Análise e melhoria do processo de reengenharia de *software*: um estudo de caso

Guilherme Mendonça de Moraes[1], Prof. Dr. Edgard Costa Oliveira[2]

[1]Programa de Pós-Graduação em Computação Aplicada (PPCA) - Universidade de Brasília (UnB), CEP: 7090-970 – ICC Centro, Módulo 14, Subsolo CSS 361, Campus Universitário Darcy Ribeiro – Brasília, DF - Brasil

[2]Faculdade de Tecnologia – Engenharia de Produção (FT/ERP) – Universidade de Brasília (UnB), DF - Brasil.

guilherme.moraes@aluno.unb.br, ecosta@unb.br

Abstract. *Although many authors seek to relate risk management to failure or success in software reengineering projects, the union of the two themes remains a major challenge for organizations that contract or provide such products and services. This article aims to analyze and improve the process of software reengineering through risk management. The methodology adopted was the case study of a project to change the technology of software in a large financial institution, which resulted in the improvement of the software reengineering process in the company.*

Resumo*. Apesar de muitos autores procurarem relacionar gestão de riscos com fracasso ou sucesso em projetos de reengenharia de software, a união das duas temáticas ainda é um grande desafio para as organizações que contratam ou fornecem produtos e serviços desse tipo. Esse artigo tem por objetivo a análise e melhoria do processo de reengenharia de software por meio da gestão de riscos. A metodologia adotada foi o estudo de caso de um projeto de mudança da tecnologia do software em uma instituição financeira de grande porte, o que resultou na melhoria do processo de reengenharia de software dessa instituição.*

1. Introdução

Com o avanço tecnológico e o crescimento das necessidades de mercado, é comum que as empresas optem por realizar a mudança de sua estrutura tecnológica atual visando a compatibilidade com outras linguagens, banco de dados e plataformas, como também o aumento do desempenho, usabilidade, acessibilidade, segurança e qualidade [Pinto e Braga, 2004; UTFPR, 2010; Machado, 2011].

A mudança da tecnologia visa não somente adaptar o processo de *software* ao de negócio [Weerakkody e Currie, 2003], mas também adequá-lo às novas tecnologias, corrigindo erros conhecidos em ambiente produtivo, como itens de qualidade expostos pela ISO 25010 (2011): funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade, fazendo com que o aplicativo cumpra requisitos específicos em tempo real [Vogel-Heuser et al., 2014].

De acordo com Pressman (2005) essa atividade de reengenharia de sistemas de computacionais consome até 70% dos esforços durante todo o ciclo de vida do sistema.

Além da reengenharia de *software* ser uma tarefa onerosa, existe um alto risco nesse processo de mudança da tecnologia [Pressman , 1995; Reis et al., 2003; Vogel-Heuser, 2010], pois o sistema legado precisa continuar funcionando sem que essas modificações causem um efeito negativo no mesmo em ambiente produtivo.

Juntamente com essas alterações existem os riscos que as mesmas representam para o negócio da instituição [Weerakkody e Currie, 2003; Sneed, 2005; Chen et al., 2009], pois a necessidade é que a toda a estrutura dos aplicativos seja alterada sem que o programa em produção fique comprometido.

Conforme a ISO 31000 (2009) risco é o efeito da incerteza nos objetivos. Em outras palavras, é algo que pode causar um desvio nos propósitos comuns. Nesse contexto, o objetivo está relacionado às expectativas de qualidade, funcionamento correto e conformidade com requisitos legais e institucionais do *software* migrado, tendo como base que suas vulnerabilidades estão ligadas diretamente à ausência de determinados controles [Elahi et al., 2010].

Dessa forma, o presente trabalho tem por objetivo a análise e melhoria do processo de reengenharia de *software* desenhando o processo atual (*AS-IS*), analisando as vulnerabilidades do processo, propondo um desenho do mesmo (*TO-BE*) com as melhorias utilizando ferramentas e técnicas para gestão de riscos.

O estudo de caso em questão será aplicado a um projeto de mudança da tecnologia de aplicativos computacionais em uma instituição financeira de grande porte, que por sigilo terá sua identificação nesse artigo como Instituição Financeira Creditar.

2. Desenho do processo atual (*AS-IS*) de reengenharia de *software*

Devido à falta de gestão de riscos nos processo de migração da tecnologia do *software*, a Instituição Financeira Creditar reportou um incidente em ambiente produtivo após migração de um aplicativo computacional e o mesmo tendo passado por todas as atividades de desenvolvimento, teste e homologação, que creditou várias vezes o salário de clientes, que por sua vez causou um grande prejuízo financeiro, dano à imagem da instituição:

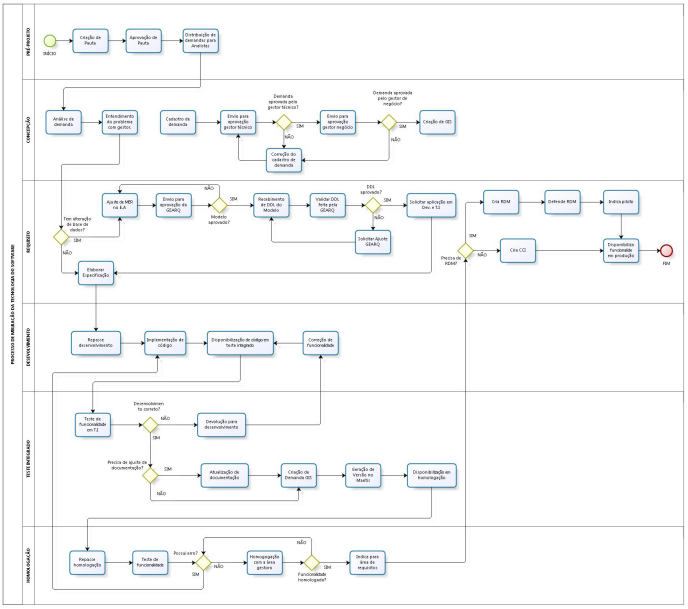

Figure 1. Diagrama do processo atual de reengenharia de *software* da Instituição Creditar

Utilizando a técnica de mapeamento de processos BPMN - *Business Process Model and Notation* [Chinosi e Trombetta, 2012] e o programa Bizagi, realizado o desenho AS-IS do processo de migração da tecnologia de *software* onde o mesmo foi dividido em 6 grandes áreas sendo: pré-projeto, concepção, requisitos, teste integrado e homologação conforme imagem anterior.

3. Identificação dos riscos no processo atual de reengenharia de *software*

Como a proposta da ISO 31000 (2009) é a identificação, análise, avaliação e tratamento dos riscos, é necessário em primeiro ponto realizar a identificação dos mesmos relacionado à todas as atividades da organização. Para tal tarefa, foi utilizada a técnica *Brainstorming* que é uma ferramenta indicada pela ISO 31010 (2009) e uma forma obter uma lista completa dessas ameaças. Após tais tarefas, foi construído um desenho com uma estrutura da análise SWOT - *Strengths*, *Weaknesses*, *Oportunities*, *Thereats*, porém, considerando apenas as oportunidades e ameaçadas identificadas no Projeto Renovação CCO da instituição estudada:

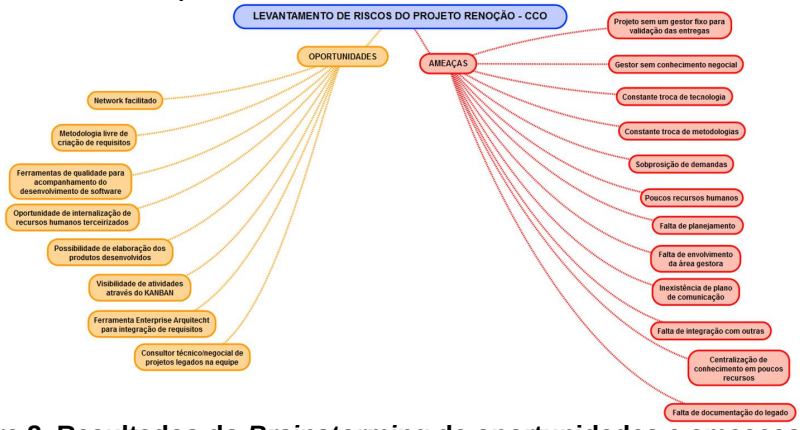

Figure 2. Resultados do *Brainstorming* de oportunidades e ameaças no processo de reengenharia de *software*

Para identificar o que foi considerado como ameaça, foram utilizados ferramentas, técnicas e melhores práticas para reengenharia de *software* e serviços de TI como: *Rational Unifield Process* [IBM, 2001], SCRUM [Schwaber e Sutherland, 2011], *Xtreme Promming* [Beck e Gamma, 2000], ITIL [OGC, 2011], ISO 25010 (2011) e o PMBOK (2018) e com isso criada uma base de conhecimento após essa revisão bibliográfica para identificar os controles aplicáveis às vulnerabilidades no processo de reengenharia de aplicativos do estudo de caso, como a seguir:

Quadro 1. Modelo CGU (2018) de base de conhecimento com a documentação dos controles

Aplicação: **Gestão de Projetos**	Ativo: **Tecnologia**	Tipo: **Norma**	*Knowledge base*: **ISO 25010 (2011)**
Descrição do controle	Deve-se criar um documento de Interoperabilidade.		
O quê?	Artefato para documentar a capacidade do produto de *software* de interagir com um ou mais sistemas especificados.		
Por quê?	Mitigação dos riscos de erro de integração do novo sistema migrado com outros *softwares* necessários.		

Após o *Brainstorming* e com ajuda de gestores do negócio, algumas partes interessadas e um gestor de riscos, chegou-se ao conhecimento da existência de cerca de 150 vulnerabilidades em todo o processo de reengenharia de *software*, das quais foram selecionadas 25 para análise, avaliação e tratamento, sendo elas de maior relevância para a instituição de acordo com os especialistas envolvidos no nessas atividades.

Foram utilizadas como referência a norma ISO 25010 (2011), PMBOK (2018), o RUP [IBM, 2001], ITIL V3 [OGC, 2011], Xtreme Promming [Beck e Gamma, 2000] e o SCRUM [Schwaber e Sutherland, 2011] para identificação dos controle aplicáveis às vulnerabilidades no processo de mudança da tecnologia do *software*. Com isso, foi realizada a documentação dos riscos como tais como esses:

Quadro 2. Identificação de riscos com a especificação do impacto e aplicação dos controles (Adaptado pelo próprio autor do modelo CGU (2018))

Etapa	Vulnerabilidades - Fatores de Risco	Impacto	Controles Aplicáveis
Pré-projeto	Divergência de planejamento e execução de pauta e especificação das necessidades e expectativas de cada projeto	Mudanças não mapeadas	É necessário identificar todas as necessidades, premissas, restrições, requisitos iniciais e principais responsáveis pelo projeto de forma a mitigar o risco de mudança de escopo, aumento do escopo, aumento de custo e prazo.
		Falta de compatibilidade com outras funcionalidades	
		Erros desconhecidos em ambiente produtivo	
		Descumprimento de requisitos negociais	
		Aumento do custo do projeto	
		Atraso no projeto	
Concepção	Falta de entendimento do sistema legado e todas as funcionalidades internas e externas impactadas por determinada mudança como também os erros existentes.	Erros desconhecidos em ambiente produtivo	Deve-se criar um documento de Interoperabilidade para documentar a capacidade do produto de *software* de interagir com um ou mais sistemas especificados.
		Atualização desnecessária de *software* ou nova funcionalidade	
		Erros desconhecidos em ambiente produtivo	Deve-se criar um documento de Capacidade do produto de *software* de prover um conjunto apropriado de funções para tarefas e objetivos do usuário especificados.
		Insatisfação do usuário final	
		Aumento do custo do projeto	Deve-se realizar a proteção frente a erros de usuários: como produto consegue prevenir erros dos usuários.
		Aumento do prazo do projeto	

4. Análise e avaliação dos riscos no processo de reengenharia de *software*

Com a utilização das ferramentas e técnicas: Análise SWOT e Análise de Restrição e Premissas [PMBOK, 2018], o gestor negocial da instituição estudada, juntamente com o gestor de riscos conseguiram identificar qual o grau de riscos de vulnerabilidade pontuando a probabilidade de determinado risco se concretizar como também o impacto que o mesmo causaria ao processo e mudança da tecnologia dos sistemas. Essas informações foram documentadas em uma matriz de probabilidade e impacto [ISO 31010, 2011] na qual foi utilizada uma pontuação de probabilidade de 0,1 a 0,9, sendo 0,1 para pouco provável e 0,9 para muito provável que determinado risco se torne um incidente, e no caso do impacto de 1 a 5, sendo 1 para impacto muito baixo e 5 para

impacto muito alto, caso o incidente aconteça. Para fins de cálculos, os valores foram multiplicados entre si gerando uma classificação de *score* com o grau de cada risco:

Tabela 1. Avaliação dos riscos (próprio autor adaptado da ISO 31010, 2009)

AVALIAÇÃO DOS RISCOS – MATRIZ DE PROBABILIDADE E IMPACTO			
Riscos	Probabilidade	Impacto	*Score*
Mudanças não mapeadas	0,7	5	3,5
Falta de compatibilidade com outras funcionalidades	0,7	5	3,5
Erros desconhecidos em ambiente produtivo	0,3	4	1,2
Descumprimento de requisitos negociais	0,1	2	0,2
Aumento do custo do projeto	0,9	2	1,8
Atraso no projeto	0,9	3	2,7
Erros desconhecidos em ambiente produtivo	0,3	5	1,5
Desenvolvimento desnecessário de atualização de *software* ou nova funcionalidade	0,2	1	0,2
Insatisfação do usuário final	0,4	5	2
Aumento do custo do projeto	0,3	3	0,9

O *score* foi avaliado tendo como base a matriz de probabilidade e impacto [PMBOK, 2018], considerando apenas as ameaçadas:

Tabela 2. Matriz de Probabilidade e Impacto (Adaptado de PMBOK (2018))

Matriz de Probabilidade X Impacto					
Probabilidade	**Ameaças**				
0,9	0,9	1,8	2,7	3,6	4,5
0,7	0,7	1,4	2,1	2,8	3,5
0,5	0,5	1	1,5	2	2,5
0,3	0,3	0,6	0,9	1,2	1,5
0,1	0,1	0,2	0,3	0,4	0,5
Gravidade	1	2	3	4	5

5. Proposta de melhoria no processo de reengenharia de *software* (*TO-BE*)

Após identificação, análise e avaliação de risco e aplicação dos controles às vulnerabilidades identificadas no processo mudança da tecnologia do *software*, como resposta a essas ameaças encontradas no anterior (*AS-IS*), foi elaborado um desenho evoluído do mesmo (*TO-BE*) como forma de mitigar os riscos identificados anteriormente, como por exemplo:

1. Na forma de elicitação de requisitos, foi identificado a necessidade de entender o sistema legado através de documentações existentes e principalmente a opinião de especialistas técnicos por *Brainstorming* e à partir daí, a elaboração de uma primeira documentação contendo as possíveis regras, telas (se aplicável) e entidades de banco de dados (se aplicável);

2. Como documentação principal de mapeamento de necessidade e desenho de funcionalidades e suas respectivas regras, foi utilizado o conceito de elaboração de interface adaptando a ideia do RUP [Kruchten, 2004; Sommerville, 2011] de forma que todos os *stakeholders* consigam opinar sobre a viabilidade, necessidades e necessidades do negócio inerente as funcionalidade migrada;

3. Processo de validação de regras, mensagens, telas (quando aplicável) e entidades de banco de dados (quando aplicável) – é cíclica iniciando no gestor técnico, equipe de desenvolvimento, representando do usuário final do *software* migrado e por último o gestor do negócio, fazendo cada vez mais a documentação e detalhamento do serviço/sistema [ITIL, 2007].

A seguir, o desenho do processo melhorado (*TO-BE*):

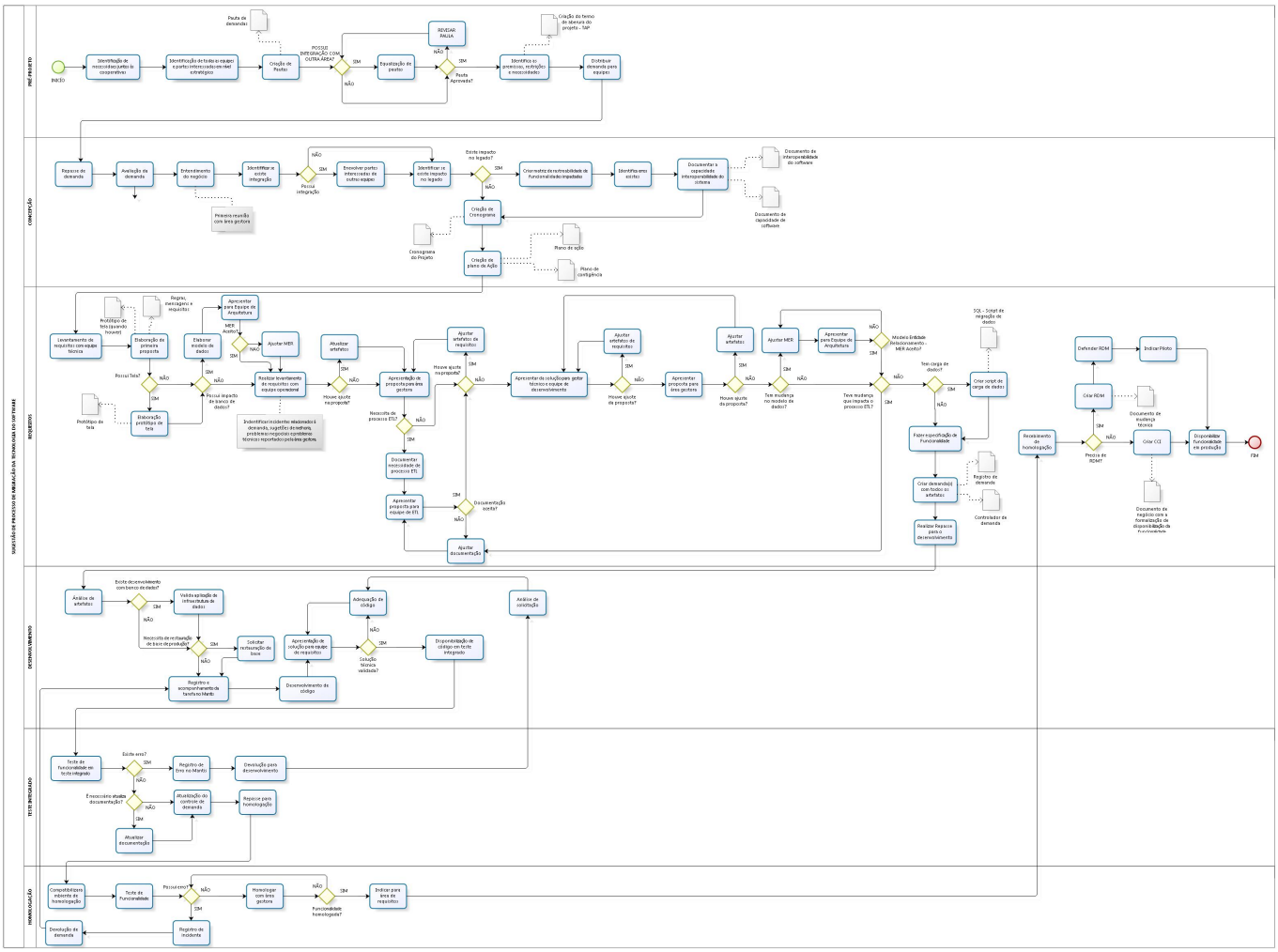

Figure 3. Desenho *TO-BE* do processo reengenharia de *software* da Instituição Financeira Creditar

Através do desenho do projeto melhorado, a área gestora pôde ter uma visão de quais atividades podem ser aplicadas para mitigar os riscos identificados no mapeamento anterior, tendo em vista que a versão aprimorada apresenta ações e documentos para reengenharia de *software* baseadas em controles de mercados como boas práticas e normas.

6. Conclusão

O estudo em questão buscou propor a aplicação da gestão de riscos como forma de análise e melhoria do processo de reengenharia de *software*, mapeamento ferramentas e técnicas para identificar, analisar, avaliar e tratar os riscos que permeiam esse processo, como também sendo as respostas à esses riscos o processo já melhorado utilizando boas práticas, normas e *frameworks*.

Foram utilizas técnicas como brainstorming, matriz de probabilidade e impacto, mapeamento de processos, apoio de especialistas assim como na revisão da literatura para identificar os pontos de atenção e riscos no processo de reengenharia de *software* e como forma de resposta a esses riscos na elaboração do processo melhorado TO-BE.

Foi utilizado como análise um projeto de mudança de tecnologia do *software* da instituição financeira em questão que possui todas as características necessárias para geração de insumo da pesquisa em questão, que além das ferramentas e técnicas, contou com o apoio de um especialista de riscos e um gestor organizacional para validação do processo melhorado. O resultado final mostrou-se bem eficiente após validação do gestor de negócio e do especialista de risco, pois conseguiu revelar os riscos que o processo de reengenharia de *software* como também mostrou as vulnerabilidades que o processo possuía, ou seja, os pontos de atenção que o , por meio de um processo melhorado, permitiu a redução dos riscos de mudança, do não cumprimento de requisitos e do não atendimento de prazo e custos e até mesmo da extinção do projeto.

Após criação e aplicação do processo melhorado, observou-se a redução dos riscos de erros em ambiente produtivo dos *software*s migrados, menor tempo de desenvolvimento, menos mudanças nos requisitos e funcionalidades e em consequência disso, diminuição do tempo e custo de desenvolvimento de todo o processo. Como recomendação para trabalhos futuros, propomos a aplicação de outras ferramentas e técnicas para gestão do risco como também uma forma automatizada para análise e avaliação do riscos identificados.

Referências

ABNT ISO GUIA 31.000:2009, Gestão de Riscos – Princípios e diretrizes, 1.ª edição, Rio de Janeiro, ABNT, 2009.

ABNT ISO GUIA 31.010:2012, Gestão de Riscos – Técnicas para o processo de avaliação de riscos, 1 ª edição, Rio de Janeiro, ABNT, 2012.

Beck, K., & Gamma, E. (2000). Extreme programming explained: embrace change. addison-wesleyprofessional.

Cagnin, M. I. (2005). Parfait: uma contribuição para a reengenharia de *software* baseada em linguagens de padrões e frameworks (Unpublished doctoral dissertation). Universidade de São Paulo.

Chaves, L. L. (2004). Sistemas legados e a aplicação de processos de reengenharia de *software*. FEA – USP , 21–23.

Chen, C. C., Law, C. C., & Yang, S. C. (2009). Managing ERP implementation failure: a project management perspective. IEEE transactions on engineering management , 56 (1), 157–170.

Chinosi, M., & Trombetta, A. (2012). BPMN: An introduction to the standard. Computer Standards & Interfaces, 34(1), 124-134.

De Bakker, K., Boonstra, A., & Wortmann, H. (2010). Does risk management contribute to it project success? a meta-analysis of empirical evidence. International Journal of Project Management, 28 (5), 493–503.

Elahi, G., Yu, E., & Zannone, N. (2010). A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities. Requirements engineering, 15(1), 41-62.

Ibbs, C. W., & Kwak, Y. H. (2000). Assessing project management maturity. Project management journal, 31 (1), 32–43.

ISO ISO/IEC 25.010 (2011), *Software* Product Quality. International Organization for Standardization.

ISO/IEC 13.335-1 (2004) Information technology - Security techniques - Management of information and communications technology security.

ITIL V3 - Information Technology Infrastructure Library (2011). OGC (Office for Government Commerce.

Kruchten, P. (2004). The rational unified process: an introduction . Addison-Wesley Professional.

MACHADO, F (2011). Análise e gestão de requisitos de *software*: onde nascem os sistemas.[sl]: Editora érica;

Ministério da Transparência e Controladoria Geral da União – CGU (2018). Metodologias de gestão de riscos;

Pinto, H. L. M., & Braga, J. L. (2004). Sistemas legados e as novas tecnologias: técnicas de integração e estudo de caso. Informática Pública, Belo Horizonte, 7 (1), 48–69.

PMBOK (2018). Guia PMBOK 6ª Edição - Um Guia do Conhecimento em Gerenciamento de Projetos. PMI.

PRESSMAN, R. S (1995). Engenharia de *Software*. São Paulo: Makron Books;

Pressman, R. S. (2011). Engenharia de *Software* – Uma abordagem Profissional. 7 ed., ISBN 978-85-8055-044-3.

Rational IBM. Rational Unified Process (2001) – Best Practices for *Software* Development Teams. Rational *Software* White Paper, rev. 11/01.

Reis, C. R., & de Mattos Fortes, R. (2003). Caracterização de um processo de *software* para projetos de *software* livre (Unpublished doctoral dissertation). Universidade de São Paulo.

Schwaber, K., & Sutherland, J. (2011). The scrum guide. Scrum Alliance, 21.

Sneed, H. M. (2005). Estimating the costs of a reengineering project. In 12th working conference on reverse engineering (wcre'05) (pp. 9–pp).

SOFTWARE ENGINEERING INSTITUTE – SEI (2010). CMMI® for Development, Version 1.3.

Sommerville, I. (2011). *Software* engineering 9th edition. ISBN-10, 137035152.

UTFPR (2010). A manutenção de *software* nas empresas. Disponível em: <http://pg.utfpr.edu.br/dirppg/ppgep/ebook/2010/CONGRESSOS/ADM/25.pdf>, Acesso em: 07 de abril de 2019.

Vogel-Heuser, B., Legat, C., Folmer, J., & Feldmann, S. (2014). Researching evolution in industrial plant automation: Scenarios and documentation of the pick and place unit (Tech. Rep.).

Weerakkody, V., & Currie, W. (2003). Integrating business process reengineering with information systems development: issues & implications. In International conference on business process management (pp. 302–320).