

FATORES DE RISCO NA FASE DE IMPLANTAÇÃO DE SOFTWARE: UMA REVISÃO DA LITERATURA

L. J. SANTOS¹, S. A. RIBEIRO², E. A. SCHMITZ³, M. F. SILVA⁴, A. J. S. M. ALENCAR⁵

Universidade Federal do Rio de Janeiro, Instituto Tércio Pacitti^{1,2,3,4,5}

Centro Federal de Educação Tecnológica do Rio de Janeiro²

lidiosantos@tic.ufrj.br¹

Submetido 20/05/2019 - Aceito 14/02/2020

DOI: 10.15628/holos.2020.8640

RESUMO

No processo de implantação de *software*, inúmeras falhas podem ocorrer, comprometendo a entrega ao cliente, tais como a falta de recursos, o surgimento de novos requisitos e a não aceitação pelos usuários das mudanças impostas devido à implantação. Estas mudanças estão associadas à adoção de novas tecnologias ou uma nova metodologia de trabalho. Este artigo mostra os resultados de uma revisão da literatura com o intuito de identificar os principais fatores de risco

envolvidos neste processo, suas causas e estratégias de contenção durante a entrega do *software* ao usuário final. A partir deste mapeamento, serão apresentados estudos primários que tratam os fatores de riscos na fase de implantação de projetos de *software*. A partir dos resultados obtidos, apresentamos onze fatores que visam contribuir na eficácia do gerenciamento de riscos de *software* durante a fase de implantação.

PALAVRAS-CHAVE: fatores de risco, estratégias de contenção, implantação de *software*, revisão da literatura.

RISK FACTORS IN DEPLOYMENT PHASE OF SOFTWARE: A LITERATURE REVIEW

ABSTRACT

In the software deployment process, there are several failures which can compromise the delivery to the customer, such as the lack of resources, the emergence of new requirements and the non-acceptance by users of the changes imposed due to the implementation. These changes are associated with the adoption of new technologies or a new work methodology. In this article, a systematic literature review was carried out to identify the main risk factors involved in this stage, their causes

and contention strategies during the delivery of the software to the end user. Based on the proposed research protocol, it was possible to identify documents that primarily deal with risk factors in the software project deployment phase. Thus, from the analysis of these documents, we present eleven factors that aiming to contribute to the effectiveness of software risk management during the deployment phase.

KEYWORDS: risk factors, contention strategies, deployment phase, literature review.

1 INTRODUÇÃO

A fase final do processo de desenvolvimento, segundo Sommerville (2016), é a entrega e implantação do *software*. Esta fase pode envolver ajustes, de modo a refletir o ambiente em que este é usado, a transferência de dados dos *softwares* existentes, bem como a preparação da documentação e o treinamento de usuários.

A etapa de implantação de *software*, aparentemente simples, pode sofrer inúmeras dificuldades durante a sua execução. Para Sommerville (2016), o ambiente de usuário pode ser diferente daquele previsto pelos desenvolvedores do sistema e adaptar o sistema para lidar com o ambiente de trabalho do usuário diversos ambientes de usuários pode ser complexo. Segundo Huang & Han (2008), a participação dos usuários finais tem um grande impacto no sucesso ou falha do projeto. É comum haver resistência às mudanças durante a implantação, especialmente se estas foram impostas externamente. Para Hijazi (2014), se todas as falhas não forem descobertas e mitigadas antes da operação do sistema, elas poderão ter um custo maior para eliminá-las.

Segundo Sommerville (2016), os problemas na implantação do *software* podem demonstrar que existe uma incompatibilidade entre o sistema e o ambiente utilizado pela equipe técnica operacional, os quais podem levar a erros técnicos durante o uso do sistema.

Muitos problemas podem invalidar todo o processo de *software*. O risco é um problema potencial, que pode ocorrer ou não. Independentemente do resultado, é aconselhável identificá-lo, avaliar sua probabilidade de ocorrência, estimar seu impacto e estabelecer um plano de contingência caso o problema realmente ocorra (Pressman, 2016).

Para Timo et al. (2014), as principais causas dos fatores de risco encontrados na fase de implantação de *software* estão associadas a baixa prioridade na execução de testes, o que pode resultar em problemas relacionados ao teste de *software*. Mudanças impostas externamente também podem resultar na resistência do usuário à mudanças e atrasos na implantação do *software*. Problemas descobertos durante a fase de teste de aceitação, onde são feitas simulações de rotina no sistema, também podem provocar o travamento ou reinício de um projeto.

As consequências de uma falha de *software* podem ser extensas. De acordo com Jones (2017), o principal motivo de atrasos e cancelamentos de *software* deve-se ao grande número de erros, cuja eliminação pode absorver mais de 60% do esforço nos grandes projetos de *software*. A média norte-americana para remoção de defeitos de *software* é de 92,5%, referente ao período de 2017. Na fase de implantação, a remoção de defeitos nos testes beta e de aceitação corresponde a aproximadamente 20%.

Segundo Pressman (2016), para evitar este tipo de falha de *software*, o gerente e os engenheiros devem evitar uma série de riscos comuns, compreender os fatores críticos de sucesso que conduzem ao bom gerenciamento e desenvolver uma estratégia de senso comum no que se refere a planejamento, monitoramento e controle do *software*.

Os fatores de risco podem ser considerados críticos para um projeto, visto que a identificação destes determina as incertezas que podem levar ao seu sucesso ou fracasso, principalmente em sua fase de implantação. Para Pressman (2016), a análise e gestão de risco são ações que ajudam uma equipe de *software* a entender e gerenciar a incerteza.

1.1 Fatores de risco no desenvolvimento de *software*

Para Baraldi (2010), um risco quando bem gerenciado, gera oportunidades de ganhos financeiros, de reputação e de relacionamento.

Segundo o PMI (2013), o risco de um projeto é um evento que poderá ocorrer de forma inesperada provocando efeitos negativos ou positivos, com relação a alterações no escopo, cronograma, custo e qualidade. Ao se fazer a análise de riscos são classificados em dois tipos: os riscos conhecidos e os desconhecidos. Os riscos conhecidos são identificados e analisados, possibilitando o planejamento de respostas e designando uma reserva de contingência para gerenciar os riscos de forma proativa. Já os riscos desconhecidos não podem ser gerenciados de forma proativa e, assim sendo, é necessário fazer uma reserva para se assegurar dos impactos, caso o problema ocorra.

Para Alencar e Schmitz (2012), um fator de risco é qualquer evento que possa prejudicar, total ou parcialmente, as chances de sucesso do projeto, isto é, as chances do projeto realizar o que foi proposto dentro do prazo e fluxo de caixa que foram estabelecidos.

Segundo o *Commite of Sponsoring of Treadway Commission* (COSO,2007), o gerenciamento de risco é definido como um processo contínuo que percorre através de todos os setores da organização, sendo conduzido por todos os profissionais em todos os níveis de hierarquia da empresa.

O gerenciamento de riscos é um processo de *software*, definido e sistemático com o objetivo de tratar os fatores de risco com a finalidade de mitigar ou minimizar seus efeitos, produzindo um produto de *software* com qualidade, que atenda às necessidades do cliente, dentro do prazo e custos estimados. (Nogueira,2009).

Em pesquisa recente feita pela empresa *Marsh Risk Consulting* (MARSH,2018), apenas 36,2% de empresas brasileiras declararam que utilizam práticas de gestão de risco. Segundo a empresa é fundamental a inclusão de uma boa política de gestão de riscos na cultura organizacional e que os gestores se conscientizem de que a implementação de processos de gestão de risco é um diferencial competitivo na prevenção de custos da organização.

Este trabalho tem por objetivo apresentar os principais fatores de risco identificados na literatura, suas principais causas e as estratégias de contenção aplicadas. A seção seguinte abordará a revisão da literatura a partir do protocolo de mapeamento utilizado, as questões e estratégias de pesquisa, a seleção das bases de dados e os critérios de inclusão e exclusão dos documentos pesquisados.

2 REVISÃO DA LITERATURA

O grande número de trabalhos científicos publicados a cada ano torna inviável e impensável a análise de todos eles. As áreas de desenvolvimento destes trabalhos estão cada vez mais multidisciplinares e abrangentes. Mesmo com os recursos tecnológicos a disposição, como *softwares*, sites de buscas e bases organizadas por áreas temáticas, esta é uma tarefa muito demorada e difícil devido ao grande número de trabalhos publicados. Além disso, o mau uso destes recursos tecnológicos pode conduzir a uma pesquisa com erros de processamentos da informação coletada. Portanto, a análise sistemática de publicações é um caminho que pode vir a facilitar e direcionar a busca pela informação correta e adequada (Ribeiro et al. 2017).

2.1 Protocolo de Mapeamento

A partir de uma revisão da literatura, este artigo tem por objetivo, obter e analisar o maior número de trabalhos primários relevantes e reconhecidos sobre fatores de risco na fase de implantação, a fim responder as questões de pesquisa. Segundo Kitchenham (2007), as revisões de literatura seguem um método específico e peculiar, que começam pela definição de um protocolo de revisão e resume as etapas de uma revisão sistemática em três fases principais: planejamento, condução e apresentação. Para Felizardo et al (2017), o mapeamento sistemático se caracteriza em prover uma ampla visão de um tópico de pesquisa, questões de pesquisa genéricas, pelo processo de busca definido pelo tópico de pesquisa e pela caracterização dos estudos primários de acordo com os sistemas de classificação.

Uma Revisão de Literatura é o meio de avaliar e de interpretar estudos importantes disponíveis para uma particular questão de pesquisa, tópico ou fenômeno de interesse (Kitchenham, 2004). A partir da pesquisa de um tópico em particular, este tipo de revisão pode induzir a identificação, seleção e produção de evidências, considerando os conhecimentos e as iniciativas existentes no campo de interesse (Mian et al. 2005).

Desta forma, este artigo foi estruturado a partir de um protocolo de revisão e questões de pesquisa, com a definição dos respectivos termos de busca. Posteriormente, estes termos foram aplicados a base de dados baseado no protocolo de pesquisa proposto por Kitchenham (2007), e a partir de critérios de inclusão e exclusão de estudos primários.

O protocolo é o instrumento que consolida o planejamento de uma pesquisa e deve abordar pontos como: o objetivo da revisão, questões de pesquisa, fontes dos estudos primários, identificação dos critérios de inclusão e exclusão na seleção dos estudos (Wohlin et al., 2013).

2.2 Questões de Pesquisa

Uma questão de pesquisa bem definida é apontada como uma das etapas essenciais para a revisão da literatura, pois permite que o pesquisador possa conduzir a busca de trabalhos relevantes e necessários (Kitchenham, 2007). As atividades de uma revisão sistemática devem incluir, segundo (Pai et al. 2004): Uma formulação de uma questão de pesquisa, identificação da necessidade de conduzir uma revisão sistemática, busca exaustiva e abrangente, com inclusão dos estudos primários, avaliação da qualidade dos estudos incluídos, extração de dados, síntese dos resultados do estudo, interpretação dos resultados para determinar sua aplicabilidade e escrita do relatório.

Neste artigo, foram elaboradas as seguintes questões de pesquisa:

Q1. Quais os fatores de risco identificados na literatura na fase de implantação de *software*?

Q2. Quais as causas associadas aos fatores de risco identificados nesta fase de implantação?

Q3. Quais as estratégias de contenção aplicadas para redução dos fatores de risco identificados na fase de implantação?

2.3 Estratégias de Pesquisa e Seleção

Após a elaboração das questões da pesquisa, o próximo passo consistiu na definição dos termos de busca (*strings* de busca) que foram elaborados a partir da identificação dos fatores de risco (*risk factor*), desenvolvimento de *software* (*software development*), *software development life cycle* (*sdlc*) e implantação (*deployment*). Por fim, foram realizadas combinações com palavras chaves e operadores booleanos.

Assim sendo, foi construída a seguinte *string* de busca com termos em inglês para cada questão:

("software development" OR "sdlc") AND ("risk factors" OR "risk list" OR "risk category" OR "risk analysis" OR "risk items" OR "risk evaluation" OR "risk assessment" OR "risk driver") AND ("deployment" OR "delivery" OR "acceptance" OR "installation" OR "maintenance")

2.4 Seleção das bases de dados

A seleção das bases se deu a partir do reconhecimento acadêmico em âmbito internacional. Com isso, as selecionadas para a pesquisa foram: *IEEE Xplore*, *Periódicos CAPES*, *Science Direct*, *Scopus* e *Springer Link*.

Vale salientar que as respectivas bases de dados possuem particularidades com relação ao seu mecanismo de pesquisa, com isso surgiu a necessidade de realizar pequenas adequações nas *strings* para se adequar a base e assim obter resultados satisfatórios.

2.5 Estratégias de Inclusão e Exclusão

A Tabela 1 apresenta os critérios de inclusão e exclusão, que são utilizados para direcionar o assunto escolhido e excluir trabalhos não relevantes para a pesquisa (Peterson et al. 2008).

Tabela 1. Critérios de inclusão e exclusão

Inclusão	Exclusão
Artigos ou periódicos publicados nos últimos 10 anos.	Documentos duplicados ou redundantes.
Artigos disponíveis nas bases <i>IEEE Xplore</i> , <i>Periódicos Capes</i> , <i>Science Direct</i> , <i>Scopus</i> e <i>Springer Link</i> .	Teses, editoriais, prefácios, entrevistas, notícias, comentários, debates e painéis.
Somente estudos publicados em inglês.	Estudos em outros idiomas publicados diferentes ao inglês.
Artigos ou periódicos publicados que descrevam riscos na fase de implantação de <i>software</i> .	Estudos que não abordam riscos na fase de implantação de <i>software</i> .

O procedimento para a inclusão e exclusão dos documentos que retornaram após a aplicação dos argumentos de pesquisa (*strings* de busca) nas bases de dados se deu em dois momentos. A pesquisa inicial identificou 737 resultados em cinco bases acadêmicas. Na primeira seleção, foram definidos no protocolo de pesquisa: título, resumo, palavras-chave e a

disponibilidade do documento em língua inglesa. Por fim, foram aceitos documentos que tratam de forma primária ou secundária os fatores de riscos na fase de implantação de projetos de *software*. Ao término desta primeira análise, restaram 115 documentos.

Em uma segunda avaliação, foi realizada uma análise do conteúdo dos artigos, com a finalidade de reduzir o grande número de resultados e confirmar se estavam de acordo com o que foi estabelecido nos critérios de inclusão e exclusão. Estes artigos foram classificados em três categorias: Irrelevante, Repetido e Incompleto. Os artigos classificados como irrelevantes não apresentavam os fatores de risco referentes à fase de implantação de *software*. Por sua vez, os artigos incompletos exibiam apenas fatores de risco referentes a outras fases do processo de desenvolvimento de *software*. Ao término dessa análise restaram 11 artigos, que foram utilizados para responder as questões propostas pelo mapeamento.

A Tabela 2, exibe os resultados obtidos durante a fase de busca e seleção dos documentos primários.

Tabela 2. Estudos identificados e resultado da seleção primária

Base	Estudos Identificados	Seleção Primária
<i>IEEE Xplore</i>	85	12
<i>Periódicos Capes</i>	120	32
<i>Science Direct</i>	160	25
<i>Scopus</i>	181	17
<i>Springer Link</i>	198	29
Total	737	115

A Tabela 3 apresenta o resultado da seleção secundária por base e a quantidade artigos selecionados na revisão da literatura.

Tabela 3. Resultado da seleção secundária

Base	Seleção Primária	Estudos Identificados			Estudos Selecionados
		Irrelevante	Repetido	Incompleto	
<i>IEEE Xplore</i>	12	7	0	3	2
<i>Periódicos Capes</i>	32	20	1	8	3
<i>Science Direct</i>	25	18	0	4	3
<i>Scopus</i>	17	13	1	2	1
<i>Springer Link</i>	29	23	0	4	2
Total	115	81	2	21	11

3 RESULTADOS E DISCUSSÕES

Após o processo de pesquisa e seleção nas bases científicas, foi possível obter uma lista, conforme apresentado na Tabela 4, referente ao período dos últimos dez anos, dos principais artigos relacionados aos fatores de risco na fase de implantação de *software*.

Tabela 4. Principais artigos relacionados aos fatores de risco

Ref.	Artigos	Ano	Autor
A1	<i>Categorization of risk factors for distributed agile projects</i>	2015	Suprika V. Shrivastava , Urvashi Rathod
A2	<i>Effects of information system planning on information system implementation process: Case Study of Universities and University</i>	2013	Moffat Kombo Barongo
A3	<i>Perceived causes of software project failures-an analysis of their relationships</i>	2014	Timo O.A. Lehtinen, Mika V. Mäntylä, Jari Vanhanen, Juha Itkonen, Casper Lassenius
A4	<i>Preeminent risk factor affecting software development</i>	2018	Hafiz Ali Hamza Gondal, Saira Moin U. Din, Sajida Fayyaz, Muhammad Dahir Zeb, Beenish Nadeem
A5	<i>Risk factors in software development phases</i>	2014	Hijazi, H, Alqrainy, S Muaidi, H
A6	<i>Risk factors in software development projects: a systematic literature review</i>	2018	Júlio Menezes JR & Cristiane Gusmão & Hermano Moura
A7	<i>Risk Impact Analysis across the Phases of Software Development</i>	2014	Bhujang, RK Engineering, V Suma
A8	<i>Risk management in enterprise resource planning implementation: a new risk assessment framework</i>	2013	Prasanta Kumar Dey, Ben Clegg & Walid Cheffi
A9	<i>Software Delivery Risk Management: Application of Bayesian Networks in Agile Software Development</i>	2015	Ieva Ancveire, Ilze Gaillite, Made Gaillite, Janis Grabis
A10	<i>The Influence of Risk Management on Software Project Success</i>	2017	Hiba Arafah, Adham El-Ahmad
A11	<i>Top Twenty Risks in Software Projects: A Content Analysis and Delphi Study</i>	2014	P. Sonchan and S. Ramingwong

Com base nos artigos identificados, são apresentados na Tabela 5, os principais fatores de risco na fase de implantação de *software*:

Tabela 5. Principais fatores de risco abordados na fase de implantação de *software*

Fatores de Risco	Referência dos Artigos
Mudanças no ambiente (<i>change in environment</i>)	A5, A7, A10
Dificuldades em utilizar o sistema (<i>difficulties in using the system</i>)	A2, A3, A5, A7, A10
Falha na Manipulação de dados (<i>insufficient data handling</i>)	A3, A4, A5, A10
Falta de recursos (<i>missing capabilities</i>)	A5, A10, A11
Surgimento de novos requisitos (<i>new requirements emerge</i>)	A5, A10
Problemas na instalação (<i>problems in installation</i>)	A2, A3, A5, A7, A9, A10
Problemas de travamento e reinício (<i>suspension and resumption problems</i>)	A2, A5, A10
Testadores não realizam um bom trabalho (<i>testers do not perform well</i>)	A5, A7, A10
Efeito no ambiente (<i>the effect on the environment</i>)	A5, A10
Muitas falhas de <i>software</i> (<i>too many software faults</i>)	A4, A5, A10
Resistência do usuário a mudanças (<i>user resistance to change</i>)	A5, A10, A11

3.1 Descrição dos Fatores de Risco identificados na literatura

Com base nos principais fatores de risco apresentados anteriormente na Tabela 5, é feita a seguir uma breve descrição dos fatores de risco identificados na fase de implantação de *software*:

Mudança no ambiente: durante a instalação do *software*, o sistema pode não ser implantado corretamente devido a mudanças de ambiente, principalmente em relação a infraestrutura tecnológica (Shahzad et al., 2009). Esta mudança de ambiente é inevitável em razão do desenvolvimento contínuo, principalmente se o tempo for maior da análise até a entrega e instalação do *software*.

Dificuldade em utilizar o sistema: segundo Menezes (2018), a falta de habilidades técnicas, a baixa experiência e alta rotatividade da equipe podem comprometer o uso do *software*.

Falha na Manipulação de dados: quando o *software* é colocado em funcionamento, ele pode estar sobrecarregado com dados dos usuários e que não podem ser manipulados devido a falhas no sistema (Hijazi, 2014).

Falta de recursos: durante a fase de implantação, os usuários finais estão em busca de novos recursos do *software* recém-implantado (**Board for Software Standardization and Control, 1995e**).

Surgimento de novos requisitos: segundo Hijazi (2014), ao operar o *software*, os usuários finais podem considerar que os novos requisitos devam ser implementados para atender suas necessidades específicas e mudanças na infraestrutura do ambiente da organização.

Problemas na instalação: podem ocorrer problemas na instalação caso os desenvolvedores não tenham experiência suficiente ou conhecimento adequado da estrutura do *software* e como este funciona. Se o sistema é complexo e distribuído e se o ambiente real é desafiador, pode ser difícil instalar o sistema ou a instalação pode ser feita de forma incorreta (Hijazi, 2014).

Problemas de travamento e reinício: durante o processo de implantação, os testadores podem ter dificuldade em decidir continuar a fazer o teste de aceitação ou suspender quando um problema for descoberto (*Board for Software Standardization and Control, 1995e*).

Testadores não realizam um bom trabalho: Durante a execução dos testes, foram identificadas questões associadas à inconsistência, falta de padrões de codificação e de testes adequados, bem como falhas na integração do *software* (Menezes, 2018)

Efeito no ambiente: durante a instalação do sistema, este pode afetar o ambiente em que ele funciona. Para Hijazi (2014), em muitos casos, não há uma aceitação do usuário. Se isso ocorrer, deve ser irrelevante.

Muitas falhas de *software*: segundo Hijazi (2014), se todas as falhas não forem descobertas e mitigadas antes da operação do sistema, elas poderão ser descobertas mais tarde. O custo de descobrir e manter tais falhas aumentará, caso as mesmas sejam descobertas antes.

Resistência do usuário à mudanças: em recentes pesquisas, observou-se que a participação dos usuários finais têm um grande impacto no sucesso ou falha do projeto. Naturalmente, os seres humanos rejeitam as mudanças na forma como eles executam, especialmente se essas mudanças foram impostas externamente. Essa rejeição afeta sua aceitação negativa para o novo sistema (Huang & Han, 2008).

3.2 Causas dos Fatores de Risco na Fase de Implantação

Após a descrição dos fatores de riscos identificados na literatura, é importante destacar as causas destes fatores na fase de implantação. A Tabela 6 apresenta a correlação entre os fatores de risco e suas causas.

Tabela 6. Fatores de risco e causas relacionadas

Fatores de Risco	Causas	Referência dos Artigos
(1) Mudanças no ambiente (<i>change in environment</i>)	(1.1) Alterações na infraestrutura (equipamentos, banco de dados).	A6
(2) Dificuldades em utilizar o sistema (<i>difficulties in using the system</i>)	(2.1) Falta de habilidades do usuário. (2.2) Falta de instruções sobre como utilizar o sistema.	A10
(3) Falha na manipulação de dados (<i>insufficient data handling</i>)	(3.1) Sobrecarga de dados de usuários no banco de dados do <i>software</i> .	A6
(4) Falta de recursos (<i>missing capabilities</i>)	(4.1) Recursos de <i>software</i> insuficientes para o usuário final.	A5
(5) Surgimento de novos	(5.1) Muitas solicitações de mudança	

requisitos (<i>new requirements emerge</i>)	são determinadas pelos clientes, o que gera ao desenvolvedor uma especificação ambígua.	A10
(6) Problemas na instalação (<i>problems in installation</i>)	(6.1) Falta de conhecimento sobre quantos arquivos devem ser configurados. (6.2) As instalações geralmente são feitas de forma distribuída e os usuários não sabem como utilizar o <i>software</i> .	A2, A9
(7) Problemas de travamento e reinício (<i>suspension and resumption problems</i>)	(7.1) Problemas descobertos durante o teste de aceitação.	A6
(8) Testadores não realizam um bom trabalho (<i>testers do not perform well</i>)	(8.1) Baixa prioridade na execução de testes. (8.2) Tempo insuficiente para realização dos testes de <i>software</i> .	A7, A10
(9) Efeito no ambiente (<i>the effect on the environment</i>)	(9.1) Falhas não descobertas antes da implantação do <i>software</i> .	A6
(10) Muitas falhas de <i>software</i> (<i>too many software faults</i>)	(10.1) Desenvolvedores não configuram os recursos implementados para o correto funcionamento do <i>software</i> . (10.2) A prioridade da correção de defeitos é muito baixa.	A4, A10
(11) Resistência do usuário a mudanças (<i>user resistance to change</i>)	(11.1) Mudanças impostas externamente.	A6

Com base na apresentação das principais causas dos fatores de risco na Tabela 5, na próxima seção será apresentado um conjunto de medidas que visam reduzir estes fatores, proporcionando o maior controle e gestão de riscos.

3.3 Estratégias de Contenção

Nesta seção, será apresentada uma lista de estratégias de contenção, com base na literatura, para cada fator de risco identificado na fase de implantação de *software*:

Mudanças no ambiente: Realização de testes em ambiente de produção (Timo et al. 2014).

Dificuldade em utilizar o sistema: Realizar treinamento do usuário (Timo et al. 2014).

Falha na Manipulação de dados: Adoção de uma nova metodologia para análise e controle dos dados existentes (Timo et al. 2014).

Falta de recursos: Atribuir um custo para cada risco identificado, com base na probabilidade de ocorrência e seu nível de impacto (Prasanta et al. 2013).

Surgimento de novos requisitos: Realizar ações que facilitem discussões do grupo, realizando demonstrações ao cliente e testes de aceitação (Supriya, 2015).

Problemas na instalação: Estabelecer novas técnicas de gerenciamento para uma integração adequada do sistema (Prasanta et al. 2013).

Problemas de travamento e reinício: Estabelecer datas de lançamento para integração dos componentes e implantação do *software* (Suprika, 2015).

Testadores não realizam um bom trabalho: Automatizar o gerenciamento de dados de teste usando ferramentas apropriadas (Suprika, 2015).

Efeito no ambiente: Adotar novas ferramentas de hardware e *software* com base nas diretrizes gerenciais da organização. (Suprika, 2015)

Muitas falhas de *software*: Priorizar a correção de defeitos do *software* (Timo, 2014).

Resistência do usuário à mudanças: Treinamento do usuário, no intuito de resolver o problema de aceitação do *software* (Moffat,2013).

3.4 Limitações

As principais limitações relacionadas a presente pesquisa são:

- Em razão da complexidade da fase de implantação de *software* e com base nas dificuldades relacionadas ao processo de desenvolvimento de *software*, não foram aplicadas avaliações estatísticas, uma vez que o trabalho foi limitado a uma revisão da literatura.
- Limitações relacionadas à especificação da fase de implantação: uma especificação mais elaborada do processo de entrega do software poderia contribuir para a formalização e gestão dos artefatos produzidos aos clientes.

3.5 Trabalhos Futuros

O processo de implantação de *software*, apesar da aplicação de técnicas para o gerenciamento de riscos, ainda é propenso a falhas. Como sugestão para trabalhos futuros, um estudo empírico pode ser realizado em instituições públicas, com a identificação e classificação principais fatores de risco existentes, no intuito de validar os resultados esperados e fornecer uma resposta aos gerentes de projeto das atividades que necessitem de maior atenção em relação aos riscos.

4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou, a partir de uma revisão da literatura, os principais fatores de risco na fase de implantação de *software*, suas respectivas causas e estratégias de contenção. Dentre os fatores identificados, destacam-se: (1) problemas durante a instalação do *software*, (2) dificuldades no uso do *software* pelo usuário final; e (3) falha na manipulação de dados.

Assim, os resultados obtidos durante esta revisão, podem contribuir para o gerenciamento de riscos e com isso, reduzir problemas durante a fase de implantação de *software*. E com a aplicação correta das estratégias de contenção, a tendência é melhorar a execução do processo e a produtividade dos projetos, aumentando a eficiência das organizações diante do número de projetos finalizados.

5 REFERÊNCIAS

- Alencar, J. (2012). Análise de risco em gerência de projetos: com exemplos em @Risk / Antônio Juarez de Alencar, Eber Schmitz. 3. ed – Rio de Janeiro: Brasport.
- Baraldi, P. (2010). Gerenciamento de Riscos. 3ª ed. Rio de Janeiro: Campus.
- Biolchini, J., Mian, P.G., Natali, A.C., Travassos, G.H. (2005). Systematic Review in Software Engineering: Relevance and Utility, Relatório Técnico ES-679/05, PESC - COPPE/UFRJ.
- Board for Software Standardization and Control. (1995e). Guide to the software transfer phase. Technical report, ESA.
- Coso (2007). Gerenciamento de Riscos Corporativos - Estrutura Integrada.
- Felizardo, Katia Romero et al. (2017). Revisão sistemática da literatura em engenharia de software: teoria e prática. Rio de Janeiro: Elsevier.
- H. A. H. Gondal, S. M. U. Din, S. Fayyaz, M. D. Zeb and B. Nadeem, Preeminent risk factor affecting software development. (2018). International Conference on Advancements in Computational Sciences (ICACS), Lahore, pp. 1-7.
- Haneen Hizagi, Shehadeh Alqrainy, Hassan Muaid and Thair Kudour. (2014). Risk Factors in software development phases. European Scientific Journal.
- Hiba Arafeh, Adham El-Ahmad. (2017). The Influence of Software Risk Management on Software Project Success. Dept. of Informatics, Lund University School of Economics and Management.
- Huang, S.-J. & Han, W.-M. (2008). Exploring the relationship between software project duration and risk exposure: A cluster analysis. *Information & Management*, 45(3), 175-182. Recuperado de <http://dx.doi.org/10.1016/j.im.2008.02.001>.
- Ieva Ancveire, Ilze Gaillite, Made Gaillite, Janis Grabis. (2015). Software Delivery Risk Management: Application of Bayesian Networks in Agile Software Development. *Information Technology and Management Science*.
- Jones, Capers. (2017). Exceeding 99% in Defect Removal Efficiency (DRE) for Software. Capers Jones, Namcook Analytics LLC.
- Jones, Capers. (2017). Wastage: The Impact of poor quality on software economics. Namcook Analytics LLC.
- Kitchenham, B.; Charters, S. (2007). Guidelines for performing Systematic Literature Reviews. In *Software Engineering*, Technical Report EBSE-2007-01, Department of Computer Science Keele University, Keele.
- Kitchenham, B. (2004). Procedures for Performing Systematic Reviews, Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia.
- Kruchten, P. (2003). *The Rational Unified Process: An Introduction (3 ed.)*. Addison-Wesley.
- Marsh. (2018) Marsh Risk Consulting. Recuperado de <https://www.marsh.com/br/services/marsh-risk-consulting.html>.

- Mata-Lima, H. (2007). Aplicação de Ferramentas da Gestão da Qualidade e Ambiente na Resolução de Problemas. Apontamentos da Disciplina de Sustentabilidade e Impactes Ambientais. Universidade da Madeira (Portugal).
- Menezes, J., Gusmão, C. & Moura, H. Software Qual J (2018). Recuperado de <https://doi.org/10.1007/s11219-018-9427-5>
- Miguel, P.A.C. (2006). Qualidade: enfoques e ferramentas.1 ed. São Paulo: Artliber.
- Moffat Kombo Barongo. (2013). Effects of information system planning on information system implementation process: Case study of universities and university campuses in Kisii, Kenya.
- Nogueira, Marcelo. (2009). Engenharia de Software. Um Framework para a Gestão de Riscos em Projetos de Software, Rio de Janeiro, Ed. Ciência Moderna.
- Pai, M., Mcculloch, M., Gorman, J., Pai, N., Enanoria, W., Kennedy, G., Tharyan, P., Colford Jr., J. (2004). Systematic reviews and meta-analyses: An illustrated step-by-step guide. The National Medical Journal of India, 17(2), pp: 86-95.
- Petersen, K. Feldt, R. Mujtaba, S. and Mattsson, M. (2008). Systematic Mapping Studies, In Anais da 12ª Conferência Internacional no Evalution and Assessment in Software Engineering, p. 68-77, Itália.
- Pmbok. (2013). A Guide to the Project Management Body of Knowledge (Pmbok guide) - Fifth edition. Pennsylvania: s.n.
- Prasanta Kumar Dey, Ben Clegg & Walid Cheffi. (2017). Risk management in enterprise resource planning implementation: a new risk assessment framework. Springer.
- Pressman, Roger S. (2016). *Engenharia de software - Uma abordagem profissional - 7ª Edição*.
- P. Sonchan and S. Ramingwong. (2014). Top twenty risks in software projects: A content analysis and Delphi study, 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Nakhon Ratchasima, pp. 1-6.
- Raghavi K. Bhujang and Suma V. (2014). Risk Impact Analysis across the Phases of Software Development. Lecture Notes on Software Engineering, Vol.2, Nº3, August.
- Ribeiro, M.P; Souza, T.P. (2005). Rational Unified Process: uma abordagem gerencial. IME - Instituto Militar de Engenharia: Rio de Janeiro.
- Ribeiro, Sildenir Alves., Eber Assis Schmitz, Antônio Juarez S. M. de Alencar, Monica Ferreira da Silva. (2017). Research Opportunities on the Application of the Theory of Constraints to Software Process Development. Journal of Software.
- Shahzad, B., Ullah, I., & Khan, N. (2009). Software risk identification and mitigation in incremental model. Proceedings of the International Conference on Information and Multimedia Technology, pages 366–370.
- Siddiqui, Q. A., Burns, N. D., & Backhouse, C. J. (2004). Implementing product data management the first time. International Journal of Computer Integrated Manufacturing, 17(6), 520.
- Sommerville, I. (2016). *Software Engineering*, Addison Wesley.

Suprika V. Shrivastava, Urvashi Rathod. (2014). Categorization of risk factors for distributed agile projects, Symbiosis Centre for Information Technology (SCIT), Symbiosis International University (SIU), Rajiv Gandhi Infotech Park, Hinjewadi, Pune 411 057, Maharashtra, India.

Task Force Report. (2004). Improving security across the software development life cycle (Technical Report). National Cyber Security Summit.

Timo O.A. Lehtinen, Mika V. Mantyla, Jari Vanhanen, Juha Itkonen, Casper Lassenius. (2014). Perceived causes of software projects failures - An analysis of their relationships. Information and Software Technology.

Yusuf, Y, Gunasekaran, A and Abthorpe, M. (2004). Enterprise information systems project implementation: a case study of ERP in Rolls-Royce.

Wohlin, C. and Prikladnicki, R. (2013). Systematic literature reviews in software engineering. Information and Software Technology, 55(6), pp. 919-920.