
TESTES DE SOFTWARE

Uma introdução

O QUE SÃO TESTES?

Submeter a um teste; pôr à prova, experimentar.

- De acordo com a definição de dicionário, testar é:
 - Então, basicamente, testar um software é utilizá-lo como se fosse em um ambiente real.
 - Assim, validando o desenvolvimento do software até o momento do teste.
-

VERIFICAÇÃO E VALIDAÇÃO

- Dois conceitos complementares, mas igualmente importantes, são verificação e validação
- A verificação lida com o que está sendo feito.
 - Está correto
 - Está de acordo com o que o cliente pediu
- Ocorre durante o desenvolvimento

VERIFICAÇÃO E VALIDAÇÃO

- A validação é voltado ao impacto do software ao cliente.
 - O software atende à necessidade do cliente
- Ocorre geralmente ao final do desenvolvimento do software.

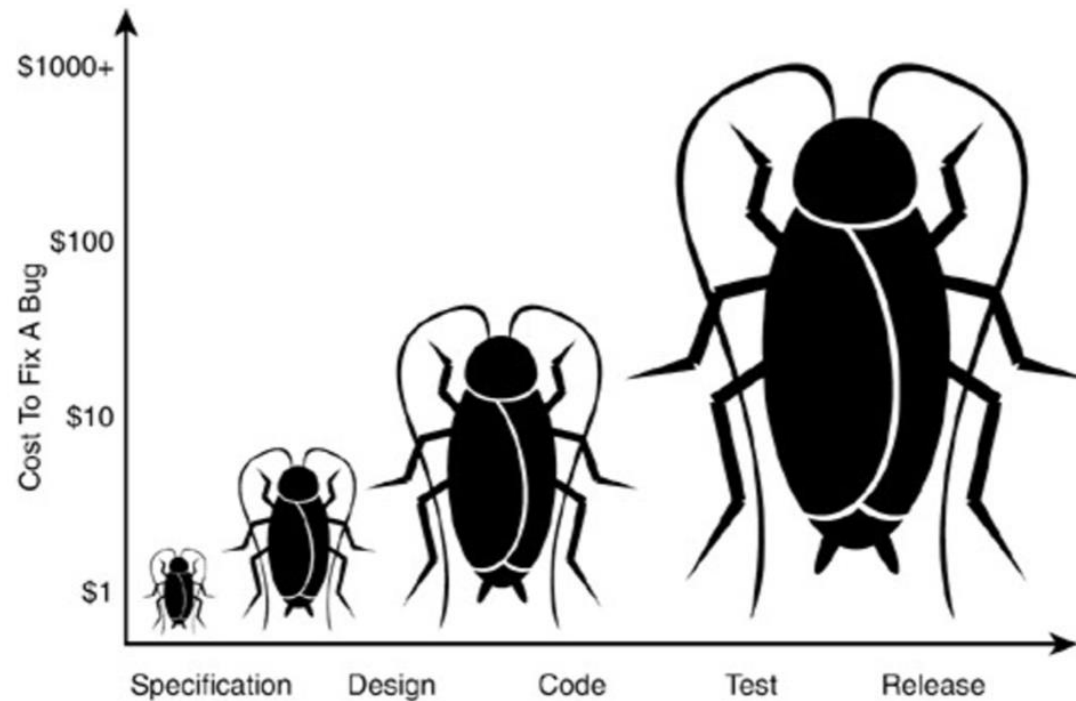
POR QUE TESTAR UM SOFTWARE

- Custos
 - O custo de realizar um teste é menor que o de realizar um recall.
- Impacto
 - Lançar um produto defeituoso pode ser uma sentença de falência
- Melhorias continuas

CUSTOS DOS TESTES

- Como tudo, executar ou deixar de realizar testes pode ter um impacto significativo
 - [Sistema de arquivos virtuais do FBI \(2003\)](#) - ~ US\$170 milhões
 - [Registros fiscais do Reino Unido \(2004\)](#) - ~ £ 85 milhões
 - [Falha no lançamento de um foguete \(1962\)](#) - ~ US\$135 milhões
 - [Global IT Outage - CrowdStrike \(2024\)](#) - ~ US\$20 bilhões

CUSTOS DOS TESTES



- Conforme o desenvolvimento do software avança, o custo médio para correção também
- No estágio de especificação de software, o custo médio por erro é de 1 dólar
- Após o lançamento, passa a ser superior a 1000 dólares.

CICLO DE VIDA DE UM SOFTWARE

- Existem duas abordagens para se desenvolver softwares:
 - Tradicional:
 - Utilizado geralmente em projetos com um escopo bem definido
 - Ágil:
 - Utilizado geralmente em projetos rápidos ou inovadores

CICLO DE VIDA DE UM SOFTWARE

- Independente da abordagem, alguns passos são comuns
 - **Coleta de Requisitos**
 - **Análise e Projeto**
 - **Implementação**
 - **Testes**
- Atualmente, as abordagens híbridas entre modelos tradicionais e ágeis são utilizados, o que torna os testes cada vez mais essenciais

CICLO DE VIDA DE UM SOFTWARE

Critério	Modelo Tradicional	Metodologias Ágeis
Flexibilidade	Baixa (mudanças custosas)	Alta (mudanças incorporadas facilmente)
Documentação	Extensa e formal	Leve e just-in-time
Feedback do Cliente	Só no final do projeto	Contínuo (a cada sprint)
Risco de Falha	Alto (se requisitos estiverem errados)	Baixo (ajustes frequentes)
Melhor Para	Projetos com escopo bem definido	Projetos dinâmicos e inovadores

TIPOS DE FALHAS

- Falhas de software usualmente são de três tipos distintos.
 - Defeitos
 - Bugs
 - Falhas

DEFEITOS E BUGS

- Anomalias de código que podem vir a causar uma falha
- Decorrem de erros de lógica, geralmente associados a falha de requisitos
- Exemplos
 - Função que realiza o cálculo de um imposto
 - Botão que não realiza nenhuma ação ao clicar.

FALHAS

- Manifestação externa de um erro ou bug
 - É percebido pelo usuário
 - Decorre de um bug/defeito não visto a tempo
-

QUADRO COMPARATIVO

Termo	Definição Formal	Exemplo Prático
Defeito (Defect)	Anomalia no código que pode causar um erro durante a execução.	Um método que calcula impostos com uma fórmula incorreta devido a um erro lógico.
Bug	Termo informal para defeito.	Um botão que não responde ao clique devido a um erro no evento onClick.
Falha (Failure)	É o resultado visível de um defeito.	O sistema exibe "Erro 500" ao tentar finalizar uma compra devido a um defeito no processamento do pagamento.

TIPOS DE TESTES DE SOFTWARE

CATEGORIZANDO

- Os testes podem ser organizados a depender do objetivo
- Ex:
 - Testes de funções individuais
 - Testes de UI
 - Testes de segurança
- Cada tipo busca validar algum tipo de situação específica

TESTES UNITÁRIOS

- São testes direcionados a avaliação de entidades únicas
 - Funções, métodos, classes
- Qual o objetivo de um teste unitário
- Garantir que o item avaliado esteja funcionando corretamente

CARACTERÍSTICAS DOS TESTES UNITÁRIOS

- Isolados, não dependem de fatores externos
- Rápidos
- Determinísticos (mesma entrada = mesma saída)
- Podem ser automatizados
 - Ferramentas como PHPUnit, Codeception podem prover automatização

TESTE NA PRÁTICA

- Considere a seguinte situação:
- Implemente uma função de validação de CPF que receba um CPF no formato "XXX.XXX.XXX-YY" e verifique apenas se os caracteres X e Y são dígitos.

TESTE NA PRÁTICA - AMPLIANDO

- Considere a seguinte situação:
- Implemente uma função de validação de CPF que receba um CPF no formato "XXX.XXX.XXX-YY" e verifique apenas se os caracteres X e Y são dígitos.

TESTE DE SISTEMA

- O teste de sistema é um teste completo
 - Todas as funcionalidades são testadas
 - Todos os fluxos
 - Ex
 - O **Sistema Interno de Votação** foi desenvolvido por alunos do curso técnico em Informática de uma escola técnica, como parte de um projeto prático voltado à aplicação de conhecimentos em desenvolvimento web. Seu objetivo é permitir que os estudantes possam votar de forma simples e digital para eleger representantes estudantis
-

TESTE DE ACEITAÇÃO

- Os testes de aceitação são os últimos a serem realizados
- Envolvem diretamente o cliente
 - Ele que dá a aceitação do produto
- Ocorre antes do processo de implantação.

EXERCÍCIO

- Desenvolva um controle de estoque simplificado. O sistema deve registrar entradas e saídas de produtos, sempre prezando pela corretude dos dados.
 - O produto deve conter nome, quantidade e preço médio do produto.
-

EXERCÍCIO

- Desenvolva um controle de estoque simplificado. O sistema deve registrar entradas e saídas de produtos, sempre prezando pela corretude dos dados.
 - O produto deve conter nome, quantidade e preço médio do produto.
 - Após desenvolvido o sistema, troque com o colega ao lado e execute os seguintes testes
 - Inserir produto com quantidade negativa
 - Inserir produto com valor negativo
 - Remover produto com valor negativo
 - Remover produto com quantidade negativa.
-