

Gerenciamento de Projetos Baseado em Cenários: uma Abordagem de Modelagem Dinâmica e Simulação

AUTOR: **MÁRCIO DE OLIVEIRA BARROS**

ORIENTADORES: **CLÁUDIA MARIA LIMA WERNER**
GUILHERME HORTA TRAVASSOS

PROGRAMA: PROGRAMA DE ENGENHARIA DE SISTEMAS E COMPUTAÇÃO
COPPE – UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

DATA E TIPO DE DEFESA: 21/12/2001 (DOUTORADO)

COPPE / UFRJ – Departamento de Engenharia de Sistemas e Computação
Caixa Postal: 68511 - CEP 21945-970 - Rio de Janeiro – RJ
Telefone: 5521 2562-8675 / Fax: 5521 2562-8676
{marcio, werner, ght}@cos.ufrj.br

Resumo

O gerenciamento de projetos é uma atividade fortemente baseada em conhecimento. Os gerentes utilizam suas habilidades e sua experiência para tomar decisões durante a execução de um processo de desenvolvimento de software. Geralmente, os gerentes experientes obtêm mais sucesso que os gerentes novatos em termos de atingir metas de cronograma, custos e funcionalidade. Estes melhores resultados são atingidos devido à experiência acumulada ao longo de situações ocorridas em projetos no passado e no conhecimento derivado destas experiências. Nesta tese, abordamos a criação e reutilização de conhecimento relacionado com o gerenciamento de projetos de software. Apresentamos uma representação para este tipo de conhecimento, os *modelos de cenários*, que permitem que um gerente verifique o impacto de teorias, ações e estratégias gerenciais que podem ser aplicadas ou impostas sobre um projeto de software. Apresentamos os resultados de estudos experimentais para análise da viabilidade das técnicas propostas e uma aplicação da abordagem proposta em um processo de gerenciamento de riscos.

Abstract

Project management is a knowledge intensive activity. Managers use their skills and experience to make decisions during the execution of a software development process. Usually experienced managers perform better than novice managers in terms of attending to project schedule, budget, and functionality. Such better results are achieved due senior managers' accumulated experiences and the knowledge that they can derive from situations faced in past projects. In this thesis, we address the creation and reuse of software project management knowledge. We present a knowledge representation, namely scenario models, which allows a manager to verify the impact of theories, actions, and strategies that can be applied or imposed upon a software project. We present an application of the proposed approach within a risk management process and the results from experimental analysis of the techniques.

1 Introdução

O estado da prática e a literatura contemporânea comprovam a persistência de um conhecido problema no desenvolvimento de software: muitos projetos consomem mais recursos do que o planejado, demoram mais tempo para serem realizados, possuem menos funções e menor qualidade do que o esperado. Relatos de insucesso na produção de sistemas de software podem ser encontrados em diversos estudos de casos e experimentos documentados ao longo dos últimos anos (Standish Group, 1994; Charette, 1996; Lafleur,

1996; Gross et al., 1999). Este panorama é particularmente perceptível em projetos complexos, de larga escala ou inovadores.

Duas linhas de estudo apresentam opiniões distintas sobre os fatores responsáveis pelos insucessos no desenvolvimento de software na indústria. A primeira linha de estudos associa estas falhas a problemas tecnológicos, agravados pela crescente complexidade dos produtos desenvolvidos (Augustine, 1982). A segunda linha de estudos transfere a responsabilidade para os problemas de gerenciamento, falta ou excesso de comunicação e dificuldades no tratamento das incertezas que se apresentam nos projetos de software (Brown, 1996; Royce, 1998; Reel, 1999).

O gerenciamento de projetos de software é uma atividade fortemente baseada em conhecimento, onde os gerentes utilizam suas habilidades e sua experiência para tomar decisões enquanto a equipe executa o processo de desenvolvimento de software. Observa-se que os gerentes mais experientes geralmente obtêm mais sucesso no controle de projetos de software do que os gerentes inexperientes. Devido às experiências acumuladas ao longo de diversos projetos realizados no passado e ao conhecimento adquirido através destas experiências, os gerentes mais experientes são capazes de perceber determinados aspectos do processo de desenvolvimento que permanecem invisíveis para outros gerentes. O planejamento e controle do projeto passam a considerar estes aspectos, que de outra forma seriam desprezados.

O “modelo de decisão baseado em reconhecimento de padrões”¹, proposto por Klein (1998), sugere que os gerentes mantêm uma coleção de padrões em suas mentes e comparam estes padrões ao contexto em que se encontram quando uma decisão se faz necessária. O conhecimento contido no repositório mental de experiências dos gerentes experientes é um recurso valioso. Sua importância para o sucesso de projetos de software cria a demanda por uma técnica para representação, armazenamento e reutilização destes padrões mentais, que permita que gerentes menos experientes compartilhem o conhecimento dos especialistas.

Os gerentes de projeto mais experientes geralmente realizam este processo de tomada de decisão mentalmente. Eles constroem um modelo mental do projeto sob análise, criam modelos mentais para os problemas e oportunidades que estão sendo investigados e buscam, dentre os padrões de seu repositório mental, ações adequadas para serem tomadas para resolver os problemas e para explorar as oportunidades. Entretanto, como um modelo mental não pode ser diretamente reutilizado por gerentes menos experientes, precisamos de uma representação explícita para o conhecimento tácito dos gerentes experientes, assim como uma técnica que permita a avaliação do impacto destes problemas e oportunidades sobre um projeto de desenvolvimento de software.

O objetivo da tese relatada neste artigo foi definir, aplicar e avaliar uma representação para o conhecimento sobre gerenciamento de projetos de software. Esta representação deve permitir a descrição do conhecimento acumulado por gerentes experientes, de forma que este conhecimento seja reutilizado em projetos controlados por gerentes novatos. Para facilitar esta reutilização, foram definidos mecanismos que suportam a automação do processo de transferência de conhecimento² entre os gerentes experientes e novatos. Estes mecanismos permitem que os gerentes novatos avaliem o impacto da aplicação do conhecimento gerencial reutilizado no contexto de seus projetos, antes que o conhecimento seja utilizado para apoiar decisões em um projeto real. A representação proposta foi utilizada em um processo de análise de riscos (Hall, 1998; ABNT, 2000) e avaliada através de estudos experimentais.

¹ No original, *recognition-primed decision model*.

² Neste contexto, definimos a transferência de conhecimento como o processo que transforma a informação, que é o elemento representado sob algum formato computacional, em conhecimento de fato, que se reflete no entendimento do gerente dos efeitos decorrentes da aplicação desta informação sobre seu projeto.

Este artigo está organizado em seis seções. A primeira seção contém esta introdução. Na segunda seção, descrevemos os mecanismos de representação e reutilização de conhecimento gerencial. Na terceira seção, apresentamos um estudo experimental planejado e realizado para analisar a viabilidade de aplicação das técnicas propostas no gerenciamento de um projeto de software. Na quarta seção, apresentamos o processo de análise de riscos que utiliza a representação proposta. Na quinta seção, apresentamos um estudo de caso da aplicação do processo de análise de riscos proposto. Finalmente, na sexta seção, apresentamos as principais conclusões deste trabalho.

2 Representação de Conhecimento Gerencial como Modelos de Cenário

O gerenciamento de projetos baseado em cenários (Barros et al., 2002a) é um paradigma para a gerência de projetos de software que sugere que um gerente especifique o comportamento esperado para um projeto de forma independente dos problemas e oportunidades que podem afetar este projeto. Entretanto, como estes eventos incertos podem ocorrer durante a execução do projeto, alterando o seu comportamento esperado, o gerente deve testar a sensibilidade do projeto à ocorrência de combinações destes problemas e oportunidades. O gerenciamento baseado em cenários envolve o conjunto de técnicas de representação de conhecimento gerencial e análise de seu impacto sobre o comportamento de um projeto de software.

A Figura 1 apresenta os principais elementos envolvidos no gerenciamento de projetos baseado em cenários, focalizando seus principais artefatos: o *modelo de projeto* e os *modelos de cenário*. O modelo de projeto, representado pelos blocos assinalados com o símbolo “PM”, define o comportamento esperado para um projeto de software. Os modelos de cenário representam problemas e oportunidades que podem afetar o comportamento do projeto, contendo o conhecimento tácito dos gerentes de projeto experientes.

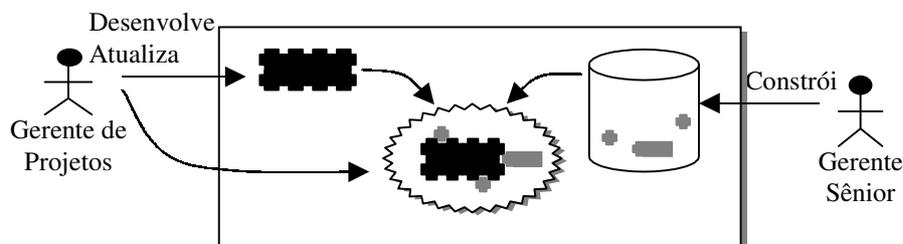


Figura 1 - Principais elementos envolvidos no gerenciamento de projetos baseado em cenários

O modelo de projeto se baseia em quatro elementos: as atividades, os desenvolvedores, os artefatos e os recursos. As atividades representam pacotes de trabalho que devem ser realizadas para a conclusão do projeto. Os desenvolvedores representam as pessoas que formam a equipe do projeto. Eles realizam as atividades utilizando recursos. Os recursos representam elementos lógicos ou físicos que são utilizados ao longo do processo de desenvolvimento, tais como computadores, pacotes de software, componentes reutilizáveis, entre outros. Finalmente, as atividades criam e evoluem artefatos de software, eventualmente consumindo outros artefatos como insumos.

Os modelos de cenário representam os eventos incertos, ações, estratégias e procedimentos gerenciais, que não podem ser considerados como parte de um projeto, mas práticas impostas ou aplicadas sobre ele. Como diversas destas práticas podem ser válidas em projetos distintos, os modelos de cenário contêm conhecimento reutilizável sobre gerenciamento de projetos. Os modelos de cenário são desenvolvidos por gerentes de projeto

sênior e armazenados em um repositório, conhecido como *repositório de cenários*, de onde podem ser recuperados e reutilizados por gerentes novatos.

Durante o planejamento de um projeto de software, o gerente inicialmente desenvolve um modelo de projeto com as características do processo que será utilizado no desenvolvimento do projeto. Em seguida, o gerente recupera os cenários relevantes no contexto do projeto, consultando o repositório de cenários. Finalmente, o gerente executa uma análise iterativa, onde combinações dos cenários recuperados são integrados ao modelo de projeto. O processo de integração ocorre em um ambiente de simulação, que permite avaliar o impacto provocado por cada combinação de cenários no comportamento do projeto. Estas análises permitem que o gerente teste a sensibilidade de seu projeto aos problemas incertos e oportunidades que podem afetar o projeto. A medida que o processo de desenvolvimento se desenrola, o modelo de projeto é periodicamente atualizado com informações sobre o estado do projeto e novas análises de cenário são realizadas, da mesma forma utilizada durante o planejamento.

Os modelos de projeto e cenário são modelos formais, representados segundo a Dinâmica de Sistemas. A Dinâmica de Sistemas é uma linguagem de modelagem para a descrição de sistemas complexos que focaliza os aspectos estruturais dos sistemas (Forrester, 1961). Esta técnica identifica e modela relações de causa-efeito e ciclos de realimentação (*feedback loops*) utilizando diagramas de fluxos. Estes diagramas são compostos por quatro elementos: repositórios, taxas, fluxos e processos. Os repositórios descrevem elementos que podem ser acumulados ou consumidos ao longo do tempo. Um repositório é descrito por seu nível, que indica o número de elementos no repositório em um instante de tempo. As taxas descrevem variações de repositórios, formulando seu crescimento e redução através de uma equação. Os processos e os fluxos atuam como variáveis intermediárias em um modelo, servindo como blocos básicos para a construção de equações complexas para taxas.

A Dinâmica de Sistemas foi previamente utilizada na descrição de projetos de software. Abdel-Hamid e Madnick (1991) desenvolveram o primeiro modelo de projeto segundo esta linguagem. Este modelo foi construído para facilitar o entendimento dos fatores envolvidos nos projetos de software e permitir a análise dos efeitos provocados por diferentes políticas de gerenciamento sobre seu custo e esforço necessário para a conclusão dos projetos. O modelo foi dividido em quatro seções: gerenciamento de pessoal, produção de software, controle e planejamento. A seção de gerenciamento de pessoal inclui as políticas de contratação, treinamento e transferência de recursos humanos do projeto. A seção de produção de software descreve as estratégias de alocação de pessoal para as tarefas de desenvolvimento, controle de qualidade e testes. A seção de controle compara o plano original do projeto, registrado na seção de planejamento, com seu estado ao longo do tempo, ajustando o plano de acordo com as demandas do projeto, tais como a determinação de um grande número de erros durante as atividades de teste, o surgimento de novos requisitos, modificações na composição da equipe, entre outros. Posteriormente, outros modelos dinâmicos (Tvedt, 1996; Lin et al., 1997) ampliaram o conjunto de aspectos abordados pelo modelo de Abdel-Hamid e Madnick. Em (Barros et al., 2000a) apresentamos exemplos de como estes modelos podem ser utilizados na avaliação de qualidade de projetos de software.

O modelo de projeto utilizado no gerenciamento de projetos baseado em cenários se diferencia dos modelos anteriormente desenvolvidos por sua capacidade de integração com os modelos de cenários. Esta característica permite que o conhecimento recorrente e reutilizável dos gerentes experientes (representado nos modelos de cenários) seja avaliado no contexto de um projeto específico (representado por um modelo de projeto). Entretanto, para que esta capacidade de integração fosse possível, foram necessárias algumas alterações no processo de

construção de modelos segundo a Dinâmica de Sistemas, como o aumento do nível de abstração e a separação de fatos e incertezas nos modelos (Barros et al., 2002a).

O aumento do nível de abstração dos modelos visa facilitar a construção e verificação dos modelos de projetos. Tradicionalmente, os modelos desenvolvidos segundo a Dinâmica de Sistemas são descritos por diversas equações matemáticas (Albin, 1997), o que dificulta seu entendimento e adaptação. Os elementos componentes do sistema no mundo real não são facilmente identificados no labirinto de equações formado pelos construtores da Dinâmica de Sistemas. A representação destes elementos está geralmente dispersa ao longo de diversas equações, o que força os desenvolvedores a analisar grande parte do modelo para determinar as equações que descrevem o comportamento de um elemento. Estas dificuldades inibem a utilização prática da Dinâmica de Sistemas na modelagem de projetos de software. Idealmente, os modelos deveriam ser construídos a partir dos conceitos do domínio do problema, sendo traduzidos posteriormente para equações da Dinâmica de Sistemas para sua avaliação. Para tal, foi definido um metamodelo de alto nível de abstração para a Dinâmica de Sistemas, que permite a definição de construtores com semântica mais próxima ao domínio do problema e sua posterior tradução para os construtores básicos da linguagem de simulação (Barros et al., 2001). A representação mais próxima do domínio do problema facilita a construção de modelos, enquanto o processo de tradução permite que não hajam perdas na capacidade de simulação. Os modelos de projeto do gerenciamento baseado em cenários são especificados a partir de construtores baseados em modelos descritivos de processos de desenvolvimento de software (desenvolvedores, atividades, recursos e artefatos).

A separação entre fatos e incertezas permite separar as informações genéricas (e potencialmente reutilizáveis) de um domínio das informações específicas de um problema representado em um modelo. Um modelo geralmente têm o objetivo de descrever um problema específico dentro de um domínio. Entretanto, o modelo contém conhecimento do domínio embutido em suas equações e parte deste conhecimento é genérico e reutilizável. Este conhecimento não é facilmente identificado entre as equações do modelo, o que inibe a criação de modelos em domínios complexos, uma vez que cada desenvolvedor de modelos deve adquirir, aprender e representar o mesmo conhecimento de domínio em cada novo modelo. Esta característica também não oferece economia de escala quando diversos modelos são construídos para o mesmo domínio porque cada modelo deve repetir a descrição do conhecimento do domínio. A utilização de modelos de cenários como extensões dos modelos de projeto, permitindo que os primeiros contenham informações específicas enquanto os primeiros conteriam informações genéricas do domínio, suporta a separação entre fatos e incertezas em um modelo.

Simulação contínua é a técnica que permite a avaliação do comportamento de um modelo de projeto. Como os modelos são descritos por formulações matemáticas (equações que descrevem as taxas de variação dos repositórios), suas equações podem ser avaliadas, revelando o comportamento subjacente do modelo. A simulação de modelos de projeto isolados descreve o comportamento esperado do projeto, sem a ocorrência dos problemas e oportunidades documentados nos cenários. A integração de uma combinação de modelos de cenário a um modelo de projeto pode provocar alterações no comportamento do projeto (através de alterações nas equações de suas taxas ou da inclusão de novas taxas associadas aos repositórios originais do modelo de projeto). A simulação de um modelo de projeto com um determinado cenário demonstra o impacto potencial da ocorrência do problema ou oportunidade representada no modelo de cenário sobre o comportamento do projeto. Este tipo de simulação, quando comparada a simulação do modelo de projeto sem a integração de cenários, apresenta ao gerente a sensibilidade do projeto à ocorrência dos eventos descritos nos modelos de cenário.

3 Análise Experimental das Técnicas Propostas

Após o estabelecimento das bases para a construção, integração e simulação dos modelos de projeto e cenário, decidimos executar um estudo de viabilidade experimental que justificasse o investimento de maiores esforços no refinamento das técnicas propostas (Barros, 2002a; Barros et al., 2002b). Neste estudo, focalizamos a aplicação dos modelos de cenários no gerenciamento de projetos de software, em contrapartida à construção de novos cenários. Este foco se justifica no objetivo do estudo, que era buscar algumas evidências da eficácia dos cenários no suporte à tomada de decisões, antes da construção de melhores ferramentas para o desenvolvimento de modelos.

Dezessete (17) alunos de cursos de mestrado e doutorado em Engenharia de Software e um (1) aluno de graduação em Ciência da Computação foram selecionados como participantes do estudo. Os participantes foram requisitados a atuar como gerentes de um pequeno projeto de desenvolvimento de software, envolvendo cerca de 67 pontos de função ajustados e seis entidades em seu projeto de alto nível. O projeto proposto teve como objetivo o desenvolvimento de parte de um sistema de controle acadêmico. Este sistema controla as informações sobre professores, alunos, áreas de pesquisa, disciplinas e inscrições em cursos.

O estudo teve como objetivo observar se os gerentes utilizando as técnicas de integração e simulação para apoiar suas decisões apresentariam melhor desempenho do que os gerentes que se basearam apenas em suas experiências e outras técnicas. O critério de desempenho selecionado foi o tempo para conclusão do projeto proposto: os gerentes foram requisitados a concluir o projeto no menor tempo possível. Realizou-se também uma avaliação qualitativa da viabilidade e usabilidade das técnicas propostas.

Os participantes foram aleatoriamente distribuídos em dois grupos: os integrantes do primeiro grupo aplicaram as técnicas propostas, enquanto o segundo grupo atuou como um grupo de controle. Os integrantes do primeiro grupo foram treinados na utilização das técnicas propostas e de um ambiente de simulação desenvolvido para oferecer suporte a estas técnicas. Os integrantes do segundo grupo receberam apenas um resumo do projeto.

Como não seria possível criar um projeto real para cada participante – devido ao custo de preparação e realização de cada projeto, que seria inviável para um estudo de viabilidade –, um emulador de projeto foi desenvolvido para o estudo. O emulador contém o modelo do projeto selecionado para o experimento, assim como os cenários que serão aplicados sobre este modelo. O emulador foi utilizado por todos os participantes do estudo, recebendo as decisões dos participantes sobre a distribuição dos desenvolvedores nas atividades do processo, o tempo de trabalho diário de cada desenvolvedor e o tempo dedicado às atividades de controle de qualidade. O emulador calculou os resultados destas decisões sobre o tempo e o custo do projeto, apresentando estes resultados para o usuário. O sistema funciona em turnos, ou seja, tomadas todas as decisões desejadas para um determinado instante do projeto, o participante comanda o avanço de um dia no projeto. A passagem de um dia gera novas informações sobre o comportamento do projeto, que podem ser utilizadas pelo participante para novas decisões. O sistema prossegue desta forma até que o projeto seja encerrado. A aplicação das técnicas de integração e simulação de modelos de projeto e cenário permitiram que os integrantes do primeiro grupo testassem a sensibilidade de seus projetos às decisões no ambiente de simulação, antes que estas decisões fossem aplicadas no emulador de projetos.

Questionários foram utilizados para coletar dados qualitativos sobre as características dos participantes, a viabilidade e a usabilidade das técnicas propostas. O emulador de projetos armazenou informações quantitativas sobre o tempo que cada participante consumiu para a conclusão do projeto, medido em dias. Após a eliminação de valores extremos, os dados quantitativos foram submetidos a um teste-T com 95% de certeza (Wohlin et al., 2000). A Tabela 1 apresenta estatísticas descritivas extraídas destes dados após a eliminação de valores

extremos. O teste-T concluiu que o tempo médio para a conclusão do projeto dos participantes que aplicaram as técnicas propostas foi menor do que o tempo médio dos participantes que não as utilizaram. Um teste não-paramétrico baseado em ranqueamento, denominado teste de Mann-Whitney (Wohlin et al., 2000), confirmou os resultados do teste-T.

Tabela 1. Resultados da análise dos dados quantitativos do estudo experimental

Tempo para Conclusão do Projeto	Participantes que aplicaram as técnicas	Participantes que não aplicaram as técnicas
Média	28,1 dias	34,3 dias
Máximo	38,0 dias	42,0 dias
Mínimo	25,0 dias	28,0 dias
Desvio Padrão	4,7 dias	4,9 dias

Além da análise quantitativa, todos os participantes do primeiro grupo concordaram que as técnicas propostas foram úteis no gerenciamento do projeto. Assim, os dados e os resultados positivos derivados da análise do estudo apresentam indícios de que as técnicas de integração e simulação de modelos de projeto e cenário podem apoiar os gerentes de projeto. Por outro lado, as informações qualitativas indicam que existe espaço para o aprimoramento destas técnicas: alguns participantes do estudo indicaram possíveis melhorias, especialmente no que concerne o treinamento e a visualização dos resultados das simulações.

4 Análise de Risco baseada em Cenários

As técnicas apresentadas na seção 2 descrevem como os cenários são formulados, integrados a modelos de projeto e como o comportamento do projeto é avaliado com e sem a integração de cenários para determinar o impacto destes cenários sobre o projeto. Cenários representam eventos incertos que podem ocorrer durante um processo de desenvolvimento de software. Sendo incertos, estes cenários podem ser vistos como riscos enfrentados pelo projeto. Baseado nesta constatação e na possibilidade de aplicar os modelos de cenários na avaliação quantitativa do impacto dos riscos sobre um projeto, utilizamos as técnicas propostas na seção 2 para definir um processo de análise de riscos. Este processo descreve como os riscos são documentados através de modelos de cenários, como são reutilizados ao longo de uma série de projetos similares e como os modelos de cenários são utilizados na avaliação do impacto dos riscos sobre uma aplicação.

A análise de riscos no gerenciamento de projetos baseado em cenários se baseia na representação de riscos através de cenários, na reutilização destes cenários ao longo de diversos projetos e na simulação de combinações de cenários para a avaliação do impacto provocado pela ocorrência dos riscos sobre o comportamento de um determinado projeto. Estas atividades representam dois diferentes objetivos em um processo de análise de riscos. Primeiro, os riscos são identificados e documentados utilizando-se de cenários. Em seguida, os riscos são reutilizados ao longo de diversos projetos, permitindo a simulação de seu impacto sobre o comportamento do projeto. Estes objetivos lembram um processo genérico de reutilização, conforme apresentado na Figura 2.

O processo genérico de reutilização se baseia em dois sub-processos: um para o desenvolvimento de artefatos reutilizáveis, que identifica e documenta estes artefatos de software, e outro para o desenvolvimento de aplicações com base na reutilização dos artefatos previamente desenvolvidos, que os seleciona e adapta a partir de uma base de artefatos reutilizáveis. O primeiro sub-processo é denominado *desenvolvimento para reutilização*, enquanto o segundo é denominado *desenvolvimento com reutilização*.

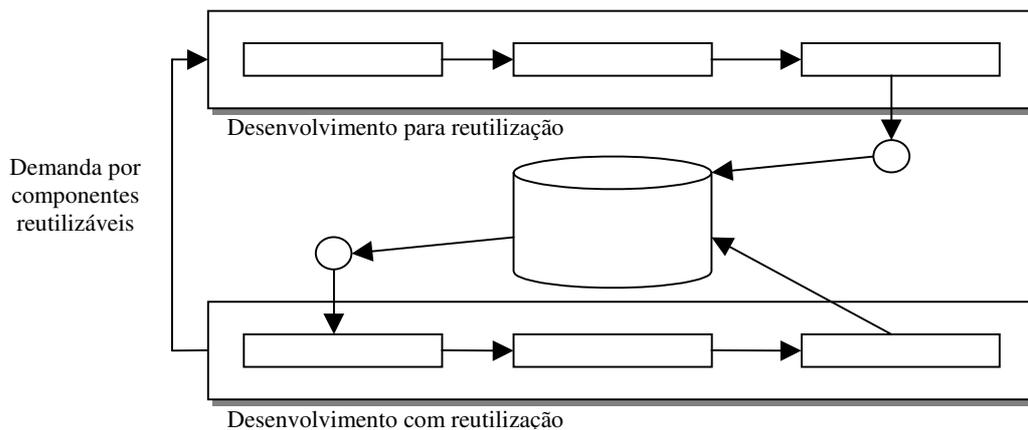


Figura 2 – O processo genérico de reutilização de software (MURTA, 1999)

Projetando-se o processo genérico de reutilização sobre o processo de análise de riscos proposto pelo paradigma de gerenciamento de projetos baseado em cenários, este processo também é dividido em dois sub-processos (Barros et al., 2002a): um para a identificação de riscos e outro para a reutilização destes riscos durante o desenvolvimento de uma aplicação. O artefato reutilizável compartilhado pelos dois processos de análise de riscos é representado pelas informações que descrevem um risco.

O principal objetivo do processo para identificação de riscos é descrever os riscos associados a utilização de um elemento de projeto específico em um projeto de desenvolvimento de software. Os elementos de projeto incluem o domínio da aplicação, as tecnologias, os papéis desempenhados pelos desenvolvedores, os artefatos de software e os recursos consumidos pelo projeto. O processo de análise de riscos organiza as informações sobre os riscos associados a estes elementos e permite sua reutilização em diversos processos de desenvolvimento de aplicações. O primeiro processo de análise de riscos é executado quando uma organização explora um novo elemento de projeto (como uma metodologia, uma linguagem de programação, um domínio de aplicação, entre outros) que pode impor riscos a diversos projetos de desenvolvimento de software.

O principal objetivo do processo de análise de riscos para o desenvolvimento de aplicações é identificar e avaliar os riscos que podem afetar um projeto de acordo com os elementos de projeto que o compõem. Este processo reutiliza os riscos associados aos elementos de projeto, identificados e documentados pelo processo de análise de riscos anterior. Ele ocorre em paralelo com o processo de desenvolvimento de uma aplicação, acompanhando sua evolução e resolvendo seus riscos.

Os processos de análise de riscos reforçam a análise quantitativa, modelando o impacto do risco e de suas estratégias de resolução como modelos de cenário. Reconhecemos as dificuldades de modelar quantitativamente todos os riscos envolvidos em um processo de desenvolvimento de software, especialmente os riscos não técnicos, como riscos legais, contratuais e políticos. Entretanto, ao sugerir a documentação formal dos riscos através de cenários, esperamos que a equipe de identificação de riscos discuta os efeitos dos riscos sobre o projeto, documentando as premissas que levaram à construção dos seus modelos de impacto. Sem a documentação formal dos riscos, o raciocínio sobre seus efeitos no projeto pode ser muito subjetivo, eventualmente sem utilidade para projetos futuros.

Os riscos são identificados e documentados através de *arquétipos de risco* (Barros et al., 1999; Barros et al., 2000b). Um arquétipo de risco é uma descrição de um problema potencial, recorrente ao longo de diversos projetos, cuja ocorrência pode causar prejuízos a um produto ou processo de desenvolvimento de software. Ele inclui uma descrição do contexto em que o

problema pode ocorrer e apresenta estratégias que podem ser aplicadas para solucionar o problema antes ou depois de sua ocorrência.

Os arquétipos de risco representam os artefatos reutilizáveis que conectam os dois sub-processos de análise de riscos. Eles identificam, documentam e fornecem informação para a avaliação dos eventos potenciais que podem afetar negativamente os projetos de desenvolvimento de software. A representação padronizada visa promover a comunicação dos riscos, seu entendimento e a distinção clara dos diversos eventos incertos. Além disso, os arquétipos de risco unem as estratégias de resolução (planos de contenção ou contingência) aos problemas potenciais por eles descritos, trazendo a descrição do problema em conjunto com suas possíveis soluções e heurísticas para aplicação destas soluções (Barros et al., 2002a). Um arquétipo de risco é composto pelos seguintes blocos:

- *Identificação*: descreve o problema potencial que é representado pelo arquétipo de risco. Este bloco contém uma descrição textual e um modelo de cenário representando o impacto do problema potencial sobre um projeto de desenvolvimento de software. Fontes de informação utilizadas para construir o modelo também são apresentadas no arquétipo;
- *Mecanismos de Identificação*: descreve os mecanismos utilizados para identificar o risco em um determinado projeto. Este bloco contém uma descrição textual do contexto em que o risco ocorre, um questionário para identificação das condições que promovem a ocorrência do risco em um projeto e uma lista de casos conhecidos de ocorrência;
- *Estratégias de Contenção*: descreve estratégias para inibir ou eliminar o problema potencial descrito pelo arquétipo antes que o risco ocorra em um projeto. Este bloco provê uma descrição textual e um modelo de cenário para cada estratégia de resolução, associando estas estratégias às condições em que elas podem ser aplicadas. Como no modelo de cenário do bloco de identificação, fontes de informação são fornecidas junto a cada modelo de cenário;
- *Estratégias de Contingência*: descreve estratégias para reduzir o impacto do problema potencial descrito por um arquétipo de risco após sua ocorrência em um projeto. Como no bloco de estratégias de contenção, um modelo de cenário e uma descrição textual são fornecidas para cada estratégia de contingência, em conjunto com fontes de informação;
- *Arquétipos Relacionados*: descreve arquétipos de riscos que ocorrem em situações similares ou arquétipos de risco que podem substituir o arquétipo atual em projetos de software.

5 Um Estudo de Caso de Aplicação do Processo de Análise de Riscos

Nesta seção, apresentamos um exemplo de aplicação do processo de análise de riscos proposto em projetos reais. Os resultados apresentados aqui derivam de um estudo observacional que utilizou os dois sub-processos em dois projetos desenvolvidos para o mesmo domínio de aplicação. O domínio selecionado foi o gerenciamento de riscos no mercado financeiro. Os projetos selecionados envolvem a especialização de um pacote de gerenciamento de riscos para o caixa, a dívida e o ativo operacional de duas grandes empresas brasileiras nos setores de siderurgia e petróleo.

Antes da realização do primeiro projeto, existia pouca informação sobre os riscos relacionados com o desenvolvimento de um software de gerenciamento de riscos para empresas brasileiras. A equipe responsável pelo projeto tinha experiência prévia no desenvolvimento de um software de gerenciamento de riscos para um banco de investimentos. Entretanto, a especialização deste software para uma empresa apresentou diversas diferenças, como diferentes horizontes para análise de riscos e a diversidade de commodities (tais como ouro, soja, metais, nafta, entre outros) que são negociados pelas empresas.

O primeiro projeto foi realizado em uma empresa do setor siderúrgico. Este projeto estava sujeito a riscos relacionados com as particularidades do gerenciamento de riscos de mercado em empresas, além de riscos genéricos de projetos de desenvolvimento de software, como volatilidade de requisitos, por exemplo. Os riscos dependentes de domínio foram documentados como arquétipos de risco e associados a um elemento de projeto denominado *domínio de software de gerenciamento de riscos no mercado financeiro*. Como exemplos destes riscos, podemos citar:

1. Modelagem de commodities: devido a sua sazonalidade e dependência de fatores externos, como condições de tempo, empresas de provisionamento, pragas e infestações, as incertezas associadas com determinados commodities são difíceis de modelar. Processos estocásticos conhecidos (e geralmente complexos) descrevem o comportamento dinâmico dos preços destes commodities, como petróleo e ouro. A mensuração dos riscos dependentes destes processos também é uma tarefa difícil, especialmente nos derivativos de commodities;
2. Avaliação de riscos baseada em simulações de Monte Carlo: estas simulações são utilizadas com frequência para mensurar os riscos associados com instrumentos do mercado financeiro que não podem ser avaliados analiticamente. Enquanto as simulações de Monte Carlo são flexíveis e representam uma plataforma confiável para a avaliação de riscos, elas exigem muitos recursos computacionais, reduzindo o desempenho da avaliação de riscos. A dependência em simulações de Monte Carlo para a avaliação de grandes carteiras de instrumentos financeiros pode reduzir o desempenho do software de análise de riscos, visto que seu usuário deverá esperar muito pela resposta de cada análise que deseje executar;
3. Volatilidade da composição das carteiras: definido aqui como a freqüente inclusão de novos tipos de instrumentos financeiros nas carteiras da empresa, a volatilidade de composição das carteiras é uma faceta dependente de domínio dos riscos impostos pela volatilidade de requisitos. Cada vez que um novo instrumento é inserido no software de gerenciamento de riscos, seu mapeamento para fatores de risco deve ser definido, projetado, codificado e testado. Eventualmente, novos fatores de risco e dados históricos sobre sua variação no tempo também precisam ser adicionados.

Mesmo sem uma avaliação formal de seus riscos, o primeiro projeto foi considerado como bem sucedido, sendo concluído com apenas uma semana de atraso em seu cronograma de quatro meses. O projeto foi dividido em quatro fases: elicitação de carteiras, análise estatística, mapeamento de instrumentos para fatores de risco e avaliação do mapeamento. Após a conclusão do projeto, arquétipos de risco foram documentados, através do processo de identificação de riscos para elementos de projeto.

O segundo projeto foi desenvolvido em uma empresa do setor petrolífero. Este projeto executou o processo de análise de riscos para o desenvolvimento de aplicações de forma simplificada, utilizando os arquétipos de risco desenvolvidos após o primeiro projeto. O segundo projeto foi maior do que o primeiro, visto que a empresa possui diversas carteiras de ativo operacional, distribuídas ao longo de suas diversas subsidiárias e diferenciados nos produtos vendidos, comprados ou distribuídos. O projeto foi planejado para ser realizado em seis meses, mas se atrasou em uma semana. Acreditamos que o projeto poderia ter se atrasado ainda mais se não fosse a análise dos arquétipos de risco e a experiência obtida na realização do projeto anterior.

Os arquétipos de risco foram utilizados como alerta de problemas no segundo projeto. Quando a equipe de análise de carteiras, composta por engenheiros financeiros, produziu uma

lista de instrumentos financeiros para a equipe de mapeamento, composta de analistas de sistemas e programadores, os arquétipos de risco produzidos pelo primeiro projeto auxiliaram os desenvolvedores na identificação dos instrumentos que representavam riscos para o projeto. Os analistas de carteiras puderam, então, se concentrar na decomposição destes elementos, replicando seu comportamento a partir de um conjunto de instrumentos de mercado mais simples sempre que possível. Simulações de Monte Carlo, necessárias para a avaliação dos instrumentos complexos, foram oferecidas como ferramenta complementar, utilizada apenas para análises mais detalhadas, enquanto aproximações analíticas foram utilizadas para o controle de riscos realizado periodicamente pela empresa. Cinco novos arquétipos de risco também foram gerados a partir do conhecimento adquirido após o encerramento do segundo projeto.

Planejamos, no futuro, realizar novos estudos observacionais utilizando modelos de cenário. A falta de informações históricas, assim como de especialistas e bibliografia específica, impediu a construção de modelos de cenários para os dois primeiros projetos. Entretanto, foram coletadas informações quantitativas para a construção de modelos de cenário no futuro.

6 Conclusões

Podemos observar a relevância da análise de riscos por referências a esta atividade em diversas normas e padrões de qualidade para o desenvolvimento de software. Em diversas normas (ABNT, 2000), nota-se também a importância da utilização de experiências e dados históricos provenientes de projetos anteriores. Esta experiência reside no repositório mental de padrões dos gerentes de projeto experientes. Devido às dificuldades relacionadas com a transferência deste conhecimento tácito, percebemos a demanda por uma representação explícita desta experiência.

Na tese relatada neste artigo, propomos a utilização de modelos formais, construídos segundo uma linguagem de modelagem denominada Dinâmica de Sistemas e avaliados por simulações contínuas. A viabilidade da utilização destes modelos no gerenciamento de projetos de software foi avaliada através de um estudo experimental. Os modelos foram aplicados em um processo de análise de riscos, que, por sua vez, foi motivo de um estudo de caso. Os resultados positivos obtidos nos estudos experimentais apresentam indícios de que as técnicas propostas podem oferecer suporte às decisões tomadas por gerentes de projeto. A análise destes estudos também apresenta direções em que as técnicas podem ser aprimoradas.

Os próximos passos desta pesquisa compreendem o desenvolvimento de novos modelos de cenários, de ferramentas de suporte à construção, integração e simulação de modelos de projeto e cenários, e a avaliação do conjunto de técnicas propostas através de diferentes estudos experimentais e da repetição do estudo descrito na seção 3 em contextos distintos.

Agradecimentos

Os autores gostariam de agradecer ao CNPq, CAPES e FINEP por seu investimento neste trabalho, a Dr. Shari Lawrence Pfleeger por suas sugestões e críticas e aos participantes do estudo experimental, por sua valiosa contribuição.

Referências Bibliográficas

- ABDEL-HAMID, T., MADNICK, S.E., 1991, *Software Project Dynamics: An Integrated Approach*, Englewood Cliffs, NJ: Prentice-Hall, Inc.
- ABNT, 2000, “NBR ISO 10006: Gestão da Qualidade – Diretrizes para a Qualidade no Gerenciamento de Projetos”, Associação Brasileira de Normas Técnicas, Rio de Janeiro, Brasil

- ALBIN, S., 1997, *Building an System Dynamics Model Part 1: Conceptualization*, IN: Relatório Técnico D-4597, MIT System Dynamics Group, Cambridge, MA
- AUGUSTINE, N.R., 1982, *Augustine's Laws*, Nova York, NY: American Institute Of Aeronautics and Astronautics
- BARROS, M.O., WERNER, C.M.L., TRAVASSOS, G.H., 1999, "Risk Analysis: a Key Success Factor for Complex System Development", *Proceedings of the 12th International Conference in Software & System Engineering and their Applications*, Paris, FR (Dezembro)
- BARROS, M.O., WERNER, C.M.L., TRAVASSOS, G.H., 2000a, "Applying Formal Modeling and Dynamic Simulation to Quality Evaluation, Prediction and Improvement of Software Processes", IN: *Anais do Workshop de Qualidade de Software do XIV Simpósio Brasileiro de Engenharia de Software (WQS'2000)*, João Pessoa, PB, Brasil (Outubro)
- BARROS, M.O., WERNER, C.M.L., TRAVASSOS, G.H., 2000b, "L' analyse de risques: un facteur de succès pour le développement de systèmes complexes", *Génie Logiciel*, ISSN 0295-6322, Setembro(24), pp. 14-20
- BARROS, M.O., WERNER, C.M.L., TRAVASSOS, G.H., 2001, "From Models to Metamodels: Organizing and Reusing Domain Knowledge in System Dynamics Model Development", IN: *Proceedings of the 19th Conference of the System Dynamics Society*, Atlanta, USA (Julho)
- BARROS, M.O., WERNER, C.M.L., TRAVASSOS, G.H., 2002a, "Project Management Knowledge Reuse Through Scenario Models", IN: *Proceeding of the 7th International Conference on Software Reuse, ICSR-7*, Austin, Texas, Estados Unidos (Abril)
- BARROS, M.O., WERNER, C.M.L., TRAVASSOS, G.H., 2002b, "Um Estudo Experimental sobre a Utilização de Modelagem e Simulação no Apoio à Gerência de Projetos de Software", IN: *Anais do XVI Simpósio Brasileiro de Engenharia de Software*, Gramado, RS (Outubro)
- BROWN, N., 1996, "Industrial-Strength Management Strategies", *IEEE Software*, Vol. 13, No. 4 (Julho)
- CHARETTE, R.N., 1996, "Large-Scale Project Management is Risk Management", *IEEE Software*, Vol. 13, No. 4 (Julho), pp. 110 – 117
- FORRESTER, J.W., 1961, *Industrial Dynamics*, Cambridge, MA: The MIT Press
- GROSS, N., STEPANEK, M., PORT, O., CAREY, J., 1999, "Software Hell", *Business Week* (Dezembro).
- HALL, E.M., 1998, *Managing Risk: Methods for Software Systems Development*, IN: SEI Series in Software Engineering, Reading, MA: Addison Wesley Longman Inc.
- KLEIN, G., 1998, *Sources of Power*, Cambridge, MA: MIT Press
- LAFLEUR, R., 1996, "Project Management: Getting Control and Keeping Control of Complex Projects", *American Programmer*, (Abril), pp. 23 – 28
- LIN, C.Y., ABDEL-HAMID, T., SHERIF, J.S., 1997, "Software-Engineering Process Simulation Model (SEPS)", *The Journal of Systems and Software*, Vol. 38, pp. 263 – 277
- MURTA, L.G.P., 1999, *Framedoc: um Framework para a Documentação de Componentes Reutilizáveis*, Projeto Final do Curso de Matemática Aplicada a Informática, Instituto de Matemática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil
- REEL, J.S., 1999, "Critical Success Factors in Software Projects", *IEEE Software*, Vol. 16, No. 3 (Maio/Junho), pp. 18 – 23
- ROYCE, W., 1998, *Software Project Management: A Unified Framework*, Reading, MA: Addison-Wesley Publishing Co.
- STANDISH GROUP, T., 1994, *The Chaos Report*, The Standish Group Report, disponível na URL http://www.pm2go.com/sample_research/chaos_1994_1.asp
- TVEDT, J.D., 1996, *An Extensible Model for Evaluating the Impact of Process Improvements on Software Development Cycle Time*, Tese de Doutorado, Arizona State University, Tempe, AZ, Estados Unidos
- WOHLIN, C., RUNESON, P., HÖST, M. et al., 2000, *Experimentation in Software Engineering: an Introduction*, Norwell, MA: Kluwer Academic Publishers