

## Taxonomia de Riscos para Manutenção de Software

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

Universidade Católica de Brasília  
SGAN 916 - Módulo B - Asa Norte, Brasília, Brasil, 70.790-160  
keniawebster@yahoo.com.br; {kathia, anquetil}@ucb.br

**Abstract.** *Risk management is an important area of software engineering that some consider as the foremost best practice in project management. Despite this importance we could realize that risk management for software maintenance –which represents more than 80% of software activity- was very ill explored. Many problems occurring in maintenance projects could come from faulty or inexistent risk management activity. This work proposes a taxonomy of possible risk for software management projects. It could be used as a guide in the identification of the potential risks of any such project.*

**Resumo.** *Gerenciamento de risco é uma das atividades fundamental da engenharia de software, considerado por alguns como a primeira dentre as melhores práticas de gerência de projetos. Apesar desta importância, o gerenciamento de riscos em manutenção de software é uma atividade pouco explorada. Embora a manutenção de software seja uma das atividades mais crítica e longa do ciclo de vida do software poucos estudos foram desenvolvidos nessa área. Muitos dos problemas encontrados nessa atividade podem ser decorrentes de falhas na gerência de risco. Este trabalho propõe uma taxonomia de riscos para manutenção de software, que possa ser usada no processo de identificação de riscos ao manter software.*

### 1. Introdução

O gerenciamento de riscos é uma área que tem crescido muito em importância no desenvolvimento de software. Alguns autores consideram que esta disciplina é a primeira dentre as melhores práticas no gerenciamento de projetos [Addison e Vallabh 2002], [Brown 1996]. No entanto, a maioria dos trabalhos realizados nesta área foca projetos de desenvolvimento e poucos estudos foram desenvolvidos para manutenção de software com o foco nesta disciplina [Charette et al. 1997].

Manutenção é uma atividade fundamental que consome a grande parte do tempo e dos recursos investidos em software, pois, de acordo com as estimativas, os custos de manutenção correspondem em até 80% do custo referente ao tempo de vida de um software [Pfleeger 2001]. A manutenção de software oferece também diferenças significativas com o desenvolvimento e os riscos encontrados nas duas atividades não são necessariamente os mesmos [Charette et al. 1997].

Por outro lado, segundo Schneidewind (2003), avaliar riscos relacionados à implementação de solicitações de mudanças, ajuda a melhorar a qualidade do software entregue, pois, os defeitos relacionados com especificações das mudanças podem ser antecipadamente identificados. Um outro estudo indica que a utilização da gerência de riscos contribui para tornar o gerenciamento de mudanças mais efetivo [CCTA 2000].

Considerando a importância da manutenção e os poucos estudos realizados sobre riscos nessa área, propomos nesse artigo a criação de uma taxonomia de risco para manutenção de software, que possa ser utilizada como uma ferramenta no processo de identificação de riscos em projetos de manutenção. Este artigo foi organizado como segue: a seção 2 aborda 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 na criação da taxonomia de riscos focada na manutenção de software, e detalha a própria taxonomia. As conclusões e trabalhos futuros são abordados na seção 5.

### **2. Riscos**

Um projeto é considerado bem sucedido quando é desenvolvido dentro das expectativas de tempo, custo, qualidade, satisfação do cliente e alta moral da equipe durante todo o ciclo de vida do projeto. Entretanto, muitas vezes, fatores internos e externos têm influência decisiva no sucesso ou fracasso de um projeto [Ropponen e Lyytinen 2000]. Dessa forma, as incertezas irão acompanhar o projeto de software durante todo o seu ciclo de vida. Essas incertezas podem ser definidas como riscos, pois, segundo Jalote (2000, 2002) e McManus (2004) risco é toda condição ou evento cuja ocorrência é incerta, mas que pode afetar os objetivos do desenvolvimento software se ocorrer.

A disciplina que faz com que se prospere em circunstâncias de grandes incertezas é chamada de gerência de risco. O gerenciamento de riscos é particularmente importante para projetos de software, devido às incertezas inerentes que a maioria dos sistemas enfrentam [Valeriano 2001]. As práticas referentes ao gerenciamento de riscos foram introduzidas com o objetivo auxiliar os gerentes a identificar problemas potenciais antes que eles ocorram, de maneira que os processos de análise e planejamento de respostas aos riscos possam ser realizados quando necessário, para minimizar a probabilidade e conseqüências que impactos adversos possam trazer para os objetivos do projeto durante o seu ciclo de vida [PMI 2002]. A gestão de riscos é considerada, portanto, um componente fundamental no desenvolvimento de software [Addison e Vallabh 2002].

Várias são as abordagens de processos de gerenciamento de riscos encontradas na literatura, tais como: Boehm [Boehm 1989], Continuous Risk Management Guidebook [Dorofee et al. 1996], PMBOK - Project Management Body of Knowledge [PMI 2002], [Schwalbe 2002], Microsoft Solutions Framework [Microsoft 2002] ou Capability Maturity Model Integration [Chrissis et al. 2002]. Embora os processos de gerenciamento de riscos possam variar de uma abordagem para outra, os processos de identificação e análise de riscos são comuns a todas as abordagens. A identificação de riscos pode ser definida como um processo que envolve determinar quais as condições, eventos, situações de incertezas, perigos, ameaças, vulnerabilidades e questões relacionadas ao projeto que se ocorrerem podem afetar os objetivos do projeto [Chrissis, Konrad e Shrum 2002], [PMI 2002]. Trata-se de uma atividade importante porque o conhecimento é pré-condição para as ações de controle. Por outro lado, a análise de risco tende a diminuir a subjetividade na avaliação do risco, mostrando aos gerentes de projeto e aos níveis diretos de uma organização qual a probabilidade do projeto alcançar ou não os seus objetivos.

A identificação dos riscos de um projeto pode ser facilitada pelo uso de uma taxonomia de risco. Segundo Sommerville (2003), taxonomia de riscos é uma

categorização dos tipos possíveis fatores de riscos. Essa ferramenta apresenta vantagem com relação às demais ferramentas, pois, sua utilização torna o processo de identificação mais rápido e simples (Hounston et al. 2001). Uma taxonomia é composta por fatores que podem ser categorizados de várias formas, tais como: econômico, técnico e comportamental (Barki et al. 1993). Esses fatores são comumente chamados de fatores de risco ou indutores de risco porque, se presentes no projeto de software, podem afetá-lo tanto positivamente quanto negativamente.

### **3. Manutenção de Software**

Durante as últimas décadas, houve uma proliferação de sistemas de software em grande escala nos ambientes de trabalho. Raros são os setores da sociedade que não são exploraram sistemas de software em suas operações diárias. Para ocorrer um aumento de confiança em sistemas de software é importante que tais sistemas realizem bem o trabalho que lhes foi proposto. Se esses sistemas não forem úteis, os usuários não os aceitam e conseqüentemente não os usam [Grubb e Takang 2003]. Alguns dos fatores que sustentam a utilidade dos sistemas de software são: funcionalidade, flexibilidade, avaliação contínua e operações corretas. As mudanças serão solicitadas geralmente para suportar estes fatores durante a vida de um sistema. Por exemplo, as mudanças podem ser necessárias para atender solicitações de melhoria de desempenho, melhorar funcionalidades e corrigir erros encontrados no sistema [Pfleeger 2001].

A manutenção de software é definida como toda atividade realizada para permitir dar um suporte eficiente a um sistema em operação [Pigoski 1996]. Segundo Pigoski (2001) a manutenção é uma atividade inevitável durante o ciclo de vida do software. Ela pode ocorrer para corrigir falhas, melhorar a performance ou outros atributos ou adaptar o produto a um ambiente [ISO/IEC 12207 apud Bennett e Rajlich 2000].

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 [Pfleeger, 2001]. Tipicamente, mantenedores devem operar sem documentação atualizada do sistema, resgatando as informações necessárias do código fonte. Essa falta de documentação ou outros tipos de informação sobre um sistema é uma grande fonte de risco para os projetos de manutenção já que os mantenedores operem modificações sem ter o conhecimento suficiente das conseqüências que essas podem ter sobre o resto do sistema, sem saber em detalhe o que o sistema deve fazer em todos os casos, ou desconhecendo as regras de negócio que se aplicam [Charette et al. 1997].

### **4. Abordagem utilizada na criação da taxonomia de riscos para manutenção de software**

Vários processos foram definidos com o propósito de gerenciar riscos no desenvolvimento de software [Chrissis et al. 2002], [PMI 2002], [Microsoft 2002], [Dorofee et al. 1996] [Boehm 1989]. No entanto, ao contrário do desenvolvimento, as práticas de gerenciamento de riscos para manutenção de software, ainda estão pouco exploradas [Charette et al.1997]. Se os processos de gerência de riscos para manutenção podem ser os mesmos do desenvolvimento, envolvendo o planejamento, identificação, análise, controle e monitoração de riscos [Charette et al. 1997], os riscos envolvidos em projetos de manutenção podem ser diferentes. Mesmo quando alguns fatores de risco do

## IV Simpósio Brasileiro de Qualidade de Software

desenvolvimento de software são semelhantes à manutenção, como por exemplo, a volatilidade de requisitos, eles podem ter conseqüências mais fortes na manutenção por causa das condições diferentes em que ela acontece (e.g. falta de documentação, ver seção precedente).

A proposta deste artigo é a criação de uma taxonomia que auxilie o processo de identificação de riscos em projeto de manutenção de software. Essa taxonomia é baseada na adaptação de taxonomias similares focadas no desenvolvimento de software e na identificação de problemas específicos à manutenção. Este trabalho, se faz necessário porque não existe, na literatura, tal taxonomia para a manutenção de software. Por exemplo, embora Charette et al (1997), defendam que existem fatores de riscos específicos para manutenção de software, o trabalho desenvolvido por eles utiliza a taxonomia de riscos proposta pelo SEI (*Software Engineering Institute*) focada no desenvolvimento de software.

Para criarmos a taxonomia de riscos para manutenção de software, pesquisamos na literatura os fatores de riscos já definidos. Encontramos 382 fatores de riscos definidos para o desenvolvimento e apenas um trabalho [Schneidewind 2003] definindo sete fatores de riscos para a manutenção. Os 382 fatores de riscos foram encontrados nas referências<sup>1</sup> apresentadas na Tabela 1.

**Tabela 1. Quantidade de fatores de riscos de desenvolvimento por referência.**

Referência	Quantidade de riscos identificados
1) [Boehm 1991], [Kwak e Stoddard 2003], [Addler <i>et al</i> 1998]	10
2) [Soeiro 1999]	18
3) [Hounston <i>et al</i> 2001]	30
4) [Keil <i>et al</i> 1998]	11
5) [Machado 2002]	62
6) [Addison e Vallabh 2002]	14
7) [Mizuno e Kikuno 2000]	23
8) [Sommerville 2003] (capítulo 4)	14
9) [McManus 2004]	20
10) [Schwalbe 2002]	21
11) [DeMarco e Lister 2003]	5
12) [Carr <i>et al</i> 1993]	64
13) [Jalote 2000, 2002]	10
14) [Pressman 2002]	10
15) [Fontoura e Price 2004]	13
16) [Leopoldino 2004]	32
17) [Farias 2002]	15
18) [Willians <i>et al</i> 1997]	10

Na falta de mais fatores de riscos específicos para manutenção, procuramos publicações listando problemas ou falhas decorrentes desta atividade, pois, segundo o PMI (2002) o conhecimento adquirido em projetos similares realizados anteriormente e outras fontes de informação, podem ser utilizados como base na elaboração dos fatores

<sup>1</sup> Os números apresentados na frente das referências são utilizados no resto do artigo para representar as referências bibliográficas.

## IV Simpósio Brasileiro de Qualidade de Software

de riscos. Além disso, realizamos um estudo de viabilidade [Webster, Oliveira e Anquetil 2004] no qual 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. Foram encontrados 84 problemas e falhas nas referências, que somando aos 7 riscos para manutenção de software proposto por [Schneidewind 2003], totalizou 91 riscos para manutenção (Tabela 2).

**Tabela 2. Quantidade de Fatores de riscos por referência bibliográfica**

Referência	Quantidade de riscos identificados
19) [Grubb e Takang 2003]	7
20) [Pfleeger 2001]	12
21) [Charette <i>et al</i> 1997]	7
22) [Lientz e Swanson 1981]	23
23) [Pigoski 1997]	18
24) [Schneidewind 2003]	7
25) [Pigoski 2001]	7
26) [Schneberger 1995]	1
27) [Sommerville 2003] capítulos 26 e 27	9

Julgamos que uma taxonomia de 473 (ou seja, 382 + 91) fatores de riscos para manutenção seria praticamente tão prejudicial quanto à inexistência de uma taxonomia, e dessa forma, decidimos integrar os 473 fatores de riscos. Essa integração consistiu na realização de três atividades: primeiramente, fizemos uma análise dos fatores de riscos pertinentes ao desenvolvimento de software para identificar possíveis duplicações. A Tabela 3 apresenta dois exemplos de integração de fatores de riscos provenientes de referências diferentes.

**Tabela 3. Quantidade de Fatores de riscos por referência bibliográfica**

Fator de riscos integrado	Fatores de riscos identificados na literatura para desenvolvimento de <i>software</i>
Requisitos de desempenho do <i>software</i> não atendidos.	Desempenho aquém do esperado (incluindo erros e qualidade) (2).
	Desempenho abaixo do esperado (11).
	Requisitos de desempenho não atendidos (13).
	Falha ou baixo desempenho na conexão (13).
Pouca confiança no desenvolvimento do sistema devido à indisponibilidade dos componentes ou erro / mau funcionamento dos componentes.	Desenvolvimento de funções de <i>software</i> não confiáveis (1).
	Falta de confiança na entrega dos sub-projetos (3).
	Desenvolvimento de funções de <i>software</i> não confiáveis (9).
	Pouca confiança no desenvolvimento do sistema devido à indisponibilidade dos componentes ou erro / mau funcionamento dos componentes / subprojetos (12).

A segunda atividade consistiu em eliminar todos os fatores de riscos para o desenvolvimento que não foram referenciados por pelo menos dois autores diferentes. Nosso objetivo nessa seleção era identificar os fatores de riscos mais pertinentes, eliminando aqueles que foram propostos por apenas um autor, possivelmente num contexto específico e dificilmente generalizável. Quando um autor reutiliza na integra a taxonomia de um outro (por exemplo, a taxonomia de Boehm ou a do SEI, são dentro das mais referenciadas e reutilizadas), decidimos não contar isso como uma segunda referência, julgando que o segundo autor podia não ter identificado especificamente os

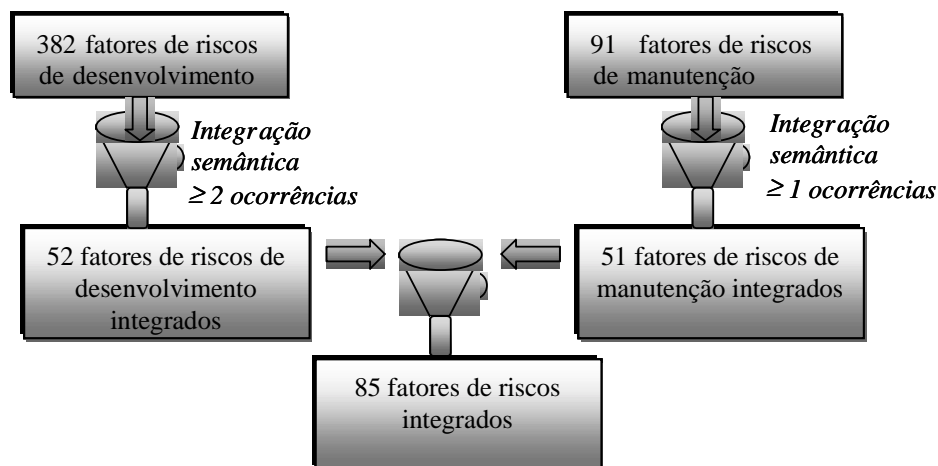
## IV Simpósio Brasileiro de Qualidade de Software

riscos do autor referenciado, mas apenas quis acrescentar novos riscos aos já conhecidos.

A terceira atividade consistiu em analisar os fatores de riscos pertinentes a manutenção de software para identificar possíveis duplicações de forma semelhante ao que foi feito com os riscos de desenvolvimento. Neste caso, não eliminamos os fatores de riscos que foram definidos em uma única referência, devido, a carência de publicações que explorem esse assunto.

A figura 1 apresenta as atividades que compõem a integração dos fatores de riscos realizada em três atividades:

- Integração e seleção dos 382 riscos de desenvolvimento. O resultado foi um grupo de 52 riscos, deixando 136 riscos com uma única referência.
- Integração e seleção dos 91 problemas e riscos de manutenção. O resultado foi um grupo de 51 riscos.
- Integração dos 52 riscos de desenvolvimento e dos 51 riscos de manutenção. O resultado foi um grupo de 85 riscos de manutenção e desenvolvimento.



**Figura 1. Integração dos vários fatores de riscos para gerar a taxonomia de riscos de manutenção**

Após a realização dessas atividades, obtivemos como resultado uma lista de 91 fatores de riscos para projetos de manutenção. Para completar a criação da taxonomia, foi necessário organizar os riscos em categorias. Escolhemos nos basear na organização proposta na taxonomia do SEI [Carr et al. 1993] que é o trabalho que foi mais referenciado (referenciado em: [McManus 2004], [Machado 2002], [Jalote 2000, 2002], [Pressman, 2002], [Charette et al. 1997], [Williams et al. 1997], [Higueira e Haimes 1996], [Dorofee et al., 1996]).

A taxonomia proposta pelo SEI é organizada em três níveis: classes são decompostas em elementos que contém os riscos. Usamos as mesmas três classes definidas pelo SEI, adaptando o nome quando necessário: Engenharia de Produto (riscos técnicos), Ambiente de Manutenção (riscos metodológicos) e Restrições de Programa (riscos organizacionais). Somente a classe Ambiente de Manutenção teve sua descrição alterada, pois, era descrita como Ambiente de Desenvolvimento.

## IV Simpósio Brasileiro de Qualidade de Software

Com relação aos elementos, utilizamos praticamente os mesmos da taxonomia do SEI com exceção de cinco adaptações:

- Juntamos dois elementos da classe Engenharia de Produto “Codificação e teste unitário” e “Teste de integração” em um novo elemento denominado “Teste” porque várias taxonomias não entram nesse nível de detalhe, mas consideram apenas o fator de risco mais geral (Teste).
- Alteração do nome do elemento Código e Teste Unitário para Código. Essa alteração foi necessária, pois, os fatores de risco associados a Código não estavam relacionados a Teste.
- Excluímos o elemento de Contrato da categoria Restrições do Programa. O único fator de risco do elemento Contrato (que permaneceu após a seleção) foi adicionado ao elemento Interfaces do Programa.
- Mudamos as descrições de dois elementos da classe Ambiente de Manutenção de “Processo de desenvolvimento” para “Processo de manutenção” e de “Desenvolvimento do Sistema” para “Manutenção do sistema”.
- Criação do elemento Legado na classe Engenharia do Produto. A criação deste elemento foi necessária, pois, a taxonomia proposta pelo SEI é muito focada no desenvolvimento de sistemas e alguns fatores de risco selecionados estavam voltados para a manutenção especificamente.

Dentro desses elementos, classificamos os 85 fatores de riscos de manutenção que identificamos. É interessante notar que destes 85 fatores de riscos, somente 35 estavam presentes na taxonomia proposta pelo SEI, 17 destes fatores de riscos são referentes ao desenvolvimento mas não foram citados na proposta do SEI. Dos 51 fatores e riscos que identificamos na literatura, apenas 33 são fatores de riscos novos, enquanto 18 já apareciam dentre os riscos de desenvolvimento que selecionamos (12 aparecem na taxonomia do SEI e 6 aparecem em outras fontes sobre riscos de desenvolvimento). Finalmente, um fator de risco estava dentro do 136 riscos de desenvolvimento que tínhamos descartado porque encontrados em uma referência só. Pelo fato de eles aparecem neste momento também como riscos de manutenção, foram re-integrados na taxonomia final, deixando assim 135 riscos de desenvolvimento que não consideramos.

Da mesma maneira que tivemos que adaptar o enunciado de alguns elementos de desenvolvimento para manutenção, as descrições de 23 fatores de riscos também foram adaptadas. A tabela 4 apresenta exemplos de tais alterações.

**Tabela 4 – Exemplo de alteração na descrição dos fatores de riscos pertinentes ao desenvolvimento**

<b>Fatores de Riscos pertinentes ao desenvolvimento de <i>software</i></b>	<b>Fatores de riscos alterados para manutenção de <i>software</i></b>
Requisitos de desempenho do <i>software</i> não atendidos.	Requisitos de desempenho do <i>software</i> a ser mantido não atendidos.
Pouca confiança no desenvolvimento do sistema devido à indisponibilidade dos componentes ou erro / mau funcionamento dos componentes.	Pouca confiança na manutenção do sistema devido à indisponibilidade dos componentes ou erro / mau funcionamento dos componentes.
Requisitos incompletos	Requisitos incompletos do <i>software</i> a ser mantido.
Adicionar mais funcionalidades do que o especificado / dar extras ao cliente	Adicionar mais funcionalidades no <i>software</i> a ser mantido que o especificado / dar extras ao cliente

## 5. Detalhamento da taxonomia para manutenção de software

Como descrito na seção precedente, a taxonomia de riscos proposta neste trabalho é composta por classes, elementos e fatores de riscos. Sendo que os dois primeiros itens foram criados para categorizar os fatores de riscos. A taxonomia tem três classes: Engenharia de Produto, Ambiente de Manutenção e Restrições de Programa. A seguir iremos detalhar melhor essas classes, os elementos que elas contem e os fatores de riscos classificados nesses elementos.

A classe Engenharia do Produto compreende as atividades de engenharia de sistema e de software relacionadas à manutenção de um produto que satisfaçam os requisitos especificados e as expectativas do cliente. Ela é dividida em seis elementos:

- **Requisitos:** agrupa os fatores de riscos relacionados com a definição do que o produto de software deve fazer, isso inclui, como o software irá se comportar e como será usado. Este elemento é composto por seis fatores de risco.
- **Projeto:** agrupa os fatores de riscos responsáveis por traduzir os requisitos do produto de software em um projeto efetivo. Este elemento é composto por dez fatores de risco.
- **Código:** agrupa os fatores de risco relacionados com as atividades de transformação do projeto de software em código para satisfazer os requisitos exigidos pelo cliente. Este elemento é composto por sete fatores de risco.
- **Teste:** agrupa os fatores de riscos relacionados com as atividades de integração das unidades de um sistema e os da validação, se o produto de software executa os requisitos exigidos pelo cliente. Este elemento é composto por dois fatores de risco.
- **Especialidade de engenharia:** agrupa os fatores de riscos relacionados com o ambiente geral dentro do qual o trabalho será executado, inclusive as atitudes das pessoas e os níveis de cooperação. Este elemento é composto por quatro fatores de risco.
- **Legado:** agrupa os fatores de risco relacionados com o estado do sistema a ser mantido (principalmente artefatos não relacionados a código) que antecedem a manutenção. Este elemento é composto por três fatores de risco.

Os fatores de riscos desta classe são listados na Tabela 5. Os números da segunda coluna correspondem às referências listadas na seção precedente (tabelas 2 e 3).

**Tabela 5 – Taxonomia de Riscos para a Classe: Engenharia do Produto**

Fatores de Riscos	Referência
<b>Requisitos</b>	
Requisitos instáveis do <i>software</i> a ser mantido.	(1), (2), (3), (5), (6), (8), (9), (11), (12), (13), (14), (15),(16),(17),(21)
Requisitos incompletos do <i>software</i> a ser mantido.	(4), (12)
Requisitos não claros (ambíguos / imprecisos).	(5), (7), (12), (13)
Requisitos mal entendidos (não refletem as expectativas do cliente).	(4), (5), (6), (7), (12), (15)
Adicionar mais funcionalidades no <i>software</i> a ser mantido que o especificado / dar extras ao cliente.	(1), (6), (9), (15)
Requisitos de desempenho do <i>software</i> a ser mantido não atendidos. Ex. Baixo desempenho.	(2), (11), (13)



## IV Simpósio Brasileiro de Qualidade de Software

**Tabela 5 (cont) – Taxonomia de Riscos para a Classe: Engenharia do Produto**

Fatores de Riscos	Referência
<b>Projeto</b>	
O entendimento do sistema a ser mantido é limitado. Ex.: o limite de cada pessoa quanto a estudar a documentação e extrair material relevante para o problema a ser resolvido.	(20), (23), (25), (26)
Características de <i>hardware</i> não confiáveis ou um suporte técnico do fabricante, dificultando a identificação da fonte de um problema do <i>software</i> a ser mantido.	(20), (22)
Alto nível de complexibilidade técnica das mudanças solicitadas.	(3), (5), (12), (24)
Desenvolvimento de interface do usuário inadequada do <i>software</i> a ser mantido.	(1), (2), (9)
Problemas de desempenho de tempo real (há tempos de respostas restritos).	(1), (9), (12)
Restrições referentes ao <i>hardware</i> designado. Ex.: capacidade de memória, acesso à base de dados e insuficiência de <i>hardware</i> .	(1), (9), (12), (22), (24)
Tecnologia nova / imatura (uso de novas tecnologias) do <i>software</i> a ser mantido.	(2), (4), (5), (6), (13), (15), (16)
Tecnologias e sistemas a serem mantidos, antiquados.	(23), (27)
Baixa qualidade do projeto. Se o software a ser mantido não for formado por componentes coesos e independentes, encontrar e corrigir a fonte de um problema pode acarretar em mudanças que criam efeitos não previstos em outros componentes.	(19), (20), (22)
Estrutura corrompida do software a ser mantido.	(27)
<b>Código</b>	
Código do programa a ser mantido complexo e desestruturado.	(23)
A execução das mudanças impacta funcionalidades de outros sistemas.	(21)
A execução de mudanças impacta funcionalidades do próprio sistema.	(21)
Grande número de linhas de código afetadas com a mudança.	(24)
A área do programa afetado é crítica.	(24)
Grande número de funções principais que são afetadas pela mudança.	(24)
Dificuldade dos mantenedores em entender os truques de programação. Ex.: truques utilizados para otimizar a utilização do espaço de memória ou a velocidade de execução.	(27)
<b>Teste</b>	
Inadequada preparação e realização dos testes. Ex.: instalações inadequadas, tempo insuficiente, falta de dados precisos para testar as mudanças ou utilização de método de teste inadequado.	(9), (12), (20), (23), (24)
Alto custo para testar uma mudança em termos de orçamento e tempo.	(25)
<b>Especialidade de engenharia</b>	
Falta de um acordo interativo entre cliente e os mantenedores relativos às especificações dos requisitos.	(7), (12)
Inadequadas especificações de mudanças. Ex.: especificações de sistema, <i>hardware</i> , <i>software</i> , interface, requisitos de teste) a qualquer nível com respeito à viabilidade de implementação e a estabilidade dos atributos de qualidade, completeza e clareza.	(7), (11), (12), (17), (20)
Escopo / objetivos pouco claros ou equivocados do <i>software</i> a ser mantido.	(6), (10), (15), (16)
Dificuldade em manter sistemas de tempo real e sistemas altamente sincronizados.	(20)
<b>Legado</b>	
Pouca ou nenhuma documentação. O software a ser mantido não possui documentação ou quando a documentação existente é insuficiente.	(20), (21), (22), (23), (27)
Pobre integridade dos dados do software a ser mantido	(22)
Os dados processados pelo sistema armazenados em diferentes arquivos, que podem ter estruturas incompatíveis. Ex.: Alguns dados podem ser duplicados, desatualizados, inexatos ou incompletos.	(27)

A classe Ambiente de Manutenção está relacionada ao ambiente que o produto de software será mantido. Ela é dividida em cinco elementos:

## IV Simpósio Brasileiro de Qualidade de Software

- **Processo de manutenção:** relaciona os fatores de riscos referentes à definição, planejamento, documentação, execução, e comunicação dos métodos e procedimentos usados para manter o produto de software. Este elemento é composto por sete fatores de risco.
- **Sistema de Manutenção:** agrupa os fatores de riscos relacionados com as ferramentas e equipamentos de suporte usados manutenção do produto, tais como, ferramentas CASE, simuladores e compiladores. Este elemento é composto por seis fatores de risco.
- **Processo de Gerência:** agrupa os fatores de riscos relacionados com o planejamento, monitoração e controle de orçamento e cronograma. Além dos fatores de riscos, relacionados com o controle de fatores que envolvem definição, implementação, testes do produto e a experiência do gerente em manutenção de software. Este elemento é composto por nove fatores de risco.
- **Métodos Gerenciais:** agrupa os fatores de riscos relacionados com os métodos, ferramentas, e equipamentos de suporte usados para gerenciar e controlar a manutenção do produto, tais como, ferramentas de monitoração, gerenciamento de pessoas, garantia da qualidade e gerência de configuração. Este elemento é composto por 12 fatores de risco.
- **Ambiente de trabalho:** relaciona os fatores de risco referente ao ambiente geral dentro o qual o trabalho será realizado, incluindo as atitudes das pessoas e os níveis de cooperação, comunicação, e moral. Este elemento é composto por seis fatores de risco.

Os fatores de riscos da classe de Ambiente de Manutenção são listados na Tabela 6.

**Tabela 6 – Taxonomia de Riscos para a Classe: Ambiente de Manutenção**

Fatores de Riscos	Referência
<b>Processo de Manutenção</b>	
Os modelos, processos, padrões, métodos e ferramentas de apoio selecionados são inexistentes / inadequados para o processo de manutenção de software.	(5), (12), (19), (23)
Manutenibilidade não é incorporada no processo de desenvolvimento de software.	(12), (25)
Falta / insuficiência de controle do processo de manutenção de software. Ex.: uso do recurso, medição do progresso referente à qualidade e metas de produtividade	(7), (12), (16)
Controle do produto inadequado. Ex.: o produto final não corresponde as expectativas do cliente.	(12), (17)
Documentação / papelada excessiva referente ao software a ser mantido	(2), (3)
Falta de aderência aos padrões de desenvolvimento de software	(22)
Dificuldade em adaptar rapidamente as mudanças referentes ao ambiente de negócios.	(23)
Dificuldade em adaptar rapidamente as mudanças referentes ao ambiente de negócios.	(23)
<b>Manutenção do sistema</b>	
O sistema de manutenção não possui capacidade suficiente. Ex.: falta de estações de trabalho, espaço de armazenamento no banco de dados insuficiente	(8), (12)
Pouca confiança no sistema de manutenção devido à indisponibilidade dos componentes ou erro / mau funcionamento dos componentes / subprojetos.	(1), (3), (9), (12)
Pobre suporte ao software que está sendo mantido. Ex.: treinamento para utilização do sistema, acesso para usuários especialistas ou consultores, conserto ou resolução de problemas por parte dos vendedores	(5), (12)
Falha ao executar software a ser mantido.	(22)
Mudanças no sistema de hardware e software.	(22)
Falta de suporte para reengenharia do software a ser mantido	(23)

## IV Simpósio Brasileiro de Qualidade de Software

**Tabela 6 (cont.) – Taxonomia de Riscos para a Classe: Ambiente de Manutenção**

<b>Processo de gerência</b>	
Planejamento inapropriado, incluindo construção e atualização do plano de contingência do software a ser mantido.	(2), (5), (7), (10), (12)
Os papéis e as responsabilidades de relacionamentos mal definidos ou mal entendidos.	(7), (10), (12)
Os gerentes de responsáveis pela manutenção do software não são experientes ou eficientes.	(5); (12) (16)
Fraca interação (comunicação) do gerente a todos com os envolvidos do projeto.	(5), (12)
Falhas ao gerenciar as expectativas do usuário final	(4), (6), (16), (22)
Ausência de um líder.	(5), (10)
Falta de metodologia efetiva para manutenção de software	(5), (6), (15), (16)
Dificuldade em gerenciar mudanças de prioridades. Priorizar soluções emergenciais para solucionar os problemas em detrimento de alternativas mais robustas.	(20), (23)
Estratégia organizacional adotada. 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 estratégia organizacional e não por uma análise objetiva dos problemas.	(19), (23)
<b>Métodos Gerenciais</b>	
Fraco monitoramento do progresso das atividades relacionadas à manutenção de software. Ex.: acompanhamento do relatório de status e manutenção de métricas progressivas.	(5), (7), (12)
Treinamento da equipe mantenedora inadequado ou indisponível.	(2), (5), (8), (12), (14), (23)
Falta de procedimentos, programas adequados e recursos para assegurar a garantia da qualidade ao manter software.	(10), (12)
Gerência de Configuração inadequada.	(2), (3), (5), (12)
Mudanças contínuas no objetivo e escopo ao manter software.	(4), (5), (16)
Erro ao estimar (tempo, custo e esforço) necessário para a manutenção do software.	(2), (3), (7), (8), (10), (14), (16), (25).
Métricas inadequadas / inexatas.	(2), (3)
Dificuldade em mensurar desempenho necessário para a manutenção do software.	(23)
Dificuldade em medir esforço.	(23)
Grande Backlog (grande número de acúmulo de trabalho a serem executados pelos mantenedores).	(22), (23)
Pobre análise de impacto do software a ser mantido e da organização com relação às habilidades críticas, documentação e processos da área.	(25)
Falta de padrão de programação durante o ciclo de vida do software.	(27)
<b>Ambiente de Trabalho</b>	
Falta espírito de equipe. Os conflitos requerem intervenção da gerência responsável pela manutenção do software.	(5), (12), (13), (17)
Pobre comunicação ao manter software. Devido à falta de conhecimento da missão, metas do projeto e informação técnicas entre pares e gerentes.	(5), (9), (12)
Baixa moral / falta de compromisso devido ao baixo nível de entusiasmo afetando assim a produtividade e criatividade. As pessoas não sentem reconhecidas e recompensadas pelos superiores.	(3), (5), (7), (12), (16), (17), (20), (22), (23), (25)
Falta de maturidade / instabilidade organizacional	(3), (5)
Baixa produtividade da equipe responsável pela manutenção do software	(3), (5), (10), (17), (20), (22)
A Equipe de desenvolvimento não tem incentivo para escrever o software de maneira que seja fácil de manter.	(27)

Finalmente, a classe Restrições de Programa consiste nos fatores externos da manutenção de software (fatores que estão fora do controle direto da equipe de manutenção), que podem influenciar o sucesso desta manutenção. Ela é dividida em dois elementos:

## IV Simpósio Brasileiro de Qualidade de Software

- **Recurso:** agrupa os fatores de riscos relacionados a restrições externas referente a cronograma, pessoas, custo e instalações. Este elemento é composto por cinco fatores de risco.
- **Interfaces do programa:** relaciona os fatores de riscos referentes a interfaces externas para o cliente, outros contratados, vendedores. Além de relacionar também os fatores de risco relacionados aos termos e condições dos contratos. Este elemento é composto por oito fatores de riscos. Os fatores de riscos da classe Restrições de Programa são listados na Tabela 7.

**Tabela 7 – Taxonomia de Riscos para a Classe: Restrições de Programa**

Fatores de Riscos	Referência
<b>Recurso</b>	
O cronograma irreal ou inadequado baseado nos eventos internos e externos do software a ser mantido.	(1), (2), (5), (6), (9), (11), (12), (13), (14), (15), (17), (21), (22)
Pressão excessiva de prazo ao manter software.	(3), (5), (22)
Problemas de pessoal. Ex.: falta experiência, pouco conhecimento, falta habilidade, falta de pessoal e indisponibilidade de pessoas chaves.	(1), (3), (4), (5), (6), (7), (8), (9), (12), (13), (14), (15), (16), (19), (21), (22), (23), (24), (25), (27)
O orçamento é insuficiente ou instável baseado nos eventos internos e externos do software a ser mantido.	(1), (2), (5), (6), (9), (12), (15), (17), (19), (25)
Instabilidade (rotatividade) e falta de continuidade das pessoas com habilidades em manutenção de software.	(3), (5), (16), (17), (19), (20), (22), (23), (27)
<b>Interfaces do Programa</b>	
Qualquer problema com o cliente, tais como: demora na aprovação de documentos, comunicação pobre, resistência à mudança, falta de comprometimento, falta de cooperação, conflitos entre clientes e departamentos.	(3), (5), (8), (12), (16), (17)
Problemas relacionados aos contratos associados.	(2), (10), (12)
Qualquer problema associado à sub-contratação.	(2), (6), (12), (15)
Fatores políticos (companhia, clientes, contratantes associados e sub-contratantes) que causam problemas para a manutenção do software.	(5), (12), (13), (17)
Qualquer problema com o usuário, tais como: falta de envolvimento/comprometimento, conflito entre usuários e departamentos, falta de entendimento com relação ao funcionamento do sistema, resistência a mudanças e falha em obter comprometimento do usuário.	(2), (4), (6), (14), (15), (16), (19), (20), (21), (22)
Falta de comprometimento da gerência sênior (alta gerência).	(4), (3), (6), (12), (15), (16), (22), (23)
Alta rotatividade dentro da organização usuária.	(22)
Treinamento inadequado dos usuários	(22)

## 6. Conclusões

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

Este trabalho propõe uma taxonomia de riscos para manutenção de software baseada na integração de fatores de riscos já levantados para o desenvolvimento de software, e em problemas conhecidos que podem acontecer em projetos de manutenção.

## IV Simpósio Brasileiro de Qualidade de Software

A lista de riscos de manutenção obtida foi classificada de acordo com a proposta de taxonomia de riscos do SEI [Carr et al 1993]. Este trabalho dá sustentação à premissa defendida por Charette et al. (1997) que alguns fatores de riscos de desenvolvimento e manutenção podem ser semelhantes.

Com relação a trabalhos futuros esta taxonomia de riscos para manutenção de software poderá ser utilizada como referência no processo de análise dos riscos, assim como poderá ser utilizada no processo de criação de planos de respostas aos riscos.

### **Agradecimentos**

O presente trabalho foi realizado com o apoio do CNPq, uma entidade do Governo Brasileiro voltada ao desenvolvimento científico e tecnológico.

### **Referências Bibliográficas**

- 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.
- Adler, T. R.; Leonard, J. G.; Nordgren, R. K. Improving Risk Management: Moving from Risk Elimination to Risk Avoidance. Technovation, Elsevier Science, 1998.
- Barki Henri; Rivard Suzanne; Talbot, Jean. Toward an Assessment of Software Development Risk. Journal of Management Information Systems, v. 10, n. 2, 1993, p. 203-225.
- Bennett, K. H.; Rajlich, V. T. Software Maintenance and Evolution: a Roadmap. The Future of Software Engineering, ICSE 2000, ACM, p. 75-87, 2000.
- Boehm, Barry. Risk management. Piscataway: IEEE Computer Society Press, 1989.
- Boehm, B. W. Software Risk Management: Principles and Practices. IEEE Software, 8(1), p.32-41, Janeiro, 1991.
- Brown, N. Industrial-strength management strategies. IEEE Software, 13(4), 1996, p.94-103.
- Carr, M. J.; Konda, S. L., Monarch, I.; Ulrich, F. C.; Wakker, C. F. Taxonomy-Based Risk Identification. Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1993. Technical report CMU/SEI-93-TR-6.
- CCTA – Central Communications and Telecommunications Agency – UK Civil Service. Risk Handbook. 2000
- Charette, R. N.; Adams, K. M.; White, M. B. Managing Risk in Software Maintenance. IEEE Software, 14(3), p.43-50, 1997.
- Chrissis, M. B.; Konrad, M.; Shrum, S. CMMI<sup>®</sup> – Guidelines for Process Integration and Product Improvement. Boston: Addison-Wesley, 2002
- DeMarco, T.; Lister, T. Waltzing with Bears: Managing Risk on Software Projects. New York, NY: Dorset House Publishing, Co., Inc., 2003.
- Dorofee, A. J.; Walker, J. A.; Alberts, C. J.; Higuera, R. P.; Murphy, R. L.; Williams, R. C. Continuous Risk Management Guidebook. Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1996

## IV Simpósio Brasileiro de Qualidade de Software

- Farias, L. L. Planejamento de Riscos em Ambientes de Desenvolvimento de Software Orientados à Organização. 2002. Dissertação de Mestrado (Engenharia de Sistemas e Computação) – Programa de Pós-graduação em Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro.
- Fontoura, L. M.; Price, R. T. Usando GQM para gerenciar Riscos em Projetos de Software. 18º Simpósio Brasileiro de Engenharia de Software, p.39-54, 2004.
- Grubb, P.; Takang, A. A. Software Maintenance: Concepts and Practice. Second Edition. Danvers, MA: World Scientific, 2003.
- Higuera, R. P.; Haimes, Y. Y. Software Risk Management (CMU/SEI-96-TR-012). Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1996.
- Hounston, A. X; Mackulak, G. T.; Collofello, J. S. Stochastic Simulation of Risk Factor Potential Effects for Software Development Risk Management. Elsevier Science, p. 247-257, 2001.
- Jalote, Pankaj. CMM in Practice: Processes for Executing Software Projects at Infosys. Boston: Addison-Wesley, 2000. cap.8, p.159-174.
- Jalote, Pankaj. Software Project Management in Practice. Boston: Addison-Wesley, 2002. cap.6, p.93-108.
- Keil, M.; Cule, P. E.; Lyytinen, K.; Schmidt, R. C. A Framework for Identifying Software Project Risks. Communications of the ACM, p.76-83, 1998.
- Kwak, Y. H; Stoddard, J. Project Risk Management: Lessons Learned from Software Development Environment. Technovation, 24(11), Elsevier Science, 2004.
- Leopoldino, C.B. Avaliação de Riscos em Desenvolvimento de Software. 2004. Dissertação (Mestrado em Administração) – Programa de Pós-graduação em Administração, Universidade Federal do Rio Grande do Sul.
- Lientz, B. P.; Swanson E. B. Problems in Application Software Maintenance. ACM, p. 763-769, 1981
- 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.
- McManus, J. Risk Management in Software Development Projects. Burlington, MA: Elsevier Butterworth-Heinemann, 2004.
- Microsoft. Microsoft Solutions Framework - MSF Risk Management. 2002. Disponível em:< <http://www.microsoft.com/msf/>>. Acesso em: 20/05/2004.
- Mizuno, O.; Kikuno, T., Takagi, Y., Sakamoto K., Characterization of Risky Projects based on Project Manager's Evaluation. IEEE Comp. Soc. Press, P.387-395, 2000.
- Pigoski, T. M. Pratical Software Maintenance: Best Practices for Software Investment. P.295. Wiley, 1996
- Pigoski, T. M. SWEBOK - Guide to the Software Engineering Body of Knowledge: Software Maintenance. IEEE, cap.6, p.1-16, 2001.

## IV Simpósio Brasileiro de Qualidade de Software

- Pfleeger L. S. Software Engineering Theory and Practice. Second Edition. River, New Jersey: Prentice Hall, 2001.
- 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.
- Pressman, R. S. Engenharia de Software. 5ª edição. Rio de Janeiro: McGraw-Hill, 2002. cap.6, p.139-156.
- Ropponen, J.; Lyytinen, K. Components of Software Development Risk: How to Address Them? A Project Manager Survey. IEEE Transactions on Software Engineering. v. 26, n. 2, p.98-112, 2000.
- Schneberger, S. L. Software Maintenance In Distributed Computer Environments: System Complexity Versus Component Simplicity. International Conference on Software Engineering, IEEE, IEEE Comp. Soc. Press, p. 304-317, 1995.
- Schneidewind, N. F. Requirements Risk and Maintainability. Idea group Inc., cap.7, p.182-200, 2003
- Schwalbe, Kath. Information Technology Project Management. Second Edition. Course Technology, Thomson Learning, Inc., 2002. cap.10, p.302-335.
- Soeiro, L. Migres – Método Integrado de Gerência em Engenharia de Software. 1999. Dissertação (Mestrado em Ciências da Computação) – Programa de Pós Graduação Ciência da Computação, UNB, Brasília
- Sommerville, I. Engenharia de Software. 6ª Edição. São Paulo: Addison Wesley, 2003.
- Valeriano, D. L. Gerenciamento Estratégico e Administração por Projetos. São Paulo: Makron Books, 2001.
- Webster, K. Oliveira, K., Anquetil, Figueriredo, R. Priorização de Riscos para manutenção de software, IV Jornadas Iberoamericanas em Engenharia Del Conocimiento, 2004, Madrid, pp. 647-660. Novembro 2004.
- Willians, R. C., Walker ,J. A., Dorofee, A.J. Putting Risk Management into Practice. IEEE Software, vol.14, n.3, pp.75-81, May/June 1997.