

Estratégia para Apoiar a Seleção de Abordagens de Teste Baseado em Modelos para Projetos de Software

Arilo Cláudio Dias Neto, Guilherme Horta Travassos

Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ
Caixa Postal 15.064 – 91.501-970 – Rio de Janeiro – RJ – Brasil.

{acdn, ght}@cos.ufrj.br

Abstract. *Software technologies, such as model-based testing approaches, have specific characteristics and limitations that can affect their use in software projects. It is very important to make available knowledge regarding such technologies aiming at to support its applicability in software projects. In particular, a choice of model-based testing approach can influence testing success or failure. Therefore, this paper presents the knowledge acquired from some secondary and primary studies regarding model-based testing approaches and the proposal of a strategy towards supporting their selection for software projects.*

Resumo. *Tecnologias de Software, tais como abordagens de teste baseado em modelos, possuem características e limitações específicas que afetam a sua utilização em projetos de software. É importante tornar disponível conhecimento sobre tais tecnologias para apoiar sua aplicabilidade em projetos de software. Em particular, a escolha de abordagem de teste baseado em modelos pode influenciar no sucesso ou fracasso dos testes. Portanto, neste artigo é apresentado o conhecimento adquirido a partir de estudos secundários e primários relacionados a abordagens de teste baseado em modelos e proposta uma estratégia visando apoiar a seleção dessas abordagens para projetos de software.*

1. Introdução

Teste Baseado em Modelos (TBM) consiste em uma técnica para geração automática de casos de teste usando modelos extraídos de artefatos de software (Dalal *et al.*, 1999). De acordo com (El-Far e Whittaker, 2001), uma série de benefícios podem ser esperados com a utilização de TBM no processo de testes em uma organização de software, tais como: cronogramas mais curtos; menores custo e esforço; melhor qualidade; facilidade na comunicação entre as equipes de desenvolvimento e testes; simplificação na exposição de ambigüidades na especificação e projeto do software; capacidade de gerar e executar automaticamente muitos testes úteis e não repetitivos; facilidades na atualização do conjunto de casos de teste após mudanças nos requisitos e; capacidade para avaliar cenários de teste de regressão.

De acordo com Dalal *et al.* (1999), a automação de uma Abordagem de Teste Baseado em Modelos (ATBM) depende de três elementos chaves: **(i)** o modelo usado para descrição do comportamento/estrutura do software, **(ii)** o algoritmo (critérios) de geração dos testes e **(iii)** ferramentas que provêm uma infra-estrutura de apoio para os testes. No entanto, outras características adicionais podem ser adicionadas a estas com o objetivo de melhorar o entendimento sobre uma ATBM e seu funcionamento, além de possibilitar uma melhor classificação e comparação entre as abordagens. Exemplos dessas características adicionais são: nível de teste avaliado por uma ATBM, nível de automação e complexidade dos passos não-automatizados.

Dias Neto *et al.* (2007a e 2007b) identificaram e caracterizaram cerca de 71 ATBM's disponíveis na literatura técnica a partir de uma Revisão Sistemática (Biolchini *et al.*, 2005), analisando tais abordagens quantitativamente e qualitativamente e considerando diferentes cenários de abstração, tais como: o tipo de modelo usado para geração dos testes, o nível de teste que a abordagem está apta a ser aplicada, se a abordagem é aplicada para teste funcional ou estrutural, a existência de ferramentas de apoio, seus critérios de geração dos testes, dentre outras informações. Considerando os resultados descritos por Dias Neto *et al.* (2007a), observou-se que a maioria das abordagens não foi avaliada experimentalmente. Portanto, não foi possível identificar conhecimento/evidência suficiente a respeito da escalabilidade, desempenho, efetividade e complexidade de tais abordagens para serem aplicadas na indústria. Dessa forma, a seleção de uma ATBM para um projeto de software torna-se uma tarefa complexa, de forma que escolhas não-adequadas podem invalidar ou tornar inviável os testes para um projeto de software (Dias Neto e Travassos, 2008).

Baseado neste cenário, este artigo descreve um conjunto de resultados obtidos a partir de estudos secundários e primários conduzidos ao longo do desenvolvimento deste trabalho. Inicialmente é descrito um estudo secundário para caracterização das ATBM's disponíveis e são apresentados fatores de risco identificados na literatura técnica que podem influenciar na seleção e utilização de ATBM's em projetos de software. Em seguida são descritos os resultados preliminares de um *survey* cujo objetivo é identificar o grau de relevância dos diferentes atributos que caracterizam uma ATBM no momento da escolha de uma abordagem para um projeto de software, e por fim, é apresentada uma proposta de estratégia a ser apoiada por uma infra-estrutura computacional, descrita inicialmente de forma resumida em (Dias Neto e Travassos, 2008), para apoiar a seleção de ATBM's para projetos de software. O principal resultado esperado com este trabalho é fornecer mecanismos e informações a respeito das diferentes ATBM's disponíveis para uso em um projeto para que a equipe de teste possa estar provida de mais conhecimento no momento da realização de uma das tarefas mais complexas do processo de teste, a seleção de uma abordagem de teste para um projeto.

Alguns trabalhos relacionados estão publicados na literatura técnica descrevendo mecanismos de apoio à seleção de técnicas de teste (ou outros tipos de técnicas) para um projeto de software. Por exemplo, em (Vegas e Basili, 2005) é descrito um mecanismo para apoiar a seleção de técnicas de teste a partir de um esquema de caracterização. Os atributos das técnicas de teste são armazenados em um repositório, e para cada projeto de software um catálogo é gerado. Os aspectos positivos da proposta de Vegas e Basili (2005) são tornar conhecimento disponível e simplificar a escolha de uma técnica de teste de propósito geral para projetos de software. No entanto, a generalidade de seus atributos torna difícil a sua customização para técnicas baseadas em modelos, o que pode representar um risco para um projeto. Em um outro trabalho (Juristo *et al.*, 2004) é descrito um estudo experimental indicando diferenças e similaridades entre técnicas de teste. Apesar de seu propósito não ser apoiar a seleção de técnicas de teste, testadores podem usar este conhecimento para apoiar esta tarefa. Outros trabalhos similares descrevem a caracterização/seleção de métodos, técnicas e ferramentas relacionadas ao desenvolvimento de software [ex: (Birk, 1997) – caracterização de tecnologias de software, (Maiden e Rugg, 1996) – seleção de técnicas de identificação de requisitos]. Apesar da importância desses trabalhos, na literatura técnica não foi encontrada a definição de alguma estratégia ou infra-estrutura para seleção de ATBM's para projetos.

Este artigo está organizado da seguinte forma: Na seção 2 é descrita uma

caracterização da área de TBM a partir de seus conceitos, uma revisão sistemática sobre ATBM's e uma análise de fatores de risco que podem influenciar a utilização de ATBM's em projetos de software. Na seção 3 é apresentada a proposta de estratégia de apoio à seleção de ATBM's para projetos de software. Finalmente, na seção 4 são discutidas as conclusões e trabalhos futuros.

2. Caracterização da Área de Teste Baseado em Modelos

2.1. Definição de Teste Baseado em Modelos

De acordo com Myers (1979), o principal fator que influencia o custo associado à atividade de teste é o número total de casos de teste. Para cada caso de teste, é necessário alocar recursos para planejar, projetar e executá-lo. Portanto, a automação da geração e execução dos casos de teste pode ser um mecanismo interessante para reduzir o custo e esforço dessa atividade.

Neste contexto, Teste Baseado em Modelos surge como uma abordagem viável para controlar a qualidade do software reduzindo o custo e esforço associado a algumas atividades do processo de teste, visto que os casos de teste são gerados a partir de artefatos (modelos) construídos ao longo do processo de desenvolvimento de software.

Uma ATBM segue, normalmente, os seguintes passos:

- Construção do modelo de software;
- Geração dos Casos de Teste;
 - Geração das entradas necessárias;
 - Geração dos resultados esperados;
- Execução dos Testes;
- Comparação entre os resultados obtidos e os resultados esperados;
- Decisão a respeito das futuras ações (se modificar o modelo de software, gerar mais casos de teste, parar os testes ou estimar a confiabilidade [qualidade] do software);

Uma estratégia de teste baseado em modelos normalmente inclui diferentes níveis de abstração, um modelo descrevendo o comportamento/estrutura do software, um relacionamento entre modelos e código fonte, tecnologia de geração dos casos de teste, a importância dos critérios de geração dos testes e uma discussão sobre o quê pode ou não ser automatizado durante os testes (Pretschner, 2005). Esses diferentes atributos tornam a identificação e caracterização de ATBM's uma atividade importante, conforme descrito na seção seguinte.

2.2. Revisão Sistemática sobre Abordagens de Teste Baseado em Modelos

Em (Dias Neto *et al.*, 2007a; Dias Neto *et al.*, 2007b) é descrita uma Revisão Sistemática (estudo secundário) com o propósito de identificar, caracterizar e analisar quantitativamente e qualitativamente ATBM's publicadas na literatura técnica. No total, 406 artigos foram encontrados. Após a leitura dos artigos, este conjunto foi reduzido para 202 artigos descrevendo ATBM's. Atualmente, a análise dos dados tem sido direcionada para 85 artigos que descrevem 71 ATBM's e que foram selecionados a partir de critérios previamente definidos, tais como o número de citações, ano de publicação e nível de teste. A lista completa com todos os artigos identificados e os critérios usados para seleção/rejeição deles está disponível em (Dias Neto *et al.*, 2007a).

Para apoiar a caracterização de ATBM's, catorze atributos identificados a partir

de uma revisão informal de literatura, principalmente a partir de (Dalal *et al.*, 1999), (Pretschner, 2005) e (Utting *et al.*, 2006), foram usados para extrair informações a respeito dos artigos selecionados: (1) Modelo Comportamental/Estrutural do Software; (2) Nível de Teste; (3) Ferramentas de Apoio; (4) Domínio do Software; (5) Critérios de Cobertura dos Testes; (6) Critério de Geração dos Testes; (7) Nível de Automação; (8) Nível de Complexidade dos Passos Não-Automatizados; (9) Tecnologia de Geração dos Testes; (10) Tipo de Técnica de Teste (Funcional ou Estrutural); (11) Característica de Qualidade de Software que a abordagem possibilita avaliar; (12) Saídas geradas pela abordagem; (13) Entradas requeridas pela abordagem; (14) Limitações/Restrições para usar a abordagem. Uma descrição detalhada sobre cada atributo pode ser obtida em (Dias Neto *et al.*, 2007b).

Baseado nos atributos descritos anteriormente, uma análise quantitativa foi realizada possibilitando observar as seguintes características:

- 63% das ATBM's são aplicadas no nível de Teste de Sistema;
- Os passos não-automatizados estão associados à construção/transformação do modelo ou seleção dos critérios de geração dos testes.
- Diagramas UML, Máquinas de Estado Finito e Grafos de Fluxo de Eventos são os modelos que têm sido usados por um grande número de abordagens (mais de 70%).
- A maioria das ATBM's possui uma ferramenta de apoio disponível (60%). No entanto, elas são normalmente protótipos e não representam produtos prontos para serem comercializados.
- 54% das ATBM's usam modelos intermediários para reduzir o escopo e a complexidade do problema antes da geração dos casos de teste.

Em outra perspectiva, os resultados obtidos a partir de uma análise qualitativa descreveram uma discussão individual e subjetiva a respeito das abordagens selecionadas, tais como: limitações, indicações de custo, esforço, nível de complexidade, entradas requeridas, pré-requisitos, formato e qualidade das saídas geradas e habilidade necessária para usar uma abordagem.

As informações descritas nesta seção possuem a intenção apenas de justificar as razões a respeito da pesquisa que está sendo conduzida e que será descrita ao longo deste artigo. Para aqueles que se interessarem em maiores detalhes a respeito dos resultados da revisão sistemática realizada, isso pode ser obtido em (Dias Neto *et al.*, 2007a; Dias Neto *et al.*, 2007b).

2.3. Fatores de Risco que influenciam o uso de Abordagens de Teste Baseado em Modelos em Projetos de Software

Uma ATBM normalmente é desenvolvida para ser aplicada exclusivamente nas atividades de teste em um projeto de software específico, e normalmente suas avaliações descritas nos artigos são realizadas para exemplos simples e direcionados a um cenário. No entanto, em projetos reais, a atividade de teste é apenas uma das atividades a serem realizadas ao longo do processo de desenvolvimento de um software. Além disso, teste, como todas as demais atividades, deve atender às restrições impostas pelo planejamento do projeto, tais como tecnologia, cronograma, orçamento e recursos.

Dado esse cenário, foi iniciado um mapeamento a partir da literatura técnica de um conjunto de fatores de riscos que podem influenciar a utilização de uma ATBM em projetos de software. Para isso, doze fatores de risco puderam ser identificados (Tabela 1). Alguns fatores são exclusivos do uso de abordagens de teste baseado em modelos e

outros são aplicados ao processo de teste, independente do tipo de técnica adotada. Independentemente disso, esses fatores indicam cenários incluídos no processo de testes que não fazem parte do escopo de uma ATBM, mas que precisam ser gerenciados pois podem tornar inviáveis os testes gerados para um projeto de software.

Tabela 1. Fatores de Risco que afetariam a utilização de ATBM's em projetos

Id	Fator de Risco	Descrição
1	Garantia da qualidade dos artefatos usados por uma ATBM para geração de testes	O sucesso dos testes está diretamente associado à qualidade dos artefatos recebidos como entrada para construção do modelo comportamental/estrutural a ser usado para geração dos testes.
2	Mecanismo eficiente para disponibilizar os artefatos de software usados como entrada para as ATBM's	A equipe de teste deve ter acesso constante às versões correntes dos artefatos usados por uma ATBM.
3	Estratégia para alocação de recursos (humanos e físicos) e planejamento do cronograma dos testes	A habilidade e grau de complexidade requerido para usar cada ATBM são diferentes. Eles impactam diretamente na seleção da equipe de teste e definição do orçamento e cronograma dos testes em um projeto de software.
4	Estratégia para seleção de ATBM's para um projeto	Diferentes atributos devem ser observados no momento da seleção de ATBM's a serem usadas em projetos de software, tais como: cobertura dos testes desejada ou as características de qualidade do produto que precisam ser avaliadas.
5	Estratégia adotada para construção do modelo comportamental/estrutural do software	ATBM's não consideram as variáveis a respeito do custo, esforço ou estratégia de construção dos modelos de software usados para geração dos testes. Estes modelos são os principais requisitos para se usar uma ATBM.
6	Estratégia adotada para a escolha dos critérios de geração/cobertura dos testes	A escolha de qual critério de geração dos testes usar em um software ou em um módulo específico do software deve ser baseada no esforço ou custo para construção e execução dos casos de testes, o número total de casos de teste gerados e, principalmente, do nível de cobertura dos testes desejado.
7	Rastreamento e análise de impacto de mudanças na especificação do software	É necessário analisar quais mudanças ocorreram na especificação do software e quais são os seus impactos no conjunto de casos de teste já gerados para um software antes que um novo modelo seja construído para re-gerar os testes.
8	Atualização dos casos de teste após mudanças na especificação do software	ATBM's não indicam como atualizar seus modelos, qual o esforço e custo, ou a viabilidade para realizar esta tarefa. Elas normalmente consideram que um novo modelo já está criado para re-gerar casos de testes, não provendo informações essenciais para o controle do processo de teste.
9	Inclusão manual de casos de teste extras	Eventualmente, pode ser necessário incluir casos de teste para avaliar funcionalidades ou características do software não cobertas por uma ATBM.
10	Controle do processo de geração e execução dos testes	A equipe de teste deve sempre ser provida com informações a respeito do estágio atual do processo de testes.
11	Rastreamento de falhas reveladas pelas ATBM's	O rastreamento de falhas apóia o controle dos testes e futuras avaliações das ATBM's e seus critérios de seleção dos testes, associando falhas aos casos de teste responsáveis pela sua identificação.
12	Avaliação de ATBM's e seus critérios de geração dos testes	A avaliação de ATBM's e seus critérios de seleção dos testes pode tornar mais simples e mais precisa as estimativas de custo e esforço e também apoiar a seleção de ATBM's para projetos de software.

Após a caracterização das ATBM's e a identificação de um conjunto de fatores de risco que podem influenciar sua utilização em projetos de software, pensou-se em mecanismos para mitigar tais riscos. A próxima seção irá discutir e apresentar a estratégia adotada para o desenvolvimento desses mecanismos visando ao desenvolvimento de uma estratégia para apoiar a seleção de ATBM's para serem aplicadas em projetos de software.

3. Estratégia para Construção de um Mecanismo de Apoio à Seleção de

Abordagens de Teste Baseado em Modelos para Projetos de Software

Como observado nas seções anteriores, existem diversas ATBM's com diferentes características disponíveis para uso em projetos de software. A seleção de qual delas aplicar de acordo com as características específicas do projeto de software deve ser realizada cuidadosamente e a partir de critérios bem definidos. Escolhas inadequadas podem aumentar significativamente a complexidade, esforço e custo para utilizar uma ATBM, e pode ainda tornar inviável os testes gerados por uma abordagem não adaptável ao projeto em questão.

As próximas subseções irão apresentar a estratégia definida visando ao desenvolvimento de uma estratégia, cuja proposta foi apresentada inicialmente de forma simplificada em (Dias Neto e Travassos, 2008), definida para apoiar a tarefa de seleção de ATBM's para projetos de software.

3.1. Corpo de Conhecimento sobre Abordagens de Teste Baseado em Modelos

O primeiro passo visando ao desenvolvimento da estratégia proposta consiste em conhecer quais ATBM's estão disponíveis e respectivamente quais são suas principais características. Isso possibilitará a criação de um corpo de conhecimento a respeito de ATBM's. A existência de um corpo de conhecimento pode simplificar a análise sobre qual abordagem é mais adequada para um projeto de software com certas características.

Atualmente este corpo de conhecimento é representado pelos 202 artigos publicados na literatura técnica e capturados através da revisão sistemática descrita em (Dias Neto *et al.*, 2007a). Ele tem sido organizado e estruturado por meio de um banco de dados da ferramenta JabRef¹, contendo todos os atributos relevantes (seção 2.2) preenchidos com informações extraídas a partir dos artigos originais. Alguns atributos representam categorias com valores pré-definidos a fim de apoiar a classificação, agrupamento e conseqüentemente localização de ATBM's (ex: o nível de testes). Outros terão seus conteúdos livres possibilitando uma caracterização individual de tais abordagens (ex: restrições ou limitações).

Após a estruturação e instanciação do corpo de conhecimento, foi identificada uma funcionalidade adicional que deveria compô-lo. Observou-se que cada atributo caracterizando uma ATBM pode ter um nível de relevância diferenciado no momento da seleção de uma ATBM para um projeto de software. Assim, pesos devem ser associados a cada atributo quando estiver sendo realizada a tarefa de seleção de uma abordagem.

No entanto, a definição dos pesos não pode ser feita aleatoriamente ou sem qualquer critério. Com isso, um *survey* foi planejado e tem sido executado visando à identificação dos níveis de relevância de cada atributo de caracterização no momento da seleção de uma ATBM para um projeto de software. A próxima subseção descreverá este *survey* com maiores detalhes.

3.2. Survey sobre Atributos de Abordagens de Teste Baseado em Modelos

O objetivo deste estudo primário é analisar um conjunto de atributos que descrevem ATBM's extraídos da literatura técnica, e apresentados na seção 2.2, com respeito à sua relevância no contexto de apoio à seleção de ATBM's para projetos de software no ponto de vista de pesquisadores ao redor do mundo trabalhando com pesquisa e desenvolvimento de soluções em TBM. E suas questões de pesquisa são:

¹ <http://jabref.sourceforge.net>

- Os atributos extraídos da literatura técnica (e apresentados na seção 2.2) são importantes (ou não importantes) para caracterizar uma ATBM?
- Existe qualquer atributo adicional considerado importante para caracterizar ATBM's que não está apresentado no conjunto inicial?
- Qual a ordem de relevância dos atributos de ATBM's que compõem o conjunto final de atributos no momento da seleção de uma abordagem para projetos de software?

Este *survey* já passou pelas fases de planejamento e execução². O questionário foi elaborado no idioma Inglês e ficou disponível na Internet no período de 10/12/2007 a 10/02/2008. Os participantes do estudo foram contatados por e-mail (ao total cerca de 100 pesquisadores identificados dentre os autores que compõem o corpo de conhecimento sobre ATBM foram convidados), onde receberam um login e senha de acesso, evitando participantes não planejados e conseqüentemente a introdução de viés nos resultados. O preenchimento do questionário é seguido em três passos:

1. Caracterização do conhecimento e habilidades dos participantes (Figura 1). Neste passo os participantes são questionados sobre seus dados pessoais (nome, e-mail, afiliação e país), nível de formação acadêmica, número de artigos publicados sobre TBM e nível de experiência em TBM na indústria;

Survey on Important Characteristics when selecting MBT Approaches for a Software Project

1 **Subject Characterization** ▶
 2 **Characterizing MBT Approaches** ▶
 3 **Selecting MBT Approaches**

How to proceed: Fill in all fields below accordingly your personal information.
(* Required fields. E-mail is required only for access control)

Name: *** E-mail:**
Affiliation: **Country:**

*** Higher Academic Degree:**
 Undergraduation Specialization Master Degree Ph.D / D.Sc

*** Number of papers published regarding Model-Based Testing:**
 1 to 5 6 to 10 11 to 20 more than 20

*** Experience Level regarding the use of MBT Approaches in software projects:**
 Low Medium High Excellent

*** Estimated number of software projects using MBT Approaches that you have participated:**

* I agree to participate of this survey.

[Start the Survey](#)

Figura 1. Tela de Caracterização e Autorização do Participante

2. Identificação dos atributos importantes ou não importantes para caracterizar uma ATBM (Figura 2). Para cada atributo, o participante deve preencher se ele é ou não importante para a caracterização de uma ATBM, ou seja, se ele deve compor o corpo de conhecimento sobre ATBM's. Além disso, o participante pode inserir 5 atributos adicionais que ele considere importante e que não estão incluídos no

² O plano do estudo pode ser obtido com os autores do trabalho.


conjunto inicial;

Survey on Important Characteristics when selecting MBT Approaches for a Software Project

① Subject Characterization ▶ ② **Characterizing MBT Approaches** ▶ ③ Selecting MBT Approaches

STEP 2: Identification of the important information to characterize a MBT Approach

How to proceed: Identify for each information, if it is important or not to characterize a MBT Approach. The importance indicates that the information is useful to describe or to define a body of knowledge regarding MBT Approaches.

PS: If you move the mouse over the icon , a description of the associated characteristic is presented.











MBT Approach Characteristics	Is it important?
 Behavioral/Structural Model used for Test Generation	<input type="radio"/> Yes <input type="radio"/> No
 Complexity Level to execute Non-automated steps	<input type="radio"/> Yes <input type="radio"/> No
 Indication of MBT Approach Experimental Evaluation	<input type="radio"/> Yes <input type="radio"/> No
 Indication of Supporting Tools availability	<input type="radio"/> Yes <input type="radio"/> No
 Inputs required to use a MBT Approach	<input type="radio"/> Yes <input type="radio"/> No
 Limitations/Restrictions to use a MBT Approach	<input type="radio"/> Yes <input type="radio"/> No
 Needs for External Tools to support steps of a MBT approach	<input type="radio"/> Yes <input type="radio"/> No
 Needs for the development of Intermediate Models between the behavioral/structural model and test cases generation	<input type="radio"/> Yes <input type="radio"/> No
 Outputs Generated by a MBT Approach	<input type="radio"/> Yes <input type="radio"/> No
 Proportion of Automated Steps per Total number of Steps composing a MBT Approach	<input type="radio"/> Yes <input type="radio"/> No




























Figura 2. Tela de Identificação dos atributos importantes para caracterizar uma ATBM

- Definição do nível de relevância de cada atributo de ATBM quando apoiando a seleção de uma abordagem para um projeto de software (Figura 3). Neste passo, apenas os atributos indicados pelo participante como importante no PASSO 2 são avaliados. Seis níveis de relevância foram definidos: (0) Sem relevância; (1) Muito Baixa Relevância; (2) Baixa Relevância; (3) Média Relevância; (4) Alta Relevância; (5) Muito Alta Relevância.

Survey on Important Characteristics when selecting MBT Approaches for a Software Project

① Subject Characterization ▶ ② Characterizing MBT Approaches ▶ ③ **Selecting MBT Approaches**

STEP 3: Relevance Level of each characteristics when selecting a MBT Approach for a software project

MBT Approach Characteristics	Relevance Level
 Behavioral/Structural Model used for Test Generation	<input checked="" type="checkbox"/>        
 Complexity Level to execute Non-automated steps	<input checked="" type="checkbox"/>        
 Indication of MBT Approach Experimental Evaluation	<input checked="" type="checkbox"/>        



 Prior Step Finish the survey 

Figura 3. Tela de Definição do Nível de Relevância para a Seleção de ATBM's

Para a análise dos dados, cada participante deve possuir um peso diferenciado de acordo com seu nível de conhecimento e habilidade. Pesquisadores com níveis maiores de experiência ou habilidade devem possuir no estudo um peso maior. Após a definição

dos pesos, as respostas de todos os participantes devem ser analisadas para cada atributo de ATBM avaliado, e ao final a ordem de relevância dos atributos será obtida.

Até o momento da escrita deste artigo, 34 pesquisadores responderam ao *survey*. Alguns resultados preliminares já foram obtidos considerando os dados já disponíveis. No entanto, eventualmente esses resultados podem mudar quando novos participantes inserirem suas respostas. Assim, a fase de coleção dos dados precisa ser finalizada para que a fase de análise possa iniciar efetivamente.

Como resultado preliminar, a Tabela 2 apresenta a ordem dos atributos avaliados a partir dos dados coletados até o momento em relação à sua importância para caracterização de uma ATBM e em relação ao grau de relevância no momento da seleção de uma abordagem para um projeto. As setas na terceira coluna da Tabela 2 indicam uma variação (ou não) da ordem do atributo na classificação por grau de relevância em relação à sua classificação pelo grau de importância. Observa-se que os resultados tendem, inicialmente, a confirmar a importância de todos os atributos avaliados para caracterização de uma ATBM (mesmo aqueles não descritos com tanta intensidade nos artigos que descrevem tais abordagens, tais como: avaliação experimental ou grau de complexidade para usar uma ATBM). Além disso, os atributos com maior grau de relevância no momento da seleção de uma ATBM para um projeto de software são aqueles relacionados aos requisitos básicos para se utilizar uma abordagem, tais como modelo estrutural/comportamental, critérios de geração/cobertura dos testes e restrições/limitações para se usar uma abordagem.

Tabela 2. Resultados Iniciais obtidos no Survey

Atributos de ATBM's	Ordem por grau de Importância	Ordem por Grau de Relevância
Modelo Comportamental/Estrutural	1	1
Critério de Geração de Casos de Teste	2	3
Critério de Cobertura dos Testes	3	2
Entradas requeridas para usar uma abordagem	4	4
Limitações/Restrições para usar uma abordagem	5	5
Característica de Qualidade de Software que a abordagem está apta a avaliar	6	9
Nível de Teste	7	8
Tipo de Técnica de Teste (Funcional ou Estrutural)	8	6
Domínio de Software	9	11
Resultados gerados pela abordagem	10	7
Ferramenta de Apoio	11	10
Resultados Históricos	12	15
Uso de Modelos Intermediários	13	12
Tecnologia de Geração dos Testes	14	13
Nível de Complexidade dos Passos Não-automatizados	15	18
Avaliação Experimental	16	17
Necessidade de Ferramentas Externas	17	14
Proporção de Passos Automatizados	18	16

Apesar de ter sido apresentada a ordem dos atributos, o resultado relevante a ser obtido com este *survey* é o peso (grau de relevância – valor entre 0% e 100%) de cada atributo no momento da seleção de uma ATBM para um projeto de software. Após definidos, esses valores irão compor o corpo de conhecimento sobre ATBM's descrito na seção 3.1. Consequentemente, eles poderão apoiar o desenvolvimento de um

mecanismo que indique abordagens mais adequadas a um projeto de software, dadas suas características, como será apresentado na próxima subseção.

3.3. Infra-estrutura de Apoio à Indicação/Seleção de ATBM's adequadas a Projetos de Software

Na infra-estrutura proposta, a seleção de ATBM's para um projeto de software será baseada em algumas etapas (ver Figura 4). Esta seção apresenta os requisitos que compõem a infra-estrutura, indicando como eles têm sido desenvolvidos.

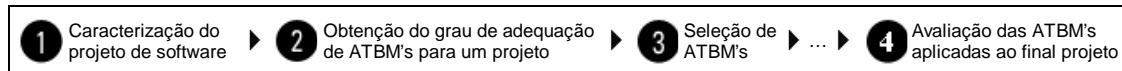


Figura 4. Etapas que compõem a Infra-estrutura de Apoio à seleção de ATBM's

3.3.1. Caracterização de Projetos de Software

A primeira etapa que compõe a infra-estrutura de apoio à seleção de ATBM's consiste em conhecer as principais características do projeto no qual uma ATBM precisa ser aplicada. A estratégia adotada para caracterização de um projeto de software neste trabalho consiste na definição de algumas informações/atributos a respeito de um projeto de software e que foram extraídas, principalmente, da funcionalidade responsável pela caracterização de projetos de software definida na ferramenta ADAPTPRO (Berger, 2003) que compõe a Estação TABA, e evoluídas a partir dos resultados do *survey* realizado.

As informações utilizadas na versão inicial da infra-estrutura apresentada para caracterização de um projeto de software são: plataforma de execução, paradigma ou linguagem de desenvolvimento, domínio da aplicação, modelo de ciclo de vida adotado, duração estimada do projeto, formato dos requisitos de software, características de qualidade que precisam ser avaliadas, tecnologia usada para modelagem do software, tamanho do projeto, complexidade do problema. No entanto, essa lista de características pode evoluir ao longo desta pesquisa e pode ainda sofrer variações de acordo com o tipo ou domínio da aplicação a ser desenvolvida no projeto.

Esta caracterização será usada para indicar ATBM's incluídas no corpo de conhecimento que são mais adequadas às características do projeto em questão.

3.3.2. Obtenção do Nível de Adequação e Indicadores para a Seleção de ATBM's

A escolha de uma ATBM para um projeto de software não pode ser realizada automaticamente, ou seja, sem a confirmação de um membro da equipe de teste, pois aspectos externos ou não-técnicos que eventualmente não são passíveis de automação podem afetar esta escolha. Portanto, a infra-estrutura proposta neste trabalho apresenta como funcionalidade a indicação de diferentes opções de ATBM's mais adequadas para um projeto a partir do cálculo de um nível de adequação de cada ATBM para o projeto de software em questão.

Este nível de adequação é calculado pela infra-estrutura para cada ATBM incluída no corpo de conhecimento. Para isso, está sendo utilizado o conceito matemático de distância euclidiana (Boldrini *et al.*, 1980), que foi adotado por Xavier *et al.* (2002) para apoiar a seleção de padrões arquiteturais em projetos de software. Segundo Xavier *et al.* (2002), este mecanismo de busca explora a possibilidade de avaliação de distâncias conceituais a partir da comparação de elementos em um espaço vetorial multidimensional. A noção de distância conceitual é realizada,

matematicamente, pela norma da diferença entre dois vetores $v1$ e $v2$. Neste trabalho, os vetores são representados pelas características do projeto ($v1$) e características de cada ATBM ($v2$). A distância entre eles indica o grau de adequação de uma ATBM para o projeto de software. Quanto mais próximos eles estiverem, mais adequada ao projeto é tal abordagem ($grau\ de\ adequação = [1 - distância] * 100$) (ver exemplo da Figura 5).

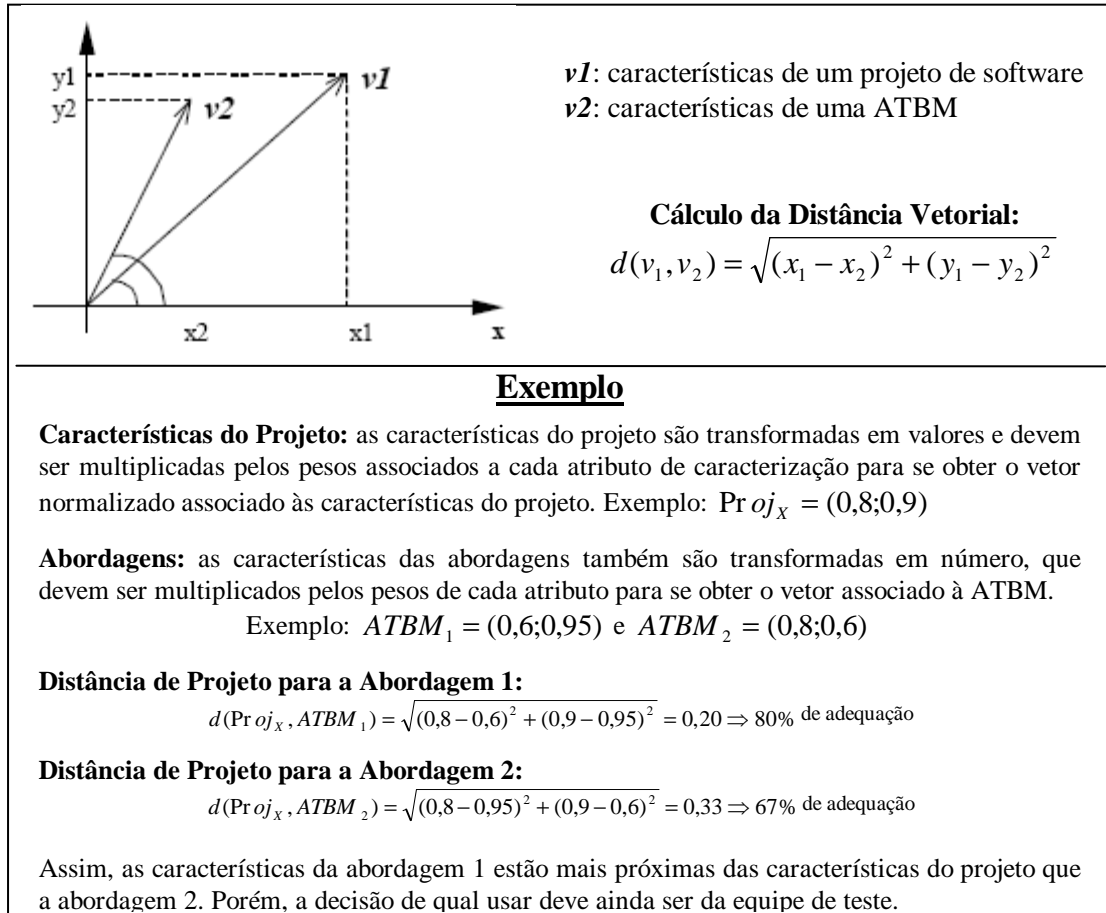


Figura 5. Distância entre vetores e exemplo do cálculo da distância entre eles

Após o cálculo da distância das características de cada abordagem em relação às características do projeto, a infra-estrutura ordena as abordagens e exibe aquelas com os níveis de adequação mais altos agrupadas por nível de teste (o número de ATBM's sugeridas pela infra-estrutura é definido como parâmetro). Além do nível de adequação calculado para cada abordagem, a infra-estrutura provê ainda informações a respeito de como o nível de adequação de uma ATBM para um projeto foi obtido a partir do cruzamento das características do projeto com as características da ATBM selecionada. A Tabela 3 exibe um exemplo de tela de detalhamento de nível de adequação.

Tabela 3. Exemplo de Tela de Detalhamento de Nível de Adequação

Características do Projeto	Valores	Nível de Adequação
Plataforma de Execução	Web	10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
Paradigma de Desenvolvimento	OO	10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
Linguagem de Programação	Java	10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Modelo de Ciclo de Vida	Incremental	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Duração Estimada do Projeto	4 meses	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Formato dos Requisitos do Software (Funcional e Não-funcional)	Textual	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Características de Qualidade do Software a serem avaliadas	Funcionalidade Segurança	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Tecnologia usada para modelagem do software	UML	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Complexidade do Problema	Alta	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Frequência de Mudança dos Requisitos	Alta	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Tamanho da Aplicação	Médio	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Recursos Humanos disponíveis para o projeto	Baixo	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █
Tempo disponível para o projeto	Baixo	10% 20% 30% 40% 50% 60% 70% 80% 90% 100% █

3.3.3. Seleção/Combinação de Abordagens de Teste Baseado em Modelos

A partir do cálculo dos níveis de adequação de ATBM's ao projeto de software, a equipe de teste poderá optar pela seleção de uma ou mais abordagens para serem aplicadas no projeto. Cada ATBM pode ser apta a cobrir uma característica específica do software para um certo domínio de aplicação, e cada uma requer um determinado nível de esforço e custo para ser aplicada em um projeto de software. No entanto, quando mais que uma abordagem é selecionada, a equipe de teste precisa ser apoiada por informações a respeito da cobertura ou nível de adequação, esforço/custo e o nível de recursos necessário para utilizar as abordagens selecionadas de forma combinada.

A infra-estrutura apresentada neste trabalho fornece tais informações combinando as características individuais das abordagens selecionadas com as características do projeto de software. No momento da seleção das abordagens, esses indicadores podem ser visualizados através de um gráfico do tipo "speedometer". Esta escolha de gráfico ocorreu por este tipo de gráfico visualizar facilmente diferentes

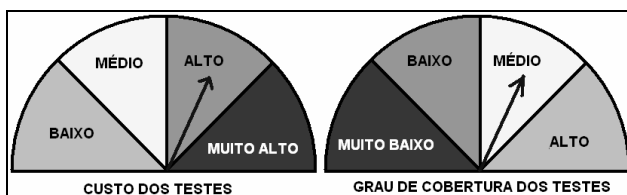


Figura 6. Exemplo de gráfico indicando custo e cobertura dos testes para uma certa combinação de ATBM's

faixas de valores representando categorias que são separadas por limites pré-estabelecidos. O marcador do "velocímetro" indica quanto o valor obtido atualmente está posicionado em relação a uma meta definida. Por exemplo, a Figura 6 apresenta um exemplo demonstrativo do uso deste gráfico no contexto deste trabalho.

3.3.4. Medição e Avaliação de Abordagens de Teste Baseado em Modelos

O corpo de conhecimento a respeito de ATBM's deve ser atualizado constantemente. Assim, a infra-estrutura apresentada provê um mecanismo de apoio à avaliação de uma ATBM's e seus critérios de geração dos testes ao final de cada projeto de software. Neste momento, dois tipos de informações são obtidos: métricas são coletadas

automaticamente do processo de testes (ex.: número de casos de teste gerados por cada critério de geração dos testes, esforço para construção dos modelos, número de incidentes de teste por caso de teste, dentre outros), além dessas informações, existe um questionário ao final de cada processo de teste a ser preenchido pela equipe de teste com itens a respeito da adequação das abordagens utilizadas de acordo com as características do projeto onde elas foram aplicadas.

A presença deste mecanismo de avaliação das ATBM's ao final do projeto permite que as informações contidas no corpo de conhecimento sobre ATBM's que compõem a infra-estrutura apresentada estejam sempre atualizadas, de forma que os indicadores apresentados por ela sejam sempre confiáveis para a equipe de teste. A Figura 7 exibe um extrato exemplificando a tela de avaliação de ATBM, separando claramente as informações obtidas automaticamente do processo de testes daquelas a serem preenchidas pela equipe de testes. Os dados apresentados são apenas ilustrativos.

Medidas coletadas automaticamente do processo de testes		
Avaliação a respeito dos Critérios de Geração dos Testes	Medidas	
- Todos os caminhos	95% de cobertura dos testes	
Número de Casos de Teste	156 casos de teste	
Número de Incidentes de Teste	46 incidentes de teste	
- Fluxos Principais	88% de cobertura dos testes	
Número de Casos de Teste	143 casos de teste	
Número de Incidentes de Teste	30 incidentes de teste	
Avaliação a respeito do Modelo Comportamental/Estrutural	Medidas	
Número de Casos de Uso	15 casos de uso	
Número de Modelos construídos	8 modelos	
Esforço para construir os modelos	100 horas	
Avaliação a respeito dos Testes de Regressão	Medidas	
Número de Modelos atualizados	5 modelos	
Esforço para atualizar os modelos	45 horas	
Informações sobre adequabilidade da ATBM ao projeto a serem preenchidas pela equipe de teste		
Características	Valores	Avaliação (preenchida por um membro)
Plataforma de Execução	Web	
Paradigma de Desenvolvimento	OO	
Linguagem de Programação	Java	
Domínio de Aplicação	Financeiro	
Modelo de ciclo de vida	Incremental	
Duração Estimada do Projeto	4 meses	
Formato dos Requisitos do Software (Funcional e Não-funcional)	Textual	
Características de Qualidade do Software a serem avaliadas	Funcionalidade	
Características de Qualidade do Software a serem avaliadas	Segurança	
Tecnologia usada para Modelagem do Software	UML	

Figura 7. Exemplo de Tela de Avaliação de ATBM

Um protótipo (ferramenta computacional) para apoiar a estratégia descrita neste trabalho já está definido. Esta infra-estrutura está sendo desenvolvida sobre a plataforma PHP+MySQL como uma extensão da infra-estrutura de apoio ao planejamento e controle de teste de software, Maraká (Dias Neto e Travassos, 2006).

No entanto, apesar das facilidades providas por um apoio ferramental, a parte essencial que compõe esta pesquisa é o conteúdo provido pela estratégia proposta, ou seja, o conhecimento científico sobre ATBM's a ser disponibilizado a fim de simplificar a implantação deste tipo de técnica de teste em organizações de software. Sendo assim, a continuidade desta pesquisa será direcionada para a avaliação experimental da infraestrutura apresentada neste artigo em relação ao conhecimento científico contido no seu corpo de conhecimento e aos indicadores providos por ela como mecanismo de apoio à seleção de ATBM's para projetos de software.

4. Conclusões e Trabalhos Futuros

Este artigo apresentou uma estratégia que apóia a seleção de ATBM's para projetos de software. A principal característica associada a este trabalho é que este tem sido desenvolvido utilizando conhecimento e resultados obtidos adotando uma metodologia científica baseada em estudos experimentais, primários e secundários.

Algumas contribuições resultantes deste trabalho podem ser observadas para a área de Teste Baseado em Modelos:

- A descrição dos resultados obtidos a respeito da caracterização de ATBM's disponíveis na literatura técnica realizada a partir de uma revisão sistemática, aliado aos resultados um *survey* realizado com pesquisadores da área de TBM que possibilitou a definição e ordenação dos atributos de caracterização de uma ATBM no momento da seleção de uma abordagem para um projeto de software. Estes resultados propiciaram a definição e construção de um corpo de conhecimento sobre ATBM's representado por meio de um banco de dados de uma ferramenta de gerência de referências bibliográficas estendida para possibilitar a caracterização de abordagens de teste baseado em modelos.
- A identificação e contextualização de fatores de risco que podem influenciar na utilização de ATBM's para projeto de software, uma vez que essas abordagens normalmente são desenvolvidas e avaliadas para cenários restritos e que consistem em simplificações de cenários reais presentes em projetos.
- A partir da identificação de tais fatores de risco, definiram-se mecanismos para mitigar seus impactos em um projeto de software, o que resultou na elaboração de uma estratégia para apoiar a seleção de ATBM's para projetos de software, que foi apresentada neste trabalho a partir dos requisitos e funcionalidades que a compõem.

Com a elaboração da estratégia proposta, o passo seguinte desta pesquisa será direcionado à obtenção e avaliação de conhecimento científico sobre ATBM's, a fim de viabilizar a implantação e avaliação da estratégia apresentada em organizações de software através do fornecimento adequado de informações que auxiliem a equipe de teste na seleção de ATBM's para um projeto de software. Este conhecimento já vem sendo obtido ao longo do desenvolvimento deste trabalho (através dos resultados da revisão sistemática citada na seção 2.2 aliados aos resultados do *survey* descrito na seção 3.2, que formam a versão inicial do corpo de conhecimento sobre ATBM's), e precisa ser constantemente evoluído.

5. Agradecimentos

Os autores agradecem ao CNPq, FAPERJ, FAPEAM e SCR pelo apoio financeiro fornecido para a realização deste trabalho. Agradecemos especialmente a Marlon Vieira e Rajesh

Subramanyan (SCR – *Siemens Corporate Research*) pela sua contribuição para a realização da Revisão Sistemática sobre TBM.

6. Referências

- Berger, P. (2003), “Instanciação de Processos de Software em Ambientes Configurados na Estação TABA”, Dissertação de Mestrado COPPE/UFRJ. Rio de Janeiro.
- Bernard, E.; Legeard, B.; Luck, X.; Peureux, F. (2004), “Generation of test sequences from formal specifications: GSM 11.11 standard case-study”, *SW Practice and Experience* 34 (10), pp. 915 – 948.
- Biolchini, J.; Mian, P.G.; Natali, A.C.; & Travassos, G.H. (2005), “Systematic Review in Software Engineering: Relevance and Utility”, Relatório Técnico ES-679/05, PESC-COPPE/UFRJ. Disponível em <http://www.cos.ufrj.br>.
- Birk, A. (1997), “Modelling the application domains of software engineering technologies”, *Proceedings of the Twelfth International Conference on Automated Software Engineering (ASE)*. Lake Tahoe, CA, Novembro.
- Blackburn, M.; Busser, R.; Nauman, A. (2004), “Why Model-Based Test Automation is Different and What You Should Know to Get Started”. In: *International Conference on Practical Software Quality and Testing, PSQT/PSTT'2004 East*, Washington D.C., USA.
- Boldrini, J. L.; Costa, S. R.; Figueiredo, V. L., et al. (1980), *Álgebra Linear*, 3 ed., capítulo 8, Harper & Row do Brasil.
- Dalal, S.; Jain, A.; Karunanithi, N.; Leaton, J. M.; Lott, C. M.; Patton, G. C.; Horowitz, B. M. (1999), “Model-based testing in practice”, In: *ICSE'99*, Maio, pp. 285--294.
- Dias Neto, A. C., Travassos, G.H. (2006) “Maraká: Uma Infra-estrutura Computacional para Apoiar o Planejamento e Controle de Testes de Software”, In: *5º Simpósio Brasileiro de Qualidade de Software (SBQS)*, pp. 250-264, Vila Velha, ES.
- Dias Neto, A.C.; Subramanyan, R.; Vieira, M.; Travassos, G.H. (2007a), “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>.
- Dias Neto, A.C.; Subramanyan, R.; Vieira, M. (2007b), Travassos, G.H. “A Survey on Model-based Testing Approaches: A Systematic Review”, In: *1st WEASEL Tech'07 – ASE2007*, Atlanta.
- Dias Neto, A.C.; Travassos, G.H. (2008), “Supporting the Selection of Model-based Testing Approaches for Software Projects”, In: *3rd International Workshop on Automation of Software Test (AST'08)*, Leipzig, Germany, Maio.
- El-Far, I. K.; Whittaker, J. A. (2001) “Model-Based Software Testing”. *Encyclopedia of Software Engineering* (edited by J. J. Marciniak). Wiley.
- Hortmann, M.; Prenninger, W.; El-Ramly, M. (2005), “Case Studies. In: *Model-Based Testing—a tutorial volume*”, pp. 439–461.
- Juristo, N.; Moreno, A.M.; Vegas, S. (2004), “Reviewing 25 years of testing technique experiments”. *Empirical SW Engineering: An International Journal*, 9(1), p.7-44, March.
- Maiden, N. A. M.; Rugg, G. (1996), “ACRE: Selecting methods for requirements acquisition”, *Software Engineering Journal* 11(3): 183Y192.
- Myers, G. (1979), “*The Art of Software Testing*”. Wiley-Interscience.
- Pretschner, A. (2005), “Model-based testing”, *Proceedings of 27th International Conference on Software Engineering, (ICSE 2005)*, pp. 722-723.
- Pretschner, A; Prenninger, W.; Wagner, S.; Kühnel, C.; Baumgartner, M.; Sostawa, B.; Zölch, R.; Stauner, T. (2005), “One evaluation of model-based testing and its automation”, in: *Proc. ICSE'05*, pp. 392–401.
- Santos-Neto, P., Resende, R., and Pádua, C. (2007), “Requirements for information systems model-based testing”, In *Proceedings of the 2007 ACM Symposium on Applied Computing* (Seoul, Korea, March 11 - 15, 2007). SAC'07. ACM, New York.

- Utting, A. Pretschner, B. Legeard, B., “A taxonomy of model-based testing”, Technical report 04/2006, Department of Computer Science, University of Waikato, Abril, 2006.
- Vegas, S.; Basili, V. (2005), “A Characterization Schema for Software Testing Techniques”, Empirical Software Engineering, v.10 n.4, p.437-466, Outubro.
- Xavier, J. R.; Werner, C.M.L.; Travassos, G.H.; (2002), “Uma Abordagem para a Seleção de Padrões Arquiteturais Baseada em Características de Qualidade”, XVI Simpósio Brasileiro de Engenharia de Software, Gramado, RS, Brasil.