

Teste de Software

Douglas O. Cardoso
douglas.cardoso@cefet-rj.br
docardoso.github.io



Objetivos da aula

- Entender os vários estágios de teste
- Familiarizar-se com técnicas para escolha de casos de teste
- Aprender sobre desenvolvimento orientado a testes
- Conhecer as diferenças entre os principais tipos de teste

Roteiro

- 1 Introdução
- 2 Testes de desenvolvimento
- 3 Desenvolvimento dirigido por testes
- 4 Testes de lançamento
- 5 Teste de usuário

Idéias básicas sobre teste de software

- Uso simulado do sistema, reproduzindo seu uso real
- Dados e condições artificiais
- Objetivo geral: verificar funcionalidades e descobrir falhas
- Testes **NÃO** garantem a ausência de erros!

Ainda sobre objetivos...

- Demonstrar que o software atende seus requisitos
 - Para sw feito sob demanda: ao menos 1 teste para cada um dos requisitos previstos
 - Para sw genérico: testar cada funcionalidade do sistema, e combinações das mesmas
 - Classe: Testes de validação
- Encontrar erros no funcionamento do sistema
 - Travamentos, encerramentos inesperados
 - Interações indevidas com outros sistemas
 - Corrupção de dados
 - Resultados indevidos
 - Classe: Testes de defeitos

Inspeção vs Teste

- Testes: verificação dinâmica do sistema, em tempo de execução
- Inspeção: verificação estática do sistema
- Alvos de inspeção: código (principalmente), documentação, requisitos, modelos
- Inspeção e testes são complementares, e não opostos

Prós e Contras de Inspeções

■ Prós

- Evita combinação de erros, devido ao seu caráter estático
- Melhor aplicabilidade a versões incompletas do sistema
- Análise mais diversa: portabilidade, manuteabilidade, eficiência, estilo de código etc

■ Contras

- Não permite verificar o atendimento de requisitos do usuário
- Não permite verificar o atendimento de requisitos não-funcionais (performance, usabilidade etc)

Fases de teste

- Existem 3 grandes fases de teste, a serem detalhadas nas próximas seções
 - Testes de desenvolvimento
 - Testes de lançamento
 - Testes de usuário

Roteiro

- 1 Introdução
- 2 Testes de desenvolvimento
- 3 Desenvolvimento dirigido por testes
- 4 Testes de lançamento
- 5 Teste de usuário

Noções básicas

- Realizados pelos desenvolvedores/programadores
- Divisão em 3 níveis de granularidade
 - Teste unitário
 - Teste de integração
 - Testes de sistema

Teste unitário

- Avaliação de funcionalidades de componentes individuais isolados
- Teste de defeitos
- Unidades podem ser:
 - Funções, ou métodos de um objeto
 - Classes
 - Até mesmo trechos de código

Automação de testes unitários

- Teste unitário: dada um entrada, verificar se a saída foi a esperada
- Fácil automação: ferramentas para isso (JUnit, pytest)
- Realização de grande número de testes, relatório de resultados
- Sequência de 3 passos: configuração, execução, verificação

Testes de integração

- Verificação da interação correta entre componentes simples
- Parte do princípio que os teste unitários foram realizados com sucesso
- Utiliza a interface para o componente composto formado pela combinação de componentes simples

Tipos de erros de interface

- Mal uso da interface: e.g., passagem de parâmetros fora de ordem
- Má interpretação da interface: equivocadamente assumir um comportamento do componente com quem se interage
- Dessincronização: interação de componentes funcionando em ritmos diferentes, levando ao acesso de informação desatualizada

Testes de sistema

- Avalia a integração de componentes compostos
- Testa o comportamento geral do sistema
- Combina subsistemas cujo desenvolvimento talvez tenha sido totalmente separado
- Mais difícil de ser automatizado que os dois tipos de teste anteriores.

Testes de Casos de Uso

- Casos de uso, que descrevem interações o com o sistema, podem ser a base para testes de sistema
- É comum mais de componente do sistema estar envolvido em um caso de um, o que obriga sua interação nos testes
- Os diagramas de sequência relacionados aos casos de uso documentam essas interações

Roteiro

- 1 Introdução
- 2 Testes de desenvolvimento
- 3 Desenvolvimento dirigido por testes**
- 4 Testes de lançamento
- 5 Teste de usuário

O que é TDD?

- Uma abordagem para desenvolvimento baseada na alternância entre codificação e teste
- Testes são definidos **ANTES** de escrever uma única linha de código
- Objetivo de programar: passar nos teste

Atividades de TDD

- Primeiro, identifica-se a melhoria de funcionalidade a ser conseguida
- Então, é preparado um teste automatizado referente a isso
- O novo teste e os já existentes são executados (haverá uma falha, já que a mudança desejada ainda não foi implementada)
- Implemente a funcionalidade e execute o teste novamente.
- Uma vez que o programa passe em todos os testes, o desenvolvimento segue para a próxima funcionalidade

Vantagens do TDD

- Cobertura do Código
- Teste de regressão
- Resolução de bugs simplificada
- Documentação do sistema

Roteiro

- 1 Introdução
- 2 Testes de desenvolvimento
- 3 Desenvolvimento dirigido por testes
- 4 Testes de lançamento**
- 5 Teste de usuário

O que são testes de lançamento?

- Testes realizados visando a disponibilização do sistema para uso fora do ambiente de desenvolvimento.
- Em geral, é um teste do tipo *black-box*: desenvolvido usando a especificação do sistema apenas.
- Pode ser visto como uma forma de teste de sistema, mas voltado para validação ao invés de defeitos.
- Geralmente é realizado por gente não envolvida no desenvolvimento.

Testes de requisitos/cenários

- Involvem examinar cada requisito e desenvolver um ou mais testes relacionados
- Exemplos
 - Autenticação através do login no sistema
 - Download e upload de registro do paciente para o notebook
 - Agendamento de consulta
 - Acesso ao banco de dados de drogas para informações sobre efeitos colaterais

Testes de performance

- Referentes a atributos como eficiência, confiabilidade
- Precisam reproduzir o tipo de uso real do sistema
- Geralmente envolvem realizar uma série de testes menores em sequência, aumentando a carga do sistema até seu limite
- No caso de um teste de estresse, a ideia é verificar como o sistema lida com sobrecarga e as falhas que ela gera

Roteiro

- 1 Introdução
- 2 Testes de desenvolvimento
- 3 Desenvolvimento dirigido por testes
- 4 Testes de lançamento
- 5 Teste de usuário**

O que são testes de usuário?

- Testes realizados pelo próprio cliente ou usuário final do sistema
- Indispensável, mesmo após a realização de extensos testes de desenvolvimento e lançamento

Tipos de teste de usuário

- Testes alfa: realizados quando o sistema ainda está em desenvolvimento, voltados para identificação de erros tão cedo quanto possível
- Testes beta: realizados usando uma versão de lançamento, mas ainda voltado para a identificação de erros
- Testes de aceitação: realizados já no ambiente real do sistema, para indicar se ele está de fato pronto para uso pleno