

## RESUMO

**Artigo:** *Software Testing: A Research Travelogue (2000–2014)*

Segundo os autores do artigo, teste de software é uma abordagem amplamente praticada para avaliação e melhoramento da qualidade de software. Em virtude desse fato, da publicação do artigo de Mary Jean Harrold sobre o assunto e da recusa desta em produzir uma atualização de seu trabalho, os autores assumiram a tarefa de produzir tal atualização, para que fosse possível documentar o estado da arte da área naquele momento.

O artigo apresenta um relato de algumas das pesquisas mais bem sucedidas na área de testes de software, cuja escolha teve como base as respostas à uma consulta feita a mais de 50 pesquisadores ativos da área e áreas que obtiveram o maior progresso e impacto.

Como resultado do trabalho, foram identificadas e descritas contribuições científicas, que são impulsionadas por atividades de pesquisa, e contribuições práticas, que são impulsionadas por esforço da indústria, além de oportunidades em ambos os casos.

As contribuições científicas, que totalizaram nove (9), foram agrupadas nas seguintes categorias:

1. geração automatizada de entrada de teste: geração de conjuntos de valores de entrada para um programa ou componente de programa, que têm sido beneficiada pelos avanços em execução simbólica, teste baseado em pesquisa, testes aleatórios e combinação de técnicas;
2. estratégias de teste: categoria que incluiu os testes combinatórios, testes baseados em modelo e mineração e aprendizado com dados de campo;
3. testes de regressão: testes que levam em consideração a versão original do programa e sua versão alterada, e o respectivo conjunto de testes para se testar a versão alterada, e cujas técnicas abordadas foram RTS (Regression Test Selection), TCP (Test Case Prioritization), TSR (Test Suite Reduction) e TSA (Test Suite Augmentation);
4. suporte para estudos empíricos: técnicas de teste tipicamente heurísticas e cujo desempenho pode variar em virtude da carga de trabalho e cenários.

Um ponto interessante com relação à execução simbólica, que executa um programa utilizando entradas simbólicas ao invés de concretas, é o considerável impacto prático do *white-box fuzzing*, implementado no SAGE (*Scalable Automated Guided Execution*) para identificação de vulnerabilidades de segurança e que foi utilizado pela Microsoft Corporation em centenas de grandes aplicações Windows.

Em relação às contribuições práticas, foram identificadas duas (2) principais:

1. Arcabouços para execução de testes: propiciam automatização de testes, como no caso do JUnit, NUnit e PHPUnit;
2. Integração contínua: prática que consiste em integrar constantemente o trabalho de diferentes desenvolvedores, que estão em cópias de um mesmo software.

Embora não existam contribuições significativas em várias outras áreas, algumas delas se apresentam como oportunidades e desafios, tais como: técnicas de testes para sistemas modernos do mundo real, oráculos (problema do oráculo), análise probabilística de programa, testes de propriedades não funcionais, testes baseados em domínio e aproveitar a nuvem (Cloud) e a multidão.

É importante observar que o artigo em questão não abordou todos os temas e esforços relacionados ao teste de software, mas apresentou os mais relevantes atualmente pesquisados.