

# Um Modelo Sistêmico para Atividade de Avaliação e Testes de Software

Ana C. Guerra<sup>1</sup>, Eduardo Vasconcelos Silva<sup>2</sup>

<sup>1</sup> Centro de Pesquisa Renato Archer - CenPRA  
Rodovia Dom Pedro I, km 143,6 Campinas SP – Brazil

<sup>2</sup>Instituto de Computação da Universidade Estadual de Campinas  
Ana.guerra@cenpra.gov.br, vasconcelos2001@yahoo.com

**Resumo.** *Uma proposta para sistematizar o processo de criação de casos de teste encontra-se no uso de uma rede Bayesiana que mapeia a arquitetura de testes a ser implementada, aliada a uma ponderação estatística de riscos a cenários. Como produto final desta proposta sistêmica, além do ferramental gráfico que possibilita a descrição dos casos de testes segundo uma seqüência lógica e simulação de cenários, têm-se uma matriz que reúne todos os casos de testes obtidos da rede e demais oriundos da análise dos requisitos, segundo o enfoque do critério de adequação.*

## 1. Introdução

O desenvolvimento de produtos de software com boa qualidade ainda é complexo e caro. No momento que a sociedade depende cada vez mais de software, os problemas históricos associados com seu desenvolvimento ainda não foram adequadamente solucionados.

A avaliação é uma atividade no processo de desenvolvimento do software que, praticado com procedimentos bem definidos, tem a característica de aumentar a qualidade dos produtos desenvolvidos. Embora a tarefa de avaliar programas de computador seja tão antiga quanto o primeiro programa produzido, casos de erros nos sistemas em operação ainda são comuns. Ao longo destes anos, surgiram várias técnicas que ajudaram a melhorar o nível de qualidade dos sistemas desenvolvidos. Entretanto, tal melhoria não conseguiu acompanhar o aumento da complexidade dos sistemas de software.

Na medida em que o emprego de sistemas de software cresceu ao ponto em que boa parte de nossa vida depende cada vez mais de software e computadores, passa a ser de vital importância a existência de software confiável – software que fornece resultados corretos quando alimentado com dados válidos e que identifica corretamente dados inválidos. Principalmente quando se trata de aplicações onde um simples defeito pode causar um grande prejuízo ou simplesmente uma catástrofe, tais como software de aplicações espaciais, software de controle de processos na área médica, controle de processos em usinas nucleares.

Uma explicação para este fato de que software tem erro é que avaliar e testar software não é uma atividade trivial[Som92]. A atividade de avaliação e teste exige conhecimentos, habilidades, e infra-estrutura específica. Um bom desenvolvedor ou projetista

de software sem esta base dificilmente realizaria uma boa tarefa de avaliação. Outra explicação é que as empresas produtoras de software normalmente se encontram bastante ocupadas nas tarefas rotineiras (principalmente corrigindo defeitos!) para dedicar tempo e esforço necessários para identificar os procedimentos, técnicas e ferramentas de software para essas atividades, tão importantes.

## **2. Objetivos e Justificativa**

Esse artigo tem como objetivo buscar mecanismos que possibilitem um tratamento sistêmico da atividade de avaliação e teste de software, trazendo uma maior racionalidade na criação e execução dos casos de testes, além da melhoria da qualidade do produto. Com este enfoque é ressaltado o modelamento do problema sob uma nova óptica, buscando levantar os cenários afetados, além dos limites do documento de requisitos. Isso assegura ao desenvolvedor de testes, segurança e domínio da corrente implementação, tendo como resultado imediato, possíveis cenários não detalhados no documento de requisitos e sua correta atualização.

O processo de testes é encarado dentro de um contexto não determinístico, como sendo um gerenciamento de risco ‘ logo um cuidado estatístico especial será utilizado para o modelamento do problema. Outro ponto importante, dentro do modelamento proposto, é o uso extensivo do engenheiro de testes, cuja experiência é incorporada ao modelo através da ponderação adequada dos fatores que influenciam o resultado final. O trabalho tem de maneira implícita a melhoria da qualidade do produto de software colaborando para a criação de produtos mais robustos e confiáveis, através da melhoria do processo, o que indubitavelmente representa vantagens financeiras interessantes.

Será sugerida uma ferramenta que possibilita a simulação de forma computacional. Também é abordado o conceito de critério de adequação, outro elemento básico para a formulação do trabalho, que reúne elementos como caso de teste, especificação funcional, e software dentro de um equacionamento baseado em critérios. Assim será proposto a unificação da rede Bayesiana que modela o domínio de entrada com o critério de adequação, sugerindo uma sistematização no processo de criação de testes. O produto final desta sistematização é uma planilha que mapeia todos os casos de testes, visando melhorar a cobertura e eliminando duplicações. Foi realizada uma análise quantitativa de quatro projetos de testes em andamento, focando no esforço de análise e cobertura e mostra a implementação prática de um design de testes.

## **3. Conceitos em Testes de Software e Metodologia**

Teste é uma forma de verificação dinâmica que consiste em executar o programa com um conjunto de dados de entrada e determinar se ele se comporta conforme o esperado, isto é, de acordo com sua especificação. Em geral, é impossível testar um programa exaustivamente; o importante é selecionar um conjunto finito de casos de testes que permitam testá-lo adequadamente [Mar00]. Pode-se afirmar que um teste bem-sucedido é aquele que descobre um erro ainda não descoberto, logo um bom caso de testes é aquele que tem alta probabilidade de encontrar um novo erro [Mye79]. Conceitos de erro, de falha e de defeito, são utilizados segundo o IEEE Std. Glossary of Software Engineering Terminology, padrão

610.12/1990, definidos como: erro: engano cometido por um desenvolvedor (analista, projetista, programador); falha: manifestação do erro (uma especificação ou código incorretos); defeito: evento notável ao usuário, ativação da falha.

### **3.1 Estudo Redes Bayesiana e Critérios de Adequação**

Critério de adequação em síntese define um relacionamento entre especificação, casos de teste e programa e redes Bayesiana que proverá um ferramental para o modelamento do teste de software, possibilitando inclusive simulações estatísticas. Um dos conceitos extremamente valiosos para a elaboração teórica do presente trabalho. Sua contribuição concentra-se na cobertura de requisitos [Pre02].

Segundo [Zhu96], critérios de adequação são regras que definem se um trecho de software foi adequadamente testado. Um vasto número de critérios de adequação tem sido propostos e investigados, podendo ser citado os seguintes: Critério baseado em fluxo de controle; Critério baseado em fluxo de dado; Critério baseado em texto de programa; Critério baseado em falhas.

## **4. Resultados Obtidos**

A abordagem proposta visa buscar um novo paradigma para o design de testes de software incorporando elementos gráficos e uma ponderação de riscos envolvidos. Nesta proposta faz-se necessário um estudo preliminar profundo sobre os novos requisitos a serem implementados e realizar um levantamento de quais casos de usos são afetados. Uma documentação de apoio como diagramas UML de casos de uso ou mesmo um diagrama de seqüência, serão materiais de inestimável valia para o correto levantamento dos cenários.

### **4.1 Produto Obtido no projeto**

Através do levantamento consistente dos cenários, e suas inter-relações, tem-se os fundamentos necessários para o modelamento Bayesiana. Nesta etapa, a preocupação está voltada para o modelamento do problema de software, buscando levantar todas as transações, ações de software e partições de entrada relacionadas. Deste conjunto de elementos constitui a arquitetura de testes, um documento estritamente direto quanto ao levantamento dos casos de testes necessários, possibilitando inclusive, simulação de cenários e conseqüentes estimativa dos testes mais indicados a identificar falha. A Metodologia foi gerada e os resultados testados e analisados [Vasconcelos06].

### **4.2 Casos de Testes e Rede Bayesiana**

Quando o modelamento de uma transação finaliza, obtém-se a arquitetura de testes e os casos de testes a serem aplicados, de forma muito direta e objetiva. A rede deve ser simulada partindo dos nós de mais alta hierarquia e para cada partição de entrada todas as partições de entrada do nó sucessor devem ser verificadas. Um novo caso de teste deve ser criado quando a probabilidade for diferente de zero e a interação corrente fizer sentido lógico. O grau de detalhamento do caso de teste está fora do escopo desse trabalho, mas vale aqui dizer que

essa tarefa sofre influência do grau de conhecimento da pessoa encarregada de executar a tarefa de teste, ou se a execução será manual ou automática.

### 4.3 Cenários não previsíveis

A construção da Rede Bayesiana constitui um ferramental muito eficiente no entendimento adequado do problema e busca de situações que deveras não estão contempladas nos requisitos. Nestes casos, o comportamento esperado não é documentado, o que acaba por exigir uma revisão dos requisitos. Desta forma a documentação do projeto, por exigência da técnica, torna-se mais robusta, completa e adequada, reduzindo assim, os problemas com manutenção no futuro.

### 4.4 Outros resultados

Desse projeto foi gerada dissertação de mestrado e serão apresentados artigos em congressos e revistas oportunamente. O projeto foi implantado na empresa com sucesso e se encontra em operação, além da capacitação de pessoas no assunto.

## 5. Aplicabilidade dos resultados

De forma a tornar elucidativo o uso dessa nova técnica foram levantadas métricas relativas ao esforço de análise, cobertura de testes e razão entre o esforço as técnicas bem como a razão de cobertura. Esses são dados reais obtidos na criação de testes de software na área de telecomunicação. Uma simulação da técnica proposta na criação de testes sobre um sistema fictício de cadastro de cartões de crédito foi realizada para validar os resultados.

A seguir é exemplificado o uso da técnica na criação de casos de testes para um sistema de banco de dados, onde foi criada uma rede e populado parcialmente a matriz de adequação. O sistema fictício de validação de cartões de crédito em questão possui os seguintes requisitos:

Identificação	Descrição
Req_1	O sistema realizará a validação de cartões de crédito das operadoras A, B e C somente, recusando as demais operadoras;
Req_2	A identificação da operadora será baseada no número do cartão, a saber: Operadora A: número do cartão inicia-se com zero; Operadora B: número do cartão não se inicia com zero e possui no total dez dígitos; Operadora C: número do cartão não se inicia com zero e possui no total doze dígitos. O sistema devera suportar até 100 validações simultâneas;
Req_3	Para as operadoras B e C deve-se validar o código de segurança do cartão (baseado num algoritmo que envolve o próprio número do cartão);
Req_4	Deve-se validar também se o cartão possui a data de validade não vencida somente para os cartões das operadoras B e C.

**Tabela 1 – Requisitos Sistema de Cartão**

Baseado nestes requisitos, uma proposta para o modelamento do problema numa rede Bayesiana. Neste experimento, toda documentação de entrada foi um documento

extremamente simples contento os quatro requisitos. Com a aplicação da técnica sistêmica proposta, foram criados trezes casos de teste, resultando num valor médio de quatro testes por requisito. Os testes baseados nos critérios positivo e negativo são os mais imediatos e obtidos da rede Bayesiana pela simples varredura seqüencial dos nós (e suas partições de entrada). Nas simulações de cenários, partições com 0% de probabilidade são casos de testes que se enquadram no critério negativo. Com relação aos critérios Limite, Volume e Interação, a matriz de adequação exige o foco em cada requisito segundo prismas diferentes, buscando identificar casos de teste que sejam aplicáveis.

## **6. Características Inovadoras**

É indiscutível o caráter inovador e os benefícios oriundo da nova proposta para melhoria da qualidade de software por meio da avaliação. Estudos mostram também que o custo despendido na manutenção em campo pode chegar a 90% do total investido no projeto [Pig97].

Outro ponto de grande importância é a necessidade de um método sistêmico que agregue ao processo de desenvolvimento de avaliação e testes, mas especificamente aqueles baseados exclusivamente em requisitos funcionais, onde ferramentas estatísticas de simulação de cenários bem como critérios de adequação [Zhu96] irão suprir uma deficiência efetiva.

## **7. Conclusões e Perspectivas Futuras**

Foram abordadas aqui a rede Bayesiana e o Critério de Adequação, fundamentos para a proposta sistêmica apresentada. A busca de uma sistematização do processo de criação de casos de teste teve em seus primórdios dois limitantes básicos: deveria agregar valor tecnicamente proporcionando um ferramental leve para o modelamento da arquitetura de testes e não poderia embutir uma sobrecarga relevante ao esforço de criação. Os dados experimentais obtidos sinalizam que o modelamento gráfico da arquitetura de testes acaba por vislumbrar cenários não cobertos no documento de requisito e que devem ser incluídos na documentação, de forma a evitar comportamento dúbio.

Essa contribuição torna-se ainda mais acentuada quando o desenvolvimento de testes inicia-se paralelamente ao design da arquitetura de software. Assim, a arquitetura do sistema acaba por sofrer influência direta da atualização dos requisitos, resultando numa implementação mais robusta.

O uso sistemático de uma metodologia para criação de casos de teste é também uma maneira de eliminar a subjetividade do processo, e permitir uma melhor controle na fase de criação, um ganho gerencial interessante no tocante a estimativas, com melhores previsões, quanto a disponibilidade dos casos de testes.

Foi feita uma análise quantitativa da técnica com dados reais. Os dados e gráficos comparativos sinalizam uma tendência positiva referente a uma melhor cobertura (quatro vezes e meio) com menor esforço de análise (um quarto), aliado a um sólido conhecimento da arquitetura de teste. Esses dados mostram-se muito atraentes frente ao desafio atual, da busca da qualidade ao menor custo, com certeza um diferencial competitivo.

Neste trabalho não foi abordado técnicas e vantagens do uso de testes exploratórios. Uma proposta para um trabalho futuro seria a união da corrente proposta sistêmica com o uso extensivo de testes exploratórios. O uso de testes exploratórios tornaria menor o esforço de escrita de testes que compartilham casos de usos comuns, com pequenas alterações do estado inicial, bem como testes de alta complexidade e não usuais. No estágio do design da matriz de adequação podem-se detalhar os testes que seriam os prováveis candidatos para exploratórios.

## **Referências**

- [Mye79] Myers, G.J.. The Art of Software Testing. John-Wiley & Sons, 1979.
- [Som92] Sommerville, Ian. Software Engineering – 4th. Edition. Addison-Wesley Inc., 2000.
- [Zhu96] Zhu,Hong. A formal Analysis of the Subsume
- [Pig97] Pigoski, Thomas M; Practical Software Maintenance: Best Practices for Managing Your Software Investment. Wiley Computer Publishing, 1997.
- [Mar00] Martins, Eliane.; Manutenção e Ferramentas CASE. IC-UNICAMP, 2000.
- [Vasconcelos06] Um Modelo Sistêmico para Atividade de Avaliação e Teste de Software. Autor: Eduardo de Vasconcelos Silva, Orientador: Ana Cervigni Guerra. Co-Orientador: Rogério Drummond Burnier P de M Filho. Dissertação de mestrado - DEPARTAMENTO DE SISTEMAS DE COMPUTACAO - Unicamp 2006.
- [Woo02] Wooff, David A. Goldstein, Michael. Coolean, Frank P. A. Bayesian Graphical Models for Software Testing. IEEE Transactions on Software Engineering Vol 28, N° 5, May 2002.
- [Pre02] Pressman, Roger S.; Engenharia de Software , 5. Edição. McGraw-Hill,Inc., 2002.