

O início de um estudo sistemático sobre ferramentas de gerenciamento de riscos para Projetos de Software

Márcia Ribeiro dos Santos¹, Luanna Lopes Lobato^{1,2}

¹Departamento de Ciência da Computação – Universidade Federal de Goiás (UFG)
Campus Catalão – GO – Brasil

²Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil

marcinharsantos@gmail.com, lll@cin.ufpe.br

***Abstract.** This paper describes a systematic review on risk management tools for software projects. Thus, we present definitions about risk management in software project development and some related works. In order to identify the research available in the literature and verify the quality of them, a systematic review is proposed. Analyzing two development methodologies: the Single System and the Software Product Line, the goal is to investigate what have been developed about risk management tools in both methodologies and to identify relevant features to development a new risk management tool, based in the characteristics identified.*

***Resumo.** Este artigo descreve uma revisão sistemática sobre ferramentas de gerenciamento de riscos para projetos de software. Para isso, são apresentados definições de gerenciamento de riscos no desenvolvimento de projetos de softwares e alguns trabalhos relacionados. Para identificar os estudos disponíveis na literatura e verificar a qualidade das mesmas, uma revisão sistemática é proposta. Analisando duas metodologias de desenvolvimento: o de Sistema Individual e da Linha de Produto de Software, o objetivo é investigar o que tem sido desenvolvido sobre ferramentas de gestão de risco em ambas metodologias e identificar características relevantes para o desenvolvimento de uma nova ferramenta de gerenciamento de risco, com base nas características identificadas.*

1. Introdução

Com a crescente demanda de mercado em se produzir mais em menor tempo e o aumento das exigências de qualidade por parte dos usuários dos sistemas, vê-se a necessidade em utilizar meios onde o conhecimento e o trabalho possam ser reutilizados. Tendo-se complexidade nos sistemas, aumenta-se a possibilidade de riscos surgirem, decorrentes do desenvolvimento de software, e interferirem o andamento do projeto, tornando cada vez mais importante a pesquisa em cima de gerenciamento de riscos. Ferramentas de gerenciamento de riscos são criadas de modo a solucionar problemas encontrados em gerenciamento de riscos dos projetos de software. Estas são criadas no contexto de projetos de desenvolvimento de software único. No entanto, para Linha de Produto essas não são tão abordadas.

O foco deste trabalho é desenvolver uma Revisão Sistemática (*Systematic Review* – SR) sobre ferramentas de gerenciamento de riscos para projetos de desenvolvimento de software. Tendo sido identificadas as características de tais ferramentas, as limitações e sugestões de melhoria, o objetivo **final** do trabalho é propor uma nova ferramenta voltada para linha de produto de software apresentando as particularidades de uma linha. Assim, torna-se necessário considerar diferentes metodologias para desenvolvimento de software, Linha de Produto de Software (*Software Product Line* - SPL) e Desenvolvimento de Sistemas Individuais (*Single System Development* - SSD) de modo a orientar na construção de ferramentas que atendam as necessidades impostas por uma SPL e ainda contemple as principais necessidades do mercado.

Este artigo está organizado da seguinte forma, na seção 2 são apresentadas as duas metodologias para desenvolvimento de software. Na seção 3, são descritos os conceitos sobre gerenciamento de riscos em projetos de software, na subseção 3.1, são sucintamente, retratados alguns trabalhos realizados com ferramentas de gerenciamento de riscos. Na seção 4, o estudo sobre revisão sistemática é apresentado. Na seção 5, são discutidos os resultados obtidos com a revisão sistemática sobre ferramentas de gerenciamento de riscos e, por fim, na seção 6, são relatados, a conclusão e os trabalhos futuros.

2. Metodologia de desenvolvimento de software

A metodologia de desenvolvimento de software tem o intuito de detalhar os passos do processo a serem seguidos, utilizando em padrão como objetivo pré-estabelecido, o qual tem garantir a qualidade nos produtos gerados, bem como cumprimento de prazos, suprimindo as necessidades impostas pelo cliente. A seguir, são apresentadas duas metodologias importantes para o estudo de desenvolvimento de softwares, as quais são a base para o desenvolvimento desta pesquisa.

2.1 Desenvolvimento de Sistemas Individuais

O SSD é o método tradicional de desenvolvimento de projetos de software, os quais são baseados na utilização de processos para desenvolvimento de sistemas individuais, com foco nas necessidades de um único cliente.

Segundo Pressman (2006), toda organização de engenharia de software deveria descrever um conjunto de atividades de arcabouço para seus processos. Mas a característica principal das metodologias atuais é que elas são divididas em etapas ou fases bem definidas e documentadas após seu término, e independente de modelo a ser seguido, os engenheiros de software têm tradicionalmente escolhido um arcabouço genérico que inclui atividades como: comunicação, planejamento, modelagem, construção e implantação, que são complementadas por atividades auxiliares como acompanhamento e controle do projeto de software, gestão de riscos, revisões técnicas, gestão de reuso e etc.

O embasamento maior em metodologias tradicionais é feito na análise e projeto que mantém tudo em documentos, tornando esta metodologia mais lenta para mudanças, e etc. Para Pressman, a engenharia de software é dividida em camadas:

- Ferramentas: fornecem apoio automatizado ou semi-automatizado para o processo e para os métodos.
- Métodos: fornecem a técnica para construção do software. Abrange tarefas como comunicação, análise de requisitos, modelagem do projeto, e outros;
- Processos: além de alicerce da engenharia de software, formam a base para controle gerencial de projetos de software e estabelecem o contexto no qual os métodos técnicos são aplicados, os produtos de trabalho como modelos, documentos e outros, são produzidos, a qualidade é assegurada e modificações são adequadamente geridas;
- Foco na qualidade: qualquer abordagem de engenharia deve se apoiar no compromisso de qualidade. Gestão de qualidade leva ao desenvolvimento de abordagens cada vez mais efetivas para engenharia de software.

O foco principal das metodologias tradicionais, que são as de desenvolvimento único, é a previsibilidade dos requisitos do sistema, que traz a grande vantagem de tornar os projetos completamente planejados, facilitando a gerência do mesmo, caracterizando o processo como bastante rigoroso, Oliveira (2004).

Como exemplos de modelos de processo de software, temos:

- Modelo em cascata: sugere uma abordagem sistemática e sequencial para o desenvolvimento de softwares, começando pela comunicação com o cliente, planejamento, modelagem, construção e implantação do sistema;
- Modelo incremental: subdividido em modelo incremental, que combina elementos do modelo em cascata de forma iterativa, e modelo RAD, que enfatiza um ciclo de desenvolvimento curto;
- Modelos evolucionários: são iterativos e caracterizados de forma a permitir aos engenheiros de software desenvolver versões cada vez mais completas do software, sendo subdividido em prototipagem, espiral, desenvolvimento concorrente;
- Modelo especializado de processo, que tem características de um ou mais modelos convencionais, além de ser subdividido em modelo baseado em componentes e métodos formais.

2.2 Linha de Produto de Software

“Uma linha de Produto de Software é um conjunto de sistemas que usam software intensivamente, compartilhando um conjunto de características comuns e gerenciadas, que satisfazem as necessidades de um segmento particular de mercado ou missão e que são desenvolvidos a partir de um conjunto comum de ativos principais e de uma forma preestabelecida”, Clements (2001).

Baseado na reutilização dos artefatos, os pontos comuns e variáveis da SPL são definidos, além de serem documentados, e assim são verificados os possíveis produtos a serem gerados a partir da linha. Com isso, possibilita-se o desenvolvimento em larga escala, o que garante o ganho em relação ao tempo na entrega e maior qualidade dos produtos (sistemas).

Conseqüentemente se aumenta a lucratividade, pois reusando partes definidas na arquitetura da linha para vários produtos, o desenvolvimento de produtos é facilitado, à medida que aumenta o número de produtos instanciados da linha, Pohl(2005).

Para garantir que a partir da SPL se desenvolvam produtos de qualidade e que atendam os compromissos de escopo, prazo e custo é necessário evitar os riscos que podem acontecer durante todo o desenvolvimento do projeto. “Maximizar a probabilidade e as conseqüências de eventos positivos e minimizar a probabilidade e conseqüências que eventos adversos possam trazer aos objetivos do projeto” (PMI, 2004).

Na figura 1 é apresentado o framework para a engenharia de linha de produto de software, onde são destacadas as etapas de engenharia de domínio e engenharia de aplicação, proposto por Weiss e Lai.

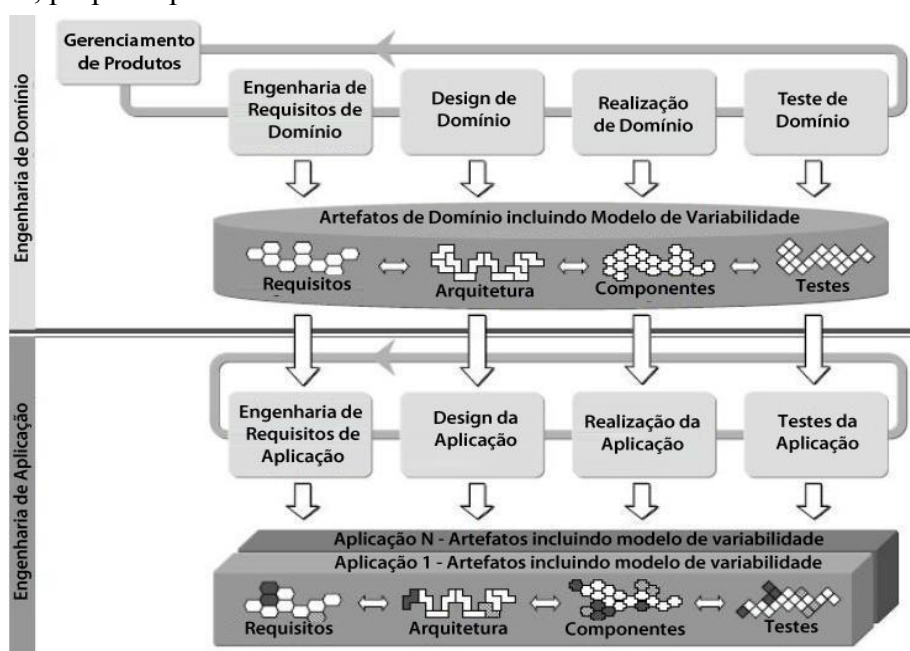


Figure 1. “Framework da Engenharia de linha de produto de software, Pohl, K. e Bockle, G. e Linden, F. V. (2005)”.

Na etapa de engenharia de domínio, são definidos os artefatos comuns e variáveis que estarão presentes na linha de produto, os quais constituirão os produtos que serão gerados. A engenharia de aplicação reusa os artefatos gerados na engenharia de domínio produzindo artefatos da aplicação. A partir da engenharia da aplicação é possível instanciar esses artefatos gerando os produtos que estão presentes ao domínio da linha de produto.

Segundo Pohl, K. and Bockle, G. and Linder, F. (2005), entre as razões da utilização de SPL, podem se destacar:

- Redução de custos de desenvolvimento;
- Melhoria da qualidade;
- Redução do tempo para entrega do produto;

- Redução do esforço de manutenção;
- Melhorar a estimativa de custo;
- Mais fácil lidar com a evolução da linha de produto;
- Maior satisfação do cliente;

Visto as características apresentadas pelas metodologias de desenvolvimento de software e baseado na necessidade em se ter gerenciamento de riscos durante o desenvolvimento de projetos, seja em SSD ou em SPL, este trabalho tem por objetivo verificar, de uma maneira sistemática, qual o cenário atual das ferramentas de gerenciamento de riscos para desenvolvimentos de projetos de software.

3. Ferramentas de gerenciamento de riscos

O risco em um projeto de software, tanto na metodologia SSD quanto na SPL, é um evento negativo que afeta o processo de desenvolvimento, bem como os produtos que são gerados. Para isso, existem os softwares de gerenciamento de riscos que consistem em avaliar e controlar estes riscos, para se ter uma medida da probabilidade de ocorrência e das perdas que podem ser causadas por estes. A melhor maneira de descobrir os riscos é definir inicialmente, os objetivos e metas do projeto, Aguiar, M.

Apesar de importante, muitas organizações ainda não são adeptas a um processo de gerência de riscos em seus projetos de software. Isso porque ainda são grandes as dificuldades para compreensão e implantação da gerência de riscos, além da falta de ferramentas ou dificuldade de acesso a elas, devido a custo e outros fatores.

Como relatado no Project Management Body of Knowledge – PMBOK(2000), o gerenciamento de riscos é o processo sistemático de identificação, análise e resposta aos riscos dos projetos. São destacados os seguintes processos que compõem o gerenciamento de riscos: planejamento do gerenciamento dos riscos, identificação de riscos, análise qualitativa e quantitativa de riscos, planejamento de resposta a riscos, monitoração e controle de riscos.

Para consolidar estes processos dentro de um projeto de software, se faz necessário o uso de uma ferramenta para este fim. Assim, este trabalho é justificado pela necessidade em investigar o que tem sido desenvolvido, com o uso da SR, de modo a identificar características relevantes para se propor uma solução de sucesso, sendo essa a proposta de ferramenta.

3.1. Trabalhos relacionados

Trabalhos focados na gerência de riscos é um tema que deve ser considerado nos projetos de software. Como poucas pesquisas a respeito foram encontradas, se faz necessário a construção de tal revisão sistemática do tema, e como o foco da SR é ferramentas de gestão de risco, abaixo mostramos algumas ferramentas que já foram desenvolvidas para este fim, das quais três destas são apresentadas.

A ferramenta *RiskFree* é uma ferramenta de gerenciamento de riscos baseada no PMBOK (*A Guide to the project Management Body of Knowledge*) e aderente ao CMMI (*Capability Maturity Model Integration*). Esse baseamento se confirma na definição do seu processo de gerência de riscos que traz as etapas de planejamento da gerência,

identificação e análise dos riscos, planejamento de resposta destes riscos, além da monitoração e controle destes, Knob (2006).

Outro exemplo de ferramenta para gerenciamento de riscos é o software TRIMS, o qual contém o questionário proposto pelo SEI (Software Engineering Institute) para avaliação de riscos de software, para utilização em projetos de software e que foi desenvolvido pelo *BMP (Best Manufacturing Practices) Center of Excellence*, uma organização patrocinada pela Marinha e pelo departamento do Comércio Norte-americano, e a Universidade de Maryland, Aguiar M.

Outra ferramenta é a *FMEA (Failure Model and Effect Analysis)* que busca, em princípio, evitar, por meio da análise de falhas potenciais e propostas de ações de melhoria, que ocorram falhas no projeto do produto ou do processo. Tal ferramenta pode ser usada para diminuir a probabilidade da ocorrência de falhas potenciais (que ainda não ocorreram), diminuir a probabilidade de falha em projetos novos, aumentar confiabilidade de produtos ou processos já em operação, diminuir riscos e erros e aumentar qualidade em procedimentos administrativos, Toledo e Amaral.

4. Revisão Sistemática

O objetivo deste trabalho é apresentar uma Revisão Sistemática em ferramentas de gerenciamento de riscos para desenvolvimento de projetos de software. A SR foi proposta como meio sistemático para identificar os trabalhos que têm sido desenvolvidos nesse contexto.

Uma SR é um meio de identificar, avaliar e interpretar toda pesquisa disponível e relevante sobre uma questão de pesquisa, um tópico ou um fenômeno de interesse. A revisão tradicional utiliza métodos informais e subjetivos de coleta e interpretação dos estudos, não descrevendo sistematicamente a pesquisa, seleção e avaliação destes, já a revisão sistemática é calcada pela definição de um protocolo, no qual são descritos alguns pontos chaves que são relevantes para a correta execução de uma SR, Kitchenham (2004) . Assim sendo, uma estratégia de busca é definida, tudo é documentado, critérios de inclusão e exclusão são estipulados, além da especificação da informação a ser obtida de cada estudo, dentre outros.

O processo de revisão sistemática é dividido nas fases de planejamento, condução e documentação da revisão. E dentre as razões para realização de uma SR podemos citar:

- Resumir evidências existentes em relação a um tratamento ou tecnologia;
- Identificar lacunas na pesquisa atual, a fim de sugerir áreas de investigação mais aprofundada;
- Fornecer um *framework/background* adequado as novas atividades de investigação.

Segundo Júnior (2007), no planejamento, é identificado a necessidade de uma revisão sistemática e o protocolo a ser seguido. A estrutura de um protocolo é dividida em: motivação, questionamento sobre a pesquisa (parte que questiona os efeitos da tecnologia, seus custos e riscos, etc.), métodos de busca (termos, fontes, etc.) e os critérios de seleção adotados.

Já a condução se trata da identificação das fontes de busca. Todas as buscas devem ser documentadas, inclusive os resultados não aprovados. A seleção dos estudos deve passar sobre os critérios de inclusão e exclusão, e a extração destes deve ser projetada em formulários, a documentação da revisão sistemática é feita pelo fato de ser importante transmitir os resultados obtidos por ela. Além de ser necessário reportar o trabalho em ao menos duas formas como, relatório técnico ou artigo de periódico.

Se uma pesquisa não é completa e justa, ela terá pouco valor científico. O que justifica a principal razão de se realizar uma SR, que requer maior esforço comparado a uma pesquisa comum, pois deve fornecer informações sobre efeitos de algum fenômeno em uma ampla gama de configurações e métodos empíricos, além de ser possível combinar dados usando técnicas de meta-análise, o que aumenta a probabilidade de detectar efeitos reais que os estudos menores são incapazes de detectar, Kitchenham (2004).

5. Resultados

Como apresentado na seção 4, a revisão sistemática é dividida em planejamento, condução e documentação. A necessidade de uma revisão sistemática em ferramentas de gerenciamento de riscos se deu por conta da importância desse tópico e a pouca informação e software disponíveis para projetos de software que usam a metodologia de desenvolvimento baseada em SPL. No protocolo adotado foram definidas questões de pesquisa, como por exemplo, “Quais as principais funcionalidades apresentadas pelas ferramentas?”, “Em que cenário as ferramentas foram desenvolvidas e utilizadas (acadêmico, industrial, ambos)?” e vem sendo desenvolvido incrementalmente de acordo com a evolução da revisão sistemática.

Para seleção dos estudos foram definidas *strings* de busca tanto em português como inglês, já que as buscas foram feitas em materiais nacionais e internacionais, como exemplo:

- “Tool AND Risk AND (SPL OR Software Product Line OR Product Family)”;
- “Ferramentas E Riscos E (LPS OU Linha de Produto de Software OU Família de Produtos)”.

Para a condução da pesquisa, as fontes, datas, títulos e autores dos estudos selecionados são registrados. Primeiramente, os estudos são selecionados através das *strings*, depois a exclusão dos mesmos é feito pela leitura de título, ou seja, se o título não tem algo relacionado com o tema de pesquisa o estudo é então descartado. Em seguida, é feita mais uma filtragem através da leitura dos *abstracts*.

Assim, os estudos que tiverem seu resumo inicial indicando que se tratam do assunto em questão da revisão, serão selecionados para a próxima fase que é a de leitura completa do artigo. Durante a leitura dos estudos, as questões definidas são respondidas e outras informações são coletadas, as quais são necessárias para o levantamento de características que uma ferramenta de gerenciamento de risco em SPL deve apresentar.

6. Conclusão e trabalhos futuros

O gerenciamento de riscos tem se tornado cada vez mais importante e decisivo para o sucesso de projetos de software, Knob (2006). Muitos riscos possíveis em SSD são

preocupação também para SPL, assim sendo, para o desenvolvimento de uma ferramenta de gerenciamento de riscos em SPL é necessário um estudo também destas ferramentas em desenvolvimento tradicional. Este estudo ainda é justificado visto que são poucas as fontes de estudo sobre gerenciamento de riscos em SPL, comparado ao tradicional.

O intuito da revisão sistemática em ferramentas de gerenciamento de riscos é prover informação para servir de incentivo ao ramo de desenvolvimento de novas ferramentas. Assim, é possível retratar o cenário da área, enfatizando como as ferramentas funcionam e respondem a um risco e assim, dispor a configuração de uma ferramenta que atenda as necessidades de qualquer projeto de software em se tratando de gerência de riscos para SPL.

Como o foco do trabalho é a revisão sistemática, e a partir de um estudo sobre esta, propor uma ferramenta capaz de atender aos projetos de software de uma SPL, o trabalho a ser realizado futuramente é a conclusão da revisão sistemática, juntamente com seu protocolo, que mostra informações como motivação, métodos de busca, critérios de seleção e outros, tendo como resultado o cenário atual das ferramentas de gerenciamento de riscos e baseado nisto, o estudo será feito e a ferramenta proposta.

Referências

- Aguiar, M. Gerenciando riscos nos projetos de software.
- Junior, E. A. O. (2007) “O processo de Revisão Sistemática”.
- Project management Institute – “A Guide to the Project Management Body of Knowledge” (PMBOK Guide) – 2000 Edition – PMI, 2000.
- Knob, F. e Silveira, F. e Orth, A. I. e Prikladnicki, R. (2006) – “*RiskFree* Uma Ferramenta de Gerenciamento de Riscos Baseada no PMBOK e Aderente ao CMMI”.
- Toledo, J.C. e Amaral, D.C. - FMEA – “Análise do Tipo e Efeito de Falha” - GEPEC – Grupo de Estudos e Pesquisa em Qualidade, DEP – UFSCar.
- Oliveira, S. R. B. e Rocha, T. A. e Vasconcelos, A. M. L. (2004) “Adequação de Processos para Fábricas de Software”, Anais do Simpósio Internacional de Melhoria de Processos de Software – SIMPROS 2004.
- Pressman, R.S. (2006), Engenharia de Software, 6ª Edição.
- Kitchenham, B. (2004) – “Procedures for Performing Systematic Reviews”.
- Pohl, K. e Bockle, G. e Linden, F. V. (2005) – “Software Product Line Engineering – Foundations, Principles, and Techniques”. Springer - Verlag, New York, USA.
- Clements, P e Northrop, L. (2001) “Software Product Lines: Practices and Patterns”, Addison-Wesley, Reading, Massachusetts.