

Seleção de Técnicas de Teste Baseado em Modelos

Arilo Claudio Dias Neto¹, Guilherme Horta Travassos²

¹ Departamento de Ciência da Computação – Universidade Federal do Amazonas (UFAM)
Campus Universitário, 69077-000 – Manaus –AM – Brasil.
arilo@dcc.ufam.edu.br

² Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ
Caixa Postal 68511 – 21941-972 Rio de Janeiro, RJ, Brasil
ght@cos.ufrj.br

Resumo. *Esta tese propõe uma abordagem para apoiar a seleção combinada de Técnicas de Teste Baseado em Modelos (TTBMs) para projetos de software, chamada Porantim, formada por (1) um corpo de conhecimento composto por 219 TTBMs identificadas na literatura técnica e (2) um processo de apoio à seleção de TTBMs que provê indicadores de adequabilidade de TTBMs a projetos de software a partir de suas características técnicas. Resultados de avaliação experimental indicam que esta abordagem contribui para melhorar a efetividade e eficiência do processo de seleção de TTBMs quando comparada a outra abordagem de seleção disponível na literatura técnica. Em complemento, é apresentada a infraestrutura computacional de apoio à seleção de TTBMs para projetos de software e sua avaliação em uma organização de software internacional.*

1. Introdução

1.1. Contexto e Motivação

A seleção de tecnologias de software (ex: técnicas, métodos, processos ou ferramentas) para projetos como tópico de pesquisa tem sido abordada desde 1991 [1]. Desde então, algumas abordagens de apoio à seleção de tecnologias de software foram desenvolvidas para apoiar diferentes áreas da engenharia de software [2][3][4][5].

O problema da seleção de tecnologias de software pode ser definido como a composição de dois problemas:

- A completude do conjunto de tecnologias de software na qual a seleção será baseada. Atualmente, existe uma larga variedade de tecnologias disponíveis para apoiar o desenvolvimento de software, porém elas estão dispersas em diferentes fontes (livros, artigos científicos ou repositórios) e há pouco conhecimento científico (evidências) a respeito destas tecnologias [2].
- A identificação apropriada das características que representam simultaneamente as tecnologias e o ambiente do projeto de software. Há uma carência de diretrizes que apoiam a seleção de tais tecnologias para um projeto de software [2].

Essas dificuldades se tornam maiores em cenários do desenvolvimento de software onde mais que uma tecnologia de software pode ser combinada com o objetivo de melhorar a efetividade do processo e, conseqüentemente, a qualidade do produto final. Apesar das abordagens de apoio à seleção de tecnologias de software atualmente disponíveis, [2][3][4][5], poderem ser aplicadas em atividades do desenvolvimento de software onde é viável combinar duas ou mais tecnologias (ex: identificar de requisitos), elas focam apenas na seleção individual de uma tecnologia de software. O que as torna

bastante limitada em cenários onde esta combinação é não apenas viável, mas desejável.

Teste de software é um exemplo de atividade do desenvolvimento de software onde a seleção combinada de técnicas pode prover melhorias em seus resultados, pois mais de uma técnica de teste de software pode ser aplicada em um mesmo projeto visando aumentar a cobertura dos testes e, conseqüentemente, a qualidade do produto final. Pode-se observar ainda que cada subcategoria de técnicas de teste possui características específicas que podem afetar a tarefa de seleção de técnicas de teste em um projeto de software. Por exemplo, Teste Baseado em Modelos (TBM) representa uma subcategoria de estratégia de teste na qual casos de teste são derivados total ou parcialmente a partir de modelos descrevendo alguma característica de qualidade do produto (ex: funcionalidade, segurança, desempenho, etc.) [6]. Contextualizando o cenário de seleção de tecnologias de software para TBM, pode-se observar a introdução de desafios adicionais para a seleção de técnicas de teste baseado em modelos (TTBMs) para um projeto de software devido às características específicas desta subcategoria de técnicas de teste, tais como a dependência de um modelo formal descrevendo estrutura/comportamento do software para geração dos testes.

De acordo com DIAS-NETO *et al.* [7], o grande número de TTBMs disponíveis e o pouco conhecimento científico sobre elas dificultam a seleção de um subconjunto de TTBMs para um dado projeto de software. Atualmente, esta tarefa é realizada baseando-se em conveniência, ou seja, normalmente são aplicadas em projetos de software aquelas TTBMs já conhecidas por uma equipe de teste, independentemente de sua adequação para tal projeto [2].

1.2. Objetivo do Trabalho

A seleção de tecnologias de software, independentemente da atividade do processo de desenvolvimento na qual esteja sendo aplicada, pode depender de diversos fatores, dentre os quais podem ser citados fatores técnicos, tecnológicos, sociais, políticos e econômicos. A abordagem proposta neste trabalho utiliza como base alguns fatores técnicos (como a tecnologia funciona na prática) e tecnológicos (em qual contexto tecnológico ela pode ser aplicada) do projeto de software com o objetivo de prover informações que possam apoiar na tomada de decisão a respeito da seleção de TTBMs em projetos de software. Esta decisão foi tomada pelo fato de esses fatores estarem descritos com mais frequência nos artigos que apresentam TTBMs, mas os demais fatores podem ser incorporados neste trabalho, precisando apenas ter suas características mapeadas nas TTBMs disponíveis na literatura técnica. Sendo assim, esta pesquisa irá focar em duas frentes para resolver o problema de seleção de tecnologias de software, contextualizando o problema para a área de Teste Baseado em Modelos:

- (1) Fornecer conhecimento baseado em resultados de estudos experimentais sobre as possíveis opções associadas às TTBMs existentes para os profissionais responsáveis pela seleção, e;
- (2) Identificar quais características ou informações dos projetos de software e TTBMs possuem influência na adequabilidade e impacto de uma ou várias TTBMs em um projeto de software.

A proposta desta pesquisa para o problema da seleção de TTBMs é o desenvolvimento de uma abordagem, batizada de *Porantim*¹, que avalia a adequabilidade

¹ *Porantim* é uma peça de Madeira usada pelos Índios Amazônicos *Sateré-Mawé* (que habitam a região do Baixo Amazonas) que funciona como uma bola de cristal prevendo acontecimentos e resolvendo conflitos internos. É também usado no ritual de passagem *Wat'amã*, cujo objetivo é selecionar os índios que serão os guerreiros da tribo.

e o impacto de TTBM a partir das características de um projeto de software onde elas seriam aplicadas. *Porantim* é formada por um corpo de conhecimento sobre TTBM e um processo de seleção que provê informações e indicadores que apoiam a equipe de teste na execução desta tarefa em um projeto de software.

1.3. Metodologia Científica

A metodologia de pesquisa seguida para o desenvolvimento deste trabalho foi fundamentada na abordagem para desenvolvimento de novas tecnologias de software baseada em estudos primários e secundários publicada em DIAS-NETO *et al.* [8]. Ela se divide em duas fases: concepção e avaliação da abordagem proposta, conforme descrito na Figura 1.

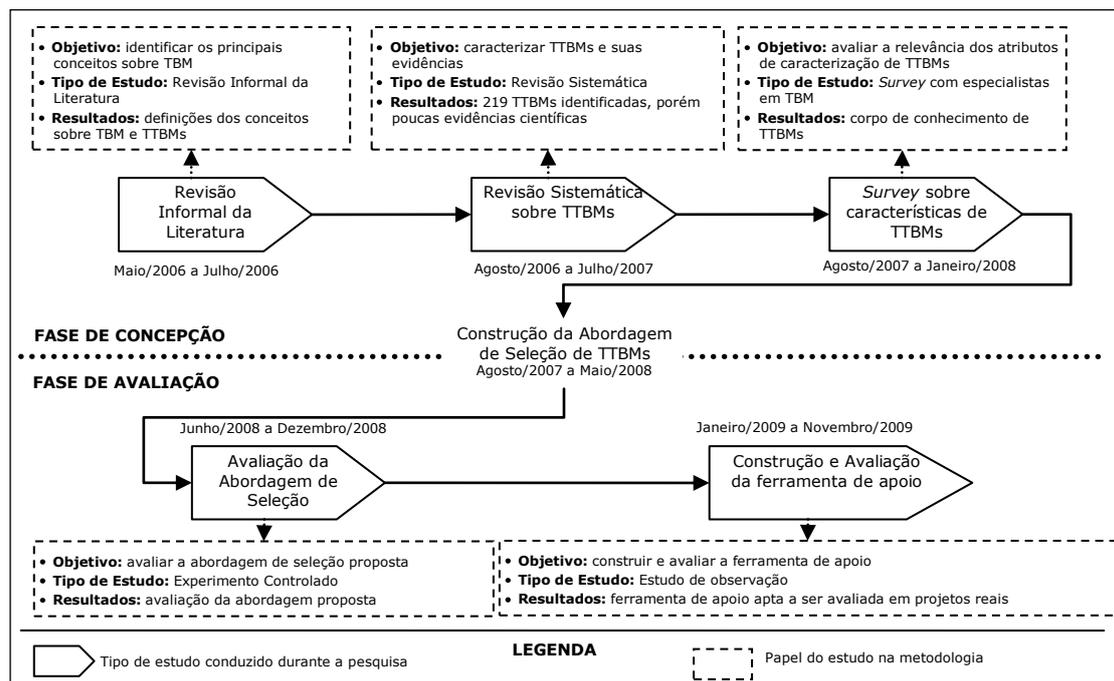


Figura 1. Metodologia de Pesquisa Adotada

A fase de concepção da tecnologia envolve a execução de estudos secundários e/ou primários com o objetivo de se obter uma proposta inicial da tecnologia proposta.

- **Revisão inicial da literatura:** executada de Março a Julho de 2006, nesta atividade realizou-se uma revisão *ad hoc* da literatura sobre a área de Teste Baseado em Modelos, seus conceitos, características, benefícios, processo, infraestruturas de apoio, principais grupos de pesquisa e fontes de informação.
- **Revisão sistemática:** executada inicialmente no período de Julho a Dezembro de 2006 e atualizada no período de Julho a Agosto de 2009, nesta atividade realizou-se uma revisão controlada da literatura cujo objetivo foi identificar e caracterizar TTBM publicadas na literatura técnica extraindo de cada técnica suas informações de acordo com os atributos de caracterização de TTBM identificados no passo anterior da metodologia.
- **Survey:** executada de Setembro de 2007 a Fevereiro de 2008, o objetivo desta atividade foi avaliar os atributos que descrevem TTBM extraídos a partir dos passos anteriores da metodologia com o propósito de caracterizá-los com respeito a sua importância para caracterizar uma TTBM e sua relevância no

contexto da seleção de TTBM's para projetos de software. O resultado de sua execução foi a geração do corpo de conhecimento sobre TTBM's com resultados baseados na opinião de especialistas.

- **Construção da Abordagem de Seleção:** executada de Março de 2008 a Novembro de 2008, com o corpo de conhecimento estabelecido, passou-se a trabalhar na abordagem proposta para apoiar a seleção de TTBM's para projetos de software.

Concluída a fase de concepção, passou-se à etapa de avaliação da abordagem proposta. Para tal fase, foram realizados dois estudos com diferentes propósitos:

- **Estudo de Viabilidade:** executado de Novembro a Dezembro de 2008, este estudo experimental teve como objetivo avaliar diversos aspectos (eficiência, efetividade, completude, usabilidade e satisfação do usuário) da abordagem proposta neste trabalho em relação a outra abordagem de apoio à seleção de técnicas de teste proposta por VEGAS e BASILI [2].
- **Estudo de Observação:** executado de Maio a Junho de 2009, este estudo de avaliação teve como objetivo avaliar o comportamento da infraestrutura computacional provida para apoiar a abordagem proposta neste trabalho com engenheiros de software que atuam no seu dia-a-dia com atividades de TBM.

1.4. Organização do Trabalho

Este artigo está organizado em 5 seções, considerando esta introdução. Na seção 2 são descritos os principais conceitos relacionados a Teste Baseado em Modelos. Na seção 3 é descrita a abordagem proposta, *Porantim*, desenvolvida para apoiar a seleção de TTBM's para projetos de software a partir de seus elementos principais e sua avaliação experimental. Na seção 4 é apresentada a infraestrutura computacional desenvolvida para apoiar a abordagem *Porantim* e sua avaliação. Finalmente, na seção 5 são descritas as conclusões do trabalho, resultados obtidos, lista de publicações e próximos passos.

2. Teste Baseado em Modelos

Teste Baseado em Modelos (TBM) consiste em uma estratégia de teste na qual casos de teste são derivados totalmente ou parcialmente de um modelo que descreve alguma característica de qualidade (ex: funcionalidade, segurança, desempenho, etc.) de um software [6]. Para sua utilização, é necessário que o comportamento ou estrutura do software (ou parte deles) tenha sido formalizado através de modelos com regras bem definidas (ex: métodos formais, máquinas de estado finito, diagramas UML com regras definidas de formação, dentre outros). A estratégia de TBM normalmente inclui diferentes níveis de abstração, um modelo comportamental/estrutural do software, o relacionamento entre modelos e código, uma tecnologia para geração dos casos de teste, a importância dos critérios de seleção dos testes (algoritmos para geração dos casos de teste) e uma discussão do que pode ou não ser automatizado durante os testes [9].

A evolução da estratégia de teste baseada em modelos tem ocorrido ano a ano. Além disso, as diversas TTBM's usam diferentes modelos para especificar o software, e esses modelos normalmente descrevem diferentes características de um produto. Esta variedade de técnicas torna a identificação, seleção e utilização de uma TTBM em um projeto de software uma tarefa complexa e de difícil decisão, o que representa uma motivação para este trabalho.

A seção seguinte apresenta a abordagem, chamada *Porantim*, que visa apoiar a seleção de TTBM's em projetos de software, seus elementos e sua avaliação experimental.

3. Porantim: Uma Abordagem para Apoiar a Seleção Combinada de Técnicas de Teste Baseado em Modelos

3.1. Visão Geral sobre *Porantim*

A abordagem proposta foi batizada de *Porantim* [11][12] e representa uma evolução da abordagem de apoio à seleção de técnicas de teste chamada de Esquema de Caracterização, proposta por VEGAS e BASILI [2]. Esta abordagem é fundamentada em dois elementos principais: (1) Corpo de Conhecimento sobre TTBM e (2) Processo de Seleção de TTBM (Figura 2). Na versão original, VEGAS e BASILI [2] propuseram um esquema de caracterização para técnicas de teste, em geral, funcionando como um repositório estático, não possuindo assim qualquer processo que apoiasse na sua seleção para projetos de acordo com suas características particulares.

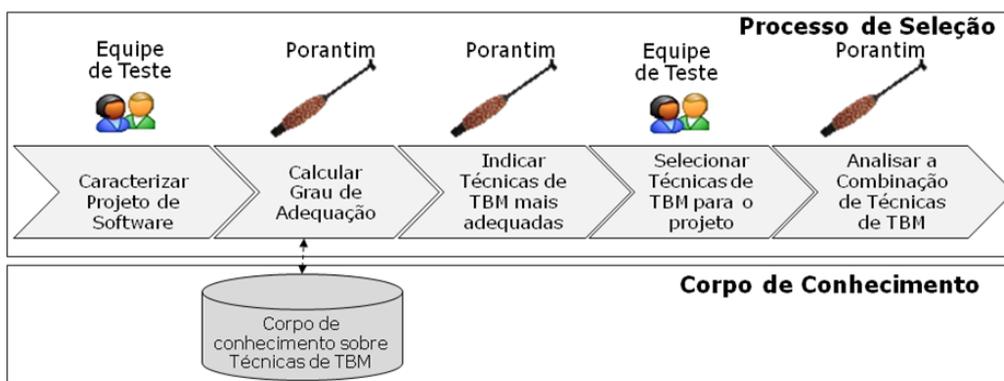


Figura 2. Esquema da Abordagem *Porantim* [11]

As próximas subseções irão descrever cada elemento que compõe a abordagem *Porantim* e a metodologia seguida para seu desenvolvimento.

3.2. Corpo de Conhecimento de TTBM

O corpo de conhecimento consiste em um repositório de TTBM desenvolvido a partir de resultados de dois estudos conduzidos no contexto deste trabalho: um para definir seu conteúdo e outro para definir sua estrutura.

Para a definição do conteúdo do corpo de conhecimento, ou seja, quais são as TTBM que irão compor o repositório, uma revisão sistemática da literatura foi executada em 2006 e atualizada em 2009 com o objetivo de identificar e caracterizar TTBM publicadas na literatura técnica. Ao total, 599 publicações foram identificadas (caracterizadas na Tabela 1), dos quais 328 artigos foram excluídos, pois não descreviam TTBM (artigos fora do escopo que foram retornados pelas máquinas de busca, algo usual em revisões sistemáticas), eram repetidos ou não estavam disponíveis eletronicamente. Dos 271 artigos restantes, todos foram analisados e eles descrevem 219 TTBM diferentes. Como a UML tem se tornado ao longo dos anos padrão para modelagem de sistemas, as TTBM identificadas foram categorias como “Utilizam UML” ou “Não utilizam UML”. Sendo assim, dessas técnicas, 76 TTBM (34,7%) adotam diagramas UML e 143 TTBM (65,3%) não adotam. A lista completa dos 599 artigos identificados ao longo das duas execuções do protocolo de revisão sistemática, o próprio protocolo de busca adotado e os resultados das análises dos artigos foram publicados em [7] e [13], e estão disponíveis no Capítulo 2 da tese [10].

Tabela 1. Classificação dos Artigos Identificados

Categorias de Artigos	Quantidade de Artigos	Percentual de Artigos
Utilizam UML	79	13,19%
Não utilizam UML	192	32,05%
Não Classificados	328	54,75%
Total de Artigos	599	100%

Para a definição da estrutura do corpo de conhecimento, um *survey* foi realizado com especialistas que trabalham com TBM (da indústria e da academia) identificados através da revisão sistemática citada no parágrafo anterior. Este *survey* foi realizado em 2007 e 34 especialistas, entre 144 convidados, em TBM responderam ao questionário. Isso nos dá uma confiança de 85% dos dados coletados usando a fórmula de cálculo do nível de confiança apresentado em [14] e descrito no Capítulo 3 da tese (Figura 3.6).

Eles avaliaram um conjunto inicial de 18 atributos de caracterização e sugeriram 3 (três) novos atributos, totalizando 21 atributos. Os detalhes a respeito do planejamento, projeto, execução e análise dos resultados do *survey* podem ser encontrados em [15] e no Capítulo 3 da tese [10]. Durante o desenvolvimento de *Porantim*, 3 atributos usados para caracterização de TTbMs precisaram ser divididos em 7 atributos para melhor caracterizar uma TTbM, totalizando 25 atributos. A lista final de atributos de caracterização de TTbMs e seus respectivos pesos estão apresentados na Tabela 2.

Tabela 2. Estrutura do Corpo de Conhecimento (Atributos de Caracterização)

Ordem	Atributo de caracterização de TTbMs	Nível de Relevância
1	Modelo Comportamental/Estrutural	83,48%
2	Critério de Cobertura dos Testes	74,79%
3	Critério de Geração de Casos de Teste	72,57%
4	Entradas requeridas para usar uma TTbM	72,54%
5	Limitações/Restrições para usar uma TTbM	69,96%
6	Tipo de Técnica de Teste (Funcional ou Estrutural)	69,89%
7	Resultados gerados pela TTbM	69,70%
8	Nível de Teste	69,70%
9	Característica de Qualidade de SW que a TTbM está apta a avaliar	68,30%
10	Custo associado à ferramenta de apoio	63,74%
11	Nome da ferramenta de apoio	63,74%
12	Plataforma em que a ferramenta de apoio opera	63,74%
13	Plataforma de execução do software	55,85%
14	Paradigma de desenvolvimento	55,85%
15	Linguagem de programação adotada	55,85%
16	Uso de Modelos Intermediários	54,47%
17	Tecnologia de Geração dos Testes	52,54%
18	Necessidade de Ferramentas Externas	49,96%
19	Resultados Históricos	48,80%
20	Proporção de Passos Automatizados	47,48%
21	Avaliação Experimental	47,43%
22	Nível de Complexidade dos Passos não automatizados	46,58%
23	Existência de um mecanismo de rastreabilidade	34,39%
24	Habilidade/conhecimento necessário	33,00%
25	Existência de um verificador de modelos	18,71%

Com isso, cada TTbM, as já identificadas neste trabalho e as que possam vir a ser identificadas no futuro, que compõe o corpo de conhecimento deve ser caracterizada de acordo com a estrutura apresentada acima.

3.3. Processo de Seleção de TTBM

A existência de um corpo de conhecimento sobre TTBM é um passo importante para apoiar a seleção de TTBM em projetos de software, porém possui função limitada, pois sozinho este provê muito conhecimento técnico sobre as TTBM, porém nenhum direcionamento a respeito de como usá-lo para selecionar TTBM mais coerentes ao contexto de um projeto de software. Sendo assim, o segundo elemento que compõe a abordagem *Porantim* é um processo que permite apoiar a análise das características do projeto de software a ser desenvolvido e das TTBM disponíveis para uso, provendo informações que apoiem na tomada de decisão sobre quais técnicas melhor se adéquam a um certo projeto de software. Tal processo é composto por cinco atividades (Figura 2), descritas a seguir:

- **Caracterizar Projeto de Software:** a equipe de teste precisa preencher um questionário descrevendo as características e requisitos de teste de um projeto de software onde TTBM serão aplicadas. As características/requisitos de teste a serem preenchidos foram extraídas da ferramenta ADAPTPRO [16], que compõe a estação TABA, do survey anteriormente citado e a partir do próprio desenvolvimento desta tese. Ao total, 16 características/requisitos de teste devem ser preenchidos: plataforma de execução, paradigma de desenvolvimento adotado, linguagem de programação adotada, tecnologia usada para modelagem, modelos providos no processo, tamanho da aplicação, habilidades da equipe de teste, níveis de teste requeridos, tipo de técnica de teste a ser aplicadas, características de qualidade de software a serem avaliadas, tipos de falhas a serem reveladas, ferramenta de apoio e custo esperado dos testes.
- **Calcular Grau de Adequação:** *Porantim* provê fórmulas matemáticas que se baseiam no conceito matemático de *Distância Euclidiana* para se calcular o *grau de adequação* (“distância”) entre o projeto de software caracterizado no passo anterior e cada TTBM disponível no corpo de conhecimento (repositório de TTBM). O *grau de adequação* é representado por um valor numérico que indica quão “próximas” são as características da TTBM em relação às necessidades do projeto de software.
- **Indicar Técnicas de TBM mais adequadas:** após calcular o *grau de adequação*, deve ser listado apenas um subconjunto com as TTBM que possuem os maiores valores de *grau de adequação* (o número de TTBM a ser listada deve ser definido pela equipe de teste), onde para cada TTBM deve ser apresentado o seu grau de adequação e as suas características.
- **Selecionar Técnicas de TBM para o Projeto:** a equipe de teste precisa selecionar um subconjunto de TTBM, dentre as sugeridas no passo anterior, para que sejam utilizadas no projeto de software.
- **Analisar a Combinação de Técnicas de TBM:** após a etapa de selecionar TTBM, *Porantim* provê fórmulas que se baseiam no conceito matemático do *Coefficiente de Jaccard* para apoiar na análise da relação entre os requisitos de teste solicitados para o projeto de software e quais destes estariam sendo providos pela combinação de TTBM selecionadas no passo anterior. Neste momento é avaliado o impacto das TTBM em algumas variáveis do processo de testes, tais como cobertura dos requisitos do projeto de software, esforço de modelagem salvo para criação dos testes e grau de preparação da equipe de teste.

Os detalhes do processo de seleção de TTBM's que compõe a abordagem *Porantim* e exemplos de seu funcionamento estão descritos no Capítulo 4 da tese [10], e foram publicados em [11] e [12].

3.4. Avaliação Experimental de *Porantim*

Nesta atividade, foi executado um estudo experimental com o objetivo de avaliar a abordagem *Porantim* em relação a outra abordagem de apoio à seleção de técnicas de teste, chamada de Esquema de Caracterização [2]. Neste momento, decidiu-se por realizar uma repetição externa (replicação de um estudo experimental com pesquisadores diferentes daqueles que executaram o estudo original) do estudo experimental publicado em [2] a fim de se aproveitar e reusar o pacote do estudo (planejamento, projeto, instrumentos, procedimentos de execução e análise dos resultados). Este estudo experimental teve como objetivo avaliar diversos aspectos (completude, eficiência, efetividade e usabilidade) da abordagem proposta.

Esta avaliação foi realizada em 2 diferentes contextos: (1) um grupo de 22 alunos de pós-graduação em engenharia de software e (2) um grupo de 34 alunos de graduação em sistemas de informação. Os estudos foram executados e analisados separadamente. Para análise dos dados, foi adotado o método de Análise de Variância (ANOVA) e o nível de significância (α) de 95%. Os resultados estão apresentados na Tabela 3.

Tabela 3. Resultado do Estudo Experimental para Avaliação de *Porantim*

Aspecto Avaliado	Variável	<i>p-value</i> ($\alpha = 95\%$)		Fator influente
		Pós-graduação	Graduação	
Completude	Percentual de informações usadas	0,0608	0,1642	Nenhum
	Quantidade de informações faltando	0,3664	0,2712	Nenhum
Efetividade	Percentual de escolhas corretas	0,0215	0,0060	<i>Porantim</i>
Eficiência	Tempo de seleção	0,0141	0,0064	<i>Porantim</i>
	Percentual de seleções corretas / tempo de seleção	0,0019	0,0007	<i>Porantim</i>
	Percentual de seleções corretas / percentual de informações usadas	< 0,0001	< 0,0001	<i>Porantim</i>
Usabilidade	Quantidade de problemas reportados	0,2074	0,4691	Nenhum

Os resultados obtidos indicam que *Porantim* seria o fator influente em todas as variáveis relacionadas aos aspectos eficiência e efetividade ($p\text{-value} < 0,05$). Pode ser ainda observado que foi obtido exatamente o mesmo resultado com alunos de pós-graduação e de graduação, o que poderia sugerir que *Porantim* possui influência na eficiência e efetividade do processo de seleção de TTBM's com engenheiros de software com diferentes níveis de formação.

O planejamento, projeto, execução e análise dos resultados deste estudo estão descritos no Capítulo 5 da tese [10], e foram publicados em [17].

4. Infraestrutura Computacional de Apoio à *Porantim*

Após a finalização da abordagem *Porantim*, percebeu-se que diversas tarefas que compõem o processo de seleção de TTBM's poderiam ser automatizadas ou semi-automatizadas, pois representam cálculos matemáticos ou filtragem de TTBM's a partir de seu repositório. Dessa forma, decidiu-se pela construção de uma infraestrutura computacional para apoiar na aplicação da abordagem *Porantim*.

Com isso, a infraestrutura computacional Maraká, desenvolvida para apoiar o planejamento e controle de teste de software [18], foi estendida em sua arquitetura e

funcionalidades a fim de prover apoio ao processo de seleção de TTbMs utilizando a abordagem *Porantim* proposta neste trabalho.

4.1. Novos Requisitos Funcionais de Maraká para implementar *Porantim*

Um conjunto de novos requisitos funcionais foi definido para implementar a abordagem *Porantim* na infraestrutura computacional Maraká, impactando dessa forma na arquitetura e funcionalidades providas originalmente pela infraestrutura Maraká. Esses requisitos estão relacionados a facilidades que possibilitam configurar e implementar os dois elementos que compõem *Porantim*, descritos na Seção 3 e Figura 2:

- RF1: Maraká deve disponibilizar um repositório contendo a caracterização de TTbMs (a lista de atributos de caracterização a serem armazenados no repositório está apresentada na Tabela 2), permitindo aos usuários de Maraká com perfil *Administrador* inserir, editar, excluir, consultar e importar TTbMs. Este requisito funcional implementa o corpo de conhecimento de TTbMs que compõe a abordagem *Porantim*.
- RF2: Maraká deve permitir que usuários de Maraká com perfil *Administrador* configurem o nível de relevância (peso) de cada atributo de caracterização a ser usado durante o processo de seleção de TTbMs sugerido por *Porantim*.
- RF3: Maraká deve permitir que usuários de Maraká com perfil *Administrador* configurem a quantidade máxima de TTbMs a serem exibidas durante o processo de seleção (Atividade 3 – *Indicar Técnicas de TBM mais adequadas* – Figura 2).
- RF4: Maraká deve permitir que usuários de Maraká com perfil *Gerente de Teste* em um projeto de software selecionem TTbMs disponíveis no repositório mantido pela infraestrutura executando o processo de seleção de TTbMs que compõe a abordagem *Porantim*. Esta tarefa deve ser inserida no processo de testes existente em Maraká na atividade *Planejamento dos Testes*.

4.2. Funcionalidades

As funcionalidades construídas na versão estendida da infraestrutura Maraká para apoiar a seleção de TTbMs estão relacionadas principalmente a 2 atividades:

○ **Funcionalidades de Apoio à Configuração de *Porantim***

Esta funcionalidade provê apoio à configuração do corpo de conhecimento (repositório) de TTbMs, sendo possível gerenciar (cadastrar, excluir, consultar e editar) individualmente as TTbMs disponíveis no repositório de Maraká.

Uma segunda funcionalidade possibilita a configuração dos parâmetros de *Porantim*, permitindo ajustar o peso dos atributos de caracterização de TTbMs previamente definidos para o processo de seleção. Tais pesos foram definidos inicialmente através de um *survey* publicado em [15], porém podem ser alterados ao longo do tempo de acordo com as características e necessidades da organização de software que esteja utilizando a abordagem *Porantim*. A soma dos pesos de todos os atributos de caracterização deve ser igual a 1, representando 100%.

○ **Funcionalidades de Apoio ao Processo de Seleção de Técnicas de TBM**

A seleção de técnicas de teste em um projeto de software normalmente ocorre durante a atividade de “Planejamento dos Testes”. Nesse contexto, a infraestrutura Maraká possui originalmente um processo de teste contendo a atividade *Planejar Testes* e uma sub-atividade chamada *Definir Técnicas de Teste*. Na versão original de Maraká,

esta sub-atividade é executada através da descrição manual das técnicas de teste que serão adotadas. Em sua versão estendida, é possível optar entre a descrição manual das técnicas de teste ou selecioná-las com o auxílio de *Porantim*. Neste último caso, o contexto restringe-se à seleção de TTBM e esta funcionalidade foi dividida em 3 passos:

- **PASSO 1) Caracterização do Projeto de Software**

Atividade onde a equipe de teste define as características e necessidades de teste para o projeto de software em que se deseja aplicar TTBM, através de um formulário contendo os atributos de caracterização de projeto de software definidos na etapa de configuração de *Porantim*.

- **PASSO 2) Seleção de Técnicas de TBM**

Atividade onde são calculados automaticamente os graus de adequação de todas as TTBM existentes no repositório em relação ao projeto de software caracterizado no passo 1, seguindo as fórmulas fornecidas por *Porantim*. As técnicas são exibidas ordenadas de forma decrescente por grau de adequação. Este passo implementa as atividades 2, 3 e 4 do elemento “*Processo de Seleção*” (Figura 2).

Como representação gráfica da adequação de uma TTBM a um projeto de software, foi adotado o *Gráfico de Radar* (Figura 3). Em *Porantim*, os vértices do gráfico correspondem aos atributos de caracterização de projetos de software/TTBMs, permitindo a visualização gráfica da adequação de uma TTBM a um projeto de software. A partir desta representação visual da adequação de TTBM, acredita-se que a equipe de teste pode ser mais capaz de tomar decisões a respeito da seleção ou não de determinada TTBM para o projeto de software em questão.

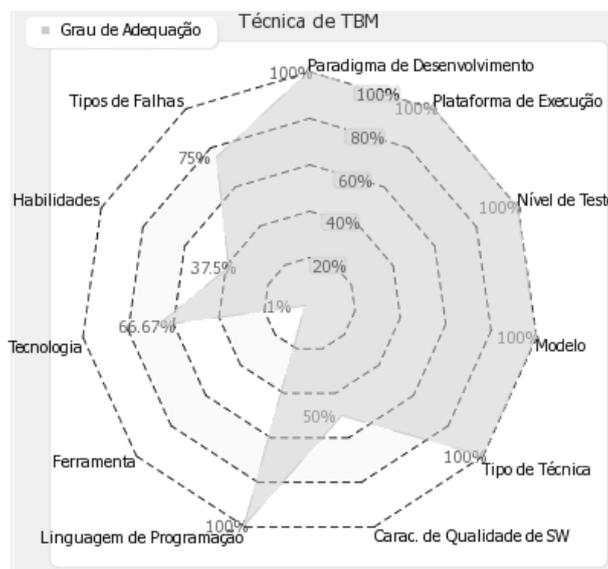


Figura 3. Tela de Análise da Adequação de TTBM via Gráfico de Radar

- **PASSO 3) Combinação de Técnicas de TBM**

Atividade onde é realizada a análise do impacto das TTBM selecionadas no passo 2, a partir do cálculo de três indicadores, representados por números que variam entre 0 e 100%: *Cobertura do Projeto de Software*, *Esforço de Modelagem Salvo para criação dos testes* e *Grau de Preparação da Equipe de Teste* selecionada para o Projeto. A análise do impacto é apresentada aos usuários através de uma tabela contendo os

dados do projeto e das técnicas selecionadas e de um gráfico representando os resultados obtidos nos indicadores analisados. Adotou-se o *Gráfico de Gauge*² para representação dos indicadores de análise de combinação de TTBM. A escolha do *Gráfico de Gauge* ocorreu pela possibilidade de ilustrar diferentes faixas de valores representando categorias separadas por limites pré-estabelecidos (Figura 4). Para simplificar a representação gráfica de cada indicador, seus valores foram divididos em quatro intervalos: [100-75%]: Alto; [75-50%]: Médio; [50-25%]: Baixo; [25-0%]: Muito Baixo.

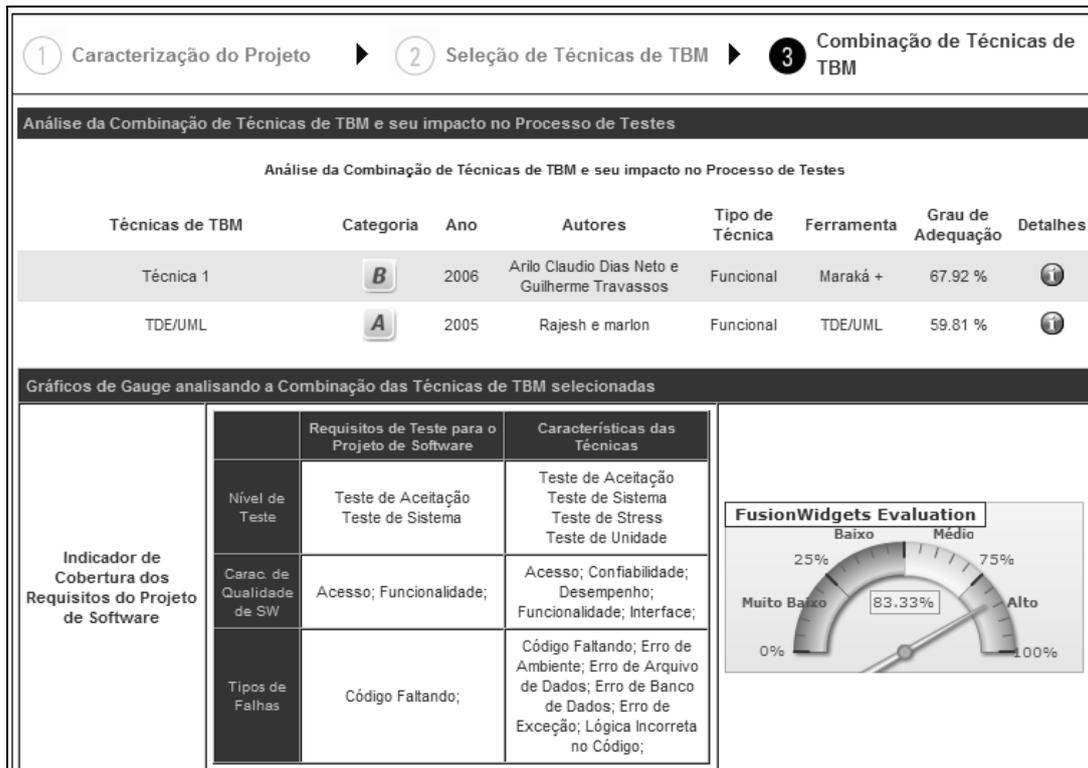


Figura 4. Tela de Análise da Combinação de TTBM

Uma versão de avaliação da infraestrutura apresentada pode ser acessada através da URL <http://lens-ese.cos.ufrj.br/porantim>, usando o login *guest* e senha *porantim*.

4.3. Avaliação da Infraestrutura Computacional

Após seu desenvolvimento, tal infraestrutura foi avaliada através de um estudo de observação por engenheiros de software que atuam com TBM em projetos reais no *SIEMENS Corporate Research*, localizado em Princeton-EUA. Os resultados desta avaliação, embora não conclusivos, indicaram a viabilidade de aplicação da infraestrutura proposta em projeto de software que adotem TBM.

O objetivo do estudo consistiu em analisar o comportamento e viabilidade da infraestrutura em relação à seleção realizada por especialistas em TBM. Participaram deste estudo 2 gerentes de teste especialistas em TBM que utilizaram 2 projetos de software caracterizados pelos pesquisadores (*Controle de Estacionamento e Gerenciamento de Vídeo Locadora*), os mesmos utilizados no Estudo Experimental

² Para exibição deste gráfico, está sendo utilizada uma versão de avaliação do componente *FusionWidgets* v3, disponível em <http://www.fusioncharts.com/widgets/>.

conduzido para avaliar *Porantim*, descrito em [17], e 5 TTbMs reais que foram disponibilizadas no repositório.

A Tabela 4 apresenta os resultados quantitativos deste estudo. Apesar de poucos, os dados indicam que o uso da infraestrutura reduziu à metade o tempo total para caracterização do projeto de software, para seleção de TTbMs e, conseqüentemente, para a execução do processo de seleção de TTbMs independente do projeto usado e do grau de experiência do profissional.

Tabela 4. Resultados obtidos no Estudo de Avaliação

Item avaliado	Participante 1	Participante 2
Fase 1 – Seleção manual de TTbMs		
Projeto usado da seleção manual de TTbMs	Vídeo	Estacionamento
Tempo para caracterização do projeto de software sem apoio computacional	10 minutos	10 minutos
Tempo para seleção manual de TTbMs	15 minutos	5 minutos
Tempo total para execução do processo de seleção	25 minutos	15 minutos
Fase 2 – Seleção de TTbMs com o apoio da infraestrutura		
Projeto usado da seleção de TTbMs	Estacionamento	Vídeo
Tempo para caracterização do projeto de software usando a infraestrutura	6 minutos	5 minutos
Tempo para seleção de TTbMs usando a infraestrutura	8 minutos	2 minutos
Tempo total para execução do processo de seleção	14 minutos	7 minutos

Os detalhes a respeito da construção da infraestrutura computacional, sua arquitetura, descrição de suas funcionalidades e todo o processo para sua avaliação estão apresentados no Capítulo 6 da tese [10].

5. Conclusões e Resultados Obtidos

Este trabalho propõe uma abordagem, chamada de *Porantim* e desenvolvida a partir de uma metodologia científica baseada na condução de estudos experimentais, que apoia a seleção combinada de TTbMs para projetos de software. Esse apoio é provido a partir da análise das principais características técnicas que descrevem as TTbMs e provendo conhecimento que auxilie na tomada de decisão a respeito de quais TTbMs são mais adequadas às características e requisitos de um projeto de software.

5.1. Contribuições

As principais contribuições desta pesquisa estão classificadas em 4 categorias:

- **Corpo de Conhecimento de TTbMs**

Disponibilização de um corpo de conhecimento contendo a caracterização de 219 TTbMs identificadas na literatura técnica a partir de uma revisão sistemática. Este corpo de conhecimento foi estruturado a partir da consulta a especialistas em TBM por meio de um *survey* (pesquisa de opinião).

- **Abordagem de Apoio à Seleção de TTbM: *Porantim***

Foi definido um processo de apoio para auxiliar a coleta de informações sobre o projeto de software no qual TTbMs devem ser aplicadas e com isso provê conhecimento técnico a respeito da adequação e impacto de um conjunto de TTbMs em relação a um projeto de software. Tal abordagem foi avaliada experimentalmente, e seus resultados sugerem que *Porantim* seria mais eficiente e efetiva que a segunda abordagem no processo de seleção de TTbMs, e para os demais aspectos possui comportamento similar.

- **Infraestrutura Computacional de Apoio ao uso de *Porantim***

Foi construída uma infraestrutura computacional a fim de reduzir o esforço da implantação da abordagem *Porantim* em organizações de software. Tal infraestrutura foi utilizada em uma organização internacional que aplica TBM em seus projetos a fim de observar a viabilidade de seu uso. Os resultados, embora não conclusivos, indicam uma redução no tempo total gasto para seleção de TTBM, além de uma avaliação positiva do apoio provido pela infraestrutura computacional pelos profissionais que a utilizam.

- **Metodologia Científica que apoia na Concepção de Tecnologias de Software**

Nesta pesquisa foi adotada uma metodologia científica (descrita na Seção 1.3) baseada na condução de estudos secundários e primários para apoiar no desenvolvimento de tecnologias de software. Esta metodologia busca agrupar resultados providos por revisões sistemáticas da literatura e a opiniões de especialistas em um tópico de pesquisa para a construção de um corpo de conhecimento sobre diferentes áreas de domínio. Esta metodologia tem sido seguida por outras pesquisas científicas desenvolvidas por membros do Grupo de Engenharia de Software Experimental da COPPE/UFRJ.

5.2. Publicações

Tais contribuições podem ser ainda evidenciadas pelas publicações obtidas durante o desenvolvimento desta pesquisa. Ao total, foram produzidas durante este trabalho 13 publicações de diferentes categorias (Tabela 5), diretamente relacionadas ao tema da pesquisa ou representando desdobramentos do trabalho.

Tabela 5. Produções Científicas

Categoria	Fonte	Ano	Referência
Revistas e Periódicos	IEEE Software Magazine	2008	[7]
	Information and Software Technology Journal	2009	[12]
Capítulo de Livro	Advances in Computers: “A picture from the Model-Based Testing area: Concepts, Techniques and Challenges”	2010	(a ser publicado)
Conferências Internacionais	Workshop on Empirical Assessment of SW Eng. Languages and Techs.	2007	[13]
	Workshop on Automation of Software Test (AST)	2008	[23]
	International Symposium on Empirical SW Engineering and Measurement	2008	[15]
	Workshop on Automation of Software Test (AST)	2009	[11]
Conferências Nacionais/Latino Americanas	International Symposium on Empirical SW Engineering and Measurement	2009	[17]
	Simpósio Brasileiro de Qualidade e Software (SBQS)	2008	[22]
	Experimental SW Engineering Latin American Workshop (ESELAW)	2008	[19]
Relatórios Técnicos	Conferência Ibero-Americana em “Software Engineering” (CIbSE)	2010	[8]
	Technical Report of SIEMENS Corporate Research	2006	[20]
	Relatório Técnico ES-713/07 (PESC-COPPE/UFRJ)	2007	[21]

5.3. Trabalhos Futuros

Os resultados obtidos até o momento representam contribuições para a engenharia de software, principalmente no que diz respeito à área de teste baseado em modelos, seleção de tecnologias e metodologia científica. Além disso, possibilitaram a organização de um arcabouço de pesquisa científica que permite explorar novas direções de investigação na área de teste baseado em modelos, conforme exemplificado a seguir:

- Prover apoio à utilização e avaliação de TTBM em projetos de software, pois *Porantim* provê atualmente apoio apenas à tarefa de seleção de TTBM.
- Investigar se a inversão do processo de tomada decisão, de forma que a partir das características de um conjunto de TTBM que se deseja aplicar em um projeto de

software, identificar quais seriam as características e requisitos mais adequados para o projeto. Por exemplo, deseja-se aplicar uma TTBM que requer modelos desenvolvidos na notação “X”, com isso, é preciso decidir se os modelos de projeto (*design*) do software a serem desenvolvidos pelos projetistas e utilizados pelos desenvolvedores já serão desenvolvidos nesta notação, pois assim reduzimos o esforço da construção dos modelos de teste requeridos pela TTBM.

- Evolução da abordagem *Porantim* no sentido de prover apoio à seleção de TTBM para um portfólio de projetos em uma organização de software.

Agradecimentos

Agradecemos à FAPEAM pelo apoio financeiro durante a realização desta tese de doutorado. O Prof. Travassos agradece ao CNPq e FAPERJ pelo apoio ao longo desta orientação.

Referências

- [1] Basili, V. R., Rombach, H. D. (1991), “Support for comprehensive reuse”. *Software Engineering Journal* 6(5): Setembro, 303-316.
- [2] Vegas, S.; Basili, V. (2005), “A Characterization Schema for Software Testing Techniques”, *Empirical Software Engineering*, v.10 n.4, p.437-466, Outubro.
- [3] Birk, A. (1997), “Modelling the application domains of software engineering technologies”, In: *International Conference on Automated Software Engineering (ASE)*. Lake Tahoe, CA, Novembro.
- [4] Maiden, N. A. M.; Rugg, G. (1996), “ACRE: Selecting methods for requirements acquisition”, *Software Engineering Journal* 11(3): pp. 183-192.
- [5] Aranda, G. N., Vizcaino, A., Cechich, A., Piattini, M. (2006), “Technology Selection to Improve Global Collaboration”, In: *International Conference on Global Software Engineering (ICGSE)*, Outubro, pp. 223-232.
- [6] Utting, M.; Legeard, B.; (2007), “Practical Model-Based Testing: A Tools Approach”, ISBN-13: 978-0-12-372501-1, Morgan-Kaufmann.
- [7] Dias-Neto, A.C.; Subramanyan, R.; Vieira, M.; Travassos, G.H.; Forrest, S. (2008), “Improving Evidence about Software Technologies: A Look at Model-Based Testing”, *IEEE Software*, Vol. 25, Issues 3, pp 10-13, Maio.
- [8] Dias-Neto, A.C., Spínola, R.O., Travassos, G.H. (2010), “Developing Software Technologies through Experimentation: Experiences from the”, No: XIII Conferência Ibero-Americana em “Software Engineering” (CibSE 2010), Abril, Cuenca, Equador.
- [9] Pretschner, A. (2005), “Model-based testing”, *Proceedings of 27th International Conference on Software Engineering, (ICSE’05)*, pp. 722-723.
- [10] Dias-Neto, A.C. (2009), “Seleção de Técnicas de Teste Baseado em Modelos”, Tese de Doutorado defendida em 23/11/2009, COPPE-UFRJ, Rio de Janeiro, RJ, disponível em http://ese.cos.ufrj.br/~acdn/index.php?option=com_docman&task=doc_download&gid=38.
- [11] Dias-Neto, A.C.; Travassos, G.H. (2009), “Porantim: An Approach to Support the Combination and Selection of Model-Based Testing Techniques”, In: 4th Workshop on Automation of Software Test, Vancouver, Maio.
- [12] Dias-Neto, A.C.; Travassos, G.H. (2009), “Model-based Testing Approaches Selection for Software Projects”, In: *Information and Software Technology (AST’08 special edition)*, Julho, DOI: 10.1016/j.infsof.2009.06.010.

- [13] Dias-Neto, A. C., Subramanyan, R.; Vieira, M.; Travassos, G.H. (2007), “A survey on model-based testing approaches: a systematic review”. Proceedings of Workshop on Empirical Assessment of Software Engineering Languages and Technologies (WEASEL'07), Atlanta, Novembro, pp 31-36. DOI= <http://doi.acm.org/10.1145/1353673.1353681>.
- [14] Hamburg, Morris (1980), “Basic Statistics: A Modern Approach”, Journal of the Royal Statistical Society, Series A (General), Vol. 143, No. 1, 2a edição.
- [15] Dias-Neto, A.C., Travassos, G.H. (2008), “Surveying on Model Based Testing Approaches Characterization Attributes”, Proceeding of International Symposium on Empirical Software Engineering and Measurement (ESEM'08), Outubro, Kaiserslautern, Alemanha.
- [16] Berger, P. (2003), “Instanciação de Processos de Software em Ambientes Configurados na Estação TABA”, Dissertação de Mestrado COPPE/UFRJ. Rio de Janeiro.
- [17] Dias-Neto, A.C., Travassos, G.H. (2009), “Evaluation of {model-based} Testing Techniques Selection Approaches: an External Replication”, Proceeding of International Symposium on Empirical Software Engineering and Measurement (ESEM'09), Outubro, Lake Buena Vista, EUA.
- [18] Dias-Neto, A. C.; Travassos, G. H. (2006), “Maraká: Uma Infra-estrutura Computacional para Apoiar o Planejamento e Controle de Testes de Software”. In: Simpósio Brasileiro de Qualidade de Software, Vila Velha-ES, Junho.
- [19] Spínola, R. O.; Dias-Neto, A. C.; Travassos, G. H. (2008), “Abordagem para Desenvolver Tecnologia de Software com Apoio de Estudos Secundários e Primários”. In: Experimental Software Engineering Latin American Workshop, Novembro, Salvador-BA.
- [20] Dias-Neto, A.C.; Subramanyan, R.; Vieira, M.; (2006), “Characterization of Model-based Software Testing Approaches”, SIEMENS Corporate Research, Technical Report, USA.
- [21] Dias-Neto, A.C.; Subramanyan, R.; Vieira, M.; Travassos, G.H. (2007), “Characterization of Model-based Software Testing Approaches”, Relatório Técnico ES-713/07, PESC-COPPE/UFRJ. Disponível em <http://www.cos.ufrj.br/uploadfiles/1188491168.pdf>.
- [22] Dias-Neto, A.C.; Travassos, G.H.; (2008), “Estratégia para Apoiar a Seleção de Abordagens de Teste Baseado em Modelos para Projetos de Software”, No: VII Simpósio Brasileiro de Qualidade e Software, Florianópolis-SC, Junho.
- [23] Dias-Neto, A. C., Travassos, G. H. (2008), “Supporting the selection of model-based testing approaches for software projects”. Proceedings of the 3rd international Workshop on Automation of Software Test (Leipzig, Germany, May). AST '08. pp. 21-24, Maio, DOI= <http://doi.acm.org/10.1145/1370042.1370047>.