

Projeto 2.22

Riscos para Manutenção de Software

Kênia Pereira Batista Webster, Káthia Marçal de Oliveira, Nicolas Anquetil

1. Introdução

Melhorar os processos das organizações tem sido uma das preocupações das empresas atualmente, pois, além de ser um diferencial no mercado é também um guia para desenvolvimento ou manutenção de produtos ou serviços. Neste contexto um dos modelos para melhoria de processos mais conhecidos é o modelo de maturidade pelo SEI (*Software Engineering Institute*), CMMI (*Capability Model Maturity Integration*). Este modelo engloba diferentes áreas de processo, entre as quais se encontra a gerenciamnto de riscos.

Gerenciamento de riscos é uma área que tem ganhado muita importância no desenvolvimento de *software*. Alguns autores consideram que esta disciplina é a primeira dentre as melhores práticas de gerência de projetos [1], [5]. No entanto poucos estudos foram desenvolvidos para manutenção de sistemas, dentre eles a adaptação de atividades de gerenciamento de riscos de desenvolvimento para manutenção de *software* [8].

Considerando a importância da manutenção e os poucos estudos realizados sobre riscos nessa área, propomos nesse trabalho buscar identificar os problemas comuns à manutenção de sistemas e utilizá-los na criação de uma lista de verificação para identificação de riscos. Na próxima seção (seção 2) serão abordados alguns conceitos referentes a riscos em desenvolvimento e manutenção de *software*. A seção 3 apresenta alguns conceitos sobre manutenção de *software*. A seção 4 descreve a abordagem utilizada para a criação da lista de verificação para identificação de riscos manutenção para manutenção de *software* utilizando pesquisa de literatura. As conclusões e trabalhos futuros são abordados na seção 5.

2. Riscos

Risco é toda condição ou evento cuja ocorrência é incerta, mas que pode afetar os objetivos do projeto se ocorrer [13]. A disciplina que faz com que se prospere em circunstâncias de grandes incertezas é chamada de gerência de riscos.

A gerência de riscos pode ser vista como um processo que acompanha o produto de *software* durante o seu ciclo de vida, desde a definição do projeto até seu término. Várias são as abordagens de atividades de gerenciamento de riscos que foram encontrados na literatura, tais como:

[4], [9], [12], [14], [20]. Embora as atividades possam variar de uma abordagem para a outra, a atividade de **identificação de riscos** é comum a todas as abordagens encontradas existindo diferentes ferramentas e métodos utilizados para apoiar esta atividade, conforme demonstrado na Alguns do métodos mais utilizados são: entrevistas e utilização de taxonomias ou listas de verificação [9], [10], [20].

Visando facilitar a atividade de identificação dos riscos em desenvolvimento de software categorias de fontes de riscos foram definidos por vários grupos como a proposta do SEI (Software Engineering Institute), do SDRF (Software Development Risk Factor) e os trabalhos de Boehm [3] e Schneidewind [23].

3. Manutenção

A manutenção de *software* é definida como uma modificação no produto de *software* após a entrega, para corrigir falhas, melhorar a performance ou outros atributos e adaptar o produto a um ambiente [2]. Segundo Pigoski (2001) a manutenção é uma atividade inevitável. Quanto a sua natureza a manutenção de *software* é dividida em quatro tipos: corretiva, adaptativa, perfectiva e preventiva [17], [19].

Manter sistemas não é uma atividade trivial já que o sistema é operacional e a equipe de manutenção tem que equilibrar as necessidades de mudança com a necessidade de manter o sistema acessível para os usuários [11]. Além desse fator, outros problemas são frequentemente encontrados nesta atividade:

- Falta de profissionais treinados na equipe com habilidades para manutenção de sistemas [11];
- Falta de ferramentas apropriadas e ambiente para apoiar a manutenção [11];
- Baixa qualidade do sistema a ser mantido [11];
- Dificuldade de reter pessoas com as habilidades necessárias para manutenção de sistemas [11];
- Entendimento limitado do sistema a ser mantido [19];
- Priorizar soluções emergenciais para solucionar os problemas em detrimento de alternativas mais robustas [19];
- O nível de especificação e o tempo para testes são inadequados [19];
- Poucos recursos disponíveis no mercado para manutenção de sistemas [19];
- Baixa moral da equipe [19];

- Pouca ou nenhuma documentação [8];
- Efeitos colaterais (impacto) das mudanças em outras funcionalidades [8];
- Efeitos colaterais (impacto) das mudanças em outros sistemas [8].

4. Lista de Verificação de Riscos para Manutenção de Software

Para propor uma lista de verificação de riscos para manutenção de sistemas inicialmente foi realizada um estudo de viabilidade [24] procurando avaliar os riscos de desenvolvimento propostos por Boehm [3] e os problemas de manutenção encontrados na literatura (citados na seção anterior) que foram descritos como potenciais riscos para manutenção. Essa avaliação tinha como objetivo verificar se tais riscos eram realmente relevantes no contexto de manutenção de software e priorizar sua importância .

A partir desse estudo inicial foi possível constatar que os problemas encontrados na literatura sobre manutenção de sistemas foram realmente identificados como riscos importantes pelo fato desses problemas nem sempre estarem presentes em todos os projetos. Dessa forma esses problemas podem ser utilizados pelos gerentes para auxiliar na identificação de problemas potenciais antes que eles ocorram auxiliando assim a gestão dos riscos.

Visando propor uma lista de verificação de riscos específica para manutenção de sistemas foi realizado um estudo mais abrangente na literatura sobre os problemas exclusivos da manutenção de sistemas e foram encontrados 92 problemas potenciais que foram integrados e refinados em 37 problemas, os mesmos foram considerados como possíveis riscos para a atividade de manutenção sistemas. A Tabela 2 apresenta a lista de verificação de riscos resultante para ser utilizada para identificação de riscos em projetos de manutenção.

Tabela 2. Lista de verificação de riscos para manutenção de sistemas

Fatores de Risco	Descrição do Risco	Referência
Baixa qualidade do sistema a ser mantido	Quando a qualidade do sistema a ser mantido é pobre e qualquer mudança pode acarretar em problemas imprevisíveis.	[11], [13], [15], [17]
Falta de ferramentas de apoio apropriadas	Falta de ferramentas apropriadas e de ambiente para apoiar a manutenção de sistemas, tais como: metodologias, padrões, procedimentos e ferramentas.	[11], [15], [17]
Falta de	Falta de profissionais na equipe treinados com	[8], [11],

Profissionais treinados	habilidades para utilização de ferramentas, metodologias e modelos requeridos para manutenção de sistemas.	[15], [17], [23]
Dificuldade em reter pessoas	A instabilidade das mudanças, a falta de controle, a falta de informação e a falta de tempo. Faz com que as pessoas não dêem continuidade nas atividades de manutenção de sistemas	[11], [15], [17]
Falta de orçamento	Falta ou insuficiência de orçamento para assegurar a implementação das mudanças.	[8], [11]
Resistência dos usuários à mudança	A resistência que os usuários tem com relação às mudanças de um produto de software, por mais importante ou lucrativa que tal mudança possa ser.	[8], [11]
Estratégia Organizacional	Determinar o orçamento de uma manutenção baseado na concorrência com outras empresas rivais. Muitas vezes o desejo de ganhar faz com que o orçamento seja determinado por estratégias organizacionais e não por uma análise objetiva dos problemas.	[11]
Prioridades de gerenciamento	A equipe de manutenção compara os desejos dos clientes com as necessidades do sistema. Frequentemente, as prioridades de gerenciamento se sobrepõem às necessidades técnicas. Algumas vezes os gerentes consideram a manutenção e o aprimoramento mais importantes que a construção de novas aplicações.	[19]
Dificuldade para realização dos testes	O nível de especificação e o tempo para a realização dos testes são inadequados ou falta de dados precisos para testar as mudanças efetuadas.	[17], [18], [19], [23]
Escassez de recursos no mercado	Poucos são os recursos experientes com habilidades em atividades de manutenção de software estão disponíveis no mercado.	[11], [15], [17], [18], [19], [23]
Entendimento limitado	O entendimento do sistema a ser mantido é limitado. Por exemplo, a taxa de limite que uma pessoa pode estudar uma documentação e extrair material relevante ao problema que está sendo resolvido.	[8], [11], [18], [19], [22]
Moral da equipe	Baixa moral e baixa produtividade da equipe pelo fato das pessoas não sentirem reconhecidas ou recompensadas pelos	[15], [18], [19]

	superiores. A equipe pode sentir desmotivada pela pouca importância dada atualmente para as atividades de manutenção de sistemas.	
Pouca ou nenhuma documentação	O sistema a ser mantido não possui documentação ou quando a documentação é existente é insuficiente ou confusa.	[8], [11], [15], [19]
Efeitos colaterais (sistemas)	A execução de mudanças impacta funcionalidades de outros sistemas.	[8], [23]
Efeitos colaterais (funcionalidade)	A execução de mudanças impacta funcionalidades do próprio sistema.	[8], [23]
Inovação tecnológica	Refere-se as mudanças de hardware e/ou software durante as atividades de manutenção de sistemas	[11], [15]
Falta de entendimento do usuário	Os usuários não entendem como o sistema funciona e eles podem fornecer dados incompletos ou errados quando relatarem os efeitos de um problema aos mantenedores.	[15], [19]
Usuários desinteressados	Falta de comprometimento ou interesse do usuário com relação às atividades de manutenção de sistemas.	[15]
Mudanças da organização usuária	Mudança da organização usuária durante a execução da manutenção do sistema.	[11], [15]
Treinamento	Treinamento insuficiente ou inadequado.	[15], [17]
BACKLOG	Grande acúmulo de trabalho a serem executados pelos mantenedores. A equipe de manutenção está sempre tentando equilibrar objetivos distintos.	[15]
Execução	Grande número de falhas no sistema ou no hardware antes da mudança ser executada.	[15], [23]
Processamento	Tempo de resposta ou requisitos de processamento restrito do sistema a ser mantido.	[15], [23]
Confiabilidade do hardware e do software.	O hardware ou software ou suporte técnico não são confiáveis e podem dificultar a solução de um problema.	[15], [19]
Apoio do suporte	Falta de apoio do suporte para ou ocorrem em tempo inoportuno.	[15]
Orçamento	Pressões orçamentárias.	[15]

Mudança de prioridade	Dificuldade em gerenciar mudanças emergenciais. Neste caso, os recursos chaves podem não estar disponíveis e na maioria das vezes as soluções emergenciais afetam o custo e o cronograma das atividades de manutenção.	[17], [19]
Dificuldade de medir desempenho	Dificuldade de medir o desempenho das mudanças realizadas.	[17]
Sistema e tecnologia antiquados	O sistema e tecnologia a serem mantidos estão obsoletos.	[17]
Plano estratégico	Plano estratégico inexistente ou inadequado	[17]
Adaptações das mudanças empresarias.	Adaptar as mudanças referentes ao ambiente empresarial rapidamente.	[17]
Integração	Integrar ou sobrepor sistemas incompatíveis.	[17]
Falta de apoio gerencial	Falta de compreensão e apoio gerencial.	[17]
Alta complexibilidade	Alta complexidade do programa a ser mantido.	[17], [23]
Métricas inexatas	As métricas são subestimadas, devido a vários fatores: dentre eles: não entendimento da mudança, complexibilidade do sistema a ser mantido, número de linhas de código do sistema a ser mantido.	[17], [18]
Falta de tempo	Falta ou insuficiência de tempo para assegurar a implementação das mudanças.	[8], [23]
Requisitos instáveis	Os requisitos de necessários para a manutenção do sistema são instáveis, ou seja, estão sempre mudando.	[8]

Analisando essa lista de verificação pode-se constatar que alguns dos riscos são muito comuns quando se mantém um sistema, como por exemplo, a baixa qualidade do sistema a ser mantido e a falta de documentação. No entanto, outros riscos vão depender muito do projeto e do contexto organizacional em que o sistema será mantido como, por exemplo, estratégia organizacional e resistência dos usuários à mudança. Isso nos permite observar que a existência ou não desses riscos vai variar de projeto a projeto devendo, portanto ser previamente analisado e monitorado numa atividade de gerência de riscos durante a manutenção de software.

5. Conclusão

Gerência de riscos tem sido considerada de grande importância na atividade de gerência de projetos. No entanto, para identificar e gerenciar riscos é necessário uma definição adequada dos riscos em projetos de software. Nesse contexto, vários trabalhos foram propostos para desenvolvimento de software, no entanto poucos estudos foram realizados na área de manutenção.

Essa pesquisa utilizou a revisão de literatura para conhecer o estado da arte sobre os problemas existentes na manutenção de *software* para apoiar a criação de uma lista de verificação para identificação de riscos para manutenção de *software*. Permitindo ajudar os gerentes responsáveis pela manutenção de sistemas a identificarem os riscos com mais agilidade e precisão.

Com relação aos trabalhos futuros essa lista será integrada com os riscos encontrados na literatura para desenvolvimento de sistemas, dentre a literatura encontrada iremos destacar as seguintes: [3], [6], [12], [16]. Essa integração baseia-se no estudo realizado por Charette, Adams e White (1997) onde eles defendem que alguns fatores de riscos podem ser semelhantes, como por exemplo, a volatilidade de requisitos, a inexperiência dos recursos humanos e restrições de custo. Apesar de semelhantes, os riscos podem ter conseqüências mais fortes na manutenção que no desenvolvimento.

6 Referências Bibliográficas:

1. ADDISON, T.; VALLABH, S. Controlling Software Project Risks – an Empirical Study of Methods used by Experienced Project Managers. Proceedings of SAICSIT, p.128-140, 2002.
2. BENNETT, K. H.; RAJLICH, V. T. Software Maintenance and Evolution: a Roadmap. ACM, p. 75-87, 2000.
3. BOEHM, Barry. Risk management. Piscataway: IEEE Computer Society Press, 1989.
4. BOEHM, B. W. Software Risk Management: Principles and Practices. IEEE, p.32-41, Janeiro, 1991.
5. BROWN, N. Industrial-strength management strategies. IEEE Software, 1996, p.94-103.
6. CARR, M. J.; KONDA, S. L., MONARCH, I.; ULRICH, F. C.; WALKER, C. F. Taxonomy-Based Risk Identification. Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1993. Technical report CMU/SEI-93-TR-6.

7. CCTA – Central Communications and Telecommunications Agency – UK Civil Service. Risk Handbook. 2000.
8. CHARETTE, R. N.; ADAMS, K. M.; WHITE, M. B. Managing Risk in Software Maintenance. IEEE Software, p.43-50, 1997.
9. CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. CMMI® – Guidelines for Process Integration and Product Improvement. Boston: Addison-Wesley, 2002.
10. DOROFEE, A. J.; WALKER, J. A.; ALBERTS, C. J.; HIGUEIRA, R. P.; MURPHY, R. L.; WILLIAMS, R. C. Continuous Risk Management Guidebook. Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1996
11. GRUBB, P.; TAKANG, A. A. Software Maintenance: Concepts and Practice. Second Edition. Danvers, MA: World Scientific, 2003.
12. HIGUEIRA, R. P.; HAIMES, Y. Y. Software Risk Management (CMU/SEI-96-TR-012). Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1996.
13. HOUNSTON, A. X; MACKULAK, G. T.; COLLOFELLO, J. S. Stochastic Simulation of Risk Factor Potencial Effects for Software Development Risk Management. Elsevier Science, p. 247-257, 2001.
14. JALOTE, Pankaj. Software Project Management in Practice. Boston: Addison-Wesley, 2002. cap.6, p.93-108.
15. LIENTZ, B. P.; Swanson E. B. Problems in Application Software Maintenance. ACM, p. 763-769, 1981.
16. MACHADO, C. A. F. A-RISK: Um Método para Identificar e Quantificar Risco de Prazo em Desenvolvimento de Software. 2002. Dissertação (Mestrado em Ciência) – Curso de Pós-graduação em Informática Aplicada - PPGIA, Universidade Católica do Paraná - PUCPR.
17. PIGOSKI, T. M. Pratical Software Maintenance: Best Practices for Software Investment. Wiley, 1997.
18. PIGOSKI, T. M. SWEBOK - Guide to the Software Engineering Body of Knowledge: Software Maintenance. IEEE, cap.16, p.1-16, 2001.
19. PFLEEGER L. S. Software Engineering Theory and Practice. Second Edition. River, New Jersey: Prentice Hall, 2001.
20. PMI – Project Management Institute. PMBOK – Um Guia do Conjunto de Conhecimentos do Gerenciamento de Projetos. Edição 2000. New Square, PA.: Four Campus Boulevard, 2002. cap.11, p.127-146.
21. SCHNEBERGER, S. L. Software Maintenance In Distributed Computer Environments: System Complexity Versus Component Simplicity. IEEE, p. 304-313, 1995.

22. SCHNEIDEWIND, N. F. Requirements Risk and Maintainability. Idea group Inc., cap.7, p.182-200, 2003
23. WEBSTER, K. P. B; OLIVEIRA, K. M. O; ANQUETIL, N. Priorização de Riscos para Manutenção de Software. JIISIC, 2004.