

Introdução aos testes de software

Testes, o que são?

- * Citem exemplos de testes em qualquer área

Exemplo de testes



Exemplo de testes



Exemplo de testes



Softwares podem matar!

Em 1983 uma terceira guerra mundial quase foi deflagrada por uma falha no software de alerta de radares soviéticos. A falha fez com que o software mostrasse que os E.U.A. tinham lançado cinco mísseis.

A guerra não aconteceu, pois o Tenente Coronel Stanislav Petrov considerou que a informação estava errada e não ordenou o contra-ataque, evitando assim o início da III guerra mundial.

Softwares causam prejuízos!

Por uma falha em um sistema de software para controle de ações em uma bolsa de valor, a empresa americana Knight sofreu perdas de \$ 460 milhões de dólares em apenas 45 minutos.

O Que pode acontecer?

- * Bancos perderem milhões
- * Telefones pararem de funcionar
- * Aviões terem suas rotas desviadas
- * Sistemas clínicos causarem problemas a pacientes

O Que pode acontecer para a empresa?



O Que pode acontecer para a empresa?



Dificuldades no teste de software

The header consists of a solid blue rectangular area at the top. Below this rectangle, there are several overlapping, wavy, light blue shapes that create a sense of depth and movement, resembling stylized waves or layers of paper.

Softwares são complexos



Por que software é complexo





Como o cliente explicou...



Como o líder de projeto entendeu...



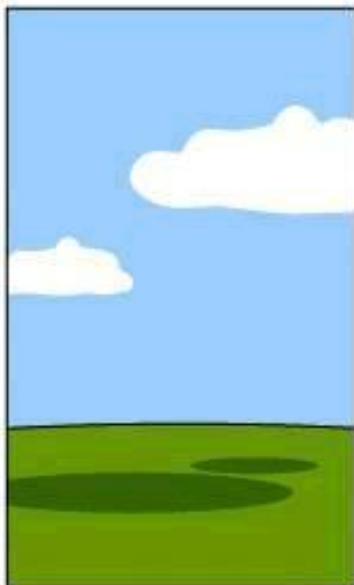
Como o analista projetou...



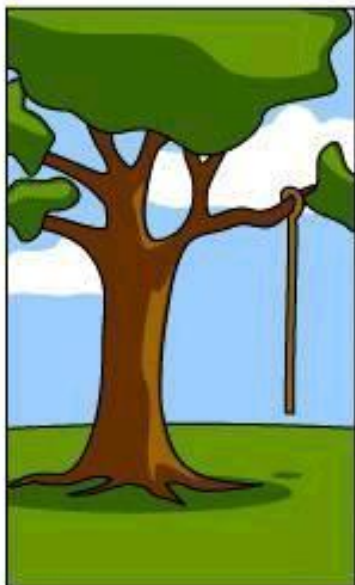
Como o programador construiu...



Como o Consultor de Negócios descreveu...



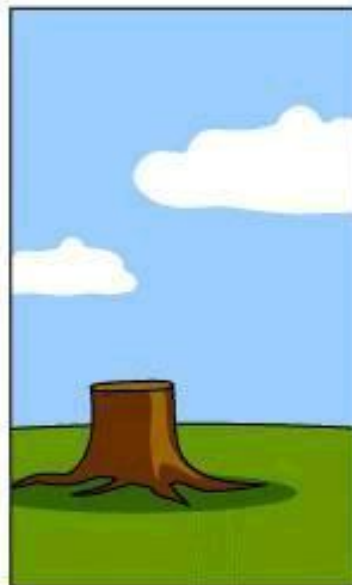
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

Várias pessoas desenvolvendo



Impaciência e tempo



Por que software

The image features a solid blue rectangular header at the top. Below the header, there are several overlapping, wavy, light blue shapes that create a sense of depth and movement, resembling stylized waves or layers. The main body of the slide is white.

Objetivo dos testes

- * Minimizar falhas
- * Atender aos objetivos especificados
- * Garantir qualidade

O Que são testes de software?

- * Verificar se o software **atende** aos seus **requisitos**
- * **Procurar** erros em programas
- * **Garantir qualidade** ao nosso produto

Qualidade na computação

- * Conformidade com requisitos funcionais e não funcionais, padrões de desenvolvimento documentados e características implícitas esperadas de todo software profissionalmente desenvolvido

Qualidade e teste de software



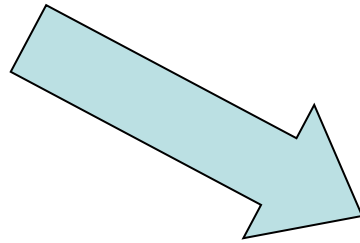
Definição de teste de software

“Teste de software é o processo de **executar o programa** com a **intenção de encontrar bugs**”

Glendford Myers, 1979

Definição de teste de software

“Testes nunca irão demonstrar a ausência de erros em softwares, apenas a sua presença.” -- E.W.Dijkstra



Principal limitação de testes é **incompletude**: é possível que teste não encontre um erro latente.

Definição de teste de software

- * “Conjunto de entradas, condições de execução e resultados esperados, desenvolvidos com o objetivo de exercitar uma parte de específica de um programa ou verificar o seu atendimento a um requisito em específico” – [IEEE]

Exemplo

```
public class Customer {  
    String getName() {...}  
}  
public class Bank{  
    static Bank createBank() {...}  
    Customer createCustomer(String name) {...}  
}
```

Exemplo

```
public class Customer {  
    String getName() {...}  
}  
public class Bank{  
    static Bank createBank() {...}  
    Customer createCustomer(String name) {...}  
}
```

```
Bank bank = Bank.createBank();  
String name = "customer1";  
Customer cust = bank.createCustomer(name);  
Assert.assertEquals(name, cust.getName());
```

Exemplo

```
public class Customer {  
    String getName() {...}  
}  
public class Bank{  
    static Bank createBank() {...}  
    Customer createCustomer(String name) {...}  
}
```

Entrada e resultado esperado:


```
Bank bank = Bank.createBank();  
String name = "customer1";  
Customer cust = bank.createCustomer(name);  
Assert.assertEquals(name, cust.getName());
```

Curiosidade

9/9

0800 Antares started
 1000 " stopped - antares ✓
 1300 (032) MP-MC ~~1.50476415~~ { 1.2700 9.032847025
 (033) PRO 2 2.130476415 9.037846995 correct
 correct 2.130676415
 Relays 6-2 in 033 failed speed test
 in relay 11.000 test.

1100 Relays changed
 Started Cosine Tape (Sine check)
 1525 Started Multi Adder Test.

1545  Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.
 1630 Antares started.
 1700 closed down.

Relay 2145
 Relay 3370

Como vocês testam seu software?



Bug Free Software

Vamos parar de gambiarra!



Como encontrar erros no software

- * Inspecionar nosso código
- * Avaliar se o software atende aos requisitos do usuário
- * Garantir a integridade do nosso código

Alguns termos importantes

- * **Defeito** – Algo que está implementado no código de forma errada
- * **Erro** – são o resultado de um defeito
- * **Falha** – comportamento da aplicação, resultados de um erro, que diferem do esperado

VV&T

- **Validação:** assegurar que o produto final corresponda aos requisitos do usuário
 - Estamos construindo o produto certo?
- **Verificação:** Assegurar consistência, completitude e corretitude do produto em cada fase e entre fases consecutivas do ciclo de vida do software
 - Estamos construindo corretamente o produto?
- **Teste:** Examina o comportamento do produto por meio de sua execução

Defeito no processo de software





Como o cliente explicou...



Como o líder de projeto entendeu...



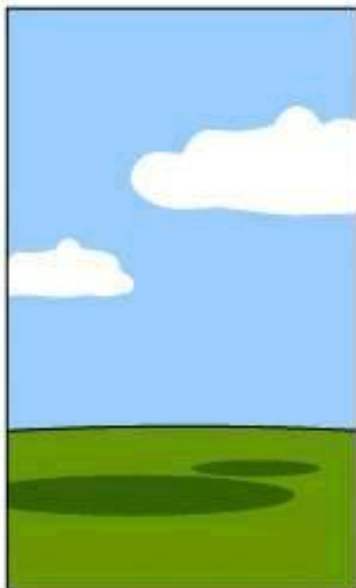
Como o analista projetou...



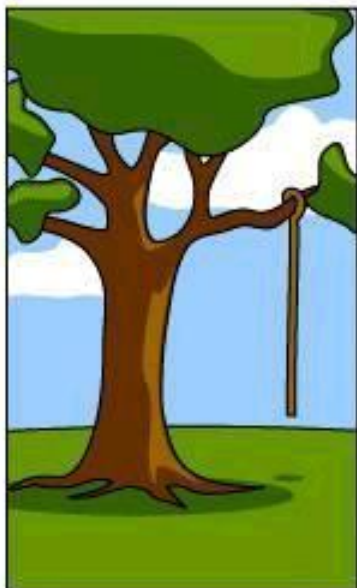
Como o programador construiu...



Como o Consultor de Negócios descreveu...



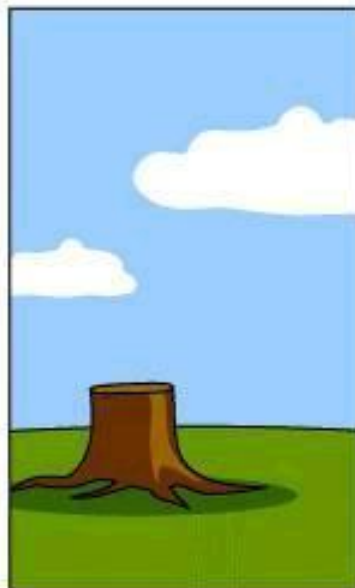
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...

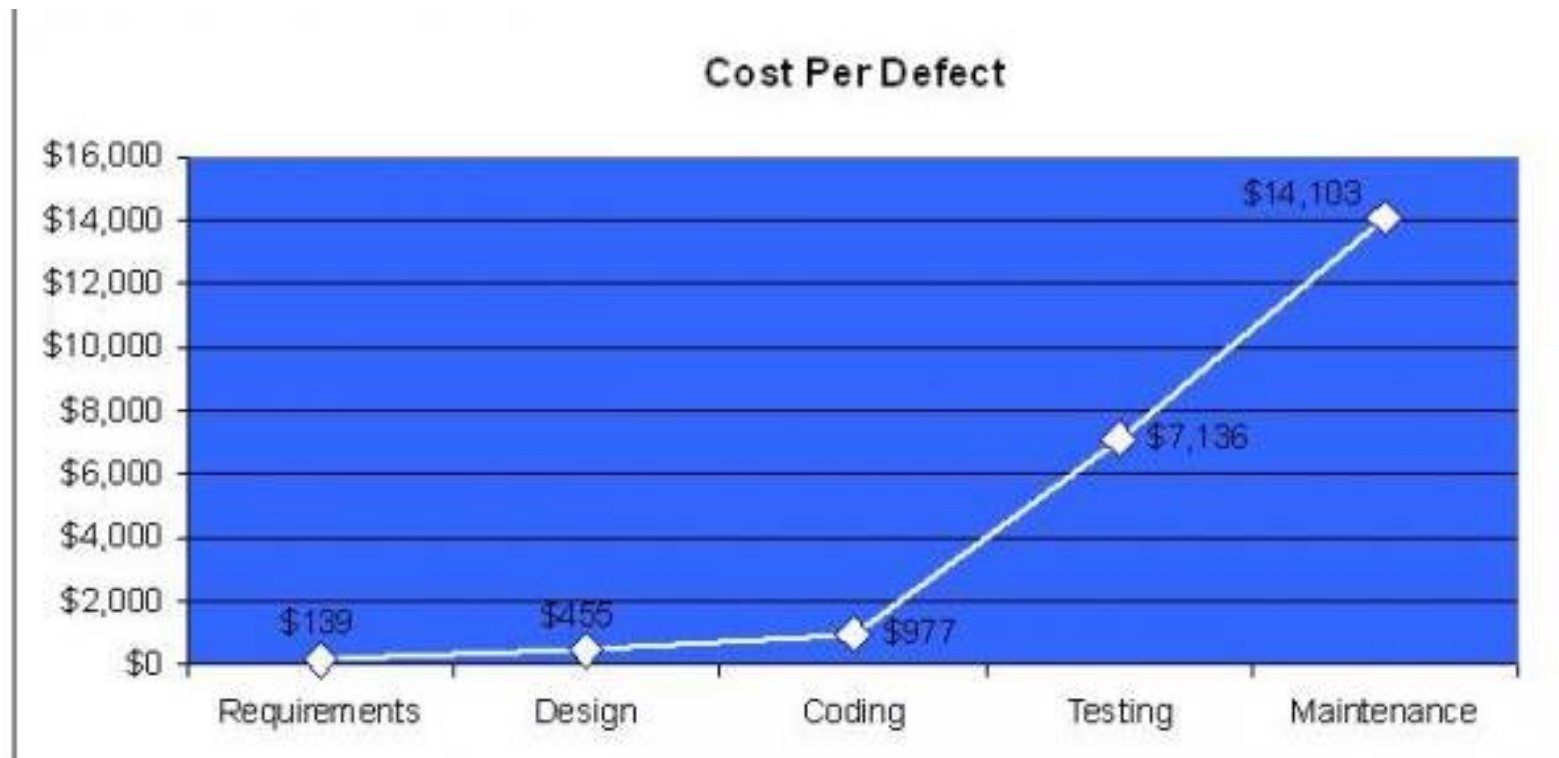


O que o cliente realmente queria...

Por que estudamos teste?

- Mais de 1/3 das falhas poderiam ser evitadas com testes [1]
- Cerca de 50% das falhas só são descobertas em produção [1]
- Segundo uma pesquisa do Departamento de Comércio dos EUA, publicada em 2002, falhas de software são tão comuns e tão danosas que se estima que causem um prejuízo anual de mais de 60 bilhões de dólares para a economia americana. [1], [2]

Por que estudamos testes?



O Que percebemos?

- Falhas de software causam grandes prejuízos
- A prática de testes ainda é pouco utilizada
- Quanto antes a presença do defeito for revelada, menor o custo da correção e maior a probabilidade de corrigí-lo
- Testes não encontram todas as falhas, mas ajudam a minimizar os erros

Quando utilizar testes?

- Desejamos oferecer uma melhor qualidade ao código
- Garantir que os requisitos estão implementados e estão corretos
- Fornecem segurança para realizar alterações no código

O Que é teste de software?

- Analisar documentos e verificar se foi produzido corretamente
- Testar funções e verificar se as entradas produzidas resultam nas saídas esperadas
- Fazer uso do software desenvolvido e verificar se está conforme os requisitos

O Que são testes de software?

- São previsíveis
- São finitos
- São (ou deveriam ser) simples
- Não são perfeitos, mas uma das melhores alternativas para validação

Quando testes não são adequados?

- Restrições impostas pelo cliente
- Tipo de software desenvolvido