

# Seleção de Casos de Teste Utilizando Conceitos de Variabilidade: Uma Revisão Sistemática

Everton N. Narciso<sup>1</sup>, Fátima L. S. Nunes<sup>1</sup>, Márcio E. Delamaro<sup>2</sup>

<sup>1</sup>Escola de Artes, Ciências e Humanidades – EACH  
Universidade de São Paulo, São Paulo – SP – Brasil

<sup>2</sup>Instituto de Ciências Matemáticas e de Computação – ICMC  
Universidade de São Paulo, São Paulo – SP – Brasil

{evernarciso, fatima.nunes}@usp.br, delamaro@icmc.usp.br

**Abstract.** *In order to increase the quality, efficiency and effectiveness in the software testing process, the selection of test cases that explore in depth the features of the system for discovering defects is essential. In this context, this article presents a systematic review of test case selection using concepts of variability, whose goal is to verify the state of the art in this field. The results, such as confirming the importance of experiments, evaluation of results and use of qualitative and quantitative analysis of papers related to the selection of test cases are also shown in this paper.*

**Resumo.** *Para aumentar a qualidade, a eficiência e a eficácia no processo de teste de software, a seleção de casos de testes que explorem profundamente as funcionalidades do sistema para a descoberta de defeitos é essencial. Dentro deste contexto, apresenta-se neste artigo a Revisão Sistemática sobre a seleção de casos de testes utilizando conceitos de variabilidade, cujo objetivo é verificar o estado da arte deste domínio. Os resultados obtidos, tais como a confirmação da importância de experimentos, da avaliação de resultados e da utilização de análise qualitativa e quantitativa em trabalhos relacionados à seleção de casos de teste também são apresentados neste artigo.*

## 1. Introdução

Desenvolver aplicativos de qualidade é uma meta almejada e necessária em todos os setores de produção de *software*. Há um longo caminho a se percorrer para atingir esta meta e, dentre as atividades cruciais para obtenção da qualidade, está a verificação e validação (V&V) do produto desenvolvido.

A V&V destina-se a mostrar que um aplicativo está em conformidade com sua especificação e que atende as expectativas do cliente. O teste, que é parte da V&V, representa uma fase do processo de desenvolvimento de *software* muito trabalhosa e dispendiosa e, segundo Sommerville (2007), as ferramentas de teste estão entre as primeiras ferramentas que deveriam ser desenvolvidas.

Um teste eficiente depende de um conjunto de condições específicas e detalhadas utilizadas para se testar o *software*, conhecidas como casos de teste [Sapna and Mohanty 2010]. Porém, na prática, os recursos de tempo e custo (*hardware*, humano e financeiro) são limitados e, em se tratando de grandes aplicações, torna-se

impraticável a utilização de todos os casos de teste possíveis. Neste contexto, a seleção de casos de teste (SCT) é muito importante para a construção e a definição de estratégias de teste [Kasurinen, Taipale and Smolander 2010], permitindo a otimização dos recursos por meio da redução de dados redundantes utilizados para testes.

O grande desafio da SCT é o desenvolvimento de métodos, técnicas e critérios apropriados, que explorem profundamente as funcionalidades do sistema para a descoberta de eventuais defeitos, com o menor esforço computacional possível [Gleirscher 2011]. Tendo em vista que o método a ser utilizado na SCT depende dos requisitos a serem testados [Levenshtein 2006], normalmente não há uma única técnica que possa responder adequadamente a uma grande diversidade de especificações e pressupostos que diferentes sistemas podem possuir [Engstrom, Skoglund and Runeson 2008]. A literatura aborda variadas técnicas de SCT para resolução de problemas específicos ao objeto de estudo, como o desenvolvimento de *software* baseado em modelos (ex. UML, BPMN), *Web-Services*, paradigma de orientação a objetos, dentre outros.

Apesar da grande variedade de técnicas de SCT encontradas na literatura, uma análise mais acurada apontou que a base de muitas estratégias de SCT é o conceito de diversidade (ou dissimilaridade), que representa algum tipo de diversificação dos valores de um conjunto de dados [Chen 2010]. Este importante fundamento é eficaz e consiste em escolher o melhor caso de teste dentre os seus similares, assim como selecionar um grupo de casos de teste dissimilares, simultaneamente, visando a aumentar a cobertura de teste [Sapna and Mohanty 2010]. Diante desse cenário, este artigo apresenta uma Revisão Sistemática (RS) com o propósito de pesquisar e identificar os métodos e técnicas existentes para SCT baseada em conceitos de variabilidade, como parte de um projeto mais abrangente que visa selecionar casos de testes para *software* com saídas gráficas [Delamaro, Nunes and Oliveira 2011]. O conceito de variabilidade aqui descrito se refere à dispersão de valores de uma variável em torno de um valor tomado como ponto de comparação, ou técnicas similares.

Para atingir o objetivo proposto neste artigo, na seção 2 são expostos os conceitos e as etapas executadas na RS. Os resultados e as discussões se encontram na seção 3 e a seção 4 conclui o artigo.

## **2. Metodologia da Pesquisa**

Neste trabalho foi utilizada a Revisão Sistemática (RS), que é uma metodologia rigorosa e confiável, desenvolvida para avaliar criticamente evidências acerca de um determinado tópico [Biolchini *et al* 2005]. Seu objetivo é identificar e selecionar pesquisas relevantes, coletar e analisar dados dos estudos e permitir a auditoria [Kitchenham 2004]. Esta pesquisa propicia um levantamento bibliográfico preciso e consciente, com propósito de aferir o estado da arte no domínio explorado. Ao confrontar diversos estudos, são apresentadas informações valiosas que podem colaborar para a pesquisa e para a elaboração de técnicas mais abrangentes e generalizadas de SCT, em detrimento à resolução de problemas pontuais. Os resultados foram sintetizados e, no final, é realizada uma avaliação quantitativa e qualitativa da RS.

Para organizar e executar os processos da RS de maneira efetiva e eficaz foram conduzidas três fases:

- **Planejamento da Revisão Sistemática:** o objetivo é auxiliar na formulação do problema e definir as especificidades que deverão ser obedecidas durante a RS. As especificações foram documentadas em um documento denominado “Protocolo da RS”, que é um documento dinâmico, adaptável e que pode ser atualizado durante a execução da RS, contendo as principais informações utilizadas na fase de planejamento, como as questões de pesquisa, as fontes de pesquisa, a estratégia de extração de informações e a estratégia para síntese dos resultados.
- **Condução da Revisão Sistemática:** nesta fase foi realizada a pesquisa propriamente dita, anotando-se informações importantes para a análise posterior. Os documentos resultantes desta fase contêm o período de execução da condução, as bases de dados pesquisadas, a *string* de busca utilizada e os arquivos retornados (ex. artigos, teses, *surveys*, outros).
- **Extração de Informações e Síntese dos Resultados:** nesta fase foram examinados minuciosamente os arquivos selecionados na fase anterior, compondo um documento que contém a análise, a interpretação e a conclusão da RS.

Os documentos citados são a base da RS e deste artigo, e são apresentados nas seções a seguir.

## 2.1. Protocolo de Revisão Sistemática

O protocolo utilizado foi adaptado às necessidades julgadas pertinentes ao objeto de pesquisa. Ele foi desenvolvido e revisado na fase de planejamento e utilizado nas demais fases da RS. Os principais tópicos do documento são apresentados a seguir.

**Abreviações:** Seleção de Casos de Teste (SCT) e Seleção de Casos de Teste baseada em Conceitos de Variabilidade (SCTbV), *Graphical User Interface* (GUI).

**Questões de Pesquisa:** 1) Quais são os métodos e técnicas existentes para SCTbV? 2) Quais os critérios utilizados para SCTbV? 3) Como são avaliados os métodos, técnicas e os critérios utilizados para SCTbV? 4) Existe algum método ou técnica utilizado para otimização de SCTbV? 5) Existem outras RS relacionadas à SCTbV?

**Seleção de Fontes:** as fontes deveriam estar disponíveis via web, em bases de dados científicas da área.

### Lista das Fontes de Pesquisa:

- Biblioteca Digital do IEEE (<http://ieeexplore.ieee.org/Xplore/>);
- Biblioteca Digital da ACM (<http://portal.acm.org/>);
- SCOPUS (<http://www.scopus.com/home.url>);
- Teses da USP (<http://www.teses.usp.br/>);
- Banco de Teses da Capes (<http://servicos.capes.gov.br/capesdw/>);

- Portal de Periódicos da Capes (<http://novo.periodicos.capes.gov.br/>);
- SciELO - Scientific Electronic Library Online (<http://www.scielo.org/>).

**Tipos das Fontes:** teses, dissertações, monografias, TCCs, artigos de revisão (*surveys*) e artigos publicados em conferências ou periódicos.

Os critérios de inclusão e exclusão de trabalhos estão discriminados na Tabela 1.

**Tabela 1 - Critérios de inclusão e exclusão de trabalhos**

Índice	Inclusão ou Exclusão	Critérios de Qualidade da Pesquisa
C1	Inclusão	Os trabalhos devem apresentar alguma proposta, aplicação prática ou experimento de SCTbV.
C2	Inclusão	O trabalho deverá ter sido publicado em periódico ou anais de eventos, nas fontes listadas anteriormente na seção “Lista de Fontes de Pesquisa”.
C3	Inclusão	Trabalhos que já possuam aprovação pela comunidade científica.
C4	Exclusão	Trabalhos práticos ou experimentais sem avaliação dos resultados.
C5	Exclusão	Trabalhos que apresentem apenas a <b>geração</b> ou <b>oráculos</b> de casos de teste.
C6	Exclusão	Trabalhos que apresentem seleção de casos de testes, mas que NÃO utilizem conceitos de variabilidade ou similares. Exemplo: baseados em UML, BPMN, <i>Web-Services</i> , Análise de Risco.
C7	Exclusão	Trabalhos de domínio muito específico que não possam ser estendidos e trabalhos fora do escopo da pesquisa.

**Seleção Preliminar de Trabalhos:** as *strings* foram submetidos às máquinas de busca. Após a leitura do resumo e aplicação dos critérios de inclusão e exclusão, o trabalho foi selecionado se confirmada a sua relevância.

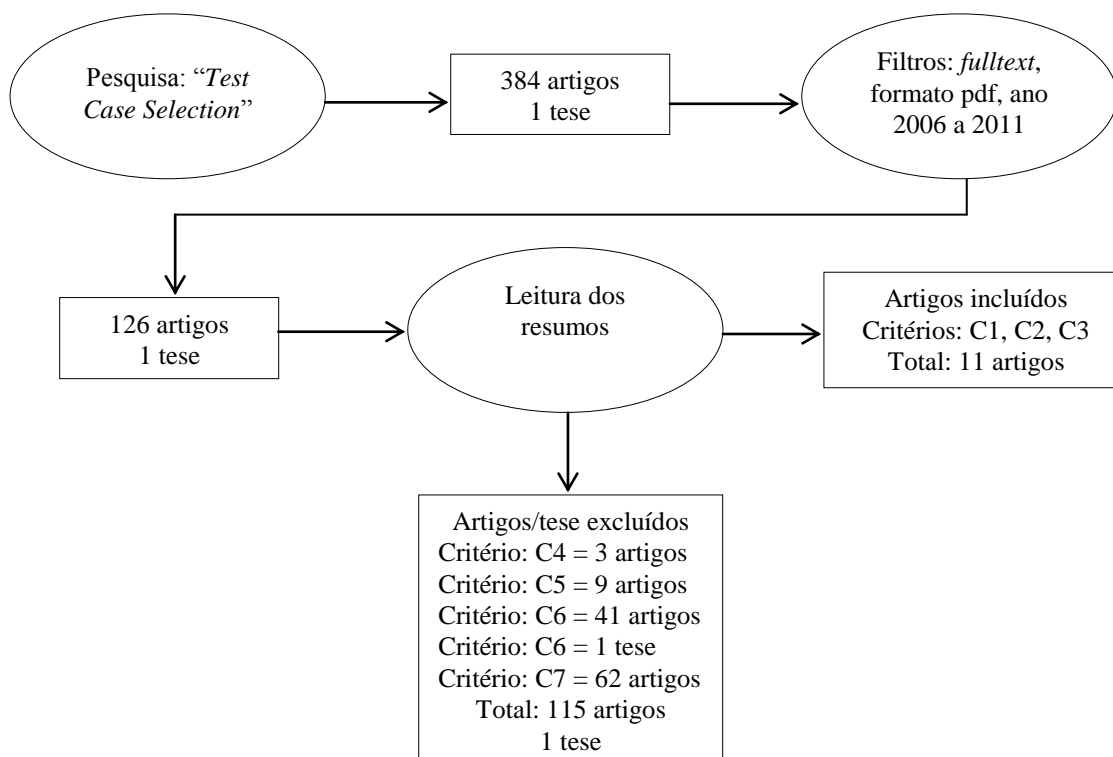
**Estratégia de Extração de Informações:** depois de definidos os trabalhos incluídos, estes foram lidos na íntegra. Foram construídos formulários com as informações mais relevantes, as opiniões e as conclusões sintetizadas. Os métodos utilizados para a avaliação e parâmetros considerados também foram anotados, quando julgado relevante.

**Síntese dos Resultados:** após a leitura e a sumarização dos trabalhos selecionados, foi elaborada uma síntese com a análise quantitativa e qualitativa sobre a pesquisa efetuada, apresentada adiante. Algumas características relevantes são: 1) tipo de aplicação a que o trabalho apresentado se destina; 2) método, técnica e critério utilizado para otimização e/ou automatização da seleção dos casos de teste, se houver.

## 2.2. Condução da Revisão Sistemática

Após a elaboração do protocolo apresentado na seção 2.1 iniciou-se a condução da RS propriamente dita. De 22 de outubro a 10 de janeiro de 2012 foi realizada a pesquisa nas bases de dados definidas no protocolo, utilizando-se a seguinte *string*: “*Test Case Selection*”.

A Figura 1 ilustra o processo de pesquisa. A elipse representa uma ação, enquanto o retângulo representa um resultado.



**Figura 1 – Processo da pesquisa nas bases de dados científicas.**

Após a aplicação de filtros nas bases de dados indexadas e a eliminação de trabalhos duplicados, aproveitou-se 126 artigos e 1 tese. Em seguida, os resumos foram lidos. Foram incluídos 11 artigos, os quais atenderam todos os critérios de inclusão pré-estabelecidos.

### 3. Resultados e Discussões

Esta seção apresenta a extração e síntese dos resultados obtidos na pesquisa e que foram incluídos baseados nos critérios de inclusão do protocolo, bem como os resultados e discussões.

A Tabela 2 apresenta a síntese dos artigos incluídos. O resultado da pesquisa mostra que existem várias técnicas diferentes para a SCT, e cada técnica pode focar diferentes aspectos ou critérios de teste, visando diminuir o número de casos de testes utilizados para teste e aumentar a cobertura de código testado. Alguns pontos relevantes para se elaborar ou adotar uma determinada técnica são: requisitos de *software* a serem testados, critérios de cobertura de teste (ex. código), tipos de falhas possíveis, dentre outros. Um ponto em comum quando se trata da SCT é que a maioria das técnicas se baseiam na diversidade dos casos de teste (também conhecida como dissimilaridade) [Chen 2010].

**Tabela 2 – Síntese dos artigos incluídos**

<b>Id</b>	<b>Fonte</b>	<b>Referência</b>	<b>Descrição das técnicas utilizadas</b>
T1	SCOPUS	Yu, Xu and Tsai (2010)	SCT utilizando SVM e conceitos de variabilidade.
T2	SCOPUS	Chen, Kuo and Liu (2008)	Nova abordagem de “ <i>Adaptative Random Testing</i> ”.
T3	SCOPUS	Kosindrdecha, N. and Daengdej J. (2010)	Abordagem sobre como diminuir a quantidade de casos de teste utilizados durante o teste.
T4	ACM	Sapna and Mohanty (2010)	SCT baseada em <i>clustering</i> .
T5	ACM	Gleirscher Mario (2011)	Abordagem de seleção baseada em “ <i>hazardous state</i> ”.
T6	IEEE	Vilkomir, Swain and Poore (2008)	Foco na modelagem das dependências dos parâmetros de entrada para posterior geração e seleção de casos de testes.
T7	IEEE	Tsai, Zhou, Paul, Chen and Bai (2007)	Apresenta uma nova abordagem: ( <i>Coverage Relationship Model, CRM</i> ).
T8	IEEE	Mirarab, Akhlaghi and Tahvildari (2011)	A aplicação é baseada em um algoritmo “Multi-Critério” iterativo que maximiza a cobertura mínima de código do <i>software</i> , utilizando várias técnicas para tal finalidade.
T9	IEEE	Zeng, Ling Li, Juan Li and Wang (2009)	Apresenta técnicas de classificação e particionamento das suítes de testes e consequente redução de casos de testes baseada na relevância do atributo, utilizando de árvores de decisão.
T10	IEEE	Zhu Xiaochun, Zhou Bo, Li Juefeng and Gao Qiu (2008)	Proposta de automação de testes funcionais para GUI, por meio de geração, seleção e execução de casos de teste utilizando o <i>Rational Functional Tester (RFT)</i> .
T11	IEEE	Simons A.J.H. and Wenwen Zhao (2009)	Proposta de melhorias em “ <i>dynamic analysis algorithms</i> ”.

Uma vez determinada a técnica a ser utilizada, é importante saber como avaliá-la. A Tabela 3 apresenta as métricas de avaliação e o tipo de aplicação a que se destina cada técnica apontada na literatura. A coluna *Id* identifica o trabalho, considerando-se a referência disponibilizada na Tabela 2.

**Tabela 3 – Métricas de avaliação e tipo de aplicação**

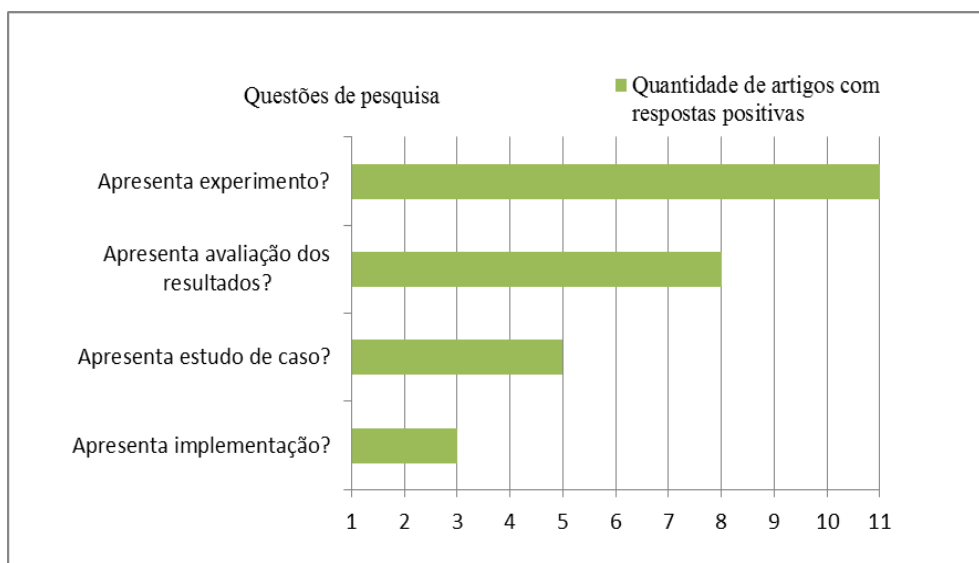
<b>Id</b>	<b>Métricas de Avaliação</b>	<b>Tipo de Aplicação</b>
T1	Média de falhas detectadas, em porcentagem.	Teste baseado em funcionalidades do sistema.
T2	P-measure (capacidade ou probabilidade de detecção de defeito).	Teste baseado em funcionalidades do sistema. Dados alfanuméricos.
T3	Quantidade de casos de teste gerados, tempo de execução dos casos de teste e porcentagem de cobertura de requerimentos críticos.	Teste baseado em requerimentos críticos do sistema. Dados alfanuméricos.
T4	Não há.	Dados alfanuméricos.
T5	Não há.	Dados alfanuméricos.
T6	Quantidade de casos de testes eliminados, em porcentagem. Número de defeitos encontrados.	Múltiplos domínios (alfanumérico, GUI, <i>WebServices</i> , outros).
T7	Relação de detecção de falhas em relação a outras técnicas.	Baseado na estrutura de código (ex. classes, métodos, blocos).
T8	Não há.	<i>Software</i> de aplicação científica, com entradas de dados estatísticos.
T9	Distribuição de casos de testes mais uniforme e maior número de detecção de falha.	Baseado na estrutura de código (região de código testada).
T10	Não há.	GUI.
T11	Diminuição de casos de testes redundantes (ou <i>rules</i> ) sob determinadas condições conhecidas.	Dados alfanuméricos para aplicações desenvolvidas em linguagem Java.

É evidente que, uma vez que existem diversas técnicas aplicáveis à SCT, também existem diversas metodologias para avaliar a eficiência e eficácia da aplicação. Os trabalhos analisados confirmam tal assertiva e apresentam diversas possibilidades de avaliação de acordo com a técnica aplicada. Ressalta-se que existem diferentes mecanismos utilizados para obtenção do conceito de variabilidade, sendo que em geral são empregadas medidas estatísticas como: chi-quadrado, média, mediana, desvio-padrão, dentre outros, o que aumenta a complexidade de pesquisa.

Verifica-se na Tabela 3 que as duas métricas de avaliação mais empregadas são a quantidade de casos de testes utilizados e a capacidade de detecção de falhas. Já os tipos de aplicações são muito variadas, porém a maioria requer, explicitamente, a entrada e saída de dados alfanuméricos. O estudo empírico aponta que não há algum experimento realizado em sistemas utilizados na prática, ou seja, sistemas que estejam em funcionamento (ex. em indústrias de *software*) para validar a técnica proposta, o que sugere que tais técnicas dificilmente podem ser generalizadas em diversos sistemas com diferentes requisitos.

### 3.1. Questões Metodológicas da Pesquisa

Com base nos 11 artigos selecionados foram sintetizadas algumas questões metodológicas importantes para a pesquisa, em relação a cada trabalho analisado: 1) o trabalho apresenta algum tipo de experimento realizado com a técnica proposta? 2) o trabalho apresenta avaliação dos resultados obtidos nos experimentos? 3) o trabalho apresenta estudo de caso? 4) o trabalho apresenta a implementação realizada e que permita a reprodução da técnica, em termos de algoritmo ou código de programação? A Figura 2 mostra o resultado obtido.



**Figura 2 – Gráfico com a quantidade de artigos incluídos que apresentam questões metodológicas importantes para a pesquisa.**

Todos os artigos incluídos na RS apresentaram algum experimento de SCT e 73% demonstram a avaliação de resultados obtidos nos experimentos, o que indica a importância da experimentação e da avaliação de resultados em trabalhos relacionados a este domínio. Outro ponto interessante é que 45% dos trabalhos também apresentam uma análise qualitativa (estudo de caso) e, por meio de uma observação empírica, é possível destacar a importância conferida pela comunidade científica quanto à utilização concomitante de metodologias quantitativas e qualitativas em pesquisas de SCT.

Baseando-se na premissa de que um artigo científico deveria permitir a reprodução da pesquisa apresentada, verificou-se que poucos trabalhos demonstram a implementação completa de sua aplicação (apenas 27%), tornado a reprodução difícil, incompreensível ou impossível.

É importante ratificar que todos os artigos apresentam diferentes implementações de SCTbV, ou seja, nenhuma implementação tinha relação direta com outra. Na maioria dos casos as métricas de avaliação também são diferentes, fatos que demonstram que cada trabalho é único e que, dentro do contexto desta pesquisa, nenhum autor dá continuidade ao trabalho de outro. Foi admitida a hipótese de que estas causas podem explicar o porquê não foram encontradas técnicas para otimização de SCT, tendo em vista a presunção empírica que uma técnica de otimização deveria ser o melhoramento de outra técnica já existente.



Comprovou-se também as diversas possibilidades de utilização das técnicas pesquisadas em diferentes domínios de aplicação. Estes fatos ajudam a evidenciar a grande amplitude da problemática da SCTbV e a dificuldade de pesquisa na área. Não foram encontrados na literatura, utilizando o protocolo apresentado neste artigo, documentos sobre SCT para *software* com saídas gráficas, tampouco outras RS que tratassem especificamente sobre SCTbV.

#### **4. Conclusão**

Neste artigo foi apresentada uma revisão sistemática acerca da seleção de casos de testes utilizando conceitos de variabilidade. Esta pesquisa focou a obtenção do estado da arte do domínio, além de possibilitar a organização de ideias, a sumarização de informações relevantes e a auditoria.

A revisão sistemática é um processo evolutivo e visa a responder algumas questões importantes para pesquisa de um determinado domínio. Baseado no conteúdo desta revisão constatou-se que: 1) existem várias técnicas e possibilidades disponíveis para resolução do problema em questão, que é a SCT; 2) a diversidade de técnicas é proveniente da diversidade de requisitos, pressupostos, funcionalidades e casos de teste que um software pode possuir; 3) há vários critérios utilizados para a SCT, tendo em vista a diversidade de técnicas apresentadas; 4) existem várias formas de avaliar os resultados alcançados com a utilização dessas técnicas, e as métricas mais empregadas são a quantidade de casos de teste utilizados e a capacidade de detecção de falhas; 5) não foram encontradas técnicas para otimização de SCT e 6) não foram encontradas outras revisões sistemáticas sobre SCTbV.

Por meio da avaliação empírica é possível afirmar que não há uma determinada técnica que seja mais utilizada ou que seja superior a outra, uma vez que cada artigo analisado apresenta a resolução de um problema pontual. Não há a demonstração de um experimento ou aplicação prática que compare as técnicas apresentadas nos onze trabalhos analisados.

Uma dificuldade encontrada foi o fato de alguns resumos não indicarem explicitamente qual o conceito de variabilidade ou quais as metodologias e métricas de avaliação foram utilizados (quando existiam). Salienta-se também que não foram encontrados trabalhos que tratassem de SCT para *software* com saídas gráficas.

Alguns tópicos importantes para trabalhos futuros são: estimular o desenvolvimento de técnicas de SCT que possam ser estendidas ou generalizadas para vários domínios de *software*; desenvolver técnicas de SCT para *software* com saídas gráficas; tornar mais claras as metodologias e métricas de avaliação de resultados para SCT; definir de maneira mais objetiva os conceitos de diversidade e de variabilidade das técnicas de SCT.

#### **Agradecimentos**

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Processo 559931/2010–7, à Fundação de Amparo à Pesquisa do Estado de São Paulo (Fapesp) – Processos 2010/15691–0 e 2010/09806–0 e ao Instituto Nacional de Ciência e Tecnologia – Medicina Assistida por Computação Científica (INCT–MACC) pelo apoio financeiro.

## Referências

- Biolchini, J., Mian, P.G., Natali, A.C.C., and Travassos, G.H. (2005). Systematic review in *software engineering*. Technical report, RT-ES679/05. System Engineering and Computer Science Dept., COOPE/UFRJ.
- Chen, T. Y., Kuo, F. and Liu, H. (2008). Distributing test cases more evenly in adaptive random testing. *Journal of Systems and Software*, 81(12), 2146-2162. [www.scopus.com](http://www.scopus.com).
- Chen, T. Y. (2010). "Fundamentals of test case selection: Diversity, diversity, diversity". *Software Engineering and Data Mining (SEDM)*, 2010 2nd International Conference on , vol., no., pp.723-724, 23-25 June 2010
- Delamaro, M. E., Nunes, F. d. L. d. S. and de Oliveira, R. A. P. (2011). Using concepts of content-based image retrieval to implement graphical testing oracles. *Software Testing, Verification and Reliability*. doi: 10.1002/stvr.463
- Engstrom, Emelie; Skoglund, Mats; Runeson, Per. (2008). Empirical evaluations of regression test selection techniques: a systematic review. In *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM '08)*. ACM, New York, NY, USA, 22-31. DOI=10.1145/1414004.1414011 <http://doi.acm.org/10.1145/1414004.1414011>
- Fanping Zeng; Ling Li; Juan Li; Xufa Wang. (2009). Research on Test Suite Reduction Using Attribute Relevance Analysis. *Computer and Information Science*, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference no, pp.961-966, 1-3 June 2009. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5223197&isnumber=5222522>
- Gleirscher, Mario (2011). Hazard-based selection of test cases. *Proceeding of the 6th international workshop on Automation of software test. AST '11*. doi: <http://doi.acm.org/10.1145/1982595.1982609>.
- Kasurinen, Jussi; Taipale, Ossi; Smolander, Kari. (2010). Test case selection and prioritization: risk-based or design-based? In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '10)*. ACM, New York, NY, USA, Article 10, 10 pages. DOI=10.1145/1852786.1852800 <http://doi.acm.org/10.1145/1852786.1852800>.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. Technical report, Keele University and NICTA.
- Kosindrdecha, N., & Daengdej, J. (2010). A test case generation process and technique. *Journal of Software Engineering*, 4(4), 265-287. [www.scopus.com](http://www.scopus.com).
- Levenshtein, V. I. (2006). An evaluation of combination strategies for test case selection. In *Empirical Software Engineering*, volume 11, 2006.
- Mirarab, S.; Akhlaghi Esfahani, S.; Tahvildari, L. (2011). "Size-Constrained Regression Test Case Selection Using Multi-Criteria Optimization," *Software Engineering, IEEE Transactions on*, vol.PP, no.99, pp.1, 0. doi: 10.1109/TSE.2011.56. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5928351&isnumber=4359464>.

- Sapna, P. G.; Mohanty, H. (2010). Clustering test cases to achieve effective test selection. In: Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India. New York, NY, USA: ACM, 2010. (A2CWIC'10), p.15:1-15:8. ISBN 978-1-4503-0194-7. <http://doi.acm.org/10.1145/1858378.1858393>.
- Simons, A.J.H.; Wenwen Zhao. (2009). "Dynamic Analysis of Algebraic Structure to Optimize Test Generation and Test Case Selection," Testing: Academic and Industrial Conference - Practice and Research Techniques, 2009. TAIC PART '09. , vol., no., pp.33-42, 4-6 Sept. 2009. doi: 10.1109/TAICPART.2009.28.
- Sommerville, Ian. (2007). Engenharia de *Software*. Oitava Edição. São Paulo, Pearson Addison Wesley: 2007.
- Tsai, W.T.; Xinyu Zhou; Paul, R.A.; Yinong Chen; Xiaoying Bai. (2007). "A Coverage Relationship Model for Test Case Selection and Ranking for Multi-version *Software*," High Assurance Systems Engineering Symposium, 2007. HASE '07. 10th IEEE, vol., no., pp.105-112, 14-16 Nov. 2007. doi: 10.1109/HASE.2007.64. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4404732&isnumber=4404712>.
- Vilkomir, S.A.; Swain, W.T.; Poore, J.H. (2008). "Combinatorial Test Case Selection with Markovian Usage Models," Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on, vol., no., pp.3-8, 7-9 April 2008. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4492446&isnumber=4492438>.
- Yu, L., Xu, L., & Tsai, W. (2010). Time-constrained test selection for regression testing. [www.scopus.com](http://www.scopus.com).
- Zhu Xiaochun; Zhou Bo; Li Juefeng; Gao Qiu; (2008). "A test automation solution on GUI functional test". Industrial Informatics, 2008, 6th IEEE International Conference on , vol., no., pp.1413-1418, 13-16 July 2008. doi: 10.1109/INDIN.2008.4618325.