Bom	[de 9 a 17]
	Regular	[de 18 a 27]
	Ruim	[acima de 27]
NPA	Excelente	[0]
	Bom	[1]
	Regular	[de 2 a 3]
	Ruim	[acima de 3]
	Excelente	[de 0 a 9]

Bom	[de 10 a 26]]
Regular	[de 27 a 59]
Ruim	[acima de 59]

Tabela 18 – Intervalos de Qualidade Java

De forma geral, o processo realizado para a análise do código fonte do SICG foi retirado do trabalho desenvolvido por Rêgo (2014), que consiste em: i) executar a ferramenta de análise estática de código fonte, Analizo, sobre o código fonte do projeto SICG; ii) Após o passo i o resultado será armazenado em um arquivo .csv⁶ com as medidas numéricas das métricas selecionadas; iii) transformar o arquivo .csv em json⁷; iv) e por fim, processar esse arquivo na solução desenvolvida por Rêgo (2014) que faz uso de um ambiente de Datawarehousing, utilizando a ferramenta Pentaho. Ao final desse processo, temos como resultado os gráficos para análise.

A fim de responder a questão específica QE12 serão mostrados a seguir os resultados do processamento do código fonte do projeto SICG no que diz respeito a qualidade interna do produto ao longo das *sprints* do projeto, de acordo com as 12 métricas selecionadas.

As Tabelas 19, 20, 21 e 22 mostram os valores de todas as métricas ao longo das *sprints* do projeto.

⁶ Comma-separated values (ou CSV) é um formato de arquivo que armazena dados tabelados

⁷ JSON (JavaScript Object Notation) é um modelo para armazenamento e transmissão de informações no formato texto, tem sido bastante utilizado por aplicações Web devido a sua capacidade de estruturar informações de uma forma bem mais compacta, tornando mais rápido o parsing dessas informações.

		OS3	OS5	OS6	OS7	OS8
Rótulo de Qualidade	Métrica	Sprint2	Sprint4	Sprint5	Sprint7	Sprint8
Bom	ACC	3	3.5	4	3	3
	LCOM4	7				
	NOC	1	1	1	1	1
	NOM	9	9	9	9	9
	RFC	20	20	19	18	15
Excelente	ACCM	1.4	1.3	1.3	1.2	1.3
	AMLOC	4.3	3.2	3.2	3	3
	ANPM	1	1	1	0.9	0.9
	DIT	1	1	1	1	1
	LOC	22	24	21	19	19
	NOM					
	NPA	0	0	0	0	0
Regular	LCOM4		8	8	8	8
Ruim	CBO	28	49	61	118	155

Tabela 19 – Valores das Métricas do Projeto SICG da *Sprint 2* a *Sprint 8*

		OS9	OS10	OS11	OS12	OS13
Rótulo de Qualidade	Métrica	Sprint9	Sprint10	Sprint11	Sprint12	Sprint13
Bom	ACC	3	3	2	3	2.5
	LCOM4					
	NOC	1	1	1	1	1
	NOM	9	9	9		
	RFC	15	16	16	15	15
Excelente	ACCM	1.3	1.3	1.3	1.3	1.4
	AMLOC	3	3	3	3	3
	ANPM	0.9	1	0.9	0.9	1
	DIT	1	1	1	1	1
	LOC	19	19	21	19	20
	NOM				8	8
	NPA	0	0	0	0	0
Regular	LCOM4	8	8	8	8	8
Ruim	CBO	158	188	218	256	267

Tabela 20 – Valores das Métricas do Projeto SICG da *Sprint 9* a *Sprint 13*

		OS14	OS16	OS17	OS19	OS20
Rótulo de Qualidade	Métrica	Sprint14	Sprint16	Sprint17	Sprint19	Sprint20
Bom	ACC	2	2	2	2	2
	LCOM4					
	NOC	1	1	1	1	1
	NOM					
	RFC	15	13	13	13	12
Excelente	ACCM	1.4	1.2	1.2	1.2	1.2
	AMLOC	3	3	3	3	3
	ANPM	1	0.9	0.9	0.9	0.8
	DIT	1	1	1	1	1
	LOC	19	18	18	18	13
	NOM	8	8	8	8	8
	NPA	0	0	0	0	0
Regular	LCOM4	8	8	8	8	8
Ruim	CBO	279	327	333	347	357

Tabela 21 – Valores das Métricas do Projeto SICG da *Sprint 14* a *Sprint 20*

		OS21	OS22	OS23	OS24
Rótulo de Qualidade	Métrica	Sprint21	Sprint22	Sprint23	Sprint24
Bom	ACC	2	2	2	2
	LCOM4				
	NOC	1	1	1	1
	NOM				
	RFC	12	13	13	13
Excelente	ACCM	1.2	1.2	1.2	1.2
	AMLOC	3	3	3	3
	ANPM	0.9	0.9	0.9	0.9
	DIT	1	1	1	1
	LOC	13	13	13	14
	NOM	8	8	8	8
	NPA	0	0	0	0
Regular	LCOM4	8	8	8	8
Ruim	CBO	364	366	366	372

Tabela 22 – Valores das Métricas do Projeto SICG da *Sprint 21* a *Sprint 24*

A seguir, serão apresentados os gráficos dos resultados de cada métrica de forma

individual ao longo das *sprints* do projeto de forma a facilitar a interpretação dos mesmos. O eixo "x" dos gráficos corresponde a *sprint* do projeto. O eixo "y" dos gráficos corresponde ao valor resultante da métrica.

LOC (Lines of Code)

A Figura 22 mostra a qualidade da métrica LOC. Esta métrica permanece com o intervalo de qualidade "Excelente" constante ao longo das *sprints* do projeto.

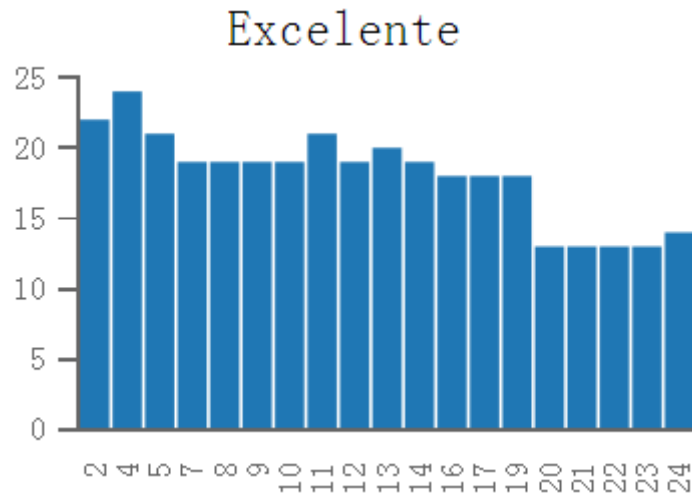


Figura 22 – Qualidade da Métrica LOC nas *Sprints* do Projeto SICG

ACCM (Average Cyclomatic Complexity per Method)

A Figura 23 mostra a qualidade da métrica ACCM. Esta métrica permanece com o intervalo de qualidade "Excelente" constante ao longo das *sprints* do projeto.

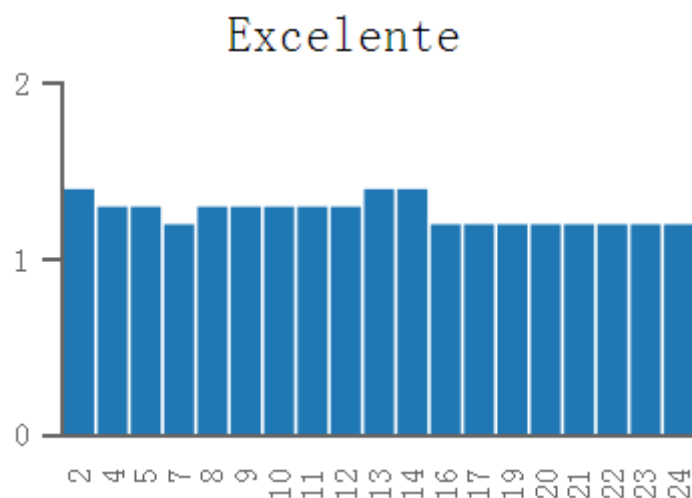


Figura 23 – Qualidade da Métrica ACCM nas *Sprints* do Projeto SICG

AMLOC (Average Method Lines of Code)

A Figura 24 mostra a qualidade da métrica AMLOC. Esta métrica permanece com o intervalo de qualidade "Excelente" constante ao longo das *sprints* do projeto.

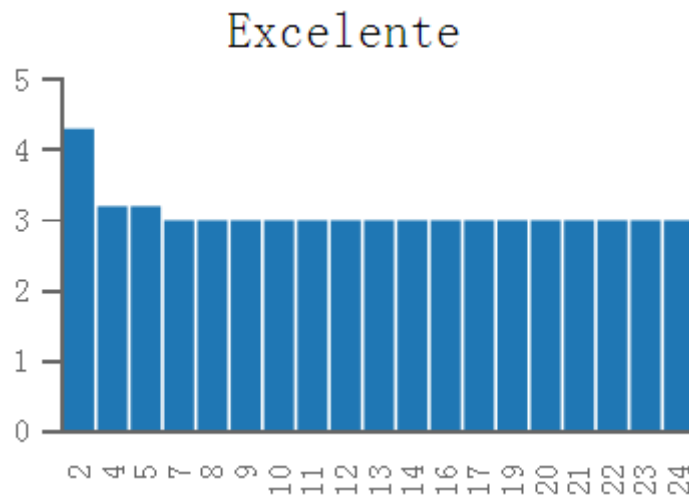


Figura 24 – Qualidade da Métrica AMLOC nas *Sprints* do Projeto SICG

ACC (Afferent Connections per Class)

A Figura 25 mostra a qualidade da métrica ACC. Esta métrica permanece com o intervalo de qualidade "Bom" constante ao longo das *sprints* do projeto.

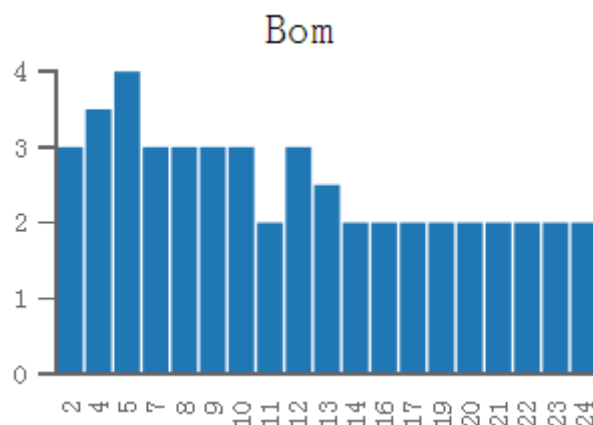


Figura 25 – Qualidade da Métrica ACC nas *Sprints* do Projeto SICG

ANPM (Average Number of Parameters per Method)

A Figura 26 mostra a qualidade da métrica ANPM. Esta métrica permanece com o intervalo de qualidade "Excelente" constante ao longo das *sprints* do projeto.

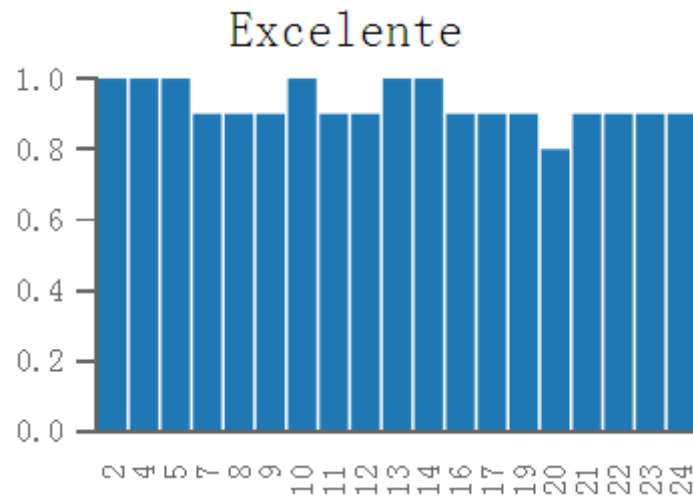


Figura 26 – Qualidade da Métrica ANPM nas *Sprints* do Projeto SICG

CBO (Coupling Between Objects)

A Figura 27 mostra a qualidade da métrica CBO. Esta métrica permanece com o intervalo de qualidade "Ruim" constante ao longo das *sprints* do projeto.

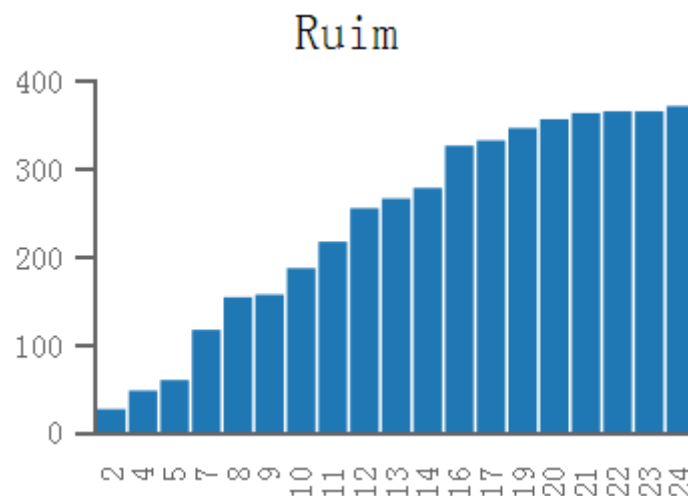


Figura 27 – Qualidade da Métrica CBO nas *Sprints* do Projeto SICG

DIT (Depth of Inheritance Tree)

A Figura 28 mostra a qualidade da métrica DIT. Esta métrica permanece com o intervalo de qualidade "Excelente" constante ao longo das *sprints* do projeto.

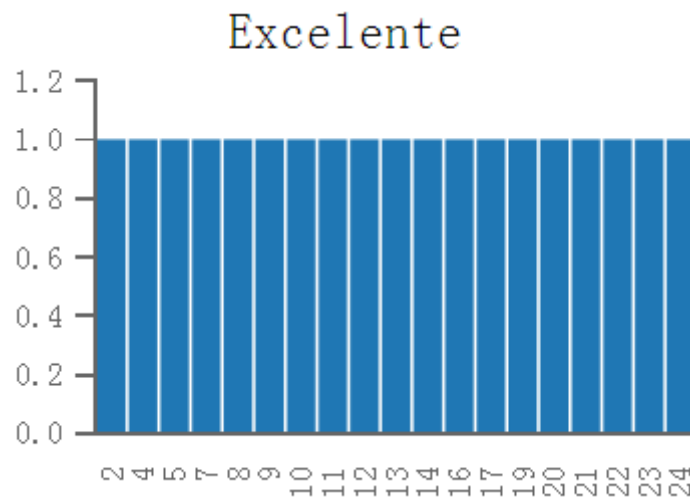


Figura 28 – Qualidade da Métrica DIT nas *Sprints* do Projeto SICG

LCOM4 (Lack of Cohesion in Methods)

A Figura 29 mostra a qualidade da métrica LCOM4. Esta métrica inicia com o intervalo de qualidade "Bom" na primeira *sprint* com entrega de *software* (Sprint 2) e, posteriormente, permanece com o intervalo de qualidade "Regular" constante ao longo das demais *sprints* do projeto.

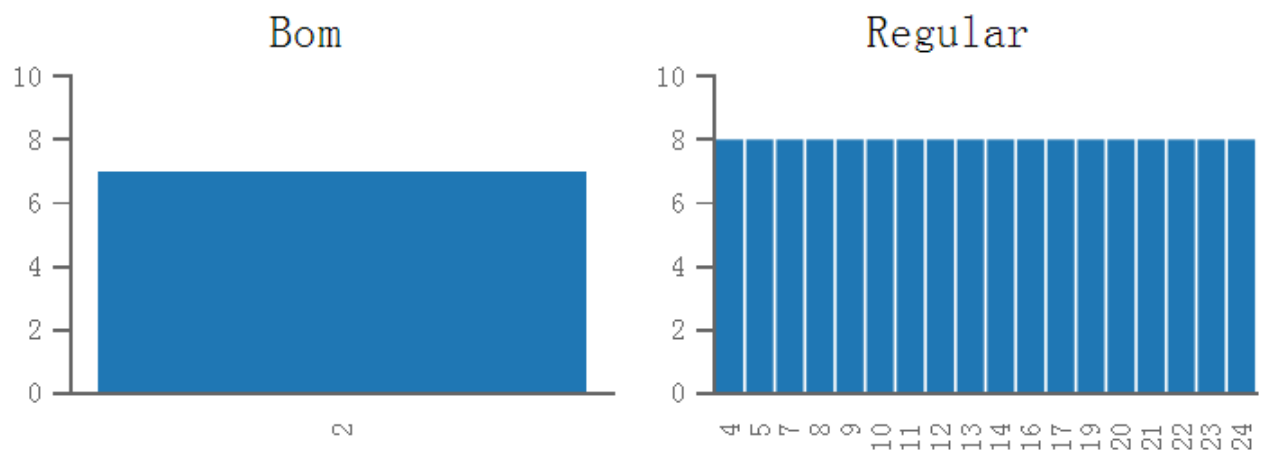


Figura 29 – Qualidade da Métrica LCOM4 nas *Sprints* do Projeto SICG

NOC (Number of Children)

A Figura 30 mostra a qualidade da métrica NOC. Esta métrica permanece com o intervalo de qualidade "Bom" constante ao longo das *sprints* do projeto.

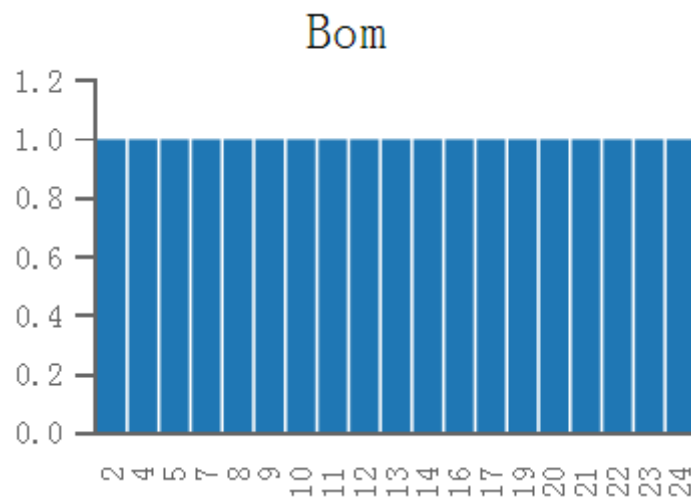


Figura 30 – Qualidade da Métrica NOC nas *Sprints* do Projeto SICG

NOM (Number of Methods)

A Figura 31 mostra a qualidade da métrica NOM. Esta métrica permanece com o intervalo de qualidade "Bom" até a *Sprint* 11 do projeto, a partir da *Sprint* 12 o código fonte obtém uma melhoria e a métrica NOM passa a ter o intervalo de qualidade "Excelente" até a última *sprint* analisada do projeto.

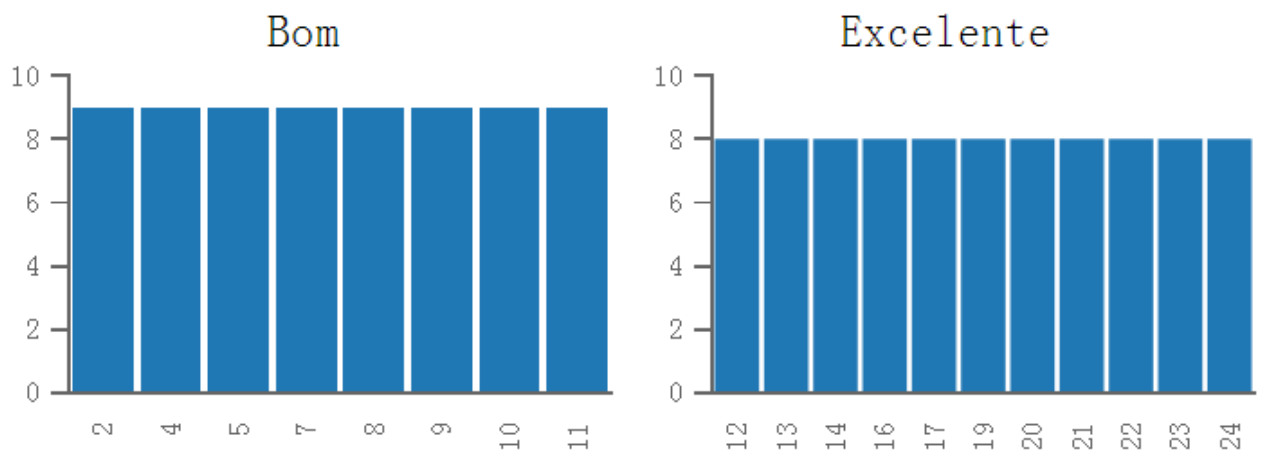


Figura 31 – Qualidade da Métrica NOM nas *Sprints* do Projeto SICG

NPA (Number of Public Attributes)

A Figura 32 mostra a qualidade da métrica NPA. Esta métrica permanece com o intervalo de qualidade "Excelente" constante ao longo das *sprints* do projeto.

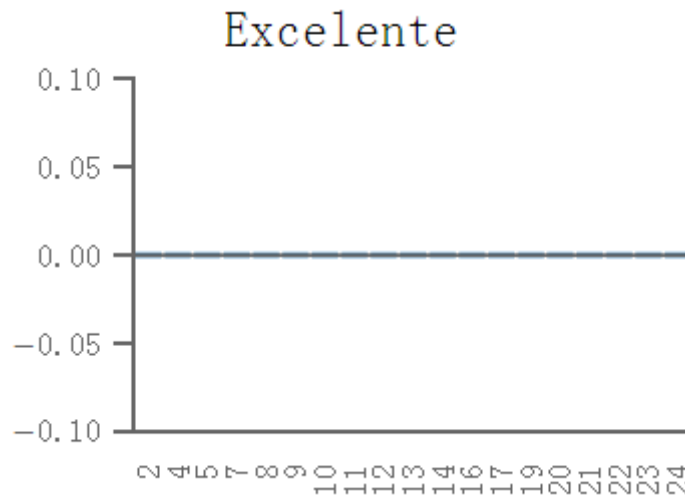


Figura 32 – Qualidade da Métrica NPA nas *Sprints* do Projeto SICG

RFC (Response For a Class)

A Figura 33 mostra a qualidade da métrica RFC. Esta métrica permanece com o intervalo de qualidade "Bom" constante ao longo das *sprints* do projeto.

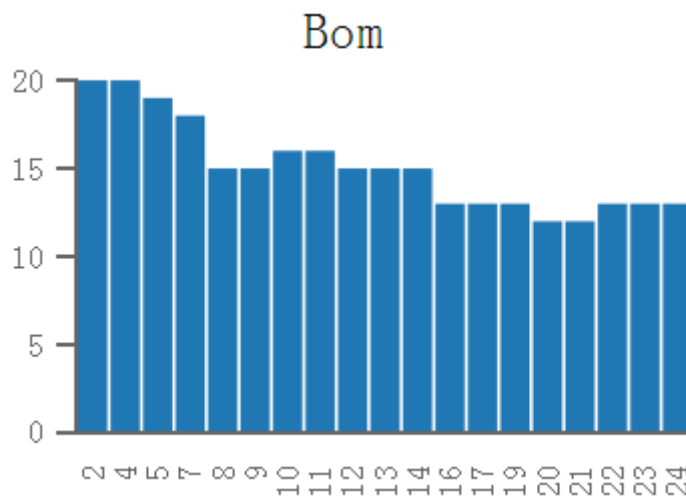


Figura 33 – Qualidade da Métrica RFC nas *Sprints* do Projeto SICG

A *Sprint* 24 é a última *sprint* do projeto e, portanto, é nela que foi entregue o código fonte final. Conforme a Fig. 34, na análise da qualidade do código fonte dessa *sprint*, observamos uma predominância do intervalo de qualidade "Excelente", o qual acontece em 7 métricas. O intervalo de qualidade "Bom" acontece em 3 métricas. O intervalo de qualidade "Regular" acontece em 1 métrica. E o intervalo de qualidade "Ruim" acontece em 1 métrica.

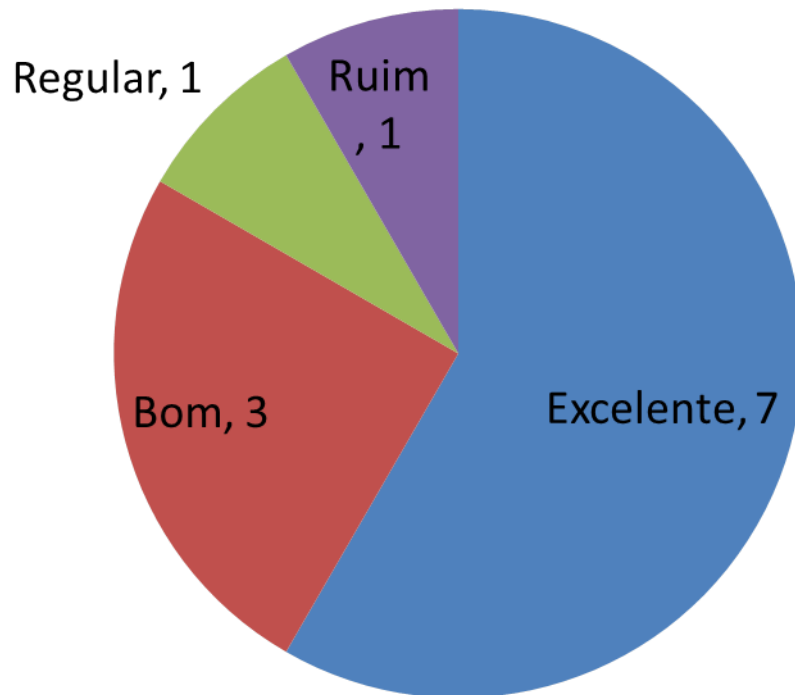


Figura 34 – Qualidade Geral do Código Fonte

É possível perceber claramente que o resultado da métrica CBO desvirtua do resultado das demais métricas. Como mostrado na Tab. (18), para que a métrica CBO seja rotulada como "Ruim" o valor dela deve estar acima de 9. Observando o gráfico da Fig. 27, vemos que na última *sprint* do projeto esta métrica está próxima do valor 400.

A Figura 35 mostra a percepção da qualidade do código fonte do SICG segundo os envolvidos no projeto (Scrum Master, Product Owner, Gestor do Contrato, Coordenador do Projeto, Fiscal Técnico e Desenvolvedores). Cerca de 66% dos envolvidos consideraram o intervalo de qualidade do código como "Excelente" e os demais consideraram o intervalo de qualidade como "Bom".

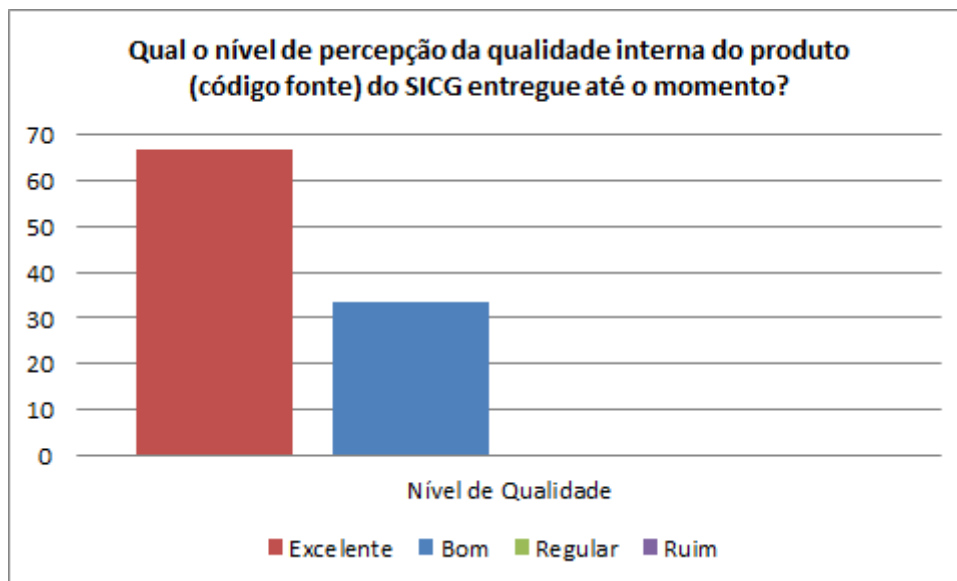


Figura 35 – Nível de Percepção da Qualidade do Código Fonte

Assim, no que diz respeito a análise estática do código realizada, a qualidade interna do produto pode ser considerada satisfatória ao compararmos com o melhor *software* livre encontrado na linguagem de programação Java, de acordo com os estudos de Meirelles (2013), sem considerarmos o impacto do resultado da métrica CBO. Além disso, há uma convergência com a também percepção satisfatória na dos envolvidos no projeto SICG, embora não tenhamos encontrado evidências de que o IPHAN realizava análise sistemática sobre tais métricas de qualidade de produto.

Ao analisarmos a qualidade interna do código fonte atribuindo um grande impacto do resultado da métrica CBO sobre a qualidade final (caso essa fosse a escolha do fiscal técnico do contrato), a qualidade do produto poderia ser considerada insatisfatória, e isso poderia contradizer a percepção dos usuários. Isso pode suscitar importantes discussões: i) o código pode estar altamente acoplado e possuir uma quantidade alta de oportunidades de refatoração no que diz respeito aos seguintes cenários: classes pouco coesa, interface dos métodos, classes com muitos filhos, classe com métodos grandes e/ou muitos condicionais, classe com muita exposição e complexidade estrutural ii) Essa alta medida pode estar sendo causada pelo uso de uma combinação de *frameworks*; iii) a configuração de métricas escolhida pode não ser adequada para a comparação com a configuração de métricas analisadas; iv) a escala da medição pode não ter sido adequada. ; v) a eficiência e eficácia da aplicação das práticas ágeis relacionadas a qualidade interna do produto não foi atingida. De qualquer modo, carecemos de estudos mais aprofundados e controlados para investigar melhor a relação de causa e efeito entre as variáveis observadas e medidas.

No entanto, segundo Meirelles (2013), o uso de uma única métrica isolada na análise pode levar a interpretações errôneas tanto do lado do cliente quanto do lado do

fornecedor. Com isso, essas métricas são reconhecidamente limitadas quando analisadas isoladamente. Assim, não é recomendado atribuir grande impacto do resultado do CBO na análise geral da qualidade interna do código fonte. O ideal é observar o comportamento de todas as métricas e extrair as informações necessárias para a melhoria da qualidade.

De acordo com as informações levantadas em entrevistas informais, o IPHAN disponibilizou o sistema desenvolvido para treinamento, por parte dos usuários, apenas na *sprint* 21 e em ambiente de produção apenas ao final do projeto. Essa implantação tardia fere o valor do Manifesto Ágil de "*Software operante...*" e poderia ter comprometido o funcionamento adequado final do sistema. Além disso, durante os ciclos do projeto, a vistoria técnica analisava se as classes estavam documentadas, como pedido no contrato, e se o modelo de dados era válido, o que evidencia a busca pela excelência da qualidade interna por parte dos gestores técnicos. Contudo, evidenciamos que a análise de qualidade do código fonte foi realizada apenas ao final do projeto por uma outra empresa contratada para aferir a qualidade, a RSI Informática Ltda.

Os resultados das métricas analisadas nesse estudo que receberam rótulos de qualidade "Excelente" ou "Bom" se devem mais a aspectos considerados pela empresa contratada do que a capacidade própria do IPHAN em aferir e considerar os resultados da análise desta como conhecimento para, por exemplo, ser utilizado para o faturamento das OS's.

O fato de padrões de qualidade de métricas de código fonte não terem sido especificados no contrato e o fato de uma análise de qualidade de código de fonte não ter sido realizada durante os ciclos do projeto figuraram como risco ao projeto, uma vez que poderia ter comprometido o sucesso e a manutenibilidade do sistema.

7 Conclusão

Neste trabalho foi constatado que é possível aplicar uma solução baseada em métodos ágeis e no pensamento *lean* sobre a gestão de um contrato.

Para a realização deste trabalho, algumas atividades foram essenciais: a análise de alguns modelos e normas, a revisão da literatura e a definição de um estudo de caso.

Com o projeto do estudo de caso, foi possível definir de forma objetiva as questões específicas de pesquisa que procuraram, por sua vez, responder a questão de pesquisa e ao problema definido neste trabalho. Com base nessas questões, os resultados das análises dos dados coletados de questionário, de documentos, do código fonte e de processos foram categorizados em três efeitos: efeitos sobre a entrega de ordens de serviço, efeitos sobre a satisfação do cliente e efeitos sobre a qualidade interna do código fonte.

Como unidade de estudo de caso, foi selecionado o contrato do Sistema Integrado de Conhecimento e Gestão (SICG) do Instituto do Patrimônio Histórico e Artístico Nacional (IPHAN). Trata-se de estudo exploratório, onde foram coletados dados qualitativos, que caracterizaram a organização contratante, a empresa contratada, o objeto do contrato e a solução de gestão de contrato definida. Também foram coletados dados quantitativos categorizados nos efeitos mencionados, os quais procuraram responder a questão de pesquisa: *Como o uso de métodos ágeis e do pensamento lean na gestão de contratos de fornecedores de desenvolvimento de software influenciaram no resultado final do contrato do ponto de vista do gestor de contrato e do fiscal técnico do contrato, que juntos gerenciam o contrato?*. Com isso, essa influência no resultado final do contrato foi analisada com relação a influência sobre as ordens de serviço, satisfação do cliente e qualidade interna do código fonte.

No que diz respeito aos efeitos sobre a entrega de ordens de serviço, conclui-se que o uso da solução, baseada em métodos ágeis e no pensamento na gestão desse contrato, resultou em uma entrega de uma versão de *software* funcional, no máximo, mensalmente; a entrega de apenas documentação figurou em apenas uma ordem de serviço do projeto, valorizando a fase de execução e entrega de produto funcional; os requisitos atendidos ao longo do projeto foram em torno de 78%; não houve aplicação de multas no projeto; e o custo final do projeto ultrapassou 1,8% o custo estimado.

No que diz respeito aos efeitos sobre a satisfação do cliente, a área requisitante ficou satisfeita com o produto entregue e a visibilidade do processo por parte do gestor do contrato foi alta, o que possibilitou tomada de decisões rápidas e uma comunicação efetiva com a empresa contratada.

No que diz respeito aos efeitos sobre a qualidade interna do código fonte, ao longo das *sprints* do projeto, a qualidade “Excelente” foi majoritária no que diz respeito às 11 métricas analisadas (LOC, ACCM, AMLOC, ACC, ANPM, DIT, LCOM4, NOC, NOM, NPA, RFC) e na percepção da maioria dos envolvidos do projeto a qualidade também foi definida como “Excelente”. No entanto, ao inserirmos a métrica CBO na análise, a qualidade é rotulada como “Ruim”. Uma investigação mais aprofundada poderia esclarecer se tal fator pode ser explicado pelo uso de uma combinação de diversos *frameworks* ocasionando um alto acoplamento do projeto, possibilitando, com isso, a apresentação de diversos cenários de limpeza de código.

Possivelmente uma solução automatizada de análise estática de código, por exemplo, como a utilizada neste estudo de caso, poderia facilitar a vistoria técnica sobre o código fonte, onde a informação resultante dessa análise pudesse auxiliar a tomada de decisão sobre o faturamento das ordens de serviço. Isto corroboraria com o Art. 25, inciso III, alínea b da Instrução Normativa nº 04 (BRASIL, 2010) que diz para realizar a *"avaliação da qualidade dos serviços realizados ou dos bens entregues e justificativas, de acordo com os Critérios de Aceitação definidos em contrato, a cargo dos Fiscais Técnico e Requisitante do Contrato"*.

Assim, a solução desenvolvida pelo IPHAN, aplicada no projeto SICG, resultou na entrega do *software* requisitado ao final do contratado. Embora esta solução seja passível de melhorias, pois há indícios de que alguns resultados apresentados não tenham sido totalmente satisfatórios, podemos concluir que o problema de pesquisa deste estudo *"Alguns contratos de desenvolvimento de software da organização não resultaram na entrega do software requisitado ao final do contrato."*, que ocorria frequentemente no órgão, não ocorreu na gestão de contratos ágil aplicada.

Como analisado no Capítulo 6, a solução de gestão de contratos definida pelo IPHAN não fere o que é determinado na Lei nº 8.666/93 ou na IN 04/2010 ou nos Princípios da APF. Ao mesmo tempo, a solução foi baseada nos valores, princípios e práticas do pensamento *lean* e de metodologias ágeis. As lições aprendidas e a criação de conhecimento fazem parte dos preceitos dos métodos ágeis e do pensamento *lean*, portanto, as recomendações, práticas e princípios que não foram evidenciados no contrato analisado nesse estudo de caso podem ser inseridos nos contratos futuros do órgão.

Como apresentado no Capítulo 5, as principais ameaças a validade de um estudo de caso são: validade do constructo, validade interna, validade externa e confiabilidade.

Com relação a validade do constructo, para a análise dos efeitos sobre as ordens de serviço, satisfação do cliente e qualidade interna do código fonte, foi utilizada a técnica GQM, a qual direcionou a definição sistemática de problema, objetivos e questões específicas de pesquisa e métricas. Além disso, foram utilizadas múltiplas fontes de evidências e informantes chaves, o que reforçou a validade do constructo. No entanto, as análises da

adequação da solução ao Pensamento *Lean*, ao Scrum, a Lei nº 8.666/93, a IN 04/2010 e aos Princípios da APF não foram realizadas a partir de uma técnica formal como o GQM, tais análises foram observadas ao decorrer desse estudo de caso a partir das múltiplas fontes de evidências coletadas. Como trabalhos futuros, pretende-se fortalecer a validade do constructo utilizando a técnica GQM para realização dessas análises de forma mais assertiva.

Com relação a validade interna, ela foi garantida por meio da triangulação de dados e de metodologia para apresentação e análise dos resultados apresentados no Capítulo 6. Os dados foram triangulados por meio de quatro fontes: documentação dos processos do projeto SICG, questionário (Apêndice A), entrevistas informais com os informantes-chaves (fiscal técnico e coordenador do projeto) e repositório do projeto (código fonte e documentos). Foram ainda coletados tanto dados qualitativos (percepções dos envolvidos no projeto) quanto dados quantitativos (qualidade do código fonte, documentação e ordens de serviço do projeto), o que figurou a triangulação de metodologia.

Com relação a validade externa, ela não foi atendida nesse estudo de caso por se tratar de um estudo de caso único com uma única unidade de análise. Além disso, este é um estudo de caso com propósitos exploratórios. Como trabalhos futuros, objetiva-se a ampliação desse estudo para múltiplos casos.

Com relação a confiabilidade, o banco de dados desse estudo de caso foi composto pelas ordens de serviço e atas de reuniões dos Processos nº 01450.011592/2010-30 e nº 01450.000845/2012-10, pelo repositório do projeto, o qual contém o código fonte e a documentação entregue em cada *sprint* do projeto, e pelos dados coletados do questionário localizado no Apêndice A. O banco de dados será disponibilizado juntamente com esse Trabalho de Conclusão de Curso, assim, garantindo a confiabilidade desse estudo de caso. Com isso, qualquer pesquisador que realizar a replicação desse estudo de caso utilizando o mesmo protocolo e as mesmas fontes de dados poderá encontrar os mesmos resultados.

Assim, como trabalhos futuros, objetiva-se a validação e refinamentos do protocolo deste estudo de caso, e em seguida, a ampliação deste estudo para outras organizações públicas brasileiras. Além disso, pretende-se realizar um relação mais sistemática entre o uso de métodos ágeis na gestão de contratos de fornecedores de desenvolvimento de *software* e os aspectos da legislação vigente.

Referências

- BASILI, V. R.; ROMBACH, H. D. *TAME: Integrating Measurement into Software Environments*. 1987. Disponível em: <<http://drum.lib.umd.edu/handle/1903/7517>>. Citado na página 126.
- BASILI, V. R.; ROMBACH, H. D. Goal Question Metric Paradigm. 1994. Citado na página 60.
- BECK, K. *Smalltalk Best Practice Patterns. Volume 1: Coding*. [S.l.]: Prentice Hall, Englewood Cliffs, NJ, 1997. Citado na página 127.
- BELL, S. C.; ORZEN, M. A. *Lean IT Enabling and Sustaining your Lean Transformation*. [S.l.: s.n.], 2011. Citado 11 vezes nas páginas 9, 130, 131, 132, 133, 134, 135, 136, 137, 138 e 139.
- BRASIL. *Lei nº 8.666/93, de 21 de Junho de 1993*: Regulamenta o art. 37, inciso xxi, da constituição federal, institui normas para licitações e contratos da administração pública e dá outras providências. [S.l.], 1993. Citado na página 23.
- BRASIL. *Instrução Normativa nº 04*: Dispõe sobre o processo de contratação de soluções de tecnologia da informação pelos órgãos integrantes do sistema de administração dos recursos de informação e informática (sisp) do poder executivo federal. [S.l.], 2010. Citado 2 vezes nas páginas 25 e 111.
- CHIDAMBER, S. R.; KEMERER, C. F. A Metrics Suite for Object-Oriented Design. *IEEE Transactions on Software Engineering*, v. 20, n. 6, p. 476–493, 1994. Citado 2 vezes nas páginas 126 e 127.
- CRUZ, C. S.; ANDRADE, E. L. P.; FIGUEIREDO, R. M. C. *Processo de Contratação de Serviços de Tecnologia da Informação para Organizações Públicas*. [S.l.: s.n.], 2011. Citado na página 23.
- CUNNINGHAM, W. 2001. Disponível em: <<http://agilemanifesto.org>>. Citado 3 vezes nas páginas 17, 45 e 46.
- FILHO, D. L. B. *Experiências com desenvolvimento ágil*. [S.l.: s.n.], 2008. Citado na página 45.
- GUEDES, P. *A gestão Lean de equipas informáticas*. 2012. Citado 2 vezes nas páginas 8 e 34.
- HIBBS, C.; JEWETT, S.; SULLIVAN, M. *The Art of Lean Software Development*. [S.l.: s.n.], 2009. Citado 4 vezes nas páginas 32, 39, 40 e 41.
- HIGHSMITH, J. *Agile Software Development Ecosystems*. [S.l.: s.n.], 2002. Citado na página 46.
- HITZ, M.; MONTAZERI, B. Measuring Coupling and Cohesion in Object-Oriented Systems. In: *Proceedings of International Symposium on Applied Corporate Computing*. [S.l.: s.n.], 1995. Citado na página 127.

- INSTITUTO DO PATRIMÔNIO HISTÓRICO E ARTÍSTICO NACIONAL. *Metodologia IPHAN de Gestão de Demandas de Desenvolvimento Ágil de Softwares*. Brasília, Brasil, 2013. Citado 4 vezes nas páginas 9, 141, 142 e 143.
- ISO/IEC 25023. *ISO/IEC 25023: Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality*. [S.l.], 2011. Citado na página 96.
- JARDIM, R. S. C. e G. *Os Cinco Passos do Pensamento Enxuto*. [S.l.: s.n.], 2010. Citado na página 135.
- JENSES, K. 2009. Disponível em: <<http://ketiljensen.wordpress.com/2009/10/31-/kanban-the-next-step-in-the-agile-evolution/>>. Citado 2 vezes nas páginas 8 e 42.
- JONES, T. C. *Applied Software Measurement: Assuring Productivity and Quality*. New York: McGraw-Hill, 1991. Citado na página 126.
- KATAYAMA, E. *A contribuição da indústria da manufatura no desenvolvimento de software*. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 16 e 33.
- KLIPP, P. *Getting Started with Kanban*. [S.l.: s.n.], 2011. Citado 3 vezes nas páginas 8, 42 e 43.
- KNIBERG, H.; SKARIN, M. *Kanban e Scrum obtendo o melhor de ambos*. [S.l.: s.n.], 2009. Citado 6 vezes nas páginas 8, 42, 44, 49, 50 e 51.
- LEAN INSTITUTE BRASIL. 2013. Disponível em: <http://www.lean.org.br/o_que_e.aspx>. Citado na página 136.
- LIMA, O.; SILVA, W. 2003. Disponível em: <<http://www2.ufersa.edu.br/portal/view/uploads/setores/241/texto%203.pdf>>. Citado na página 16.
- LORENZ, M.; KIDD, J. *Object-Oriented Software Metrics*. [S.l.]: Prentice Hall, 1994. Citado na página 127.
- MCCABE, T. J. A Complexity Measure. *IEEE Transactions Software Engineering*, v. 2, n. 4, p. 308–320, December 1976. Citado na página 126.
- MCCABE, T. J.; DREYER, L. A.; WATSON, A. H. Testing An Object-Oriented Application. *Journal of the Quality Assurance Institute*, v. 8, n. 4, p. 21–27, October 1994. Citado na página 126.
- MEIRELLES, P. R. M. *Monitoramento de métricas de código-fonte em projetos de software livre*. Tese (Doutorado) — Instituto de Matemática e Estatística – Universidade de São Paulo (IME/USP), 2013. Citado 3 vezes nas páginas 96, 108 e 126.
- MELO, C. et al. *Métodos ágeis no Brasil: estado da prática em times e organizações*. [S.l.: s.n.], 2012. Citado 2 vezes nas páginas 16 e 51.
- MINISTÉRIO DO PLANEJAMENTO, ORÇAMENTO E GESTÃO. *Guia Prático para Contratação de Soluções de Tecnologia da Informação*. Brasília, Brasil, 2011. Citado na página 22.

- MINISTÉRIO DO PLANEJAMENTO, ORÇAMENTO E GESTÃO. 2013. Disponível em: <<http://arquivo.governoeletronico.gov.br/mcti/index.htm>>. Citado 3 vezes nas páginas 8, 27 e 30.
- MORGAN, J. M.; LIKER, J. K. *The Toyota product development system: integrating people, process, and technology*. [S.l.: s.n.], 2006. Citado na página 137.
- NONAKA, I.; TAKEUCHI, H. *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation*. [S.l.: s.n.], 1995. Citado na página 37.
- PEREIRA, C.; SOUSA, C. *Metodologias de Desenvolvimento de SW - Metodologias "Ágeis"*. [S.l.: s.n.], 2011. Citado na página 17.
- POPPENDIECK, M.; POPPENDIECK, T. *Implementando o Desenvolvimento Lean de Software: Do conceito ao dinheiro*. [S.l.: s.n.], 2011. Citado 6 vezes nas páginas 16, 33, 35, 36, 37 e 38.
- RÊGO, G. Monitoramento de Métricas de Código-Fonte com suporte de um ambiente de Data Warehousing. 2014. Citado na página 98.
- RODRIGUES, M. J. F. *Implementação de práticas Lean numa linha de produção eletrônica*. [S.l.: s.n.], 2012. Citado na página 139.
- ROSENBERG, L. H.; HYATT, L. E. Software Quality Metrics for Object-Oriented Environments. *Crosstalk - the Journal of Defense Software Engineering*, v. 10, 1997. Citado na página 127.
- RUAS, F.; VAZ, W. 2013. Disponível em: <<http://www.slideshare.net/faruas/apresentacao-agile-brazil-2013>>. Citado na página 57.
- SALDANHA, B. L. F. *O Gerenciamento e suas origens*. [S.l.: s.n.], 2005. Citado na página 16.
- SCHWABER, K. *Scrum Development Process*. [S.l.: s.n.], 1996. Citado na página 47.
- SHARBLE, R.; COHEN, S. The Object-Oriented Brewery: A Comparison of Two Object-Oriented Development Methods. *Software Engineering Notes*, v. 18, n. 2, p. 60–73, 1993. Citado na página 128.
- SHIH, T. et al. Decomposition of Inheritance Hierarchy DAGs for Object-Oriented Software Metrics. In: *Workshop on Engineering of Computer-Based Systems (ECBS 97)*. [S.l.: s.n.], 1997. p. 238. Citado na página 127.
- SUTHERLAND'S, J. *Scrum Handbook*. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 47 e 48.
- TRIBUNAL DE CONTAS DA UNIÃO. *Guia de boas práticas em contratação de soluções de tecnologia da informação*. Brasília, Brasil, 2012. Citado na página 22.
- TRIBUNAL DE CONTAS DA UNIÃO. *Acórdão nº 2314*. Brasília, Brasil, 2013. Citado 8 vezes nas páginas 16, 18, 46, 52, 53, 55, 86 e 91.
- VIEIRA, E. et al. *A Teoria Geral de Sistemas, Gestão do Conhecimento e Educação a Distância*. 2005. Disponível em: <<http://www.redalyc.org/articulo.oa?id=273520153003>>. Citado na página 16.

WILLIAMS, S. 2009. Disponível em: <http://wheels.blogs.nytimes.com/2009/09/04/toyota-quality-control-includes-greensleeves-and-popeye/?_r=1>. Citado 2 vezes nas páginas 9 e 139.

YIN, R. K. *Estudo de Caso - Planejamento e Métodos*. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 19 e 65.

Apêndices

APÊNDICE A – Questionário Gestão do Contrato



Um Estudo de Caso sobre a Gestão de Contratos de Fornecedores de Desenvolvimento de Software

No Brasil, o cenário das contratações públicas de fornecedores de desenvolvimento de software vem passando por mudanças. Algumas organizações públicas tem compactuado entendimento de que os instrumentos contratuais, hoje em vigor, não priorizam a entrega de software, tampouco sua qualidade interna e valor de negócio. Isso contribui para que projetos terminem sem sucesso, o que onera os cofres públicos.

Com isso, iniciaram investimentos para adotar contratações de serviços de desenvolvimento de software utilizando métodos ágeis. Contudo, cada uma dessas organizações têm vivenciado diferentes experiências e vêm compartilhando dificuldades comuns no que diz respeito às limitações impostas pelo normativo de contratação de software.

A utilização de metodologias ágeis e semelhantes em contratações de serviços de tecnologia da informação está ganhando espaço nas organizações públicas brasileiras e gerando questões importantes de estudo para a academia, e, portanto, tornam-se necessários estudos que evidenciem os resultados advindos do uso de métodos ágeis neste contexto.

Com isso, o objetivo do trabalho de conclusão de curso em Engenharia de Software da Universidade de Brasília, da aluna Aline Gonçalves dos Santos, sob orientação do professor Hilmer Rodrigues Neri, é realizar uma investigação empírica do uso de tais abordagens na gestão de contratos públicos de terceirização de software. Para tanto, pretende-se realizar um estudo de caso no Instituto do Patrimônio Histórico Artístico e Nacional (IPHAN).

Objetivo do Estudo de Caso

O objetivo do estudo de caso é analisar a influência do uso de métodos ágeis e do Pensamento Lean no contrato do Sistema Integrado de Conhecimento e Gestão (SICG) com a empresa EGL - Engenharia a partir dos dados coletados da documentação, observação e entrevistas.

O processo de realização de estudo de caso contará com as etapas de planejamento, coleta, análise e relato dos resultados. Após a coleta dos dados, estes serão categorizados e organizados de forma que possam ser utilizados na análise. Após a análise, todo o processo utilizado e os resultados serão relatados no trabalho de conclusão de curso.

Com isso, uma das principais contribuições desse trabalho será evidenciar para os gestores de contratos, para os demais envolvidos no processo de gestão e desenvolvimento de software terceirizado e para a academia os resultados do uso de métodos ágeis no contrato do SICG.

Esta pesquisa não é capaz de identificar os respondentes, sendo garantido o anonimato.

Continuar »

50% concluído

*Obrigatório

Questionário Gestão de Contrato do SICG

Função/Papel: *

Organização: *

IPHAN

EGL - Engenharia

Outro:

Qual a quantidade de pessoas do IPHAN que trabalha com o contrato SICG? *

1 a 3

4 a 6

7 a 9

10 a 12

Maior que 12

Quais desses atores existem dentro do projeto SICG por parte do IPHAN? *

Gestor do contrato

Fiscal técnico do contrato

Fiscal administrativo do contrato

Fiscal requisitante do contrato

Área requisitante

Área de tecnologia da informação

Equipe de planejamento da contratação

Dono do Produto (Product Owner)

Outro:

Qual a quantidade de pessoas da EGL - Engenharia que trabalha com o contrato SICG? *

1 a 3

4 a 6

7 a 9

10 a 12

Maior que 12

Quais desses papéis existem dentro do desenvolvimento do software do projeto SICG na EGL - Engenharia? *

- Desenvolvedor
- Analista de requisitos
- Analista de qualidade
- Designer
- Testador
- Scrum Master
- Gestor do Projeto
- Preposto
- Outro:

Quais metodologias foram utilizadas na gestão de contrato do IPHAN antes do projeto SICG? *

Marque quantos itens desejar.

- [MPS.BR](#)
- RUP
- Nenhuma metodologia
- Desconheço
- Outro:

Qual metodologia foi utilizada no desenvolvimento de software da EGL - Engenharia no Projeto SICG? *

- Scrum
- Extreme Programming (XP)
- Scrum/XP (Híbrido)
- Cascata
- RUP
- Nenhuma metodologia
- Desconheço
- Outro:

Qual era a experiência do fiscal técnico do contrato do IPHAN com o uso de métodos ágeis e do pensamento lean antes do início do contrato? *

- Pouca (não conhecia nada sobre esses métodos e nunca tinha trabalhado com eles)
- Regular (conhecia os métodos mas nunca tinha trabalhado com eles)
- Alta (conhecia os métodos e já tinha trabalhado com eles)
- Desconheço

Qual era a experiência do gestor do contrato do IPHAN com o uso de métodos ágeis e do pensamento lean antes do início do contrato? *

- Pouca (não conhecia nada sobre esses métodos e nunca tinha trabalhado com eles)
- Regular (conhecia os métodos mas nunca tinha trabalhado com eles)
- Alta (conhecia os métodos e já tinha trabalhado com eles)
- Desconheço

Qual era a experiência da empresa EGL - Engenharia com o uso de métodos ágeis e do pensamento lean antes do início do contrato? *

- Pouca (não conhecia nada sobre esses métodos e nunca tinha trabalhado com eles)
- Regular (conhecia os métodos mas nunca tinha trabalhado com eles)
- Alta (conhecia os métodos e já tinha trabalhado com eles)
- Desconheço

Qual era a experiência dos desenvolvedores da empresa EGL - Engenharia com a linguagem de programação utilizada no desenvolvimento? *

- Pouca (conhecimento básico sobre a linguagem e poucos trabalhos desenvolvidos)
- Regular (conhecimento regular da linguagem e alguns trabalhos desenvolvidos)
- Alta (alto conhecimento da linguagem e muitos trabalhos desenvolvidos)
- Desconheço

O que motivou o uso de métodos ágeis e do pensamento lean na gestão de contratos no IPHAN? *

Marque quantos itens desejar.

- A entrega de software funcional era pouco frequente nos contratos anteriores.
- A visibilidade do processo era baixa.
- O fiscal técnico do contrato e o gestor do contrato não estavam exercendo os seus papéis como deveriam.
- A satisfação do cliente era baixa.
- A qualidade do produto era preocupante.
- Os requisitos de software não era atendidos.
- O custo era alto
- Nenhum contrato anterior foi finalizado com sucesso (entrega do produto requerido).
- Documentação desnecessária estava sendo produzida e entregue.
- Desconheço
- Outro:

Quais foram as práticas ágeis de engenharia de software utilizadas no projeto SICG? *

- Padrões de Codificação
- TDD (Test Driven Development)
- BDD (Behavior Driven Development)
- Programação em Par
- Refatoração de Código
- Integração Contínua
- Controle de Versão
- Entregas Frequentes
- Código Limpo
- Revisão de Código
- Testes Unitários
- Testes de Integração
- Testes de Aceitação
- Testes de Fumaça (Smoke Testing)
- Histórias de Usuário
- Planning Poker
- Planejamento das Releases
- Planejamento das Iterações
- Backlog do Produto
- Backlog da Sprint
- Quadro Kanban
- Limite de Trabalho em Progresso (WIP)
- Burndown Charts
- Retrospectivas
- Definição de Pronto
- Tempo de Ciclo (Lead time)
- Reunião Diária
- Equipes Auto-organizadas
- Times Pequenos
- Scrum
- Programação Extrema (XP)
- Lean
- Outro:

Qual era a frequência de entrega de software antes do projeto SICG no IPHAN? *

Em caso de mais de um projeto, mais de uma opção pode ser marcada.

- Mensal
- Trimestral
- Semestral
- Anual
- Não havia entrega
- Desconheço

Qual é a frequência de entrega de software funcional no projeto SICG? *

- Mensal
- Trimestral
- Semestral
- Anual
- Não houve entrega
- Desconheço

De forma geral, qual foi a porcentagem de requisitos atendidos conforme solicitados em cada ordem de serviço? *

- Menos de 10%
- De 10% a 30%
- De 30% a 50%
- De 50% a 70%
- De 70% a 90%
- Mais de 90%
- Desconheço

Quantas multas foram aplicadas até o momento no projeto SICG? ***Qual o nível de percepção da qualidade interna do produto (código fonte) do SICG entregue até o momento? ***

- Excelente
- Bom
- Regular
- Ruim
- Desconheço

O quanto de visibilidade do processo o gestor do contrato tem durante a execução do projeto Sistema Integrado de Conhecimento e Gestão (SICG)? *

Visibilidade das funcionalidades que estão sendo desenvolvidas, do que já foi feito e do que será feito, dentre outras particularidades do processo.

- Alta
- Média
- Baixa
- Desconheço

Qual o nível de satisfação com o software entregue até o momento no projeto SICG? *

- Muito Satisfeito
- Satisfeito
- Neutro
- Insatisfeito
- Muito Insatisfeito
- Desconheço

Quais as principais barreiras enfrentadas na adoção de metodologias ágeis na gestão de contrato do IPHAN? *

- Colaboração do Cliente
- Resistência à Mudança
- Habilidade de mudar a cultura organização
- Limitações de custo
- Complexidade ou tamanho do projeto
- Normativos
- Princípios da Administração Pública Federal
- Falta de experiência no uso dos métodos
- Não teve barreiras
- Desconheço
- Outro:

Você desejaria informar outros efeitos percebidos da aplicação da solução explicada abaixo no que diz respeito a satisfação do cliente, entrega de ordens de serviço e qualidade interna do produto (código fonte)?

A solução com métodos ágeis e o pensamento lean consiste na utilização dos valores e princípios ágeis na gestão do contrato e na utilização da ferramenta Kanban.

APÊNDICE B – Métricas de Código Fonte

Neste trabalho foram selecionadas doze métricas de código fonte levantadas e categorizadas por Meirelles (2013): métricas de tamanho e complexidade e métricas de orientação à objetos.

As métricas de tamanho e complexidade selecionadas foram:

LOC (Lines of Code): número de Linhas de Código foi uma das primeiras métricas utilizadas para medir o tamanho de um software. São contadas apenas as linhas executáveis, ou seja, são excluídas linhas em branco e comentários (JONES, 1991).

ACCM (Average Cyclomatic Complexity per Method): média da Complexidade Ciclomática por Método mede a complexidade dos métodos ou funções de um programa. Essa métrica pode ser representada através de um grafo de fluxo de controle (MCCABE, 1976). O uso de estruturas de controle, tais como, if, else, while aumentam a complexidade ciclomática de um método.

AMLOC (Average Method Lines of Code) - essa medida indica se o código está bem distribuído entre os métodos. Quanto maior, mais pesados são os métodos. É preferível ter muitas operações pequenas e de fácil entendimento que poucas operações grandes e complexas (MEIRELLES, 2013).

As métricas de orientação à objetos selecionadas foram:

ACC (Afferent Connections per Class): conexões Aferentes por Classe é o número total de classes externas de um pacote que dependem de classes de dentro desse pacote. Quando calculada no nível da classe, essa medida também é conhecida como Fan-in da classe, medindo o número de classes das quais a classe é derivada e, assim, valores elevados indicam uso excessivo de herança múltipla (MCCABE; DREYER; WATSON, 1994) (CHIDAMBER; KEMERER, 1994).

ANPM (Average Number of Parameters per Method): calcula a média de parâmetros dos métodos da classe. Seu valor mínimo é zero e não existe um limite máximo para o seu resultado, mas um número alto de parâmetros pode indicar que um método pode ter mais uma responsabilidade (BASILI; ROMBACH, 1987)

CBO (Coupling Between Objects): é o número total de classes dentro de um pacote que dependem de classes externas ao pacote. Quando calculada no nível da classe, essa medida também é conhecida como Fan-out da classe (CHIDAMBER; KEMERER, 1994)

DIT (Depth of Inheritance Tree): profundidade da Árvore de Herança é o número de superclasses ou classes ancestrais da classe sendo analisada. São contabilizadas apenas as superclasses do sistema, ou seja, as classes de bibliotecas não são contabilizadas. Nos casos onde herança múltipla é permitida, considera-se o maior caminho da classe até uma das raízes da hierarquia. Quanto maior for o valor DIT, maior é o número de atributos e métodos herdados, e, portanto, maior é a complexidade (SHIH et al., 1997).

LCOM4 (Lack of Cohesion in Methods): falta de Coesão entre Métodos. Originalmente proposto por Chidamber e Kemerer (1994) como LCOM não teve uma grande aceitabilidade. Após críticas e sugestões a métrica foi revisada por Hitz e Montazeri (1995), que propôs a LCOM4. Para calcular LCOM4 de um módulo, é necessário construir um gráfico não-orientado em que os nós são os métodos e atributos de uma classe. Para cada método, deve haver uma aresta entre ele e um outro método ou variável que ele usa. O valor da LCOM4 é o número de componentes fracamente conectados nesse gráfico.

NOC (Number of Children): número de Filhos é o número de subclasses ou classes filhas que herdam da classe analisada (ROSENBERG; HYATT, 1997). Deve se ter cautela ao modificar classes com muitos filhos, pois uma simples modificação de assinatura de um método, pode criar uma mudança em muitas classes.

NOM (Number of Methods): número de Métodos é usado para medir o tamanho das classes em termos das suas operações implementadas. Essa métrica é usada para ajudar a identificar o potencial de reuso de uma classe. Em geral, as classes com um grande número de métodos são mais difíceis de serem reutilizadas, pois elas são propensas a serem menos coesas (LORENZ; KIDD, 1994).

NPA (Number of Public Attributes): mede o encapsulamento. Os atributos de uma classe devem servir apenas às funcionalidades da própria classe. Portanto, boas práticas de programação recomendam que os atributos de uma classe devem ser manipulados através dos métodos de acesso (BECK, 1997)

RFC (Response For a Class): respostas para uma Classe é número de métodos dentre todos os métodos que podem ser invocados em resposta a uma mensagem enviada

por um objeto de uma classe (SHARBLE; COHEN, 1993).

Anexos

ANEXO A – Princípios e Práticas do Lean na Manufatura

A.1 Princípios

A.1.1 Constância de Propósitos

O princípio de Constância de Propósitos diz respeito a manter a clareza dos objetivos importantes de longo prazo e é um dos constituintes da base da pirâmide dos princípios porque prover a persistência necessária para influenciar o comportamento de todos dentro da organização. Quando diariamente o comportamento muda, a cultura da organização também muda. A Constância de Propósitos tem o foco no pensamento e no comportamento, fazendo com que todos caminhem na mesma direção. As pessoas devem ser encorajadas a desafiar o modo como as coisas são feitas e a perceber as dificuldades e oportunidades de melhoria, a partir disso, é possível ter a resolução de problemas de forma colaborativa (BELL; ORZEN, 2011).

Com isso, líderes executivos têm a responsabilidade de definir objetivos estratégicos e criar constância de propósitos. Os gestores tem como responsabilidade eliminar impedimentos, estabilizar os processos e ajudar os trabalhadores a desenvolver habilidades de resolução de problemas, para que eles possam sentir-se donos do seu trabalho e possam assumir responsabilidade sobre a melhoria contínua (BELL; ORZEN, 2011).

A.1.2 Respeito às Pessoas

O segundo princípio base é o respeito às pessoas. Todos os indivíduos possuem uma única coleção de experiências e fazem distintas contribuições quando participam de um processo de melhoria. A resolução de problemas coletiva só ocorre quando existe respeito pelas pessoas em todos os níveis hierárquicos de uma organização. O respeito direciona o desenvolvimento dos trabalhadores, encoraja participação e melhora a relação entre fornecedor e cliente (BELL; ORZEN, 2011).

Além disso, respeito às pessoas encoraja alcançar a excelência profissional e o pontencial criativo. Em um ambiente de aprendizado e desenvolvimento colaborativo as pessoas sabem que suas ideias possuem valor e sentem-se mais estimuladas em melhorar diariamente. Pessoas estimuladas e respeitadas não só geram sucesso individual como também sucesso coletivo e organizacional.

A.1.3 Melhoria Contínua e Perfeição

O último princípio da base da pirâmide é a melhoria contínua em busca da perfeição. As soluções imediatas, embora possam ser adequadas para hoje, são na melhor das hipóteses temporárias. A mudança é constante e novas ideias são necessárias sempre que o padrão de trabalho atual não produz mais os resultados esperados. Em uma cultura Lean, os trabalhadores devem aceitar as mudanças como inevitáveis e de forma proativa enfrentarem os desafios. Cada indivíduo deve ver o seu trabalho como tendo dois componentes inseparáveis: trabalho diário e melhoria diária (BELL; ORZEN, 2011).

No entanto, as pessoas possuem hábitos diferentes. Muitas pessoas gostam da mudança, mas não gostam de serem mudadas, pois para mudar é preciso sair da zona de conforto. Com isso, as pessoas costumam resistir a mudanças e impedir que a criatividade e inovação sejam desenvolvidas dentro de si. Quando Lean é visto como um programa ou projeto, com início e fim, elas costumam aceitar e realizar o que é pedido. Quando o projeto termina, voltam a praticar os mesmos hábitos e comportamentos de antes. Porém, quando coletivamente as pessoas reconhecem que se não estão melhorando, estão ficando para trás, a compreensão do trabalho diário muda radicalmente. Esta percepção inspira novas ideias e a reinvenção diária, influenciando no sucesso de toda a organização (BELL; ORZEN, 2011).

A.1.4 Comportamento Proativo

O princípio que fica sobre a base da pirâmide é o de comportamento proativo, que significa tomar iniciativa, assumindo pessoal responsabilidade pela qualidade do próprio trabalho e pelo ambiente de trabalho. Ser proativo significa aproveitar a oportunidade para fazer a diferença dia a dia (BELL; ORZEN, 2011).

Em os 7 Hábitos de Pessoas Altamente Eficazes, o Dr. Stephen Covey introduz um modelo chamado Matriz de Gerenciamento de Tempo, dividindo o trabalho em quatro quadrantes com base na importância e urgência, como ilustrado na Fig. (36).

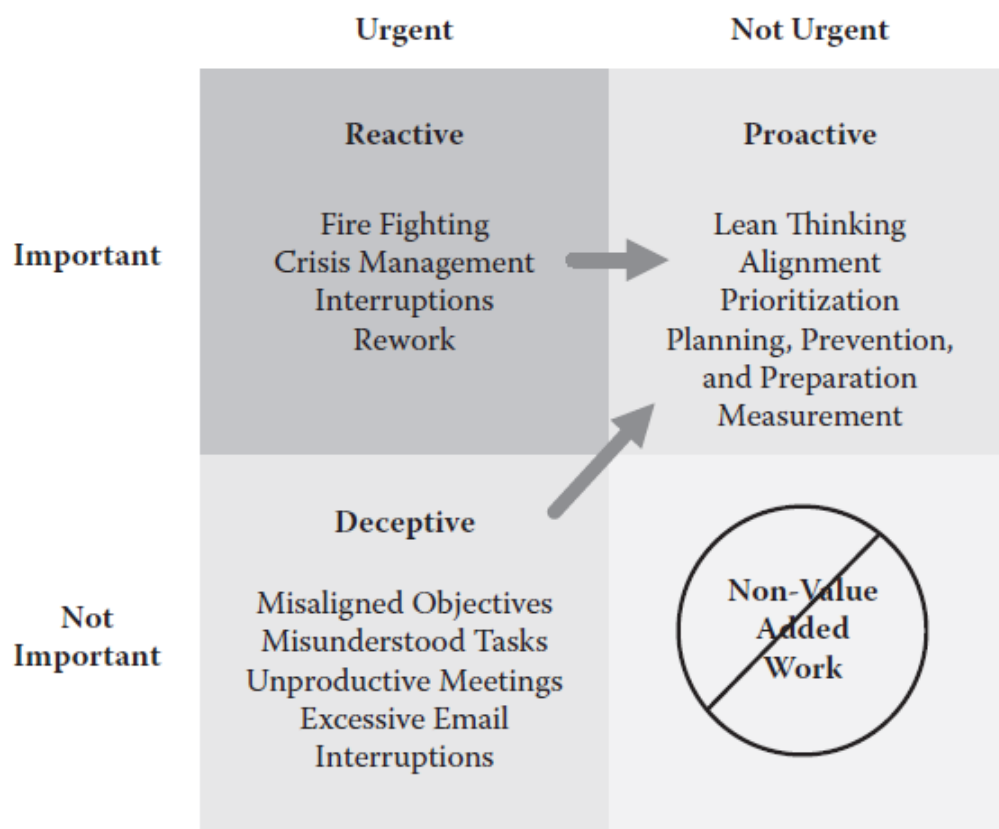


Figura 36 – Matriz de Gerenciamento de Tempo (BELL; ORZEN, 2011)

Ele ressalta que o trabalho de maior valor encontra-se no quadrante importante e não urgente, onde o planejamento proativo, a prevenção, a preparação e a aprendizagem serão mais fortemente desenvolvidos. Trabalho não planejado concentra-se no quadrante importante e urgente (combate de forma reativa) ou não importante e urgente (atividades que parecem importante devido à sua urgência). Em ambos os casos, eles roubam tempo e recursos que devem ser usados para abordar proativamente o trabalho importante e não urgente. O comportamento reativo, geralmente, deixa desperdício e impede o progresso.

Lean desloca-se da ênfase em sempre realizar trabalho não planejado para o trabalho proativo de melhoria contínua. Esta é uma abordagem eficaz porque quanto mais tempo é gasto na melhoria contínua, menos trabalho não planejado e urgente aparecem e mais tempo é liberado para trabalhar de forma proativa.

O próximo nível da pirâmide está relacionado ao conhecimento em três perspectivas: voz do cliente, qualidade na raiz e pensamento sistêmico.

A.1.5 Voz do Cliente

A maioria dos processos têm tanto os clientes internos que irão receber os produtos de trabalho e o rendimento deles quanto os clientes externos (parceiros comerciais e

usuários finais) que recebem o produto final, serviço ou informação. Para compreender claramente a voz do cliente é preciso estar envolvido com os clientes, seja ele quem for. Ao ouvirmos constantemente a voz do cliente temos a garantia que estamos focados nas questões certas e fazendo melhorias que serão valiosas para os clientes atuais e futuros. Entender as necessidades e os desejos dos clientes mais claramente que o concorrente faz com que a empresa seja mais competitiva e ágil (BELL; ORZEN, 2011).

A.1.6 Qualidade na Raíz

A qualidade na raiz significa fazer as coisas certas da primeira vez sempre, o trabalho com problemas não é enviado para etapa seguinte ou para o cliente. A abordagem de concertar mais tarde um problema é praticado em muitas organizações devido a pressão do prazo, inadequado treinamento e conhecimento não adequado do processo. A melhor hora de resolver um problema é no momento que ele acontece, pois a causa ainda está fresca, os trabalhadores estão atentos e isso previne a adição de defeitos até que a causa do problema seja encontrada. O foco do trabalho diário deve ser produzir qualidade desde o início (BELL; ORZEN, 2011).

Em um ambiente Lean existe a obrigação de parar e corrigir problemas e um comprometimento coletivo de não deixar que os defeitos conhecidos cheguem até o cliente. Por meio do trabalho padronizado e do treinamento, todo o esforço é feito para garantir que o problema ou defeito não seja repetido novamente. As pessoas assumem a responsabilidade do trabalho que elas passam adiante, independentemente de onde veio o problema. Essa mudança fundamental na atitude reduz o desperdício e a frustração. Com isso, quando a qualidade na raiz toma conta, mais tempo é disponibilizado para desenvolver o que o cliente está pagando, o que por sua vez melhora a produtividade, custo, qualidade e moral (BELL; ORZEN, 2011).

A.1.7 Pensamento Sistêmico

O terceiro elemento do terceiro nível da pirâmide é o pensamento sistêmico: a capacidade de visualizar a interligação entre os processos que compõem a cadeia de valor inteira embora esteja ciente da interdependência de causa e efeito que podem criar valor ou criar desperdícios (BELL; ORZEN, 2011).

A cadeia de valor é composta de todos os processos, atividades e tarefas usados para gerar um produto ou serviço desde a sua concepção até a sua entrega ao cliente, e inclui todas as informações, procedimentos e materiais. Para evitar soluções que criem otimização localizada, o Pensamento Lean requer conhecimento da natureza de todas as regras de negócio e fluxos de informações.

Esta não é a forma natural que a mente humana e as organizações trabalham,

as pessoas tendem a concentrar-se em partes específicas de um quebra-cabeça em vez de concentrar-se em toda a imagem. Medidas inadequadas e os incentivos muitas vezes reforçam essa estreita forma de concentração.

Quando a solução de problemas baseia-se em uma compreensão clara da cadeia de valor global e dos clientes que são atendidos por ela, as empresas evitam o erro comum de realizar melhorias locais, que muitas vezes transferem ineficiências e desperdícios de uma área para a outra. Equipes multifuncionais fornecem a amplitude de entendimento necessário para o pensamento sistêmico, abrangendo o fluxo de valor, ligando o conhecimento e a compreensão de cada membro da equipe.

A perspectiva holística mostrada aqui pode ser desconhecida para muitos profissionais que passaram suas carreiras a aperfeiçoar as habilidades em uma área especializada da empresa. Um pensamento sistêmico estimula o potencial criativo dos trabalhadores, ampliando seus horizontes e desafiando a amplitude de sua percepção. Para fazer melhorias que impactam no que é recebido pelo cliente externo, a cadeia de valor deve ser vista como um sistema. O mapeamento do fluxo de valor e o pensamento sistêmico são complementares, ambos ajudam as pessoas a ver processos de negócios em um novo contexto: fluxo. Assim, o pensamento sistêmico permite ver o todo, criando um fluxo de valor para o cliente (BELL; ORZEN, 2011).

A.1.8 Fluxo Contínuo, Produção Puxada e *Just in Time*

O próximo nível da pirâmide concentra-se no fluxo: a progressão ininterrupta de materiais, serviços e informações. Como Jeffrey Liker enfatiza no “Toyota Way”: "permitir que o trabalho flua sempre que puder, quando o fluxo é interrompido, utilize sinais para puxar o início do trabalho".

O fluxo é conseguido por meio da eliminação de atrasos e interrupções durante toda a cadeia de valor. O mapeamento do fluxo de valor é uma ferramenta eficaz para identificar, quantificar e eliminar o desperdício. O fluxo de informações produz a transparência e visibilidade necessária para coordenar de forma eficiente o fluxo de trabalho. Quando a informação é usada para nivelar a demanda, equilibrar a capacidade e melhorar a qualidade, o fluxo é melhorado e valor é entregue ao cliente de forma rápida. Quando não há interrupções em uma série de tarefas, o trabalho flui sem problemas. Tipicamente, o trabalho irá fluir até encontrar uma barreira que impeça sua continuidade, por exemplo, o transporte para o outro local (BELL; ORZEN, 2011).

A produção puxada diz respeito ao cliente puxar a cadeia de valor. Ou seja, o cliente determina qual é o produto que ele deseja. Com isso, a produção é feita sobre demanda. A produção em massa deveria ser eliminada, pois ela empurra o produto para o cliente sem que ele tenha oportunidade de decidir o que realmente deseja. Assim, a

produção puxada inverte o valor produtivo: as empresas não mais empurram os produtos para o consumidor por meio de descontos e promoções. O consumidor é que passa a “puxar o fluxo do valor”, reduzindo estoques e valorizando o produto (BELL; ORZEN, 2011).

O *Just in Time* como dito na seção anterior é um dos pilares do Pensamento Lean e estar relacionado aos dois conceitos anteriores. Ele tem como objetivo eliminar todas as fontes de desperdício e tudo o que não acrescenta valor à organização. O princípio para atingir este fim é simples: só produzir o que é pedido pelo cliente e só quando ele o pretende, ou seja, não manter estoques, seja de produtos acabados ou intermediários.

A.1.9 Cultura

O princípio localizado no topo da pirâmide é a cultura: ela representa crenças compartilhadas de uma organização e o valor, os quais manifestam-se na atitude e no comportamento dos membros da organização. Cultura é o resultado da mudança de comportamento, em Lean ela está relacionada a melhoria contínua e a proatividade das pessoas na resolução de problemas. A mudança cultural pode resultar em um desempenho superior, na vantagem competitiva e em bons resultados financeiros (BELL; ORZEN, 2011).

A evolução de uma cultura Lean geralmente começa com adoção de práticas de melhoria contínua, seguida pela formação de um comportamento orientado a sistemas e orientado por valores e princípios comuns. As práticas do Lean fornecem estrutura e capacitação. Os sistemas desenvolvem práticas em comum. E os princípios fornecem a base que reforça os padrões culturais e o comportamento diário.

A.2 Conceitos de Valor e Desperdício

O principal foco do Lean é a resolução de problemas com o propósito de entregar valor ao cliente, com base na sistemática eliminação de desperdícios ao longo do fluxo ou da cadeia de valor. Portanto, esses três conceitos: valor, fluxo de valor e desperdícios precisam estar claros.

Para um entendimento mais conciso do Pensamento Lean é importante ainda ter em mente que o termo desperdício recebe uma conotação específica e uma autêntica subordinação à ideia de valor. Ou mais especificamente, ao valor percebido pelos clientes, considerando suas expectativas, necessidades e desejos. A melhor maneira para se identificar os desperdícios, segundo o pensamento Lean, é colocar-se na posição do cliente e refletir criticamente sobre os processos de produção na forma como são realmente feitos (JARDIM, 2010).

A.2.1 Valor

O ponto de partida para o Pensamento Lean consiste em definir o que é valor e o fluxo do valor. Diferente do que muitos pensam, não é a empresa, e sim o cliente quem define o que é valor. Para ele, a necessidade gera o valor e cabe às empresas determinarem qual é essa necessidade. A partir disso, pode-se procurar satisfazer essa necessidade e cobrar um preço específico sobre ela (LEAN INSTITUTE BRASIL, 2013). Assim, valor é aquilo que o cliente deseja e pelo que ele paga.

A.2.2 Cadeia de Valor

O fluxo de valor é composto por todo o ciclo de vida dos processos requeridos para gerar serviços, produtos e informação do conceito ao cliente. Isto inclui todas as atividades, que criam ou não valor. O pensamento sistêmico ajuda as pessoas a perceberem o processo e entender o valor da perspectiva do cliente.

Para isso, é preciso dissecar a cadeia produtiva e separar os processos em três tipos: os que efetivamente geram valor, aqueles que não geram valor, mas são importantes para a manutenção do processo e para a qualidade do processo e do produto, e aqueles que não agregam valor e que devem ser eliminados (os desperdícios) (LEAN INSTITUTE BRASIL, 2013).

A.2.3 Os Três Ms

De acordo com o Lean, os desperdícios podem ser divididos em três categorias, conhecidas como os três Ms: mura, muri e muda. O Mura significa irregularidade e variação, ele representa a inconsistência no fluxo de trabalho causada pelas mudanças, variedade e qualidade desejadas pelo cliente. É preciso saber minimizar os impactos causados pelo mura por meio da padronização do processo. O Muri significa sobrecarga, que representa a carga excessiva sobre as pessoas ou sobre os equipamentos, o que pode causar stress, erros e retrabalho. É preciso saber remover sobrecargas por meio da padronização do processo e gerenciamento adequado da demanda. E o Muda, que diz respeito ao desperdício em si, o qual na produção da Toyota foram categorizados em sete tipos: superprodução, estoque, tempo de espera, transporte, processos inadequados, movimentação de pessoas e correção devido à defeitos (BELL; ORZEN, 2011).

A.3 Práticas e Ferramentas

O Pensamento Lean sugere um conjunto de práticas e ferramentas que podem ser aplicadas na organização a fim de atingir os objetivos defendidos por ele. Vale ressaltar que as práticas aqui apresentadas são aquelas consideradas principais para este trabalho,

a organização deve buscar aquelas que melhor se adequem ao seu contexto. Como dito anteriormente, as ferramentas servem de estrutura e meio de capacitação, é importante que os trabalhadores, sobretudo, vivam diariamente os princípios da cultura Lean.

A.3.1 A3 Thinking

O pensamento A3 é a aplicação consistente do PDCA (*Plan, Do, Check and Act*) na resolução de problemas para identificar o melhor caminho para enfrentar desafios e oportunidades. O pensamento A3 guia as atividades da equipe em direção a uma correta definição dos problemas ou oportunidades. A ferramenta usada no pensamento A3 é o relatório A3, que se refere a um formato padronizado de comunicação que expressa o processo de resolução de problema em uma folha de papel em tamanho A3.

O uso de apenas uma folha para expressar todo o conhecimento faz com que as pessoas refinem seus pensamentos de modo que todas as questões e soluções sejam expostas de forma simples (MORGAN; LIKER, 2006).

A.3.2 Mapeamento da Cadeia de Valor

O mapeamento da cadeia de valor é uma das ferramentas mais utilizadas no Lean, ele permite identificar todas as atividades de um processo da organização que criam ou não criam valor do ponto de vista do cliente, ou seja, permite visualizar todo o fluxo, ao longo da cadeia de valor, desde a concepção até à entrega ao cliente (BELL; ORZEN, 2011).

O mapeamento do fluxo de valor representa visualmente o fluxo de informações e materiais com ênfase na quantificação de desperdícios e na quantificação de tempo e qualidade. Ele pode ser feito com grande nível de detalhe, porém, tipicamente, o foco está em um nível mais macro do que no mapeamento de processos e não inclui tarefas e decisões individuais.

A.3.3 Kaizen

O Kaizen é uma das práticas sugeridas pelo Lean. A palavra Kaizen é a junção de duas palavras japonesas: Kai que significa mudança e Zen que significa bom, porém é comum traduzi-la para melhoria contínua. O Kaizen pode ser considerado uma forma de atingir os objetivos do Lean (BELL; ORZEN, 2011).

Esta melhoria é obtida por todos os trabalhadores e tem foco na eliminação de todo tipo de desperdício. Apesar de este ser um processo lento e incremental, os ganhos em longo prazo são grandes.

Um dos conceitos utilizados pelo Kaizen é a metodologia PDCA, desenvolvido por William Edwards Deming. Esta metodologia é composta por quatro etapas:

- Planejar: refere-se à definição do problema, bem como suas possíveis causas e soluções, o estabelecimento de um plano corretivo e os objetivos de forma clara;
- Fazer: refere-se à implementação do plano e à recolha dos dados para análise;
- Verificar: refere-se a verificar se os dados recolhidos vão de encontro aos objetivos definidos e é feito o registro dos resultados;
- Agir: é onde são padronizados os resultados que foram eficazes e, se houver medidas não eficazes, o ciclo é refeito.

Assim, o Kaizen é um processo incremental e contínuo que abrange toda a organização e o envolvimento de todos os trabalhadores. Todos devem trabalhar de forma proativa em busca da melhoria diária e acreditar que bons resultados virão em longo prazo.

A.3.4 Metodologia 5S

Os 5S pode ser considerada a ferramenta mais básica que o Lean sugere, é considerado um passo a mais em direção à melhoria da qualidade e da produtividade. O objetivo desta prática é organizar o ambiente de trabalho de forma adequada para que o trabalhador tenha, apenas, os materiais e ferramentas necessários para executar seu trabalho (BELL; ORZEN, 2011).

O nome desta metodologia é originado de cinco palavras japonesas: Seiri, Seiton, Seiso, Seiketsu e Shitsuke. Para que o método funcione é preciso que as pessoas realmente entendam a importância dele e que ele seja um processo rotineiro e não apenas aplicado de forma isolada.

O primeiro conceito, Seiri, consiste na remoção de todos os materiais e ferramentas não necessários para a execução das tarefas diárias, é importante que os itens sejam identificados por frequência de utilização para que sua prioridade seja percebida. O segundo conceito, Seiton, consiste na organização dos itens de trabalho, de forma que eles estejam acessíveis ao trabalhador de forma rápida, aumentando a eficácia e a eficiência das atividades, os itens devem estar alocados em lugares próximos e instalados e configurados no ambiente de trabalho de forma correta, no caso de *softwares*. O terceiro conceito, Seiso, consiste na limpeza do local de trabalho diariamente, proporcionando um ambiente confortável, limpo e ergonômico. O quarto conceito, Seiketsu, consiste na padronização e sistematização de todas as atividades ditas anteriormente, de forma que esteja disponível para todos os trabalhadores. E o quinto conceito, Shitsuke, diz respeito à sustentabilidade

e à disciplina. Para que os resultados sejam visíveis em longo prazo existe a necessidade de acompanhamento e disciplina. Quando um problema é identificado não se deve julgar ou culpar os trabalhadores, mais sim realizar reuniões para que o problema seja resolvido de forma colaborativa e harmoniosa (BELL; ORZEN, 2011).

A.3.5 Kanban

Uma das ferramentas de grande importância associado ao Pensamento Lean é o sistema Kanban (Fig. 37) , palavra japonesa que significa cartão ou registro visível. Sendo mais um dos conceitos desenvolvidos pela Toyota, este sistema tem como objetivo o balanceamento da produção e a minimização de estoque. Por meio da gestão visual, os Kanbans fornecem de forma simples e intuitiva indicações aos trabalhadores relativas a fluxos de materiais, recursos e informação.

Este sistema é implementado com o objetivo de atingir a produção *Just-in-Time*, ou seja, produzir na quantidade certa, na altura devida e o produto correto. O Kanban limita o trabalho em progress, o que fornece previsibilidade de tempo em ciclos e faz com que as entregas sejam mais confiáveis. A abordagem de "parar a linha de produção", para superar os obstáculos e os erros encontrados, também resulta em níveis mais elevados de qualidade e uma queda rápida de retrabalho. Além disso, o Kanban implica um modelo de produção do tipo *pull*, ou seja, este sistema desencadeia ordens de produção vindas do cliente(RODRIGUES, 2012).



Figura 37 – Quadro Andon, com o mesmo objetivo do Kanban, na fábrica da Toyota (WILLIAMS, 2009)

ANEXO B – Metodologia IPHAN de Gestão de Demandas de Desenvolvimento Ágil de *Software*

O MIDAS foi baseado nos princípios ágeis advindos do Manifesto Ágil, tais princípios foram adaptados ao contexto do órgão. Os princípios norteadores deste modelo foram:

- Envolvimento do requisitante (cliente): as áreas requisitantes devem estar profundamente envolvidas no processo de desenvolvimento. Seu papel será fornecer e priorizar novos requisitos dos sistemas e avaliar as iterações;
- Entrega incremental: o *software* é desenvolvido em incrementos e o cliente especifica os requisitos a serem incluídos em cada incremento;
- Foco nos resultados e não no processo: a equipe de desenvolvimento deve desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos;
- Aceitação de mudanças: ter em mente que os requisitos de um sistema podem mudar fará com este seja projetado para acomodar as mudanças;
- Manter a simplicidade: sempre que possível, a complexidade de um sistema deve ser eliminada concentrando-se na simplicidade tanto do sistema quanto do processo de desenvolvimento.

Os papéis definidos no MIDAS são: área de TI, área de negócio (requisitante) e o fornecedor contratado. Ao relacionamos estes papéis com os papéis do Scrum temos de forma geral que os representantes da área requisitante serão o *Product Owner*, um representante da área de TI do órgão e um representante do fornecedor representarão o papel de Scrum *Master* e a equipe do fornecedor contratado representa a Equipe de Desenvolvimento do Scrum. Se mapearmos ainda os papéis do Scrum com os papéis específicos definidos pela IN 04, temos como resultado a Tab. (23).

Papéis MIDAS/SCRUM	Papéis IN 04/2010
Product Owner	Área Requisitante da Solução, Integrante Requisitante, Fiscal Requisitante, Gestor do Contrato
Scrum Master IPHAN	Área de TI, Integrante Técnico, Fiscal Técnico, Gestor do Contrato
Scrum Master Contratada	Preposto
Equipe de Desenvolvimento	Equipe do fornecedor contratado

Tabela 23 – Papéis MIDAS x Papéis IN 04

As demandas para desenvolvimento de sistemas são divididas em cinco tipos dentro do MIDAS: sistema novo, manutenção evolutiva, manutenção corretiva, documentação de sistemas legados e refatoração. O sistema novo e a manutenção evolutiva devem seguir todo o processo do MIDAS. A refatoração e a documentação de sistemas legados podem seguir todo o processo do MIDAS ou apenas o subprocesso Sprint. Já a manutenção corretiva não segue nenhum dos processos, ela seguirá uma ordem de serviço específica de acordo com sua urgência.

O processo MIDAS é dividido em três fases: planejamento, desenvolvimento e encerramento, e deve ser executado de forma incremental, com entregas frequentes e progresso medido continuamente. O macro fluxo do processo MIDAS é representado pela Fig. (38).

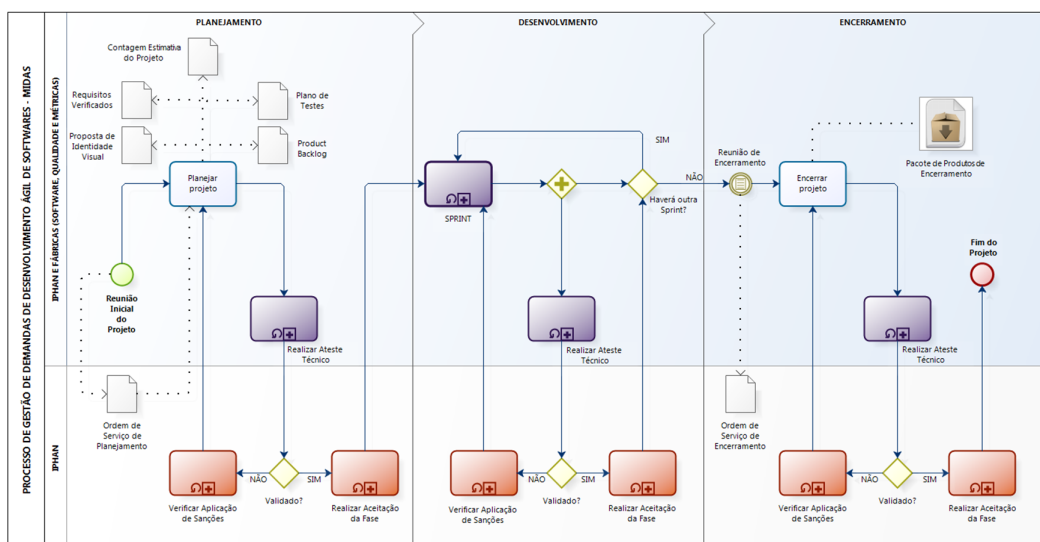


Figura 38 – Processo MIDAS (IPHAN, 2013)

O primeiro subprocesso advindo do planejamento é o subprocesso Realizar Ateste Técnico, trata-se de um subprocesso padrão que será executado em todas as fases do processo MIDAS. Nele estão contidos outros dois processos: Medição de Sistemas e Controle

de Qualidade, cujo detalhamento não cabe no escopo deste trabalho. O subprocesso de Realizar Ateste Técnico tem o objetivo de verificar se os requisitos de cada fase foram satisfeitos e se os produtos de trabalho atendem às especificações de entrada e aos planos e regras estabelecidos. O fluxo deste subprocesso está representado na Fig. (39) com suas atividades, entradas e saídas.

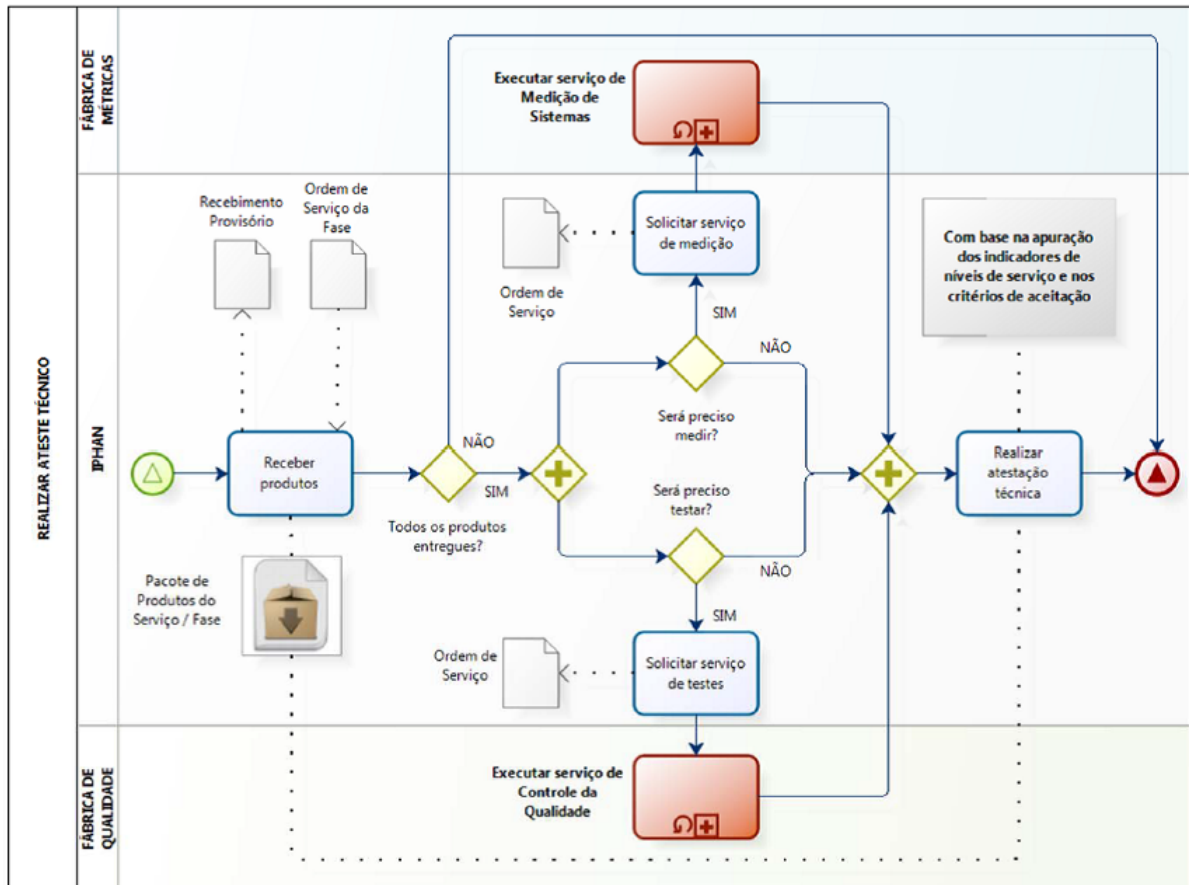


Figura 39 – Subprocesso Realizar Ateste Técnico (IPHAN, 2013)

O subprocesso Sprint é o principal processo da fase de desenvolvimento e tem o objetivo de realizar o ciclo de trabalho de desenvolvimento. Sprint é um ciclo de desenvolvimento onde requisitos são implementados tendo como resultado um incremento do produto que está sendo desenvolvido. A quantidade de ciclos totais do projeto é definida na atividade de “Planejar Projeto”, a qual é realizada na fase de planejamento. O fluxo deste subprocesso está ilustrado na Fig. 40.

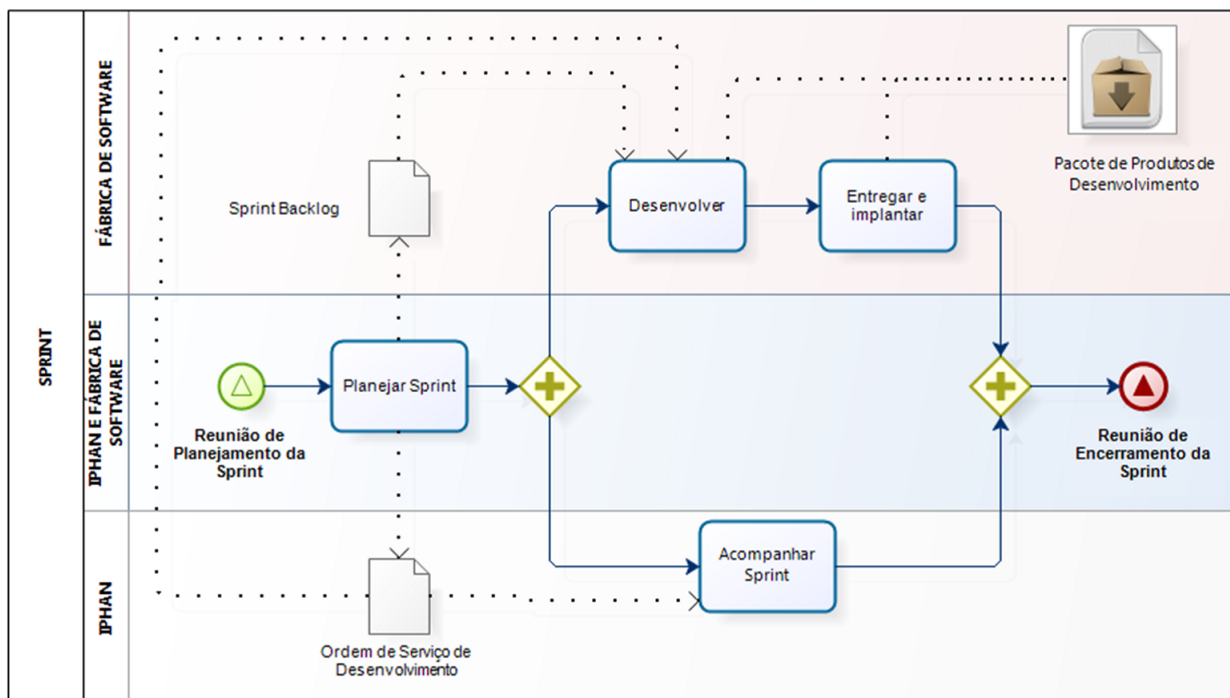


Figura 40 – Subprocesso Sprint (IPHAN, 2013)