

Um Estudo de Caso sobre a Adoção de Métodos Ágeis na Gestão de Contratos de Fornecedores de Desenvolvimento de Software em uma Organização Pública Brasileira

Hilmer R. Neri¹, Aline Gonçalves¹, Guilherme B. Rêgo¹

¹Faculdade UnB Gama - Universidade de Brasília (UnB), Brasil

hilmer@unb.br, alinegsantoss@gmail.com, gbre.111@gmail.com

Resumo. Atualmente, algumas organizações da Administração Pública Federal iniciam investimentos para adotar contratações de serviços de desenvolvimento de software utilizando métodos ágeis, motivado pelo entendimento de que os instrumentos contratuais, hoje em vigor, apresentam limitações que causam impacto nos custos dos projetos e na entrega do produto. O objetivo desse trabalho foi analisar a influência da adoção de métodos ágeis na gestão de contratos públicos de desenvolvimento de software. Para isso, foi realizada uma investigação empírica, descritiva, por meio da execução de um estudo de caso exploratório. Analisamos os efeitos sobre a entrega de ordens de serviço, a qualidade interna do produto e a satisfação do cliente da solução desenvolvida na organização. Após a análise dos dados coletados concluímos que neste caso foi possível a aplicação de métodos ágeis na gestão de contratos públicos de desenvolvimento de software, e que isso pode melhorar a atividade da gestão do desenvolvimento, em que pese, haja restrições face ao normativo vigente.

Palavras-chave: Gerenciamento, contratações, software, organizações públicas, lean, métodos ágeis, desenvolvimento, produção, estudo de caso, engenharia de software empírica.

1. Introdução

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* [BRASIL 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*.

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 precursores 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 [adm].

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. 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.

Com isso, com objetivo de trabalhar a problemática atual de adoção de métodos ágeis na gestão de contratos em um trabalho de conclusão de curso foi realizado um estudo de caso em uma organização pública brasileira, a qual utilizasse métodos ágeis e o pensamento lean na sua solução de gestão de contrato de fornecedores de desenvolvimento de software, para análise da influência e efeitos do uso dessas metodologias nos resultados da gestão do contrato.

2. Metodologia de Pesquisa

Na metodologia de pesquisa adotada neste trabalho, 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.

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 ([Yin 2010]). Portanto, as etapas Plano, Projeto e Preparação definidas por Yin foram agrupadas na fase Planejamento nesse estudo de caso.

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:

- 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 pela autarquia pública 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.

3. Projeto do Estudo de Caso

3.1. Definição

Este trabalho consistiu 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 foi a análise dos efeitos sobre a entrega de ordens de serviço, a satisfação do cliente e a qualidade interna do código do contrato selecionado, o qual é alinhado 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, os quais guiaram o pesquisador durante a fase de coleta de dados.

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 and 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:

ID	Questão	Métrica
QE1.	Qual a quantidade total de ordens de serviço?	quantidade total de ordens de serviço
QE2.	Qual a quantidade de ordens de serviço que tiveram entrega de <i>software</i> funcional?	quantidade de ordens de serviço de <i>software</i>
QE3.	Qual a quantidade de ordens de serviço que tiveram entrega apenas de documentação?	quantidade de ordens de serviço de documentação
QE4.	Qual a proporção de entrega de <i>software</i> funcional?	quantidade de ordens de serviço de <i>software</i> que tiveram entrega de <i>software</i> funcional /quantidade total de ordens de serviço
QE5.	Qual a duração média de entrega de <i>software</i> funcional?	duração total das sprints/duração total das sprints que tiveram entrega de <i>software</i> funcional
QE6.	Qual a quantidade de ordens de serviço que não teve entrega de <i>software</i> funcional e de documentação?	quantidade de ordens de serviço sem entrega de <i>software</i> e de documentação
QE7.	Qual a porcentagem de requisitos atendidos em cada ordem de serviço?	(requisitos atendidos/requisitos pedidos) * 100
QE8.	Quantas multas foram aplicadas no contrato?	quantidade de multas
QE9.	Qual o custo de cada <i>sprint</i> e ordem de serviço do projeto?	custo/sprint
QE10.	O quanto de visibilidade do que estava sendo feito o gestor do contrato teve durante o contrato?	alto, médio ou baixo
QE11.	Qual o nível de satisfação com o <i>software</i> entregue ao final do contrato?	muito satisfeito, satisfeito, neutro, insatisfeito, muito insatisfeito

Tabela 1. Questões Específicas do Processo

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 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:

ID	Questão	Métrica
QE12.	Qual a qualidade interna do produto entregue até o momento ?	bom, excelente, regular e ruim

Tabela 2. Questões Específicas do Produto

3.2. 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 traz apenas um levantamento do que foi feito de uso de métodos ágeis pelos órgãos e autarquia analisados. No entanto, não houve uma coleta e análise de dados, tal qual realizada neste trabalho.

3.3. Design

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.

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).

3.4. 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.

4. 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 do repositório de código fonte do contrato do Sistema Integrado de Conhecimento e Gestão (SICG) com a empresa EGL - Engenharia. O Sistema Integrado de Gestão do Conhecimento (SICG) teve como objetivo automatizar o processo de trabalho decorrente da metodologia de inventário, cadastro, normatização, fiscalização, planejamento e análise e gestão do patrimônio material. Esta solução de software foi construída na linguagem Java com a utilização de frameworks conhecidos no mercado, como por exemplo o VRaptor e Hibernate.

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. Os 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.

- Amostra: foi aplicado o questionário para todos da área de sistemas do IPHAN (3 pessoas) e para 3 integrantes da equipe da EGL-Engenharia.

4.1. Validade

As principais ameaças aplicáveis aos estudos de caso são: validade do constructo, validade interna, validade externa e confiabilidade [Yin 2010].

A validade de constructo diz respeito ao uso de múltiplas fontes de evidência e seleção de informantes chave. 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 da autarquia 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. 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 Para [Yin 2010]. 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 deste protocolo do estudo de caso e o desenvolvimento de um banco de dados do estudo foram táticas utilizadas para garantir a confiabilidade.

4.2. 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.

4.3. 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. Resultados do Estudo de Caso

Os resultados dos dados coletados foram 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.

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 3 apresenta os resultados das métricas para cada uma das questões específicas anteriormente mencionadas.

ID	Resultado da Métrica
QE1.	25 ordens de serviço
QE2.	20 ordens de serviço
QE3.	1 ordem de serviço
QE4.	80%
QE5.	3.7 semanas
QE6.	4 ordens de serviço
QE7.	Tabela 4
QE8.	0 multas
QE9.	Tabela 5

Tabela 3. Resultados Métrica de QE1. a QE9.

A Tabela 4 apresenta a porcentagem de requisitos atendidos em cada ordem de serviço ou *sprint*, respondendo assim a questão específica QE7.

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 5 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

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

Tabela 4. Porcentagem de Requisitos Atendidos por Ordem de Serviço

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

<i>Sprint</i>	0	1	2	3	4	5	6
OS	1	2	3	4	5	6	6.1
Custo	R\$ 49,500.00	R\$ 0.00	R\$ 53,316.25	R\$ 0.00	R\$ 36,382.50	R\$ 31,927.50	R\$ 0.00
<i>Sprint</i>	7	8	9	10	11	12	-
OS	7	8	9	10	11	12	-
Custo	R\$ 45,478.13	R\$ 31,741.88	R\$ 29,143.13	R\$ 65,896.88	R\$ 64,597.50	R\$ 107,291.25	-
<i>Sprint</i>	13	14	15	16	17	18	-
OS	13	14	15	16	17	18	-
Custo	R\$ 56,801.25	R\$ 39,909.38	R\$ 63,669.38	R\$ 61,627.50	R\$ 31,185.00	R\$ 0.00	-
<i>Sprint</i>	19	20	21	22	23	24	-
OS	19	20	21	22	23	24	-
Custo	R\$ 53,831.25	50,490.00	R\$ 63,112.50	R\$ 13,612.50	R\$ 22,027.50	R\$ 36,630.00	-

Tabela 5. 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, a autarquia 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 foi aplicada. Além disso, o fato da empresa ter seu fluxo de caixa interrompido, já se caracterizou como um tipo de penalização e fez com que esta promovesse adequações ao seu processo de trabalho. Outrossim, evidencia-se uma boa versação dos recursos do erário.

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. Ao executar 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. Cabe destacar que a métrica de pontos por função não se caracteriza como uma métrica pertencente ao domínio de métricas ágeis. Este estudo não teve o foco de discutir o uso de pontos por função em contratos com uso de metodologias ágeis. Afora, a autarquia estava seguindo as diretrizes e cumprindo os deveres prescritos no normativo.

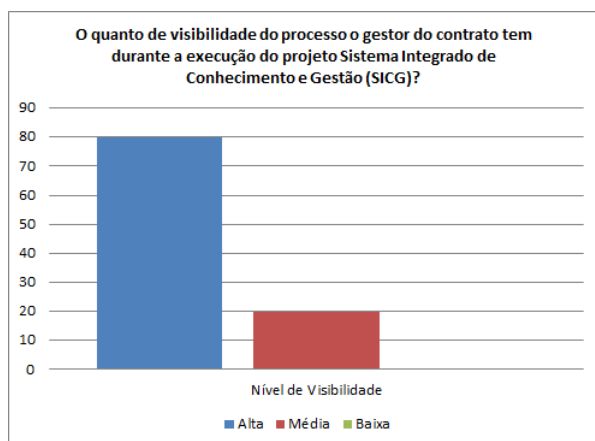
Ainda pelos dados coletados neste estudo, não foi evidenciada a utilização de

uma técnica de gerenciamento de custos de projeto, como exemplo, a análise do valor agregado. Uma solução a ser investigada seria o uso de uma ferramenta que automatizasse o processo de monitoramento de custos. Assim, métricas relacionadas a custo poderiam ser acessadas tempestivamente de forma 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.

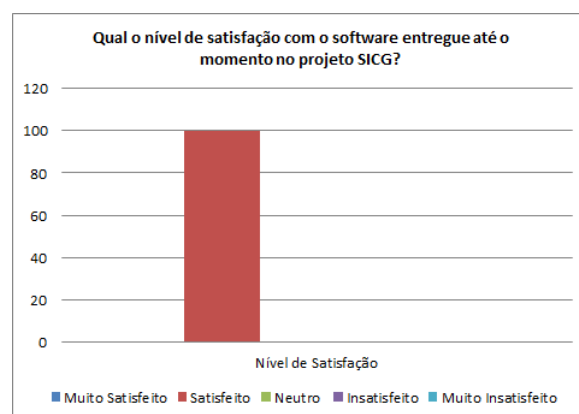
5.2. Efeitos sobre a satisfação do cliente

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.

A Figura 1(a) 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. A Figura 1(b) 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.



(a) Nível de Visibilidade do Processo do Projeto SICG



(b) Nível de Satisfação do Produto Entregue do Projeto SICG

5.3. Efeitos sobre a qualidade do código

Para cada uma das *sprints* do SICG, coletou-se o código fonte disponibilizado pelo IPHAN. A qualidade do código fonte do SICG foi aferida por meio de métricas que evidenciam a qualidade interna do produto de software [ISO/IEC 25023 2011]. Neste trabalho foram selecionadas doze métricas de código fonte analisadas por Meirelles [Meirelles 2013].

Um dos objetivos científicos do estudo de Meirelles foi a identificação das distribuições estatísticas dos valores das métricas apresentadas em 38 projetos de *software* livre, onde foram analisadas 344 mil e 872 classes das aplicações mais utilizadas de *software* livre, tais como, Chrome, Firefox, OpenJDK e VLC. A partir de cada uma das

distribuições estatísticas foram classificadas 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, estes rótulos de intervalos de frequência foram interpretados em rótulos qualitativos: excelente, bom, regular e ruim, segundo Machine [Machini et al. 2010].

As medidas definidas nos intervalos utilizados para se chegar as esses indicadores de qualidade foram retirados do estudo de Meirelles [Meirelles 2013]. Para este trabalho foram utilizados os intervalos encontrados para a linguagem Java, uma vez que o o objeto desta análise é um código fonte escrito em Java.

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, na Linguagem Java, o Open JDK8.

De forma geral, o processo realizado para a análise do código fonte do SICG consistiu 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 .csv¹ com as medidas numéricas das métricas selecionadas; iii) transformar o arquivo .csv em json²; iv) e por fim, processar esse arquivo na solução desenvolvida em um ambiente de Datawarehousing, proposto por [Rêgo 2014].

A fim de responder a questão específica QE12 são 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: LOC (Lines of Code), ACCM (Average Cyclomatic Complexity per Method), AMLOC (Average Method Lines of Code), ACC (Afferent Connections per Class), ANPM (Average Number of Parameters per Method), CBO (Coupling Between Objects), DIT (Depth of Inheritance Tree), LCOM4 (Lack of Cohesion in Methods), NOC (Number of Children), NOM (Number of Methods), NPA (Number of Public Attributes) e RFC (Response For a Class).

A *Sprint* 24 foi a última *sprint* do projeto e, portanto, é nela que foi entregue o código fonte final. Conforme a Fig. 1, 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.

Entretanto, segundo Meirelles, a análise de métricas de código fonte de forma isolada pode levar em interpretações errôneas. Além disso, por meio da utilização de métricas de código fonte é possível identificar partes do código que podem ser melhoradas, por meio de refatoração. As métricas apresentadas foram combinadas de forma a identificar cenários de limpeza de código fonte, auxiliando com isso, a tomada de de-

¹Comma-separated values (ou CSV) é um formato de arquivo que armazena dados tabelados

²JSON (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.

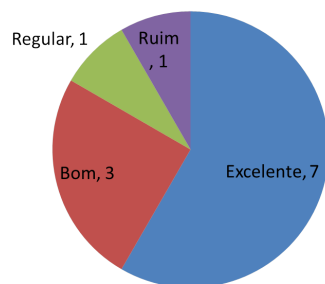


Figura 1. Qualidade Geral do Código Fonte

cisão sobre refatoração, ou mesmo sobre a aferição e interpretação da qualidade interna do produto Para realização deste mapeamento foram utilizados como base os cenários de limpeza propostos nos estudos de Machini Machini et al. 2010, que por sua vez, foram implementados na solução de Datawarehousing proposta por [Rêgo 2014].

Dentre os tipos de cenários de limpeza identificados, os mais incidentes foram: Complexidade Estrutural, Classe Pouco Coesa e Interface dos Métodos. Outros tipos de cenários de limpeza como Classe com Muita Exposição, Classe com Muitos Filhos e Classe com Método Muito Grande também foram identificados, porém, com menos frequência.

O cenário de limpeza de código fonte Complexidade Estrutural é aquele que o tratamento deveria ser priorizado, pois ele correspondeu de 55% a 68% da quantidade total de cenários identificados. Essa informação permitiria, por exemplo, ao fiscal técnico, priorizar atividades de refatoração junto a contratada, ou até mesmo decidir por não realizar a homologação.

Foi analisada ainda 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), onde 66% dos envolvidos consideraram o intervalo de qualidade do código como "Excelente" e os demais consideraram o intervalo de qualidade como "Bom".

Assim, de acordo com as análises realizadas, a qualidade interna do produto pode ser considerada satisfatória ao compararmos o Opens JDK8. Além disso, há uma convergência com a percepção satisfatória dos envolvidos no projeto SICG, embora não tenhamos encontrado evidências de que o IPHAN realizava análise sistemática sobre tais métricas de qualidade de produto.

O fato de padrões de qualidade de métricas de código fonte não terem sido especificados no contrato e o fato de uma aferição de qualidade de código de fonte não ter sido realizada durante os ciclos de desenvolvimento do projeto, figuraram como risco, uma vez que poderia ter comprometido o sucesso e a manutenibilidade do sistema. Por fim, não evidenciamos a entrega de uma suíte de testes automatizada, o que também poderia ter contribuído para a qualidade interna do produto, assim como sua segurança e robustez.

6. 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 público de desenvolvimento de software.

Possivelmente uma solução automatizada de análise estática de código, por exemplo, como a utilizada neste estudo de caso, e a exigência de entrega de uma suíte automatizada de testes 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 na autarquia, não ocorreu na gestão de contratos ágil aplicada.

Ainda, 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.

Por fim, destacamos que essa aproximação da academia e uma organização pública, que contrata serviços de desenvolvimento de software, mostrou-se benéfica para ambas partes, e consequentemente para o estado.

Referências

- Basili, V. R. and Rombach, H. D. (1994). Goal Question Metric Paradigm.
- BRASIL (2013). *Acórdão nº 2314*. Tribunal de Contas da União, Brasília, Brasil.
- Cunningham, W. (2001).
- ISO/IEC 25023 (2011). ISO/IEC 25023: Systems and software engineering - systems and software quality requirements and evaluation (square) - measurement of system and software product quality. International Standard 25023, International Organization for Standardization and International Electrotechnical Commission.
- Machini, J. a., Almeida, L., Kon, F., and Meirelles, P. R. M. (2010). Código Limpo e seu Mapeamento para Métricas de Código-Fonte.
- Meirelles, P. R. M. (2013). *Monitoramento de métricas de código-fonte em projetos de software livre*. PhD thesis, Instituto de Matemática e Estatística – Universidade de São Paulo (IME/USP).
- Melo, C., Santos, V. A., Corbucci, H., Katayama, E., Goldman, A., and Kon, F. (2012). *Métodos ágeis no Brasil: estado da prática em times e organizações*.

Rêgo, Guilherme Baufaker, G. B. N. (2014). Monitoramento de métricas de código-fonte com suporte de um ambiente de data warehousing: um estudo de caso em uma autarquia da administração pública federal.

Yin, R. K. (2010). *Estudo de Caso - Planejamento e Métodos*.