ESCOLA POLITÉCNICA
DE PERNAMBUCO

Departamento de Sistemas e Computação

# RBT*Tool*: Proposta de Melhorias e Evolução na Automação das Atividades de Teste baseado em Riscos

## Trabalho de Conclusão de Curso

## Engenharia da Computação

**Júlio Venâncio de Menezes Júnior**
**Orientador: Profª Dra. Cristine Gusmão**
**Co-orientador: Profª MSc Ellen Souza**

UNIVERSIDADE
DE PERNAMBUCO

**Júlio Venâncio de Menezes Júnior**

# RBT*Tool*: Proposta de Melhorias e Evolução na Automação das Atividades de Teste baseado em Riscos

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Recife, Junho de 2009.**

*Dedico este trabalho a todos que contribuíram de maneira direta ou indireta para que eu conseguisse obter o título de Engenheiro da Computação.*
*Aos meus pais, que me deram toda a base necessária para que eu pudesse alcançar meus objetivos.*

# Agradecimentos

Este trabalho fecha com chave de ouro tudo o que foi vivenciado durante esta importante etapa na minha vida. Foram muitos momentos bons e ruins que serviram de lição e, sem dúvidas, são únicos.

Agradeço a todos os professores do DSC, que mostraram comprometimento a ponto de motivarem seus alunos a sempre acreditarem, proporcionando ao curso de Engenharia da Computação um nível de qualidade invejável.

Agradeço muito também às minhas orientadoras e amigas Cristine Gusmão e Ellen Souza. Hoje, posso afirmar categoricamente que foram as principais responsáveis pela minha evolução como profissional e pessoa.

No mais, agradeço aos colegas de curso e amigos que torceram muito por mim.

Muito obrigado!

# Resumo

A atividade de Teste de Software é reconhecidamente essencial para a garantia de qualidade de software. As organizações que desenvolvem software sabem dessa importância e buscam sempre a conformidade do produto final e/ou parcial com os requisitos e padrões especificados, ou até mesmo implícitos. No entanto, esta atividade requer um esforço considerável. A técnica de Teste baseado em Riscos (*Risk-based Testing* – RBT) permite justamente priorizar esforços e alocar recursos para os requisitos de software que necessitam ser testados prioritariamente, com o objetivo de otimizar as atividades relacionadas ao Teste de Software. Ainda assim, os engenheiros de teste, ainda encontram dificuldades em aplicar a técnica na prática, pela ausência de conhecimentos sólidos sobre as atividades da Gerência de Riscos de Software e, principalmente, pela ausência de ferramental de apoio às atividades necessárias para utilização desta técnica por completo. Dentro deste contexto, este trabalho apresenta a RBT*Tool*, uma ferramenta de apoio às atividades de Teste baseado em Riscos. A primeira versão, já disponibilizada, contempla apenas a atividade de Identificação de Riscos. Este trabalho visa dar continuidade ao desenvolvimento da RBT*Tool* através de melhorias e adição de novas funcionalidades de apoio às atividades de Análise de Riscos, Planejamento de Teste e Projeto de Casos de Teste. Também, será realizado um estudo de avaliação mais aprofundado, aplicado em ambiente de desenvolvimento de software, como forma de avaliação dos requisitos desenvolvidos.

# Abstract

Testing activity is admittedly essential for software quality assurance. Organizations which develop software know about this importance and always seek the fact that will exist agreement between the final/partial product and the specified, or even implicit, requirements and patterns. However, this activity requires significant efforts. Risk-based technique justly allows prioritizing efforts and allocating resources for the software requirements which need to be tested carefully, to take the software testing activities optimized. Nevertheless, testers continually encounter difficulties in effectively applying risk-based in practice, because they normally do not have Risk Management knowledge and, specially, there is no software tool to support the approach inside all the risk-based testing activities. In this light, this work presents the RBT*Tool*, a support tool for risk-based testing activities. First version was developed and it just was considered Risk Identification activity. This work aims to continue the RBT*Tool* development, improving the currently features and also adding new features which supports Risk Analysis, Test Planning and Test Cases Project activities. Also will be made an evaluating study applied in a software development environment, in order to evaluate the developed requirements.

# Sumário

# Índice de Figuras

# Índice de Tabelas

# Tabela de Símbolos e Siglas

CASE – *Computer-aided Software Engineering*

GARA – Gestão Ágil de Riscos de Ambiente

NASA – *National Aeronautics and Space Administration*

RBT – *Risk-Based Testing*

RBT*Tool* – *Risk-based Testing Tool*

RCP – *Rich Client Platform*

RE – *Risk Exposure*

RiteDAP – *Risk-based test case Derivation And Prioritization*

RUP – *Rational Unified Process*

SATC - *Software Assurance Technology Center*

SEI – *Software Engineering Institute*

SQA – *Software Quality Assurance*

TBQ - *Taxonomy Based Questionnaire*

# Capítulo 1

# Introdução

As organizações em geral enfrentam um grau de competitividade muito alto, onde um dos itens considerados essenciais para sua sobrevivência está fortemente relacionado à garantia de qualidade dos produtos e/ou serviços prestados. Esta realidade não poderia ser diferente em ambientes de desenvolvimento de software. Existem diversas normas, metodologias e processos adotados pelas empresas, cujo objetivo principal é avaliar a qualidade do software que está sendo desenvolvido.

Neste contexto, a atividade de Teste de Software surge como umas das formas de melhorar a qualidade dos produtos de software, procurando sempre garantir que haja a conformidade do produto final e/ou parcial com requisitos e padrões especificados, ou até mesmo implícitos através da descoberta de defeitos.

Da mesma forma, a Gerência de Riscos em projetos de desenvolvimento de software - em especial os riscos técnicos - tem se mostrado fundamental no contexto de melhoria dos processos de Qualidade de Software. Através da detecção de potenciais problemas relacionados ao projeto, implementação, interface, verificação e manutenção, a gerência de riscos pode auxiliar bastante na definição de estratégias que busquem atingir os objetivos com o menor tempo e custo, sem abrir mão da qualidade do produto final.

A técnica de Teste baseado em Riscos (*Risk-based Testing* – RBT) consegue destacar a importância da atividade de gerência de riscos aplicada ao processo de Teste de Software, permitindo a priorização de esforços e alocação de recursos para os componentes de software que necessitam ser testados mais cuidadosamente a partir da identificação, análise e controle dos riscos técnicos associados ao requisito do software [Souza e Gusmão 2009].

O objetivo deste capítulo introdutório é apresentar a relevância do tema abordado, como também suas motivações, objetivos e metodologia utilizada para a realização deste trabalho.

## 1.1 Motivação

O processo de Teste de Software é reconhecidamente essencial para a garantia de qualidade de software (*Software Quality Assurance* – SQA). As organizações que desenvolvem software sabem dessa importância e buscam sempre a conformidade do produto final e/ou parcial com os requisitos e padrões especificados, ou mesmo implícitos.

No entanto, o processo de teste requer esforço considerável e é caro realizá-lo. Muito frequentemente, as atividades de teste chegam a custar até 40% do valor inicial do produto [Pressman 1995]. Vale destacar, que quanto mais tarde um erro é encontrado, mais caro é o custo de correção, que cresce de forma exponencial.

Diante das restrições de tempo e de custo para entrega do produto final ao cliente, é comum não se dar a devida atenção ao processo de teste durante o projeto de desenvolvimento de software. Contudo, ao mesmo tempo, as organizações não querem perder clientes por conta de um produto de má qualidade. Desta forma, é necessário haver alguma maneira de priorizar esforços e alocar recursos para os componentes do software que necessitam ser testados mais cuidadosamente.

A técnica de Teste baseado em Riscos (*Risk-based Testing* – RBT) surgiu como uma abordagem para minimizar alguns desses problemas de forma geral. De acordo com Souza, a abordagem de RBT consiste em um conjunto de atividades que favorecem a identificação, análise e controle de fatores de riscos associados aos requisitos do produto de software [Souza 2008].

Mesmo com a simplicidade de utilização da abordagem RBT, a comunidade de testes ainda encontra muitas dificuldades em aplicá-la na prática [Goldsmith 2006], por não conhecer técnicas para identificação e análise de riscos para testes de software e, principalmente, pela ausência de ferramentas específicas que auxiliem por completo na utilização dessas técnicas [Souza 2008].

Desta forma, este trabalho visa melhorar e evoluir a **RBT*Tool*** - uma ferramenta de apoio às atividades de teste baseado em riscos. A versão inicial

proposta por Oliveira, compreende as atividades de identificação de riscos técnicos [Oliveira 2008]. Logo, esse trabalho tem como objetivos avaliar e dar continuidade à versão inicial através da implementação de melhorias e de novas funcionalidades relacionadas às atividades de análise de riscos, planejamento de testes e projeto de casos de teste.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

O objetivo geral deste trabalho é evoluir a RBT*Tool,* iniciada por Oliveira [Oliveira 2009], de forma que esta atenda também aos requisitos relacionados à análise de riscos, planejamento de testes e projeto de casos de teste.

### 1.2.2 Objetivos específicos

Para que o objetivo geral seja atingido, algumas metas foram identificadas:

- Pesquisar sobre as principais abordagens de teste baseado em riscos, identificando vantagens, desvantagens, pontos em comuns e elaborando um estudo analítico.
- Identificar os requisitos de forma que atendam as principais atividades das abordagens estudadas. Dando continuidade, desta forma, à construção da RBT*Tool*.
- Avaliar a RBT*Tool* através de estudos experimentais..
- Disseminar a técnica de teste baseado em riscos e a RBT*Tool* através da escrita de trabalhos técnicos e científicos.

Este último item dos objetivos específicos foi obtido através da publicação de artigo que propõe métricas específicas de controle e medição para casos de teste e atividades RBT:

Souza, E.; Gusmão, C.; Alves, K.; Venâncio, J., Melo, R.; **Measurement and Control for Risk-based Test Cases and Activities**. In: 10th IEEE Latin American Test Workshop, LATW, 2009, Búzios, Rio de Janeiro - Brasil.

## 1.3 Metodologia

Foi definido um conjunto de etapas, cada qual com um conjunto de estratégias de forma que o desenvolvimento deste trabalho ocorresse de maneira linear e objetiva:

**Fase 1: Estudo sobre o tema**

Foi realizada uma revisão bibliográfica sobre a área de teste baseado em riscos, visando identificar as principais abordagens. Em paralelo, houve um estudo dos conceitos fundamentais necessários e relacionados às áreas: Testes de Software e Gerência de Riscos de Software.

Ao final, houve um estudo analítico das principais abordagens RBT identificadas, com o objetivo de identificar pontos em comum, pontos fortes e fracos e aplicações.

**Fase 2: Levantamento e Revisão dos Requisitos e Modelagem**

Parte dos requisitos já havia sido levantada, documentada e modelada por Oliveira , inicialmente [Oliveira 2008]. Foi realizada uma revisão destes artefatos, visando realizar algumas melhorias na versão atual da RBT*Tool*.

Em seguida, foram levantados e detalhados os requisitos restantes através da definição de quais atividades das abordagens estudadas na fase 1 serão suportadas pela RBT*Tool*. E finalmente, os requisitos restantes foram documentados e modelados.

**Fase 3: Implementação da RBT*Tool***

Levantados e revisados os requisitos, a implementação foi realizada nesta fase. A RBT*Tool* é livre e de código aberto para que possa ser utilizada em qualquer

ambiente de testes, bem como adaptada e/ou melhorada pelas empresas que a adotarem.

**Fase 4: Avaliação da Ferramenta**

Nesta fase, a RBT*Tool* foi avaliada através de estudos experimentais.

## 1.4 Estrutura do Documento

Este trabalho está dividido em cinco capítulos, incluindo este capítulo introdutório. O restante deste documento está estruturado da seguinte maneira:

**Capítulo 2 – Teste de Software baseado em Riscos:** Este capítulo apresenta revisão do estado da arte em relação à técnica de Teste baseado em Riscos, inicialmente fornecendo uma visão geral e depois apresentando as principais abordagens estudadas. Será realizado também um estudo analítico, destacando pontos fortes, fracos e em comum entre as abordagens estudadas, fornecendo subsídios para o levantamento de requisitos da ferramenta de forma que esta seja o mais genérica possível.

**Capítulo 3 – RBT*Tool*:** é apresentada a ferramenta desenvolvida, destacando suas funcionalidades e características. Também serão apresentadas as tecnologias, ambiente de desenvolvimento necessários para a construção e modelagem da mesma.

**Capítulo 4 – Avaliação da RBT*Tool*:** Este capítulo apresenta o estudo de avaliação da RBT*Tool*, mostrando os cenários dos estudos, características e resultados da avaliação.

**Capítulo 5 – Considerações finais:** Com base nos resultados do estudo experimental, serão apresentadas as considerações finais. Neste capítulo, serão apresentados também trabalhos relacionados, dificuldades encontradas e sugestões de trabalhos futuros.

Ao final desta monografia ainda serão apresentadas as referências bibliográficas e apêndices.

# Capítulo 2

# Teste Baseado em Riscos

No contexto de projetos, risco é um evento que possui probabilidade de ocorrência, e, caso ocorra, poderá trazer impactos positivos ou negativos. Considerando apenas os impactos negativos, pode-se afirmar que a atividade de execução de Teste de Software em geral é uma maneira indireta de reduzir e controlar o risco.

A técnica de teste baseado em riscos – RBT – permite justamente o planejamento e projeto de casos de teste com foco direto nos riscos técnicos. De acordo com Souza, a técnica de Teste de Software baseado em Riscos é a composição de um conjunto de atividades que buscam tirar proveito da identificação de riscos associados aos requisitos de software [Souza 2008]. Uma vez identificados, os riscos são priorizados de acordo com a sua probabilidade de ocorrência e impacto. Os casos de testes, por sua vez, são projetados com base nas estratégias para tratamento dos fatores de riscos identificados. E, durante todo este processo, os esforços de teste são continuamente ajustados de acordo com técnicas de controle e medição das atividades RBT.

Desta forma, a técnica RBT permite identificar e priorizar os componentes do software que realmente necessitam ser mais bem testados. Este argumento é válido, pois é sabido que em ambientes de desenvolvimento de software, a atividade de execução de testes é normalmente realizada ao final do projeto ou da iteração do projeto, quando o tempo e os recursos já estão escassos. Assim, a técnica RBT consegue fazer melhor uso do pouco tempo e recursos disponíveis.

Segundo Redmill [Redmill 2004], focar em RBT significa fazer julgamento sobre:

i) **Cobertura de teste** – Esta atividade permite a alocação de esforços de teste, permitindo medir o quanto é necessário para se testar algo em um determinado momento de maneira eficiente.

ii) **Seleção da quantidade de testes a ser conduzida** – De forma que seja suficiente e, preferencialmente, proporcional à quantidade de riscos encontrados.

iii) **Escolhas dos tipos de testes e de revisões** – Ou seja, definir o tipo de teste e revisão capaz de minimizar ou mitigar o risco do produto do software desenvolvido.

iv) **O uso e balanceamento entre teste dinâmico e análise estática** - Pelo fato de que a abordagem RBT permitir o planejamento e execução de teste também no início do projeto [Amland 1999], é preciso analisar com cuidado o que vai ser avaliado na fase inicial e o que vai ser avaliado na fase final do projeto. Esta prática visa evitar sobrecarga de testes.

## 2.1 Principais Atividades

A Figura 2.1.representa bem o conjunto de atividades RBT em geral. Os retângulos representam o processo de Gerência de Riscos, definido por Karolak [Karolak 1996], enquanto que as elipses foram inseridas por Amland [Amland 1999], sendo artefatos produzidos a partir das atividades de Gerência de Riscos.

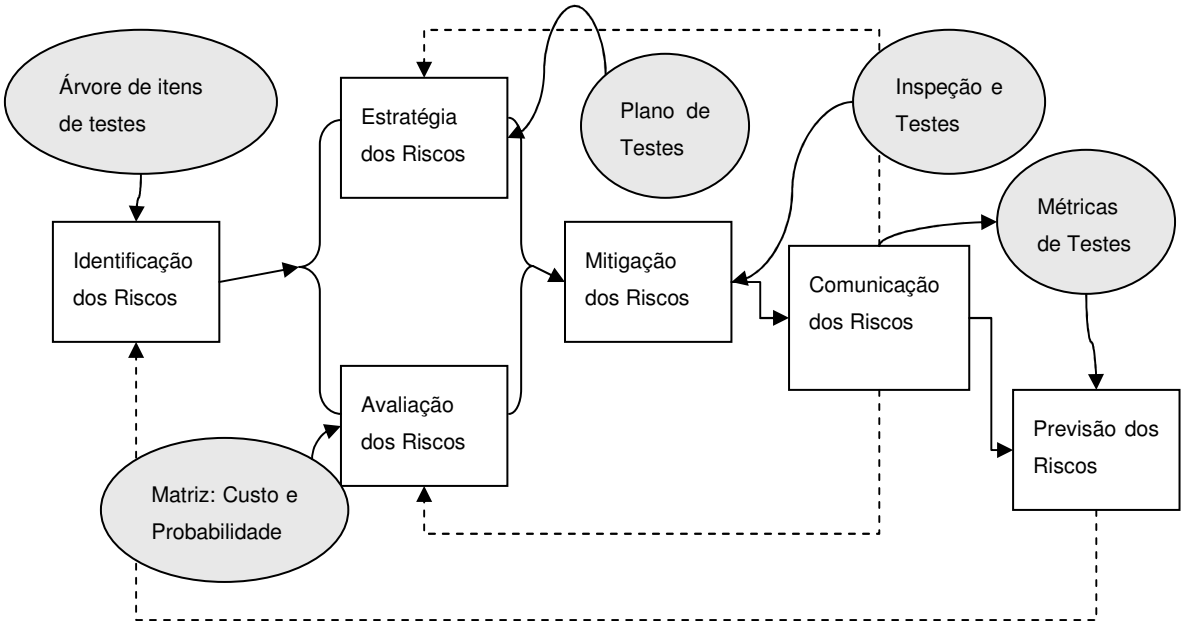



**Figura 2.1.** Atividades RBT [Amland 1999]

Para cada atividade do ciclo de vida do processo de Teste de Software, há uma atividade da Gerência de Riscos associada, com o intuito de fornecer o tratamento e acompanhamento adequado para os riscos técnicos identificados.

A atividade de identificação de riscos é realizada com base nos itens de teste e produz como resultado um lista de riscos associados aos requisitos a serem testados. A partir da análise qualitativa e/ou quantitativa, os riscos são priorizados pelo grau de exposição ao risco. Em seguida, estratégias para tratamento e mitigação dos riscos são elaboradas de acordo com a prioridade relacionadas à sua execução, gerando o artefato plano de testes.

Quando os testes ou inspeções são realizados, os riscos são mitigados ou minimizados através da definição de estratégias para que o impacto dos fatores de riscos seja menor. A comunicação dos riscos fornece os dados para definição e acompanhamento dos riscos através de um conjunto de métricas, ajudando no processo de monitoramento e qualidade do processo de testes adotado.

## 2.2 Principais Abordagens

Após visão geral, serão apresentadas de forma resumida as principais abordagens RBT identificadas durante o processo de revisão bibliográfica.

### 2.2.1 Abordagem Baseada em Heurística

Bach [Bach 1999] define uma abordagem baseada em heurística para a atividade de identificação de riscos sugerindo duas formas diferentes e complementares: "*Inside-out*" e "*Outside-in*".

Na "*Inside-out*", o contexto em que o produto está inserido é detalhado e ele é repetidamente questionado sobre o que pode dar errado. O autor sugere três perguntas genéricas para cada componente do software a ser avaliado:

1. Que defeitos ou possíveis falhas existem no componente?

2. Que entradas ou situações podem desencadear o aparecimento de uma vulnerabilidade no componente?

3. Quem será atingido caso o defeito surja e em que grau de severidade?

Na "*Outside-in*" é utilizada uma lista pré-definida de riscos, onde o produto é verificado de acordo com a ocorrência dos itens da lista. Bach utiliza três tipos de listas:

1. **Categorias de critérios de qualidade**: Requisitos do sistema a ser avaliado são agrupados em diferentes categorias de critérios de qualidade. Exemplos de categorias: usabilidade, confiabilidade, desempenho, etc.

2. **Listas genéricas de riscos:** São utilizadas listas de riscos comuns a qualquer sistema. Exemplos de riscos genéricos: complexidade, nova funcionalidade, modificações, dependências, etc.

3. **Catálogo de riscos**: São utilizados catálogos de riscos para um domínio de aplicação específico. Ou seja, os riscos não são mais tão genéricos e variam de acordo com o sistema a ser analisado.

Finalizada a identificação de riscos utilizando uma das duas abordagens explicadas anteriormente, o autor sugere três formas diferentes de comunicar os riscos identificados e organizar a atividade de teste ao redor destes riscos:

1. **Lista de riscos:** Consiste em uma lista de riscos que será revisada periodicamente.

2. **Matriz de Riscos/Atividades:** É definida uma tabela com duas colunas. A primeira coluna apresenta o risco, enquanto que a segunda apresenta atividades que visam mitigar o risco associado.

3. **Matriz de Riscos/Componentes:** É definida uma tabela com três colunas. A primeira coluna apresenta o componente a ser testado, a segunda apresenta o fator de risco, e a terceira apresenta as heurísticas associadas. Como pode ser visualizado na Tabela 2.1, um

componente é qualquer elemento que é passível de teste, enquanto que a heurística indica os riscos importantes naquele componente. O fator de risco é uma avaliação implícita e subjetiva do produto probabilidade x impacto realizada.

**Tabela 2 1.** Matriz de riscos/componentes.

| Componente | Fator de Risco | Heurísticas |
|---|---|---|
| Impressão | Médio | Sistema distribuído, popular |
| Geração de relatórios | Alto | Nova funcionalidade, complexidade, funcionalidade crítica |
| Instalação | Baixo | Popular, usabilidade, modificações |

### 2.2.2 Abordagem Baseada em Métricas

Nesta abordagem, Amland [Amland 1999] propõe métricas de análise de riscos, com o objetivo de auxiliar no processo de Teste de Software. Tais métricas visam gerenciar melhor o processo de teste e agregar valor às decisões tomadas. As métricas propostas pelo autor levam em consideração que funcionalidades complexas, novas, construídas com pouca qualidade e grandes (tamanho) possuem maior probabilidade de apresentar falhas.

Amland propõe uma técnica de priorização das funcionalidades de acordo com o valor da exposição ao risco utilizando a seguinte fórmula, conforme Equação 2.1:

$$\mathrm{Re}(f) = P(f) * \frac{C(c) + C(v)}{2}$$

**Equação 2.1.** Cálculo de exposição ao risco [Amland 1999]

Onde:

Re(f) = valor da exposição ao risco da funcionalidade

P(f) = probabilidade de ocorrência de falha em uma funcionalidade

C(c) = custo da falha do ponto de vista do cliente

C(v) = custo da falha do ponto de vista do fornecedor do serviço

Para o cálculo da probabilidade P(f), inicialmente é calculada a média dos valores atribuídos às métricas: “Nova funcionalidade”, “Projeto/qualidade”, “Tamanho” e “Complexidade”. Cada métrica possui um peso associado, definidos pelo autor. Em seguida, essa média é dividida pelo maior valor possível que o valor da média pode assumir.

**Tabela 2.2.** Cálculo de Exposição ao Risco [Amland 1999]

| Funcionalidade | CUSTO | | PROBABILIDADE | | | | | | RE ( f ) |
|---|---|---|---|---|---|---|---|---|---|
| | C ( v ) | C ( c ) | Nova Peso 5 | Projeto / Qualidade Peso 5 | Tamanho Peso 1 | Complexidade Peso 3 | Média | P(f) | |
| Identificar Riscos Manualmente | 1 | 1 | 1 | 1 | 1 | 2 | 4,25 | 0,4 | 0,4 |
| Criar Plano de Mitigação | 2 | 2 | 2 | 2 | 2 | 2 | 7 | 0,6 | 1,3 |

A Tabela 2.2 exibe um exemplo de como é calculada a exposição ao risco para priorização de funcionalidades. Neste caso a funcionalidade “Criar Plano de Mitigação” apresentou maior valor, e por isso deve ser vista com maior atenção durante as tarefas de testes.

O autor também propõe algumas métricas para controlar o progresso e qualidade do projeto que adotar a técnica RBT. Ele chegou a realizar um estudo de caso em uma aplicação financeira e, segundo ele, os resultados foram satisfatórios, pois o tempo gasto na execução dos testes e o número de recursos utilizado foram menores do que o estimado em uma técnica tradicional de testes.

### 2.2.3 Abordagem para Métricas de Software Orientado a Objetos

Rosenberg e demais autores [Rosenberg et al 1999] fornecem uma abordagem para identificação de classes que estejam mais propensas a erros. Os autores partem do princípio de que quanto mais complexo o código, maior a incidência de erros ou problemas.

Segundo os autores, através da aplicação de métricas de complexidade de software orientado a objetos, pode-se chegar a classes que possuem maior probabilidade de falha. Seis métricas foram identificadas e aplicadas pelo *Software Assurance Technology Center* (SATC) do NASA *Goddard Space Flight Center* durante três anos e, de acordo com os resultados informados, as métricas definidas forneceram ótimo embasamento para o planejamento dos testes.

### 2.2.4 Abordagem para Teste de Regressão

Chen [Chen 2002] propôs um método baseado em especificação para seleção de casos e cenários de teste de regressão baseado em riscos. A autora supõe que os casos de teste já estão prontos.

Para seleção dos casos de teste é necessário seguir os seguintes passos:

1. Estimar o custo de cada caso de teste, utilizando a fórmula proposta por Amland (Seção 2.2.2).

2. Derivar a probabilidade de severidade de cada caso de teste. Este valor é obtido a partir do número de defeitos e suas severidades.

3. Calcular a Exposição ao Risco (RE) de cada caso de teste, obtida através do produto entre o valor do custo e o valor da probabilidade de severidade.

4. Selecionar os casos de teste que tiveram os maiores valores de exposição ao risco.

Para seleção dos cenários baseado em riscos é preciso obedecer duas regras: i) selecionar os cenários que cobrem os casos de teste mais críticos e ii) garantir que cenários vão cobrir um maior número de casos de teste. A seleção dos cenários segue os seguintes passos:

1. Calcular a exposição ao risco para cada cenário.

2. Selecionar os cenários com maior valor da exposição ao risco.

3. Atualizar matriz de rastreabilidade, removendo cenários selecionados e casos de teste cobertos, e recalculando o valor da RE.

4. Repetir os passos 2 e 3 quando necessário.

A autora destaca que foi realizado um estudo de caso em organização industrial e os resultados mostraram que a técnica RBT adotada foi eficiente para encontrar defeitos mais cedo.

### 2.2.5 Abordagem Baseada em Percentual de Uso

Proposta por Besson [Besson 2004], esta abordagem utiliza a informação de percentual de uso da funcionalidade para priorização. O autor sugere uma metodologia simples, baseada na idéia de obter o mínimo de esforço de teste possível que maximiza a redução de riscos.

A metodologia proposta por Besson é dividida em cinco passos:

1. Identificar as funcionalidades “vitais”, que podem prevenir o usuário de usar o software caso o defeito seja encontrado.

2. Projetar casos de teste para as funcionalidades identificadas no passo 1.

3. Estimar o esforço de teste para execução dos casos testes projetados no passo 2.

4. Ordenar os casos de teste em ordem ascendente, onde serão priorizados os que requerem esforço mínimo.

5. Executar os casos de teste de acordo com a ordem definida no passo 4.

### 2.2.6 Abordagem Baseada em Modelos

Stallbaum e demais autores [Stallbaum et al 2008] desenvolveram uma técnica automática para priorização e geração de cenários de casos de testes utilizando diagramas de atividades como modelos de testes. A abordagem proposta,

denominada RiteDAP (*Risk-based test case Derivation And Prioritization*) utiliza modelos de teste acrescidos de informação sobre o risco. O risco neste caso é determinado através da função *R(P,D) = P*D*, onde *P* é a probabilidade de ocorrência de falta na entidade pertencente ao diagrama e *D* é o dano causado por esta falta. Esta informação é acrescentada a cada entidade pertencente ao diagrama de atividades. A Figura 2.2 apresenta um exemplo de diagrama representando um modelo de teste acrescido de informação sobre o risco. Nela percebe-se que, para cada decisão tomada dentro de um caso de teste, representado pelo diagrama, vai existir uma probabilidade P e um dano D associado, resultando no valor do risco R.

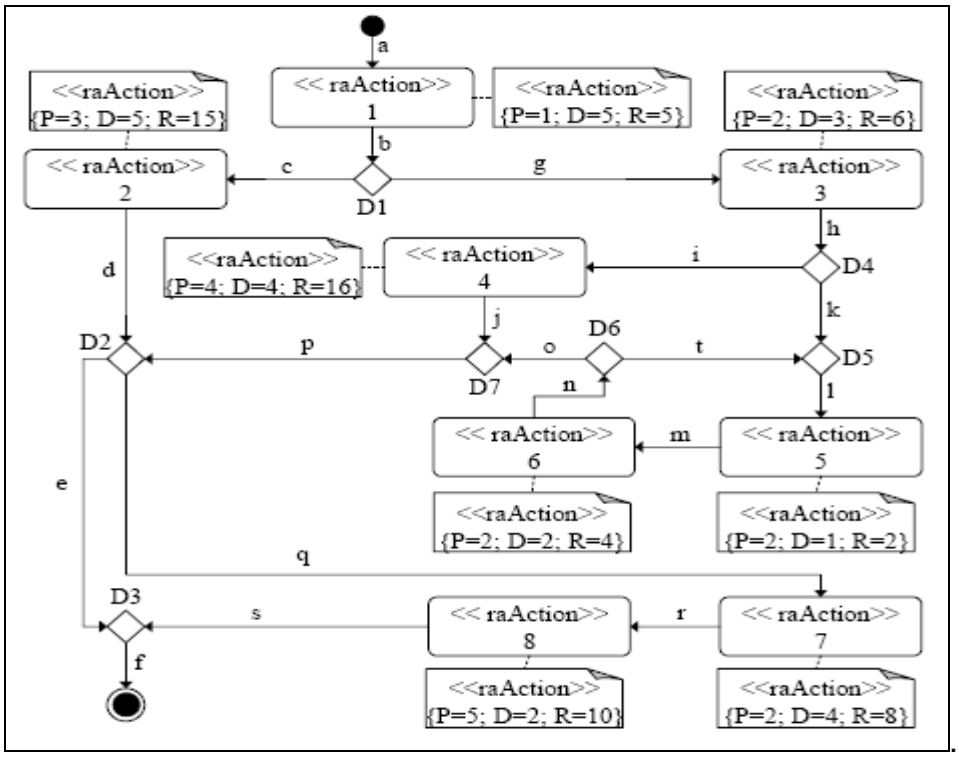



**Figura 2.2.** Exemplo de diagrama de atividades com informação extra do risco

Esta abordagem é dividida em duas atividades: i) Derivar cenários de caso de teste e ii) Priorizar e ordenar os cenários de caso de teste. A primeira atividade não é detalhada pelo autor, pois ele deixa que a equipe de teste selecione a melhor maneira de gerar casos de teste. Já para derivar a ordem de execução dos cenários, inicialmente é realizada a soma dos valores dos riscos de cada diagrama que representa cada cenário. Logo depois a priorização é efetivamente realizada através de uma das duas estratégias sugeridas pelo autor: i) uma estratégia é ordenar os

cenários de acordo com o valor da soma dos riscos; ii) a outra estratégia é executar apenas os cenários com riscos que ainda não foram cobertos pelos cenários anteriores.

### 2.2.7 RBT*Process*

Proposto por Souza, em sua dissertação de Mestrado em Engenharia da Computação – UPE, o *RBTProcess* é um Modelo de Processo de Teste de Software baseado em Riscos. Ele é baseado no RUP, iterativo, orientado a riscos, modelado no nível "M1" do *Software Process Engineering Metamodel* (SPEM[1]) e descrito, utilizando o *Eclipse Process Framework* (EPF[2]) [Souza 2008].

O RBT*Process* foi motivado pela ausência de processos com atividades, papéis e artefatos que possam guiar os engenheiros de teste no uso da técnica RBT.

O *RBTProcess* possui quatro fases distintas juntamente com seus marcos e atividades, como pode ser visto na Figura 2.3 : **i. Planejamento:** Consiste no planejamento inicial dos testes com base na análise dos riscos. Tem como marco a priorização dos requisitos através da identificação e análise dos riscos; **ii. Projeto:** Os casos de testes planejados são projetados com base na análise dos riscos. O marco desta fase é a criação de casos de testes que verificam a existência ou não dos riscos identificados; **iii. Execução:** Os casos de testes planejados e projetados são executados segundo o cálculo de exposição ao risco e **iv. Controle:** Os resultados da execução dos testes são coletados, avaliados e controlados.

Mais detalhes sobre o RBT*Process* estão disponíveis no endereço http://dsc.upe.br/~eprs/rbt/process/.

[1] SPEM na web: www.omg.org/technology/documents/formal/spem.htm

[2] EPF na web: www.eclipse.org/epf/

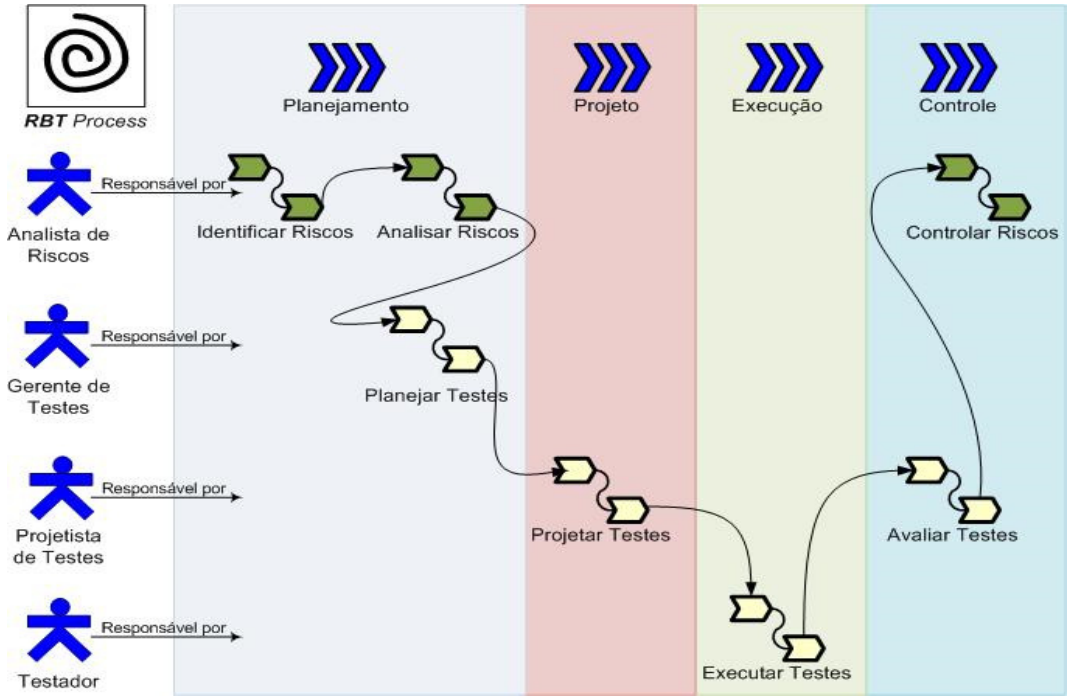



**Figura 2.3**. Fases e atividades do RBT*Process*

## 2.3 Estudo Analítico

Após o estudo detalhado de cada abordagem (seções 2.2.1 a 2.2.7), foi possível realizar um estudo comparativo com o objetivo de identificar pontos positivos e pontos negativos de cada abordagem. Este estudo analítico é resultado do aprofundamento de outro estudo realizado por Souza e demais autores [Souza et al 2008].

Bach define uma abordagem baseada em heurística para a atividade de identificação, análise e controle de riscos [Bach 1999]. No entanto, o autor não fornece nenhuma indicação de como os casos de teste são gerados.

Amland propõe métricas de análise de riscos, com o objetivo de auxiliar no processo de Teste de Software [Amland 1999]. Tais métricas visam gerenciar melhor o processo de teste e agregar valor às decisões tomadas. As métricas propostas pelo autor levam em consideração que funcionalidades complexas, novas, construídas com pouca qualidade e grandes (tamanho) possuem maior probabilidade de apresentar falhas. Também, o autor não fornece nenhuma

orientação sobre a criação dos casos de testes, focando apenas na análise de riscos.

Rosenberg e demais autores fornecem uma abordagem para identificação de classes que estejam mais propensas a erros [Rosenberg et al 1999]. Através da aplicação de métricas de definição da complexidade de software orientado a objetos, pode-se chegar a classes que possuem maior probabilidade de falha. No entanto, os autores não propõem estratégias de testes para o código tampouco formas de rastreamento das funcionalidades impactadas por classes com alto risco de falhas.

Chen propôs um método baseado em especificação para seleção de casos e cenários de teste de regressão baseado em riscos [Chen 2002]. Esta abordagem se mostra mais viável quando há poucos casos de teste ou pode ser realizada de forma automática. Contudo, a autora não faz menção à atividade de identificação de riscos e assume que os casos de teste já estejam prontos.

Besson utiliza a informação de percentual de uso da funcionalidade para priorização [Besson 2004]. No entanto, não há garantia de que as funcionalidades que estão sendo atacadas são as mais propensas a apresentarem falhas e precisam ser mais bem testadas.

Stallbaum e demais autores desenvolveram uma técnica automática para priorização e geração de cenários de casos de testes utilizando diagramas de atividades como modelos de testes [Stallbaum et al 2008]. A análise de risco é bastante subjetiva, levando em consideração apenas a probabilidade de ocorrência e o dano. Dependendo do tamanho dos modelos e do número de funcionalidade do software, a análise de cada transição do diagrama pode levar muito tempo ou mesmo tornar-se inviável. Também não é feita nenhuma menção à atividade de identificação de riscos.

Souza propõe um Modelo de Processo de Teste de Software baseado em Riscos – RBT*Process* - [Souza 2008]. Seu trabalho inclusive envolveu o estudo das abordagens apresentadas anteriormente, e sua contribuição maior foi fornecer uma abordagem mais RBT completa, de forma que atenda às principais atividades, que são Identificação de Riscos, Análise de Riscos, Planejamento de Testes, Projeto de

Testes, Execução de Testes, Avaliação de Testes e Controle de Riscos. Apesar da de abranger a todas estas atividades, o RBT*Process* ainda não foi aplicado por completo. Os estudos de caso realizados pela autora focaram mais nas atividades de identificação e análise de riscos, enquanto que atividades de melhoria contínua das atividades de teste com base no controle e monitoramento dos riscos não foram realizadas. Desta forma, é necessária a aplicação do processo de maneira mais aprofundada para verificar o real impacto da sua adoção pelas organizações.

O estudo analítico foi bastante importante no levantamento de requisitos para a RBT*Tool*, pois permitiu identificar pontos em comuns entre as diversas abordagens estudadas, bem como auxiliou bastante na definição das iterações no processo de desenvolvimento e priorização de funcionalidades. Inclusive, serve de referência para trabalhos futuros na implementação de novas funcionalidades na RBT*Tool*.

A Tabela 2.3 apresenta, de forma sintetizada, o estudo analítico das principais abordagens apresentadas. Para o estudo, foram listadas atividades da Gerência de Riscos de software e atividades mínimas de Teste de Software existentes em qualquer processo.

**Tabela 2.3.** Análise comparativa das principais abordagens RBT [Souza 2008].

| ABORDAGEM | IDENTIFICAR RISCOS | ANALISAR RISCOS | PLANEJAR TESTES | PROJETAR TESTES | EXECUTAR TESTES | AVALIAR TESTES | CONTROLAR RISCOS |
|---|---|---|---|---|---|---|---|
| **Baseada em Heurística [Bach 1999]** | Listas de critérios de qualidade, genéricas e catálogo de riscos | Equação de Barry Boehm | Inexistente | Não fornece detalhes | Matriz de rastreabilidade | Inexistente | Não fornece detalhes |
| **Baseada em Métricas [Amland 1999]** | Inexistente | Métricas específicas para Teste de Software | Inexistente | Inexistente | Ordem de exposição ao risco | Inexistente | Fornece um conjunto de métricas para controle de progresso dos testes |
| **Baseada em Uso [Besson 2004]** | Inexistente | Utiliza a informação de percentual de uso | Inexistente | Inexistente | Percentual de uso da funcionalidade e tempo de execução do caso de teste | Inexistente | Não fornece detalhes |
| **Testes de Regressão [Chen 2002]** | Inexistente | Métricas propostas por Amland | Não fornece detalhes | Leva em consideração que os casos de teste estão prontos | Ordem de exposição ao risco | Inexistente | Não fornece detalhes |
| **Código Fonte Orientado a Objetos [Rosenberg et al 1999]** | Inexistente | Métrica para código fonte OO | Inexistente | Inexistente | Inexistente | Inexistente | Inexistente |
| **Baseada em Modelo [Stallbaum et al 208]** | Inexistente | Análise de probabilidade e dano | Inexistente | Deriva testes a partir do modelo | Ordem de exposição ao risco | Inexistente | Não fornece detalhes |
| **RBT*Process* [Souza 2008]** | Lista de Riscos, Questionário e *Brainstorming* | Métricas propostas por Amland | Priorização com base na análise de riscos | Mitiga os riscos identificados | Ordem de exposição ao risco | Gera relatórios das execuções | Métricas para controle de progresso dos testes |

## 2.4 Resumo do Capítulo

Este capítulo faz uma revisão geral sobre a técnica de teste baseado em riscos. Inicialmente foi introduzida a idéia da técnica RBT através da apresentação de conceitos básicos e atividades contempladas por esta técnica de uma forma geral.

Logo depois foram apresentadas as principais abordagens de maneira resumida. Para cada abordagem foram enumeradas atividades identificadas e a partir disto, foi elaborado um estudo analítico com base nas abordagens estudadas. O objetivo principal deste estudo analítico foi identificar pontos positivos e negativos de cada uma.

# Capítulo 3

# RBT*Tool*

Gerência de Riscos é um processo crítico em ambientes de desenvolvimento de software, um dos fatores decisivos para que os gerentes e membros de equipes alcancem seus objetivos na execução de um projeto. Ao disponibilizar estratégias dos riscos levando em conta a relação custo-benefício e também favorecer o compartilhamento das informações geradas, as atividades de Teste baseados em Riscos permitem a tomada de decisão mais bem fundamentada. Da mesma maneira, a automação das atividades de Teste de Software permite trazer diversos benefícios, entre os quais se destacam: **i)** produção de um sistema mais confiável, **ii)** melhoria da qualidade dos esforços de teste e **iii)** redução do esforço de teste e minimização do cronograma [Dustin et al 1999]. Baseada nestas premissas, foi concebida a RBT*Tool*[3].

Neste contexto, este capítulo introduz a RBT*Tool* na Seção 3.1. Em seguida, na Seção 3.2, são apresentadas as principais funcionalidades. Na Seção 3.3, é exposto o projeto de software abordado para sua concepção, detalhando tecnologias utilizadas e modelo arquitetural. E finalmente, na Seção 3.4, é apresentado o ambiente de desenvolvimento, destacando material e método utilizados no desenvolvimento da RBT*Tool*.

## 3.1 Introdução

A RBT*Tool* é uma aplicação *desktop*[4], de código aberto, que está em processo de desenvolvimento, com o objetivo de fornecer suporte às atividades da abordagem de

[3] RBT*Tool* na web: http://pma.dsc.upe.br/rbttool

[4] Aplicação *desktop* é toda aquela que foi projetada para funcionar localmente. Logo, não é utilizada a arquitetura cliente-servidor. Sua vantagem é a facilidade de instalação e por permitir uma interface gráfica mais atrativa.

Teste baseado em Riscos, auxiliando os engenheiros de teste especialmente nas atividades relacionadas ao gerenciamento de riscos, envolvendo as etapas de identificação, análise e controle de riscos técnicos associados aos requisitos de software.

A RBT*Tool* visa fornecer maior qualidade e produtividade na atividade de Teste de software, mais precisamente, na atividade de Teste baseado em Riscos, através da automação de tarefas repetitivas, diminuindo a possibilidade de inconsistência e erros de resultados, favorecendo uma execução sistemática e contínua.

A proposta da RBT*Tool* é pertinente pelo fato de ainda existirem lacunas no uso de ferramentas para automação de casos de teste baseado em riscos. Também, é importante salientar que, na literatura estudada, não foram encontradas referências sobre ferramentas de suporte à abordagem RBT. Apenas Jørgensen [Jørgensen 2005] desenvolveu um protótipo contendo algumas funcionalidades, porém não foram contemplados requisitos relacionados às atividades de identificação e análise de riscos de requisitos, planejamento e projeto de teste, importantes para o monitoramento, controle e progresso dos testes. Além disso, o protótipo desenvolvido foi descontinuado e visava dar suporte a apenas uma abordagem RBT, proposta pelo próprio autor.

## 3.2 Funcionalidades

A RBT*Tool* está dividida em dois grandes módulos: o módulo de apoio às atividades da Gerência de Riscos, que compreende as atividade de identificação, análise e controle dos riscos, e o módulo de apoio ao processo de Teste de Software que compreende as atividades de planejamento, projeto, execução e avaliação dos testes.

O módulo de apoio a Gerência de Riscos foi priorizado, pois é o que fornece um maior ganho para atividade RBT, contemplando atividades que não são de domínio dos engenheiros de teste e que não são apoiadas por ferramenta alguma. A

Figura 3.1 exibe diagrama contendo os principais casos de uso para cada um dos módulos apresentados e que farão parte da segunda versão da RBT*Tool*.

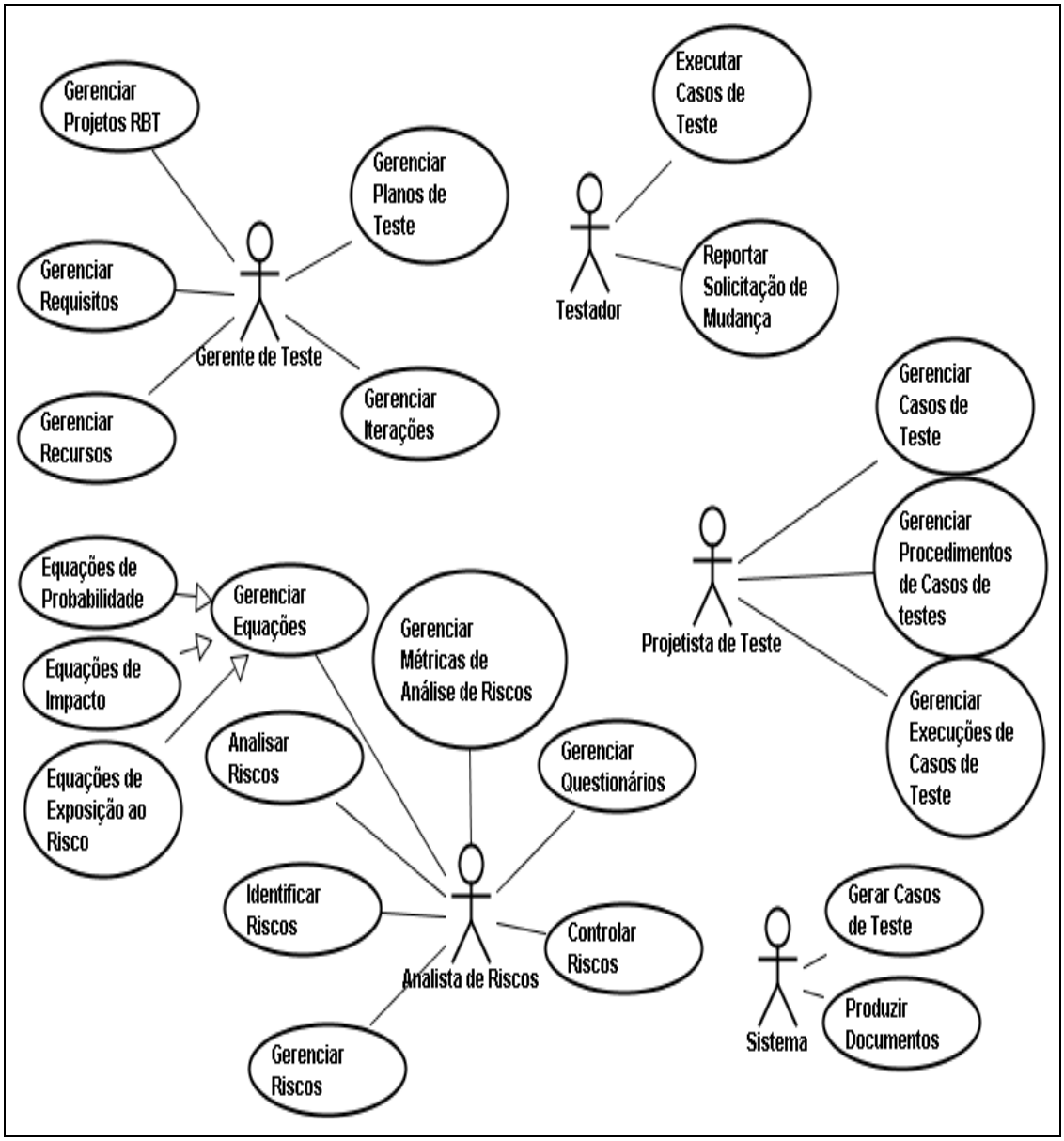



**Figura 3.1.** Diagrama de Casos de Uso da RBT*Tool*

Algumas telas da RBT*Tool* exemplificando as principais funcionalidades são apresentadas no Apêndice A.

Como já citado anteriormente (seção 1.1), a primeira versão da RBT*Tool* foi iniciada por Oliveira [Oliveira 2008], e seu trabalho estava relacionado à atividade de identificação de riscos. Para facilitar a delimitação do escopo deste trabalho em relação ao trabalho de Oliveira, a Tabela 3.1. exibe as funcionalidades

implementadas na versão atual da RBT*Tool*, destacando as que não sofreram modificações, as que sofreram modificações e as adicionadas. Lembrando que foram consideradas na tabela 3.1 as modificações do ponto de vista funcional.

**Tabela 3.1**. Delimitação entre o escopo deste trabalho e o de Oliveira [Oliveira 2008]

| **Funcionalidade** | **Descrição** | **Status** |
|---|---|---|
| Gerenciar Projetos RBT | Torna possível a inclusão, exclusão e alteração dos projetos de RBT. | Não-modificado |
| Gerenciar Requisitos | Torna possível a inclusão e exclusão dos requisitos pertencentes a um projeto. | Não-modificado |
| Gerenciar Questionários | Permite a inclusão, exclusão e alteração de questionários que podem ser utilizados na etapa de itentificação de riscos. | Não-modificado |
| Gerenciar Riscos | Torna possível a inclusão, exclusão e alteração de riscos para cada requisito. | Modificado |
| Identificar Riscos | A identificação de riscos é feita através da resolução de questionários pelos usuários, utilizando a técnica de Questionário baseado em Taxonomias de Riscos [Carr et al 1993]. | Modificado |
| Analisar Riscos | Permite a priorização dos requisitos. | Adicionada |
| Gerenciar Planos de Teste | Permite a inclusão, exclusão e alteração de planos de teste. | Adicionada |
| Gerenciar Casos de Teste | Permite a inclusão, exclusão e alteração de casos de teste com base nos riscos identificados. | Adicionada |

A seguir serão detalhadas as funcionalidades que sofreram modificações e as que foram adicionadas. Os detalhes e descrição dos demais casos de uso estão disponíveis na *web*[5].

[5] Documentação de requisitos disponível em: http://pma.dsc.upe.br/rbttool/requirements/

**Gerenciar Riscos:** Na versão anterior, o risco era fixo, sendo definido unicamente pelo atributo de risco, definido pela taxonomia de riscos proposta pelo SEI (*Software Engineering Institute*) para os riscos de requisitos da Engenharia de Produto [Car et al 1993]. Não era permitida a inclusão ou exclusão de riscos.

Na atual versão o risco é composto por dois elementos: i) atributo de risco e ii) justificativa para a existência do risco. Esta modificação foi necessária porque somente o atributo de risco sendo ligado ao requisito pode ser insuficiente e vago para definir estratégias de geração de casos de teste. Esta modificação permite também que o mesmo atributo de risco apareça mais de uma vez em cada requisito, e a justificativa da presença do risco consegue fundamentar e especificar melhor a ocorrência de um risco.

**Identificar Riscos:** A identificação de riscos é feita através da resolução de questionários pelos usuários, utilizando a técnica de Questionário baseado em Taxonomias de Riscos (TBQ - *Taxonomy Based Questionnaire*) [Carr et al 1993]. O seu preenchimento consiste de respostas binárias, sim ou não, e através destas é possível saber qual risco atinge um determinado requisito. Foi abordado na ferramenta o uso de questionários para identificar riscos por ser uma técnica que não requer conhecimento prévio sobre Gerência de Riscos por parte de quem responde o questionário.

A esta funcionalidade foi adicionada a possibilidade do usuário justificar a sua resposta, caso o risco atinja um determinado requisito. Esta informação será bastante importante no momento da geração de casos de teste.

A RBT*Tool* também permite a importação/exportação das respostas do questionário no formato XML [XML 2009].

**Analisar Riscos:** Utilizando métricas e equações pré-definidas para cálculo de exposição ao risco os participantes da análise preenchem os valores das métricas. Quando concluídos, estes valores são processados e sumarizados resultando na lista de priorização de requisitos.

As fórmulas utilizadas pela ferramenta para priorização dos riscos foram as propostas por Amland [Amland 1999] (seção 2.2.2). Além das métricas definidas por

Amland, foi utilizada mais uma chamada Dependência, para que o cálculo de exposição ao risco alcançasse maior confiabilidade [Souza 2008].

A RBT*Tool* também permite a importação/exportação dos valores da exposição ao risco de cada funcionalidade no formato XML [XML 2009].

**Gerenciar Planos de Teste:** Permite a inclusão, exclusão e alteração de planos de teste. Através dos planos de teste são definidas as iterações de testes com base no valor da exposição ao risco de cada requisito, calculado na fase de análise de riscos. Para exemplificar melhor a atividade de definição de plano de teste, observa-se que na Tabela 3.2 foram definidas três iterações, onde cada uma destas possui um conjunto de funcionalidades definido por um intervalo de valores da exposição ao risco de cada uma. A Iteração I, por exemplo, contempla as funcionalidades no intervalo de valores entre 1,74 e 1,21. Esta funcionalidade seguiu a idéia de criação de plano de testes definida por Souza no RBT*Process* [Souza 2008].

**Tabela 3.2**. Exemplo de conjunto de iterações definido no plano de teste

| | **Funcionalidades** | **Valor da Exposição ao Risco** |
|---|---|---|
| Iteração I | FUNC10 | 1.74 |
| | FUNC02 | 1.64 |
| | FUNC11 | 1,44 |
| | FUNC07 | 1.21 |
| Iteração II | FUNC05 | 1.10 |
| | FUNC04 | 1.10 |
| | FUNC03 | 1,04 |
| Iteração III | FUNC09 | 0,84 |
| | FUNC08 | 0,81 |
| | FUNC06 | 0,75 |
| | FUNC01 | 0,65 |

**Gerenciar Casos de Teste:** Permite a inclusão, exclusão e alteração de casos de teste com base nos riscos identificados. Uma parte da geração de casos de teste é de forma automática através dos resultados gerados na etapa de identificação de riscos. A proposta de geração de caso de teste foi baseada em

[Souza 2008]. Para esclarecer melhor a forma de geração automática de casos de teste, suponha o seguinte exemplo:

"Foi identificado um risco para a funcionalidade "Gerenciar projeto". Este risco estava ligado ao atributo de estabilidade e a justificativa fornecida era: "Módulo de Edição Sofreu Muitas Modificações"."

Desta forma, o caso de teste a ser gerado terá as informações descritas na Figura 3.2.

| | |
|---|---|
| Identificador: CT001_NOME_GerenciarProjeto_Estabilidade | |
| Descrição: **MÓDULO DE EDIÇÃO SOFREU MUITAS MODIFICAÇÕES** | |
| Requisito Associado:<br><br>**RF002 – GERENCIAR PROJETO** | Atributo de Risco Associado:<br><br>**ESTABILIDADE** |
| Pré-Condições:<br><br>• | Pós-Condições:<br><br>• |

**Figura 3.2.** Modelo de Caso de Teste a ser gerado pela RBT*Tool*.

Os campos em negrito são gerados automaticamente pela RBT*Tool*. A ferramenta faz um mapeamento entre os riscos identificados para cada requisito e gera os casos de teste. Percebe-se que para cada risco ligado a um requisito vai haver um caso de teste com o objetivo de mitigar, ou pelo menos minimizar, aquele risco. Os campos em branco deverão ser depois preenchidos pelo usuário.

## 3.3 Projeto de Software

O projeto da RBT*Tool* é totalmente amparado por tecnologias livres e de código aberto, com a finalidade de permitir uma maior apadtabilidade e portabilidade por parte dos usuários, principalmente em ambientes organizacionais, permitindo também a evolução da própria ferramenta. As seções seguintes apresentam de maneira resumida o modelo arquitetural e as tecnologias adotadas. Mais detalhes sobre o projeto da RBT*Tool* podem ser encontrados em [Oliveira 2008].

### 3.3.1. Arquitetura

A arquitetura da RBT*Tool* foi baseada no padrão de projeto arquitetural Camadas (*Layers*) [Layers and Packages 2009], estando dividida em três camadas: **i**. camada de apresentação (interface gráfica), **ii**. camada de negócio (fachada, regras de negócio e classes básicas) e **iii**. camada de persistência (repositórios).

A implementação dessa arquitetura foi bastante útil, pois facilita tanto a inclusão dos módulos subseqüentes, como a manutenção evolutiva da RBT*Tool*.

### 3.3.2. Tecnologias

As seguintes tecnologias foram utilizadas:

**Java para desenvolvimento** – a linguagem Java foi escolhida pelo fato se ser orientada a objetos, livre e independente de plataforma.

**XML para persistência dos dados** – formato padrão recomendado pela W3C[6] (*World Wide Web Consortium*) e se caracteriza pela simplicidade e portabilidade. Para utilização da linguagem XML, foi utilizado o framework XStream[7], o qual é uma biblioteca para serialização de objetos para XML.

## 3.4 Ambiente de Desenvolvimento

O ambiente de desenvolvimento escolhido foi o Eclipse RCP[8], que é uma plataforma projetada para construção de aplicações, utilizando toda a biblioteca de interface gráfica usada pela própria plataforma de desenvolvimento Eclipse, usando a linguagem Java. As aplicações RCP são baseadas na arquitetura de *plug-in* comum do Eclipse, com a ressalva de que funcionam sem a necessidade de instalação da IDE Eclipse. O uso deste ambiente permite que a RBT*Tool* funcione em diferentes

[6] Disponível em: http://www.w3.org/XML/

[7] Disponível em: http://xstream.codehaus.org/

[8] Disponível em: http://wiki.eclipse.org/index.php/Rich_Client_Platform

sistemas operacionais, sem haver a necessidade de instalação de componentes adicionais.

**Ferramentas CASE** - Algumas ferramentas CASE (*Computer-aided Software Engineering*) foram bastante úteis no auxílio ao desenvolvimento do projeto da RBT*Tool*. Para levantamento e documentação dos requisitos foi utilizada a ferramenta **REM** [Duran 2000], enquanto que para modelagem e criação de diagramas de caso de uso foi utilizado o **JUDE**[9].

**Processo de Desenvolvimento** - O processo de desenvolvimento adotado é baseado no RUP – *Rational Unified Process* [Kruchten 2003], sendo iterativo, incremental e dirigido a caso de uso. A cada iteração, os requisitos são detalhados e revisados, são criados diagramas de análise e projeto e os casos de uso são implementados e testados.

**Processo de Gerenciamento de Riscos** - Para o gerenciamento dos riscos, durante o desenvolvimento da RBT*Tool*, foi adotado o processo Ágil de Gestão de Riscos de Ambientes – o GARA [Ribeiro e Gusmão 2008].

## 3.5 Resumo do Capítulo

Este capítulo tratou de apresentar a RBT*Tool*. Foram apresentadas as melhorias e evolução da ferramenta através das funcionalidades adicionadas e/ou modificadas para este trabalho. Também foi apresentado o projeto de software adotado, enumerando tecnologias, modelo arquitetural, bem como o ambiente de desenvolvimento adotado, contendo ferramentas CASE, processo de desenvolvimento e processo de gerenciamento de riscos.

[9] Disponível em: http://jude.change-vision.com

# Capítulo 4

# Avaliação da RBT*Tool*

Após implementação das propostas de melhorias e evolução apresentadas no Capítulo 3, a RBT*Tool* foi avaliada por especialista em Gerência de Riscos e através de um estudo experimental. Ambas as avaliações foram planejadas e executadas a fim de verificar se a ferramenta atende corretamente aos requisitos funcionais e para identificar possíveis melhorias e mudanças no escopo do projeto. Neste capítulo, serão apresentadas as características dos estudos empíricos e resultados alcançados.

## 4.1 Avaliação por Especialista em Gerência de Riscos

Nesta avaliação foi utilizada uma instância do processo de Avaliação por Especialista em Gerência de Riscos, que por sua vez é baseado no LAPS[10] - Laboratório de Avaliação de Produtos de Software. O LAPS desenvolveu um formato de avaliação que propõe integrar os aspectos técnicos com especialistas nas áreas de domínio do software, que irão identificar e definir as necessidades fundamentais do usuário.

### 4.1.1 O processo de Avaliação por Especialista em Gerência de Riscos

O processo em questão foi modelado por uma aluna de graduação, como proposta de Trabalho de Conclusão do Curso de Engenharia da Computação

[10] LAPS na web: http://www.cin.ufpe.br/~laps/

(POLI/UPE) [Silva 2009a], sendo dividido em três atividades, como pode ser visto na Figura 4.1.

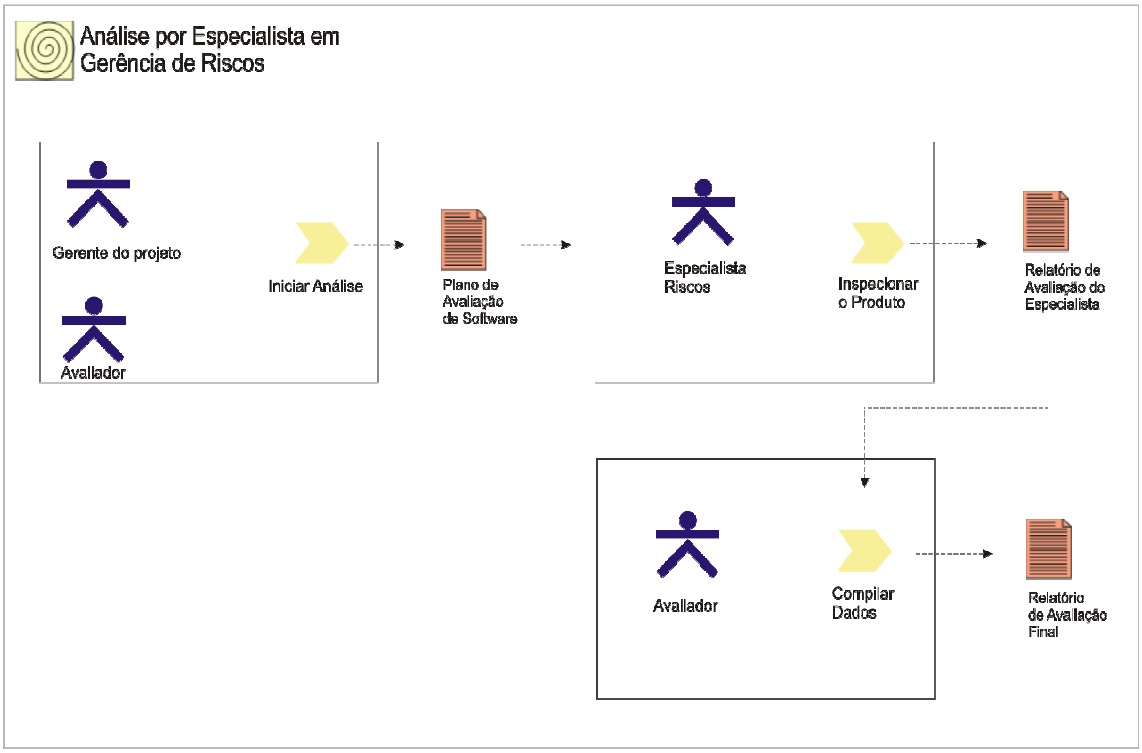



**Figura 4.1.** Fluxo de Atividade por Especialista em Gerência de Riscos

**Iniciar Análise:** Esta atividade tem o objetivo de investigar a documentação do sistema a ser avaliado, fornecida pelo Gerente do Projeto a ser avaliado, dando subsídios para que o avaliador obtenha informações necessárias para o início da avaliação do produto gerando o documento de plano de avaliação de software, como também entrar em contato com o especialista em Gerência de Riscos para execução do processo.

**Inspecionar o Produto:** O objetivo desta atividade é realizar a análise do produto que será avaliado a partir do programa executável, gerando um relatório de avaliação do especialista de Gerência de Riscos.

**Compilar Dados:** Nesta atividade os dados são compilados pelo avaliador a fim de produzir o artefato Relatório de Avaliação Final.

### 4.1.2 Características da Avaliação

O objetivo principal desta avaliação foi verificar se a ferramenta realmente atendia às expectativas de um especialista em Gerência de Riscos de requisitos de software.

A avaliação contou com a participação de três voluntários, sendo dois discentes da graduação e um docente do curso de Engenharia da Computação, todos da Universidade de Pernambuco – UPE. O docente em questão tem sua área de pesquisa fortemente ligada à Gerência de Riscos, logo assumiu o papel do Especialista em Gerência de Riscos. Com relação aos alunos, um assumiu o papel de Avaliador e outro de Gerente de Projeto da RBT*Tool*.

Também foi elaborado um pequeno roteiro de execução da RBT*Tool* a fim de auxiliar ao especialista em riscos no uso da ferramenta.

### 4.1.3 Resultados Obtidos

Com relação a regras de negócio ausentes, não foram encontradas não-conformidades na RBT*Tool*. E do ponto de vista do especialista em riscos, a utilização das funcionalidades foi considerada objetiva para o uso de um conhecedor da área de gerenciamento de riscos.

Para um aplicativo que se propõe a auxiliar o testador na definição dos principais requisitos a serem tratados, inicialmente, as funcionalidades estão dentro do escopo especificado. No entanto, o gerenciamento de riscos, como a área de medidas de software – garantia da qualidade de software, deve fazer uso de técnicas e procedimentos múltiplos de gerenciamento de riscos, como forma de minimizar possíveis erros ao longo das atividades previstas.

Também, foram verificadas a existência de regras de negócio implementadas incorretamente e fora do escopo, como também, falhas de interoperabilidade. No entanto, não foram constatadas funcionalidades incorretas, além de todas as funcionalidades implementadas estarem dentro do escopo a que se propõem.

Verificando as funcionalidades necessárias inexistentes, ainda não está implementada a facilidade de uso direto da identificação e análise de riscos sem a

necessidade da importação/exportação dos arquivos XML. Também houve a ocorrência de funcionalidades inacuradas em que foi mencionada uma não-conformidade durante a execução do cenário de teste relacionado, mas que não impactou no resultado final esperado.

Esta avaliação foi de suma importância para identificação de pontos de melhorias em relação às funcionalidades relacionadas à Gerência de Riscos. Algumas idéias foram identificadas e podem ajudar a evoluir o conceito e aplicação das atividades de gerenciamento de riscos na RBT*Tool* :

- Identificação de Riscos – incremento de técnicas que auxiliem a identificação dos riscos nos requisitos, pois riscos podem passar despercebidos;
- Análise e priorização dos riscos – não ficou claro através do uso da RBT*Tool* como, baseados na exposição ao risco, os riscos seriam priorizados na geração dos casos de testes, pois apenas os requisitos são priorizados;
- Avaliação completa dos riscos – disponibilizar as estratégias de tratamentos dos fatores de riscos identificados;
- Visualizar o histórico do risco ao longo do desenvolvimento do produto de software e seu respectivo conjunto de riscos identificados/analisados, casos de testes gerados/executados e estratégias utilizadas;
- Permitir a visualização do risco efetivado;
- Gerar relatórios analíticos sobre cada projeto executado com visões diversas dos riscos.

## 4.2 Estudo Experimental

A RBT*Tool* também foi avaliada através da aplicação do Modelo de Processo de Teste de Software baseado em Riscos – o RBT*Process* [Souza 2008] (seção 2.2.7) – no projeto *m*PRIME *Tool*[11].

O objetivo do estudo experimental foi verificar contribuição da RBT*Tool* na automação de um processo de Teste de Software baseado em riscos e também identificar seus pontos de melhorias, para que esta atenda melhor às funcionalidades necessárias à aplicação da técnica RBT. Nas seções subsequentes será inicialmente introduzido o projeto *m*PRIME *Tool*. Logo depois será apresentado todo o processo de execução do estudo, incluindo metodologia e material utilizado. Ao final, serão apresentados os resultados do estudo experimental.

### 4.2.1 A *m*PRIME *Tool*

A *m*PRIME *Tool* é uma ferramenta para Gestão de Riscos em Ambientes de Múltiplos Projetos de Desenvolvimento de Software, dando suporte à avaliação, tratamento e controle dos riscos em um ambiente organizacional. Esta ferramenta foi desenvolvida como *add-in* para o *Microsoft Project*. Sua definição teve por base estudos acadêmicos, em nível de doutorado e mestrado, do Centro de Informática da Universidade Federal de Pernambuco – CIn/UFPE.

A metodologia de desenvolvimento utilizada para a construção da *m*PRIME *Tool* foi o XP (*eXtreme Programming*), sendo desenvolvida utilizando a plataforma .NET.

Atualmente, a *m*PRIME *Tool* está sendo remodelada para ser uma aplicação livre e de código aberto, como trabalho da aluna do Mestrado em Ciências da Computação, Centro de Informática (CIn/UFPE) [Silva 2009b]. Este novo projeto se chama Open*m*PRIME e consiste na criação de um *framework* para gerenciamento de riscos em ambientes de múltiplos projetos de *software*. O Open*m*PRIME ainda

[11] *m*PRIME *Tool* na web: http://cin.ufpe.br/~suppera

está em fase de revisão bibliográfica, e, por este motivo, não foi levado em consideração neste estudo experimental.

### 4.2.2 Execução do Estudo Experimental

O estudo experimental inicialmente foi planejado conforme mostra as atividades presentes na Tabela 4.1. Vale destacar que das atividades do RBT*Process*, as atividades Executar Testes, Avaliar Testes e Controlar Riscos não foram executadas, pois a RBT*Tool* ainda não dá suporte a estas atividades.A seguir serão detalhadas as atividades realizadas neste estudo.

**Tabela 4.1.** Atividades realizadas no estudo experimental

| **Atividades** | | **Participantes** | **Observações** |
|---|---|---|---|
| Instanciação do RBT*Process* | - Definir atividades<br>- Definir processo<br>-Seleção de participantes | - Analista de riscos | - Será definido de acordo com a realidade do projeto.<br>- Seleção dos módulos a serem testados<br>- Seleção de recursos de acordo com a disponibilidade<br>- Cadastro do projeto na RBT*Tool* |
| Identificar riscos | - Revisar fontes e categorias de risco | - Analista de riscos | - Montagem do questionário baseado em taxonomia de riscos (TBQ) |
| | -Responder questionário | - Gerente do Projeto | - Cada membro da equipe responderá o questionário individualmente |
| | -Realizar *brainstorming* | - Gerente de Testes<br>- Projetista de Testes<br>- Gerente do Projeto | - Revisão dos riscos identificados |
| Analisar riscos | - Calcular exposição ao risco | - Gerente do Projeto | - Atribuições de valores às métricas<br>- Exportação de arquivo XML |
| | - Priorizar requisitos pela exposição ao risco | - Analista de Riscos | - Fornecida pela RBT*Tool* via importação de arquivo XML |
| Planejar testes | - Definir iterações de testes | - Gerente do Projeto<br>- Gerente de Testes | - Realizada de acordo com os resultados da identificação e análise de riscos.<br>- É definido o número de iterações |
| Projetar testes | - Gerar os casos de teste | - Gerente do Projeto<br>- Projetista de Testes | - Casos de teste gerados pela RBT*Tool* |
| Avaliação da RBT*Tool* | - Responder a questionário | - Gerente do Projeto | - Pequeno questionário para avaliar a aplicação da ferramenta do ponto de vista do usuário. |

**- Instanciação do RBT*Process*:** Escolhido o projeto *m*PRIME *Tool*, foi realizada uma análise do projeto através de conversa com a gerente de projeto e leitura da documentação do software. O objetivo principal desta atividade foi instanciar o RBT*Process* para a realidade do projeto *m*PRME *Tool*. Também foi definido o

cronograma de atividades. O participante selecionado para participar do estudo, de acordo com a disponibilidade foi a gerente do projeto *m*PRIME *Tool*. Logo, o estudo contou com a participação de dois voluntários, e os papéis definidos e utilizados no RBT*Process* foram organizados da seguinte maneira:

- Autor: Analista de riscos, Gerente de teste e Projetista de teste.
- Gerente de projeto: membro do projeto *m*PRIME *Tool*.

Os módulos escolhidos para serem avaliados no processo foram os considerados essenciais na documentação fornecida pela gerente do projeto.

**- Identificar Riscos:** Atividade do RBT*Process* que consistiu em três passos:

1. *Revisar fontes e categorias de riscos:* Obtida através da revisão da documentação do *m*PRIME *Tool*, servindo para montagem do questionário para ser respondido pela gerente de projeto. Após a montagem do questionário, a RBT*Tool* foi disponibilizada para a gerente do projeto poder usá-la no restante do estudo experimental.

2. *Responder questionário:* Atividade realizada pela gerente de projeto, consistiu no uso da funcionalidade Identificar Riscos presente na RBT*Tool* (Seção 3.2). Finalizada a resolução do questionário, foi exportado um arquivo XML contendo as respostas, que foi enviado para o autor por e-mail. Esta atividade teve duração de 5 minutos.

3. *Realizar brainstorming:* Teve como objetivo validar os riscos identificados. Inicialmente foi importado o arquivo XML com as respostas do questionário para a RBT*Tool* poder exibir os riscos identificados por funcionalidade. Durante o *brainstorming*, foi identificado um risco de aderência de uma funcionalidade ao especificado, que tinha passado despercebido quando da resolução do questionário. Esta atividade teve duração de 20 minutos.

**- Analisar Riscos:** Atividade do RBT*Process* que consistiu em dois passos:

1. *Calcular exposição ao risco:* Está relacionada à funcionalidade Analisar Riscos, presente na RBT*Tool* (Seção 3.2). Finalizada esta atividade, foi

exportado um arquivo XML contendo os valores da exposição ao risco de cada requisito calculados, e este arquivo foi enviado ao autor por e-mail. Esta atividade teve duração de 2 minutos.

2. *Priorizar requisitos pela exposição ao risco:* Consistiu na importação do arquivo XML contendo os valores da exposição ao risco de cada requisito calculados. A RBT*Tool* calcula o valor médio da exposição ao risco de cada requisito e o exibe. Existe uma opção na RBT*Tool* para mostrar a priorização dos requisitos por este valor, tanto em ordem ascendente como descendente. Após a importação, o cálculo é realizado de maneira instantânea.

**- Planejar e Projetar Testes:** Estas atividades, também presentes no RBT*Process*, foram realizadas em conjunto, uma vez que quando o plano de teste é criado, os casos de teste são automaticamente gerados. Como os testes não chegaram a ser executados, não houve preocupação com o número de iterações e apenas uma foi definida, abrangendo todas as funcionalidades. Com relação aos casos de teste, como a *m*PRIME *Tool* já havia sido testada e validada anteriormente, apenas seis casos de teste foram gerados. Mesmo assim, a gerente de projeto concordou que a maioria das funcionalidades que tiveram casos de teste eram as mais críticas. Outro fator que merece destaque é o fato de que os requisitos com maior valor da exposição ao risco apresentaram maior quantidade de riscos identificados.

**- Avaliação da RBT*Tool*:** Ao final da aplicação do RBT*Process*, foi enviado um questionário de avaliação da RBT*Tool* para ser preenchida pela gerente do projeto. O objetivo deste questionário foi ter uma visão do usuário sobre a RBT*Tool*, indicando pontos fortes, pontos fracos e contribuição para a área. O modelo de questionário utilizado está disponível no Apêndice B.

### 4.2.3 Resultados Obtidos

O maior ganho foi identificado na questão da manipulação da grande quantidade de dados, especialmente os advindos das atividades de Gerência de Riscos. A consolidação dos dados referentes à identificação e análise de riscos (que consistiu apenas na importação dos arquivos das respostas dos questionários e da análise de riscos no formato XML) para dar suporte à criação do plano de teste e

geração de casos de teste realmente forneceu um indicativo de que o tempo em relação a uma atividade realizada de forma manual – planilhas, por exemplo – foi bastante reduzido. Este ponto reforça bastante a necessidade da automação das atividades RBT, principalmente no momento de geração dos resultados, pois, caso realizado de forma manual, exigiria grande capacidade de concentração dos envolvidos, estando ainda suscetível a erros.

Mesmo com a não execução dos testes, os resultados poderão contribuir na construção do Open*m*PRIME, principalmente nas atividades de Gerência de Riscos realizadas. O resultado final poderá auxiliar na priorização de ações a serem tomadas com base nos riscos identificados e priorizados, deixando os envolvidos no projeto cientes de possíveis eventos adversos que poderão vir a ocorrer neste novo projeto.

Outros pontos positivos também foram identificados pela gerente do projeto *m*PRIME *Tool*:

- O tempo gasto na realização das atividades de Gerência de Riscos foi considerado baixo, não chegando a gerar uma sobrecarga no processo de teste.

- Dado o roteiro de execução fornecido, a RBT*Tool* acabou sendo considerada uma ferramenta de fácil uso e objetiva, com interface agradável, e que atendeu às expectativas geradas.

- Os resultados apresentados nas atividades de identificação e análise de riscos, planejamento e projeto de testes estavam condizentes com a realidade.

- A técnica RBT foi vista como uma forma de auxílio na identificação de possíveis eventos adversos, ajudando, desta forma, a priorizar ações.

Pontos de melhorias também foram identificados:

- Necessidade de implementação de outras técnicas de identificação de risco, pois o questionário pode não abranger todos os riscos, além de que

a resolução de questionários pode se tornar uma tarefa muito repetitiva e cansativa. A metodologia heurística [Bach 1999], pode ser uma boa alternativa de implementação.

- Deixar mais clara para o usuário a opção de priorização dos requisitos pelo valor da exposição ao risco.

- Melhorias em alguns pontos da interface gráfica. A plataforma RCP se mostrou como uma ótima alternativa para desenvolvimento de aplicações *desktop*, contudo a curva de aprendizado inicial para interface gráfica é muito alta, fator que pode ter contribuído para o surgimento desde ponto de melhoria.

## 4.3 Resumo do Capítulo

Este capítulo apresentou duas avaliações realizadas em cima da RBT*Tool*. O objetivo maior das avaliações foi identificar pontos de melhorias para ajustes. A primeira avaliação consistiu na aplicação de um processo de avaliação por especialista em Gerência de Riscos, enquanto que a segunda consistiu em um estudo experimental em cima de um projeto de desenvolvimento de software.

# Capítulo 5

# Considerações Finais

Este capítulo relata as conclusões obtidas neste trabalho. Nas demais sub-seções deste capítulo serão apresentadas as principais contribuições para a comunidade de testes e resultados alcançados (Seção 5.1), trabalhos relacionados (Seção 5.2), dificuldades encontradas (Seção 5.3) e trabalhos futuros (Seção 5.4).

## 5.1 Contribuições

Este trabalho apresentou a RBT*Tool*, uma ferramenta que se propõe a fornecer uma maior qualidade e produtividade aos envolvidos nas atividade do Teste de Software baseado em Riscos, através da automação de tarefas repetitivas, diminuindo a possibilidade de inconsistência nos resultados, falhas humanas e, principalmente, proporcionando maior rapidez na execução de tais tarefas. Foram propostas e implantadas melhorias e evolução de modo que a RBT*Tool* seja mais completa no sentido de auxiliar na execução das atividades RBT.

A avaliação realizada forneceu um indicativo de que a proposta da RBT*Tool* é bastante válida no contexto de Teste baseado em Risco. Entretanto, é necessária a realização de uma avaliação em um projeto de software em desenvolvimento para avaliar o real impacto do uso da abordagem RBT de forma automatizada.

Atualmente, a RBT*Tool* é capaz de auxiliar aos engenheiros de testes nas atividades de Identificação de Riscos, Análise de Riscos, Planejamento de Testes e Projeto de Casos de Teste.

## 5.2 Trabalhos Relacionados

Um dos principais motivos para a concepção da RBT*Tool* foi justamente a carência de ferramentas que dêem suporte a abordagem RBT.

Houve uma tentativa de construção de uma ferramenta de apoio realizada por Jørgensen. O autor levantou requisitos e construiu um protótipo, usando tecnologias *web*, que abrangia atividades de análise de riscos para priorização dos módulos do produto do software considerados mais críticos e planejamento de testes, priorizando os testes mais necessários. Em seu trabalho [Jørgensen 2005], o autor propôs uma abordagem RBT e implementou uma parte dela no protótipo. A versão implementada foi avaliada por um testador e o mesmo apresentou dificuldades na sua utilização, principalmente pela pouca familiaridade com termos utilizados e falta de clareza nas funcionalidades.

O trabalho de Oliveira [Oliveira 2008] foi o ponto de partida para a proposta de melhorias e evoluções apresentadas. Com Oliveira [Oliveira 2008], a RBT*Tool* foi concebida e sua maior contribuição foi fornecer uma técnica de identificação de riscos que não exigisse conhecimento tão aprofundado em Gerência de Riscos, mas mesmo assim permitisse dar suporte à comunidade de testes de software, auxiliando na disseminação da abordagem RBT.

## 5.3 Dificuldades Encontradas

A principal dificuldade encontrada foi durante o processo de desenvolvimento da RBT*Tool*. Foram necessárias mudanças profundas na arquitetura, principalmente nas funcionalidades relacionadas à identificação de riscos, para fornecer uma forma aceitável de geração de casos de teste.

A implementação da interface gráfica também foi uma grande dificuldade, pois o conhecimento da plataforma Eclipse RCP apresenta curva de aprendizado bastante acentuada.

## 5.4 Trabalhos Futuros

Foram identificados alguns trabalhos futuros interessantes para continuidade do projeto da RBT*Tool* e a técnica RBT em geral:

- Suporte a outras abordagens estudadas no Capítulo 2.

- Suporte a outras técnicas de identificação e análise de riscos, tais como listas de verificação e priorização dos riscos via cálculo de impacto e probabilidade.
- Implementação da atividade de controle de riscos através de métricas específicas para a abordagem RBT [Amland 1999; Souza et al 2009].
- Implementação de arquitetura distribuída para que as atividades sejam realizadas de forma *online*.
- Pesquisar e propor outras técnicas de geração automática de casos de teste com base nos riscos.
- Pesquisar e propor outras técnicas de planejamento de testes com base nos riscos.
- Utilização de técnicas de Inteligência Artificial para que a RBT*Tool* para auxiliar nas atividades de identificação e análise de riscos, bem como projeto dos casos de teste.
- Estudar a possibilidade de integração da RBT*Tool* com outras ferramentas de gerência de testes ou de riscos.
- Fornecer um ambiente que torne a RBT*Tool* uma ferramenta de fácil uso, para que não seja mais necessário prover um roteiro externo de execução da mesma.
- Realizar um estudo sobre o modelo arquitetural adotado, a fim de definir qual a melhor arquitetura para a evolução RBT*Tool*.

# Referências Bibliográficas

[Amland 1999] Amland, S. **Risk Based Testing and Metrics: Risk analysis fundamentals and metrics for software testing including a financial application case study**. In: 5o International Conference EuroSTAR'99, 1999.

[Bach 1999] Bach, J. ***James Bach on Risk-Based Testing: How to conduct heuristic risk analysis***. In: Software Testing & Quality Engineering Magazine, p.23-28, 1999.

[Besson 2004] Besson S. ***A Strategy for Risk-Based Testing***. In StickyMinds.com, agosto de 2004.

[Car et al 1993] Carr, M. J., Konda, S.L., Monarch, I., Ulrich, F. C., Walker, C. F. **Taxonomy Based Risk Identification**. Software Engineering Institute, Carnegie Mellon University, EUA, 1993.

[Chen 2002] Chen, Y. **Specification-based Regression Testing Measurement with Risk Analysis**. Dissertação de Mestrado, University of Ottawa/Canada, 2002.

[Duran 2000] Duran, A. **A Methodological Framework for Requirements Engineering of Information Systems,** 2000.

[Dustin et al 1999] Dustin, E. ; Rashka, J. ; Paul, J. **Automated Software Testing : Introduction, Management and Performance**. 1ª Ed., Addison-Wesley, pp 37 – 52, 1999.

[Goldsmith 2006] Goldsmith, R. **Early and Effective: The Perks of Risk-based Testing**. Software Test & Performance Magazine, Volume 3, p.24-30, 2006.

[Jørgensen 2005] Jørgensen L. K. U. **A software tool for risk-based testing**. Dissertação de mestrado, Norwegian University of Science and Technology/Noruega, 2005.

[Karolak 1996] Karolak D. W. **Software Engineering Risk Management**, IEEE Computer Society Press, 1996.

[Kruchten 2003] Kruchten, P. **Introdução ao Rup: Rational Unified Process**. 2ª Ed. Ciência Moderna. São Paulo. pp 25-36, 2003.

[Layers and Packages 2009] **Architecture: Layers and Packages**. Disponível em: <http://www.cs.purdue.edu/homes/apm/courses/BITSC461-fall03/uml-slides/design-4-apm.ppt> Acesso em: março de 2009.

[Oliveira 2008] Oliveira, K. **RBT Tool: Uma Ferramenta para Identificação de Riscos no Teste de Software**. Trabalho de Graduação em Engenharia da Computação, Universidade de Pernambuco/Brasil, 2008.

[Pressman 1995] Pressman, R. **Engenharia de Software**. 1ª Ed. Makron Books. São Paulo, pp 786-793, 1995.

[RBTTool 2008] **RBT*Tool* – Ferramenta de Apoio ao Teste de Software baseado em Riscos**. Disponível em <http://pma.dsc.upe.br/rbttool/requirements/>. Acesso em Abril de 2009.

[Redmill 2004] Redmill, F. **Exploring risk-based testing and its implications**. In: Software Testing, Verification and Reliability, v.14, p.3-15, 2004.

[Ribeiro e Gusmão 2008] Ribeiro, L.; Gusmão, C.; **Definição de um Processo Ágil de Gestão de Riscos em Ambientes de Múltiplos Projetos**. In: Hífen Magazine (Uruguaiana), 2008.

[Rosenberg et al 1999] Rosenberg, L. H.; Stapko, R.; Gallo, A. ***Risk-based object oriented testing*.** In Twenty-Fourth Annual Software Engineering Workshop, NASA SEW24, 1999.

[Silva 2009a] Silva, L. **Processo de Avaliação por Especialista – Um estudo de Avaliação**. Relatório Técnico de Trabalho de Conclusão de Curso em Engenharia da Computação, Universidade de Pernambuco/Brasil, 2009.

[Silva 2009b] Silva, S. **Um Framework para Gerenciamento de Riscos em Ambientes de Múltiplos Projetos de Software**. Proposta de Mestrado em Ciência da Computação, Universidade Federal de Pernambuco/Brasil, 2009.

[Souza 2008] Souza, E. **RBT*Process*: Modelo de Processo de Teste de Software baseado em Riscos**. Dissertação de Mestrado em Engenharia da Computação, Universidade de Pernambuco/Brasil, 2008.

[Souza et al 2008]. Souza, E.; Gusmão, C.; Rocha, H; **RBT*Process* – Proposta de Modelo de Processo de Teste de Software baseado em Riscos**. In: III Encontro Brasileiro de Teste de Software, EBTS, Recife, Pernambuco – Brasil, 2008

[Souza et al 2009] Souza, E.; Gusmão, C.; Alves, K.; Venâncio, J., Melo, R.; **Measurement and Control for Risk-based Test Cases and Activities**. In: 10th IEEE Latin American Test Workshop, LATW, Búzios, Rio de Janeiro – Brasil, 2009.

[Souza e Gusmão 2009] Souza, E.; Gusmão C. **Melhorando a Qualidade do Software e Otimizando Recursos com Teste Baseado em Riscos**. Engenharia de Software Magazine, 10 ed., p.36-44, 2009.

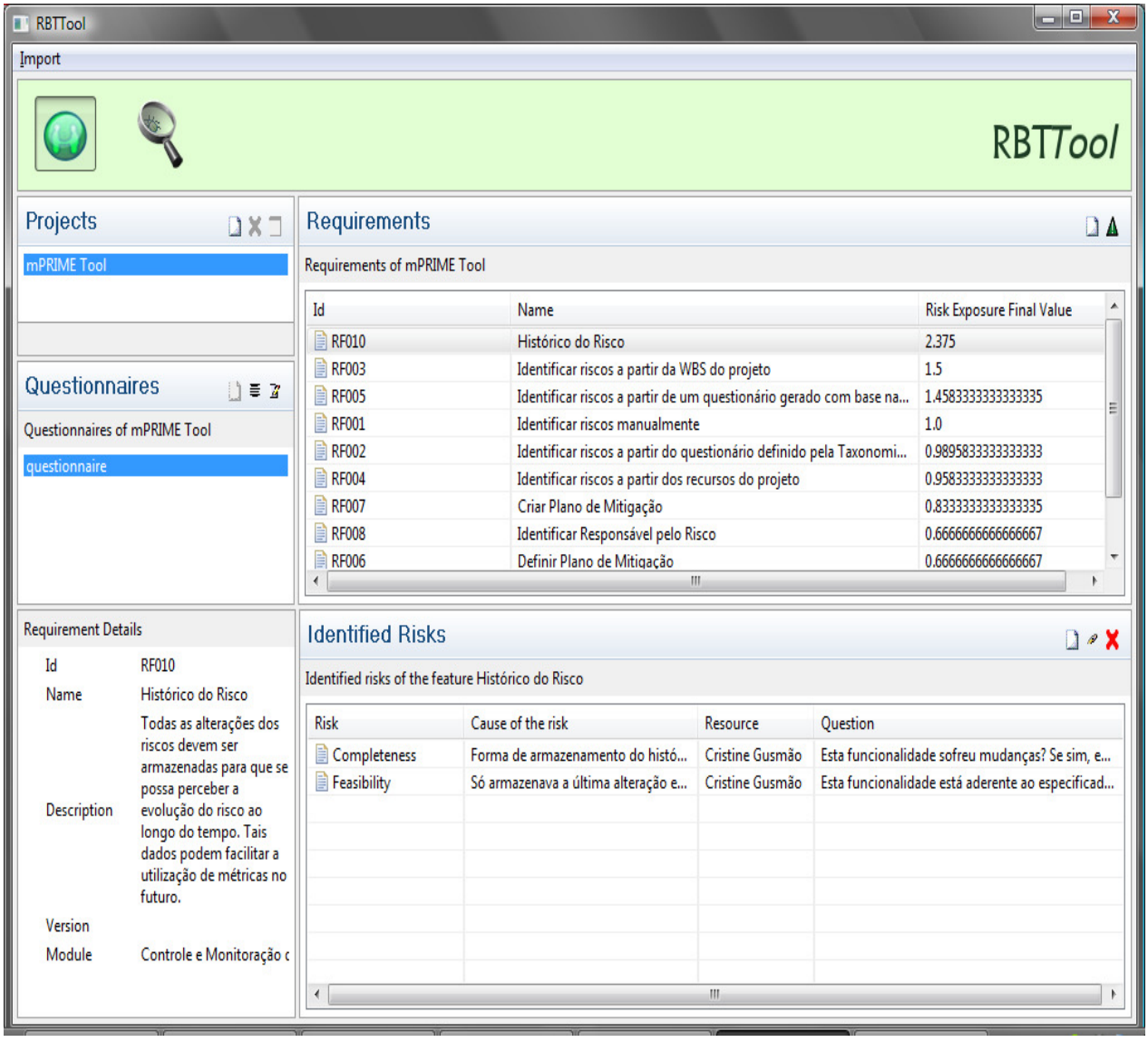
[Stallbaum et al 2008] Stallbaum, H.; Metzger, A.; Pohl, K. **An Automated Technique for Risk-based Test Case Generation and Prioritization**. In: 3rd Workshop on Automation of Software Test, AST, 2008.

[XML 2009] Extensible Markup Language. Disponível em <http://www.w3.org/XML/>. Acesso em Abril de 2009.

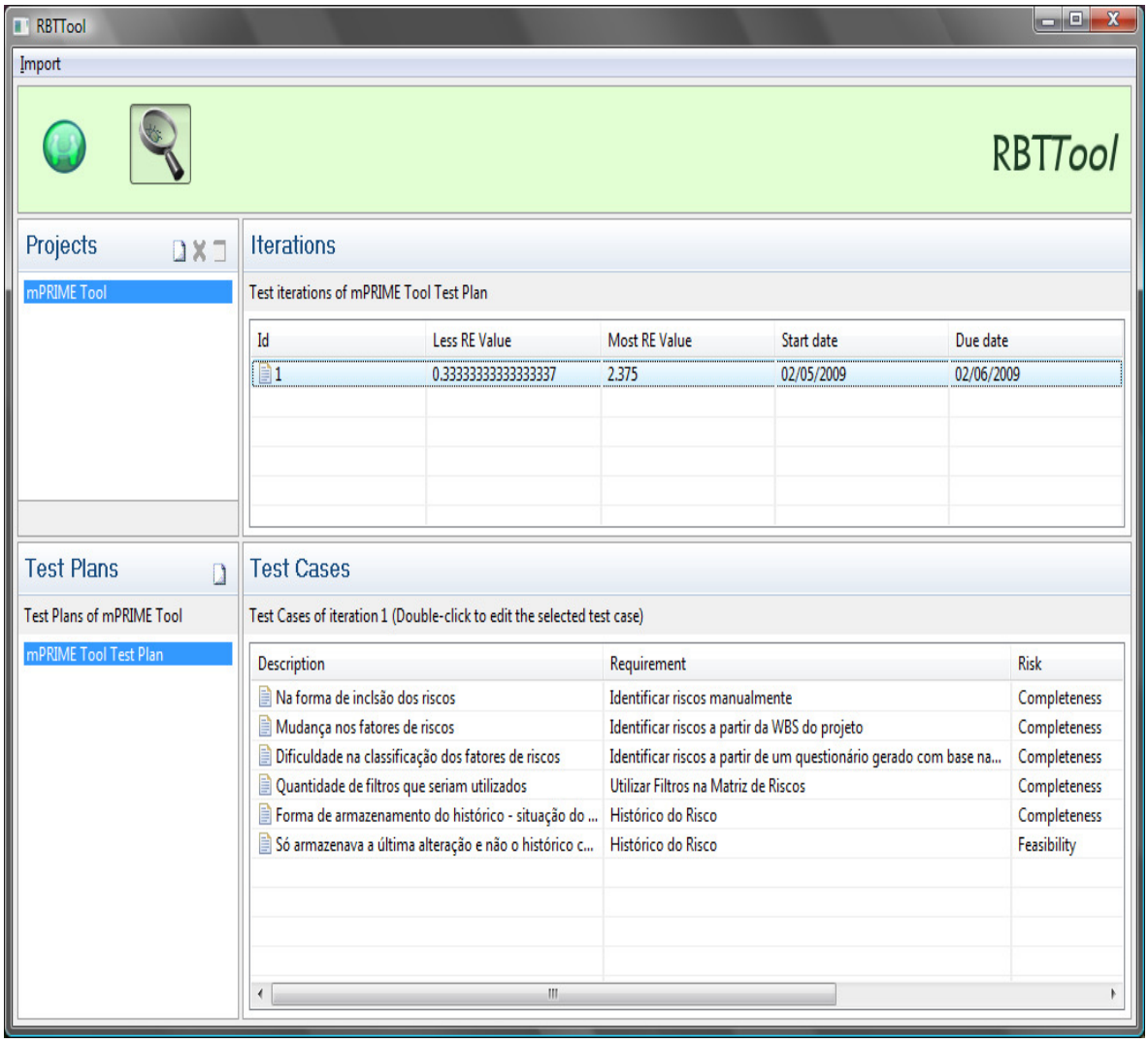
# Apêndice A

# Telas da RBT*Tool*

### Tela principal de Gerência de Riscos

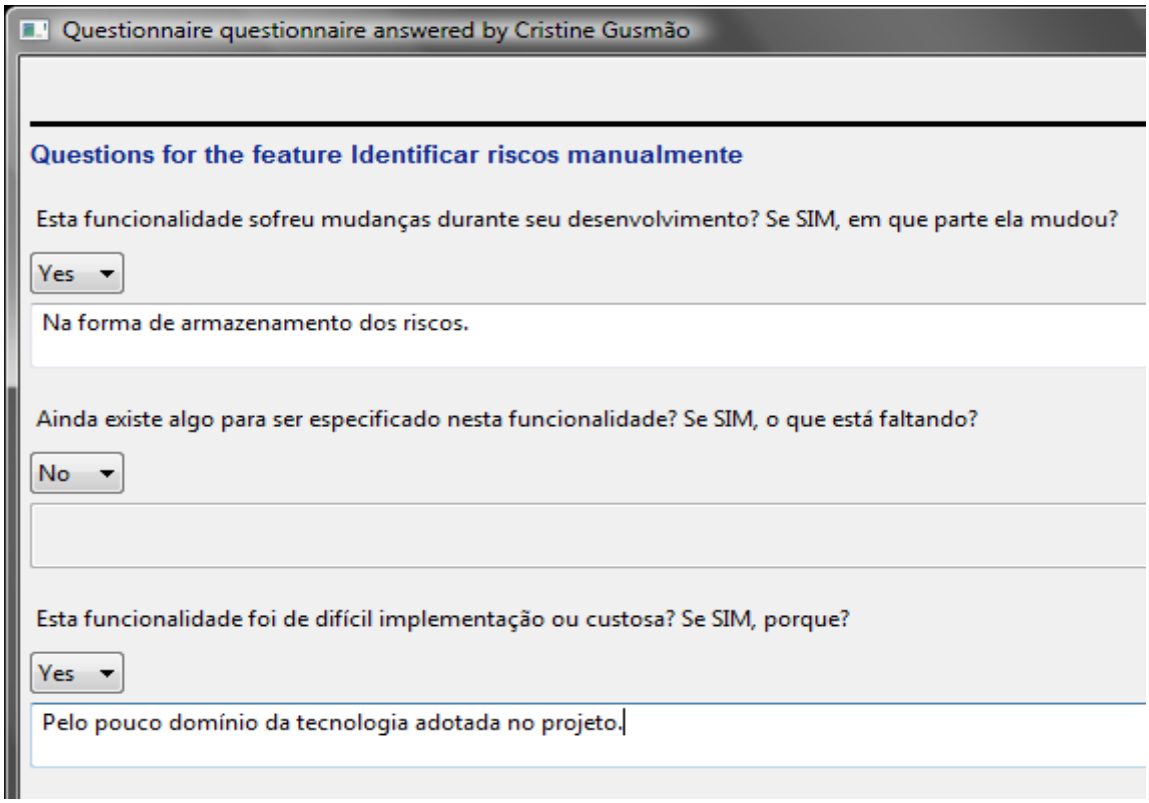


## Tela principal de Gerência de Testes

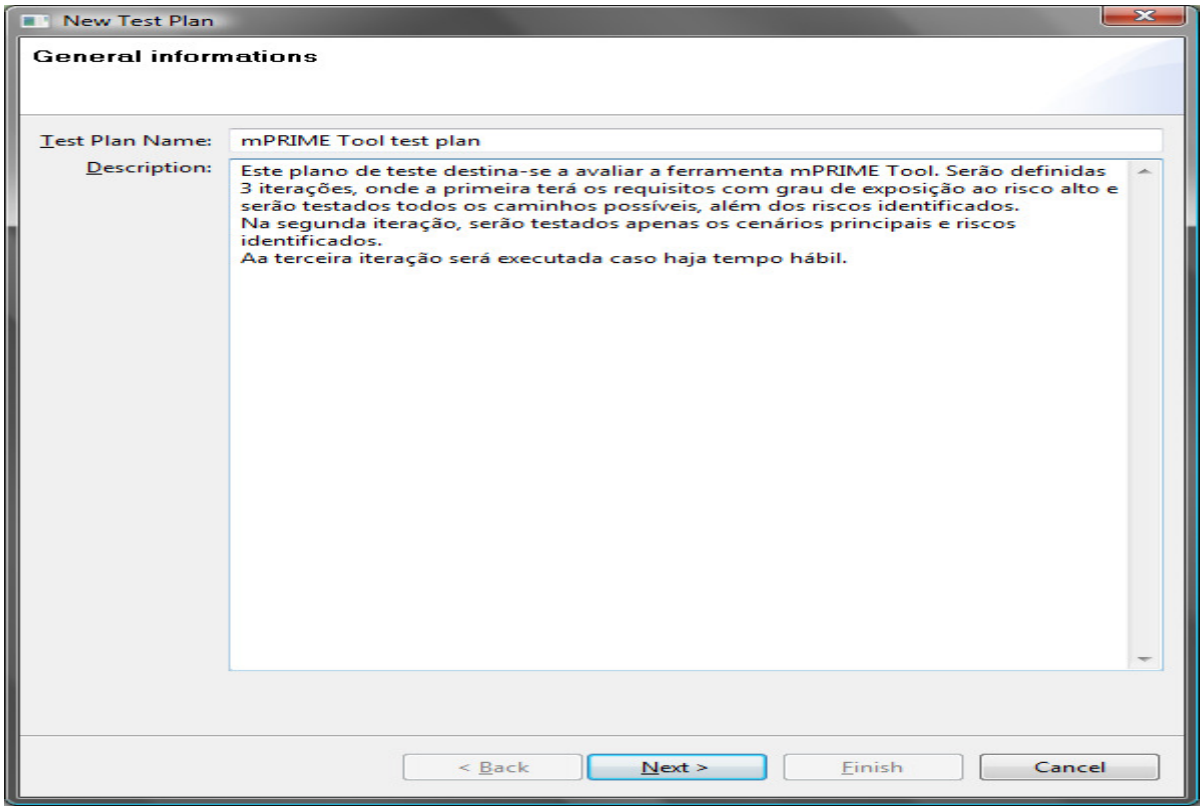


## Etapa de análise de riscos dos requisitos

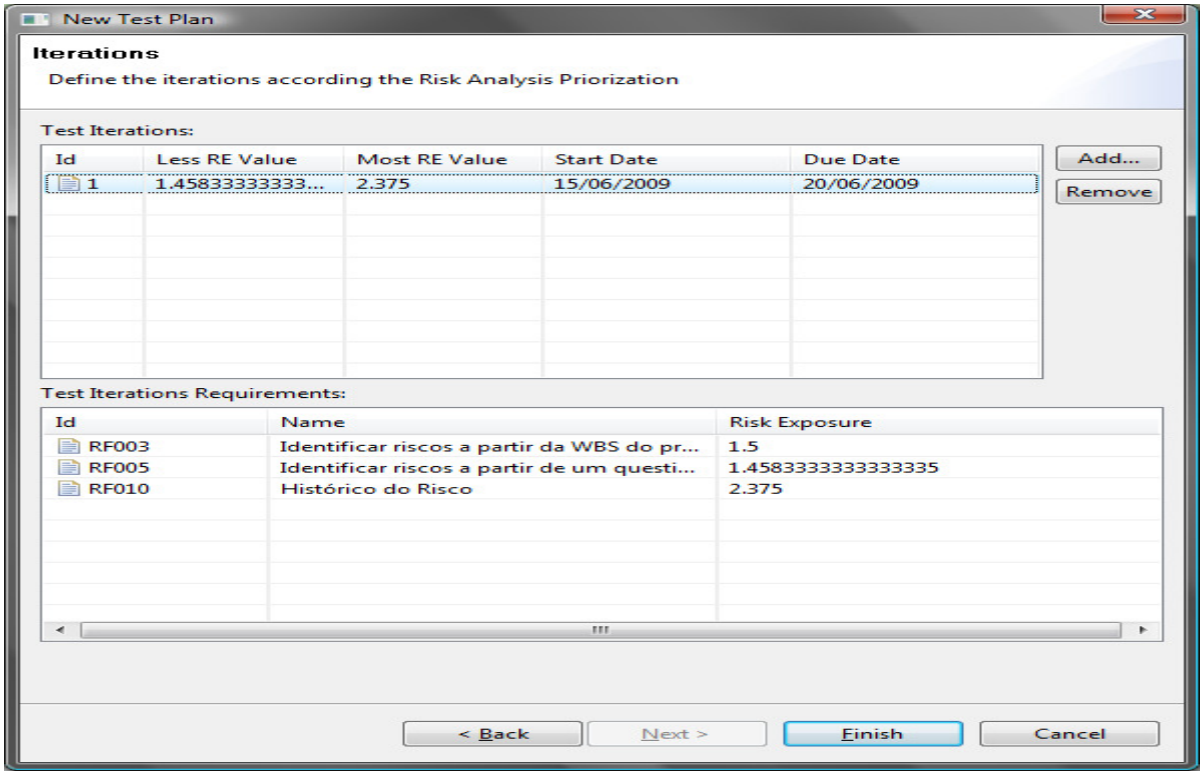

## Etapa de identificação dos riscos dos requisitos através de questionário



## Criação de plano de teste




## Definição das iterações do plano de teste



# Apêndice B

# Questionário de Avaliação da RBT*Tool*

Este questionário serve para avaliar o desempenho da RBTTool do ponto de vista do usuário

1. Nome:
2. Como você classifica o tempo gasto para Identificação e Análise de Riscos, Validação dos riscos identificados via Brainstorming, Planejamento e Projeto de Testes no geral?

| | |
|---|---|
| | Foi gasto pouco tempo. |
| | Foi gasto um tempo normal/aceitável. |
| | Foi gasto muito tempo. |

3.Você acha que a aplicação da técnica RBT suportada pela RBTTool pode contribuir em Projeto de desenvolvimento de software?

| | |
|---|---|
| | Não |
| | Contribui muito pouco |
| | Sim |
| | Caso contribua, forneça detalhes do tipo de contribuição. |
| | |

4. Você sentiu dificuldades em utilizar a RBTTool?

| | |
|---|---|
| | Não. |
| | Sim. Cite as dificuldades encontradas. |
| | |

5.Quais os pontos negativos da RBTTool?

| | |
|---|---|
| | Não foram identificados pontos negativos. |

| | |
|---|---|
| | Foram identificados pontos negativos irrelevantes. |
| | Foram identificados pontos negativos relevantes. Cite-os: |

6.Quais os pontos positivos da RBTTool?

7.Que melhorias você sugere para a RBTTool?

8. Você indicaria o uso da abordagem RBT em outros projetos? Justifique sua resposta.

9. Você concorda com os resultados apresentados pela RBTTool? O resultado reflete a realidade?

| | |
|---|---|
| | Sim. |
| | Não. Explique: |