

**UNIVERSIDADE SÃO JUDAS TADEU
BACHARELADO EM SISTEMA DE INFORMAÇÃO**

**Geovane Augusto Costa dos Santos
Natan Fernandes Araujo Ibiapina
Henryk Bagdanovicius Roza
Olivia Frankiw
João Luiz Santana Borean**

CONCEITOS E ESTRATÉGIAS DE TESTES DE SOFTWARE

**São Paulo
2025**

Introdução

Roger S. Pressman, no livro Engenharia de Software, aborda a importância crítica do teste de software dentro do ciclo de desenvolvimento, destacando que essa atividade deve ser planejada e executada de forma sistemática. De acordo com Ian Sommerville, testar um software significa executá-lo sob condições controladas para identificar falhas e garantir que ele atende aos requisitos especificados.

Conceitos de Testes

Os testes de software são atividades fundamentais para garantir que o software funcione conforme o esperado e que erros sejam detectados antes da entrega ao cliente. Porém, como enfatiza Pressman, o teste não pode provar a ausência de defeitos, apenas demonstrar sua presença. Ele também observa que "a ausência de erros não significa que o software está correto", reforçando a necessidade de validação para garantir que o software atenda aos objetivos do usuário. Sommerville define o teste de software como "o processo de executar um programa com a intenção de encontrar erros". Esse processo é essencial para detectar defeitos antes da implantação e minimizar riscos.

Estratégias de Testes

As estratégias de teste são abordagens organizadas que asseguram a avaliação eficaz de todas as partes do sistema. Exemplos incluem:

- **Teste baseado em requisitos:** Verifica se os requisitos definidos foram atendidos.
- **Teste baseado na estrutura do código:** Avalia a cobertura do código fonte.
- **Teste baseado em modelos:** Utiliza representações formais para guiar os testes.

Pressman destaca que o teste deve ser bem planejado, cobrindo diferentes possibilidades de erro e seguindo uma abordagem incremental, desde os testes menores, como unitários, até testes em maior escala, como os de sistema.

Verificação e Validação

Sommerville distingue os conceitos de verificação e validação:

- **Verificação:** Confirma que o software foi desenvolvido corretamente conforme as especificações.

- **Validação:** Garante que o software atende às necessidades e expectativas do usuário final.

De forma resumida, verificação pergunta "Fizemos o software corretamente?" e validação pergunta "Fizemos o software certo?".

Testes de Software

Englobam todas as atividades de teste, desde os testes iniciais até os finais.

- **Teste de Regresso:** Assegura que novas alterações não causaram defeitos nas funcionalidades existentes.

- **Teste de Desempenho:** Mede a eficiência do sistema sob diferentes condições de carga.

- **Teste de Segurança:** Avalia vulnerabilidades e protege contra ameaças cibernéticas.

- **Testes Unitários:** Focados em componentes individuais, como funções e classes, normalmente conduzidos por desenvolvedores.

- **Testes de Integração:** Avaliam a interação entre módulos, garantindo que funcionem corretamente juntos.

Pressman reforça que "o objetivo dos testes não é provar que o software funciona, mas sim encontrar erros antes que os usuários os encontrem". Testes eficazes devem ser planejados para identificar falhas com eficiência.

Teste de Validação, Teste de Sistema e Depuração

- **Teste de Validação:** Confirma se o sistema cumpre os requisitos do usuário, frequentemente envolvendo testes de aceitação.

- **Teste de Sistema:** Avalia o software como um todo, verificando funcionalidades, desempenho e segurança em ambientes reais ou simulados.

- **Depuração:** Processo investigativo para localizar e corrigir erros identificados durante os testes. Pressman ressalta que "a depuração é um processo diferente do teste de software", sendo necessária análise e experimentação para solucionar problemas.

Por fim, Pressman observa que "nenhum software é completamente testável". Devido às infinitas combinações de entradas e estados possíveis, é essencial adotar estratégias de teste eficientes para abordar os casos mais relevantes.