



Instituto Nacional de Ciência
e Tecnologia para Engenharia
de Software

Rede de Laboratórios de Produtividade de Software

Testes de Software - Introdução

Programa de Capacitação
em Testes de Software



Por que é necessário testar?

- Reparar os problemas depois que os produtos forem entregues aos clientes, não é mais aceitável [Ian, 2003]



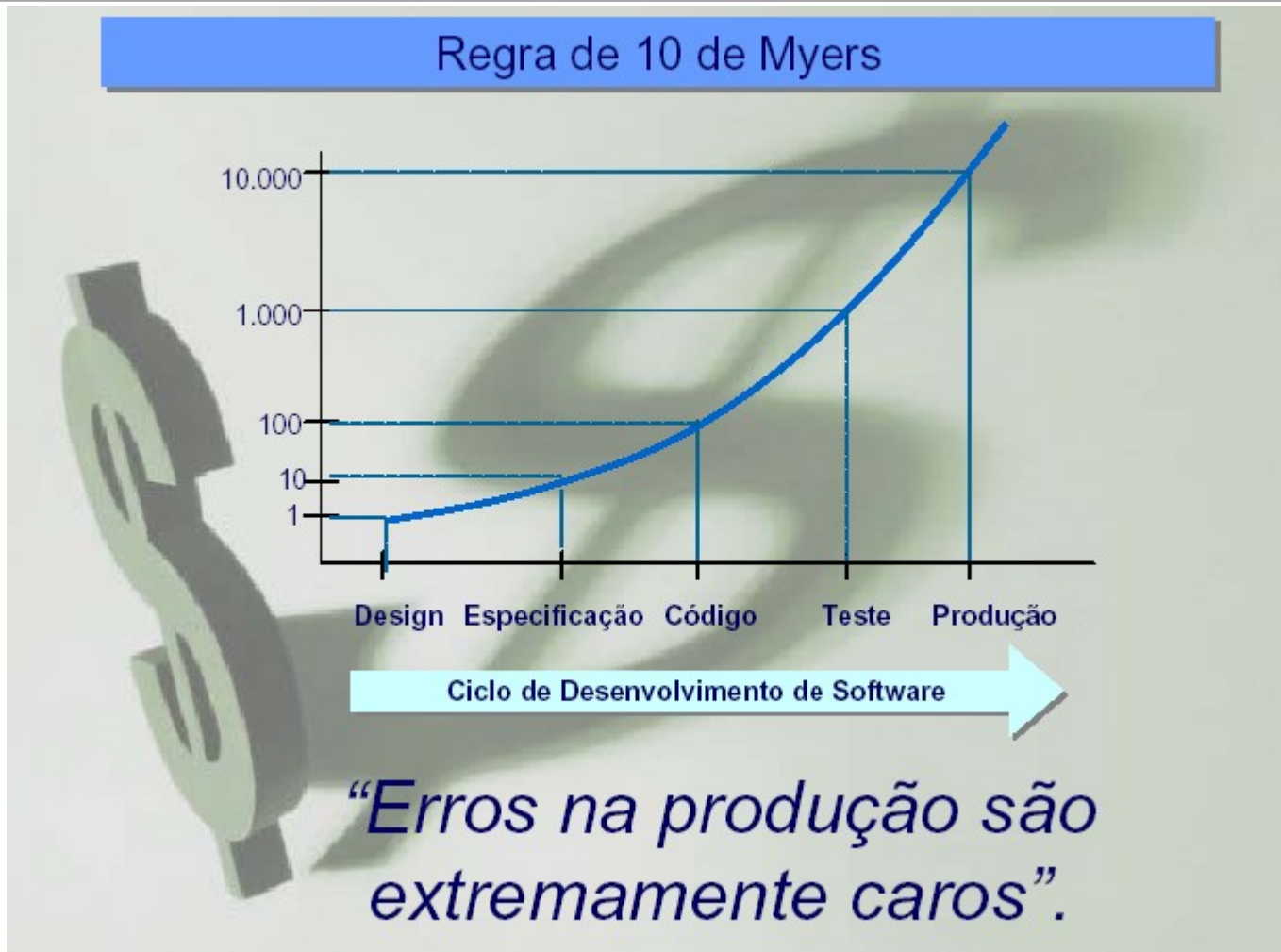
Necessidade de produzir software de qualidade.

Por que é necessário testar?

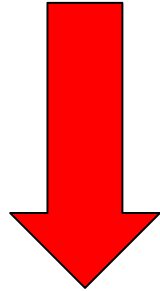
Evolução do Processo de Desenvolvimento e Testes de Software

| | 1960 | 1970 | 1995 |
|--|-------------------|------------------|---------------------|
| Tamanho do Software | Pequeno | Médio | Muito Grande |
| Complexidade do Software | Baixa | Média | Alta |
| Tamanho do Time de Desenvolvimento | Pequeno | Médio | Grande |
| Padrões e Metodologias de Desenvolvimento | Interno | Moderado | Sofisticado |
| Padrões e Metodologias de Testes | Interno | Primitivo | Emergente |
| Organizações de Testes | Bem Poucas | Algumas | Muitas |
| Reconhecimento da importância dos Testes | Pequeno | Algum | Significante |
| Tamanho do Time de Testes | Pequeno | Pequeno | Grande |

Por que é necessário testar?



Por que é necessário testar?



- **Credibilidade**

- **Prejuízos Financeiros**

- “Falha deixa Skype inoperante no mundo” (<http://tinyurl.com/yzufj98>)
- “Descobertas falhas de segurança no IE e software de messaging da microsoft” (<http://tinyurl.com/yfnfdyx>)
- “Falha em software permite download gratuito de filmes” (<http://tinyurl.com/yfuahbg>)
- “T-Mobile compensará usuários do Sidekick por perda de dados” (<http://tinyurl.com/ye8lvbe>)
- “Problema no software de gerenciamento do sistema de transmissão gera 'recall' de veículos” (<http://tinyurl.com/y8s8b93>)

Por que é necessário testar?

EUA **perdem** US\$59,5 bilhões por ano com **bugs** em **software**.

Testes mais detalhados nos programas **poderiam reduzir** esse montante em cerca de **um terço**, ou US\$ 22,5 bilhões

(National Institute of Standards and Technology, NIST - 2007)

Por que é necessário testar?

Defeitos podem ocorrer por diversas razões, tais como:

- Requisitos;**
- Complexidade do Software;
- Seres humanos são passíveis de falha;
- Pressão nos prazos de entrega;
- Mudança de tecnologia;
- Auto-Confiança excessiva;
- Pobre documentação de código;
- Ferramentas de desenvolvimento de software;
- Condições Ambientais (Radiação, Magnetismo, poluição, etc)

Por que é necessário testar?

Onde estão localizadas as falhas?

56 % - Requerimentos

27 % - Desenho / Modelagem do Sistema

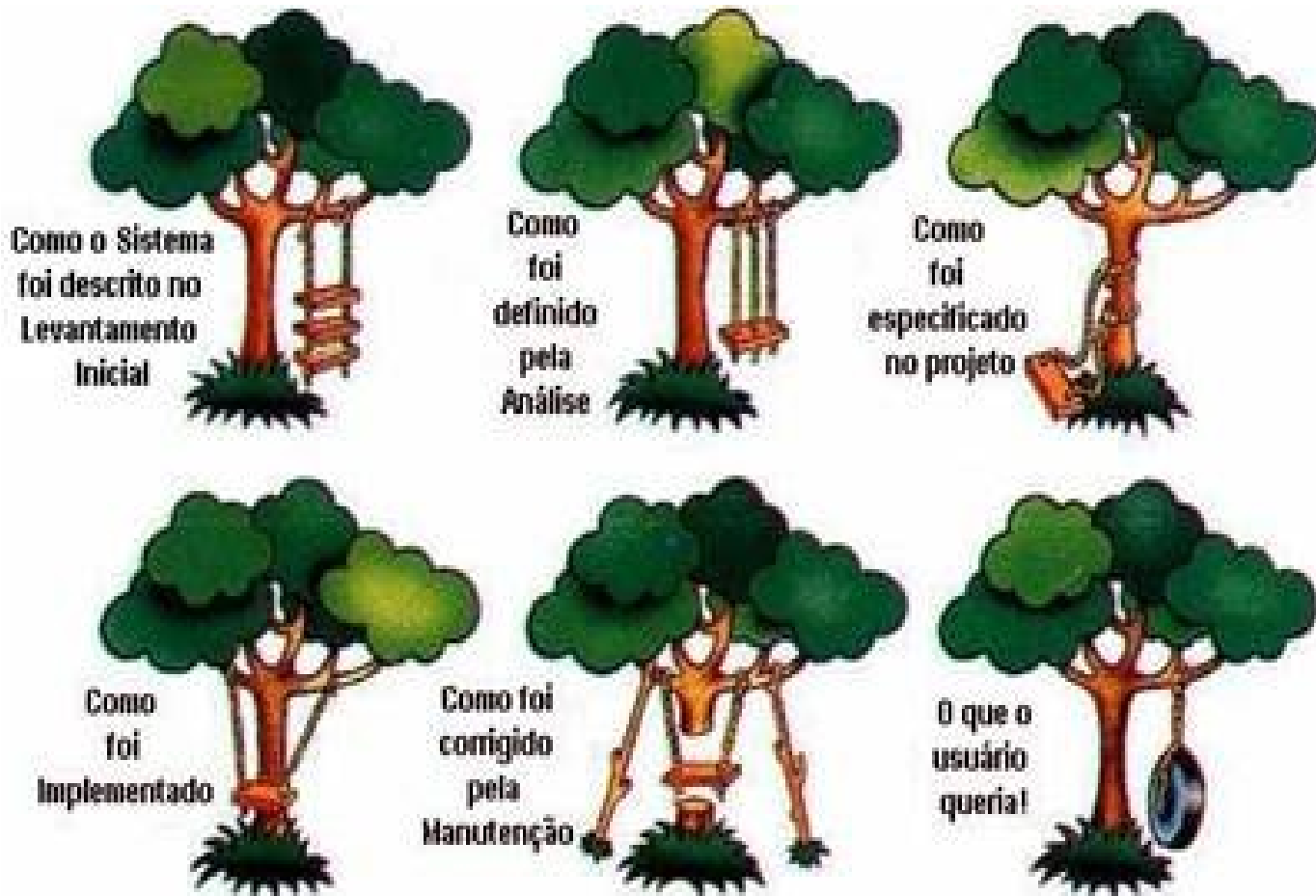
10 % - Ambiente, conversões, etc.

7% - Código

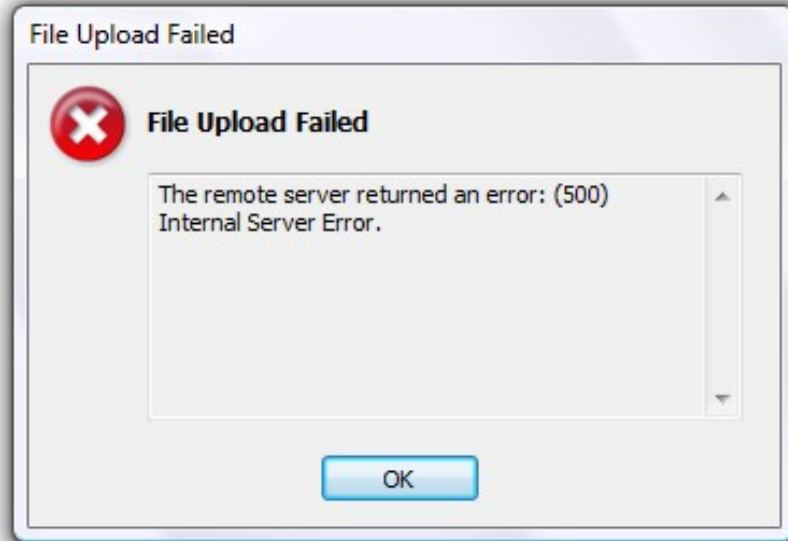


Fonte: Dick Bender – Writing Testable requirements

Por que é necessário testar?



Por que é necessário testar?



Testes
mostram a
presença de
defeitos.

Rex Black

Objetivos do Teste

Objetivos Gerais:

- Executar um programa com a intenção de **descobrir erros**;
- Encontrar defeitos **o quanto antes**;
- **Contribuir com a qualidade** do software;
- **Garantir** que o software está **de acordo** com os seus **requisitos** ou padrões específicos;
- **Questionar** um produto.

Teste x Qualidade



Qualidade é o valor agregado à **alguém**.

Teste x Qualidade

O que é um BUG?



É qualquer coisa que **implique** no **valor do produto**.

Teste x Qualidade

- Os testes **podem** AJUDAR a medir a qualidade do software em termos de defeitos encontrados;
- O teste **pode** aumentar a confiabilidade do software se poucos ou nenhum defeito for encontrado;
- Testar bem **reduz o risco** de ter um sistema com má qualidade!

Teste x Qualidade

- A qualidade do sistema **pode** aumentar quando os defeitos são consertados;
- Lições **podem** ser aprendidas: defeitos já conhecidos podem ser prevenidos.
- Teste pode ser considerado como uma das atividades de garantia da qualidade.

O Engenheiro de Testes

Por que o testador é necessário?

- Os testes feitos por desenvolvedores
 1. Tendem a verificar apenas o “caminho feliz”
 2. Normalmente são otimistas
 3. Não são sofisticados
- Organizações iniciais tem usualmente 5 testes de “caminho feliz” para cada teste de “caminho alternativo”
 - Organizações maduras usualmente tem o oposto - 5 de caminho alternativo para cada um de caminho feliz
- Desenvolvedores normalmente só conseguem enxergar 50-60% dos casos de teste do seu código

O Engenheiro de Testes

Provemos **informações**
que ajudam a **decidir**
sobre a possível
qualidade de um
produto e na tomada de
decisões



Erros Clássicos em Testes

Podemos não testar tudo?



- Hoje é amplamente aceito que você não pode testar tudo
- Você pode encontrar testadores exaustos, mas testes exaustivos, não!
- Testes completos nem teoricamente nem praticamente são possíveis

O Engenheiro de Testes

Papéis:

- Arquiteto de Testes;
- Projetista de Testes;
- Implementador de Testes;
- Executor de Testes;
- Líder de Testes

O Engenheiro de Testes

Procurar falhas num sistema requer:

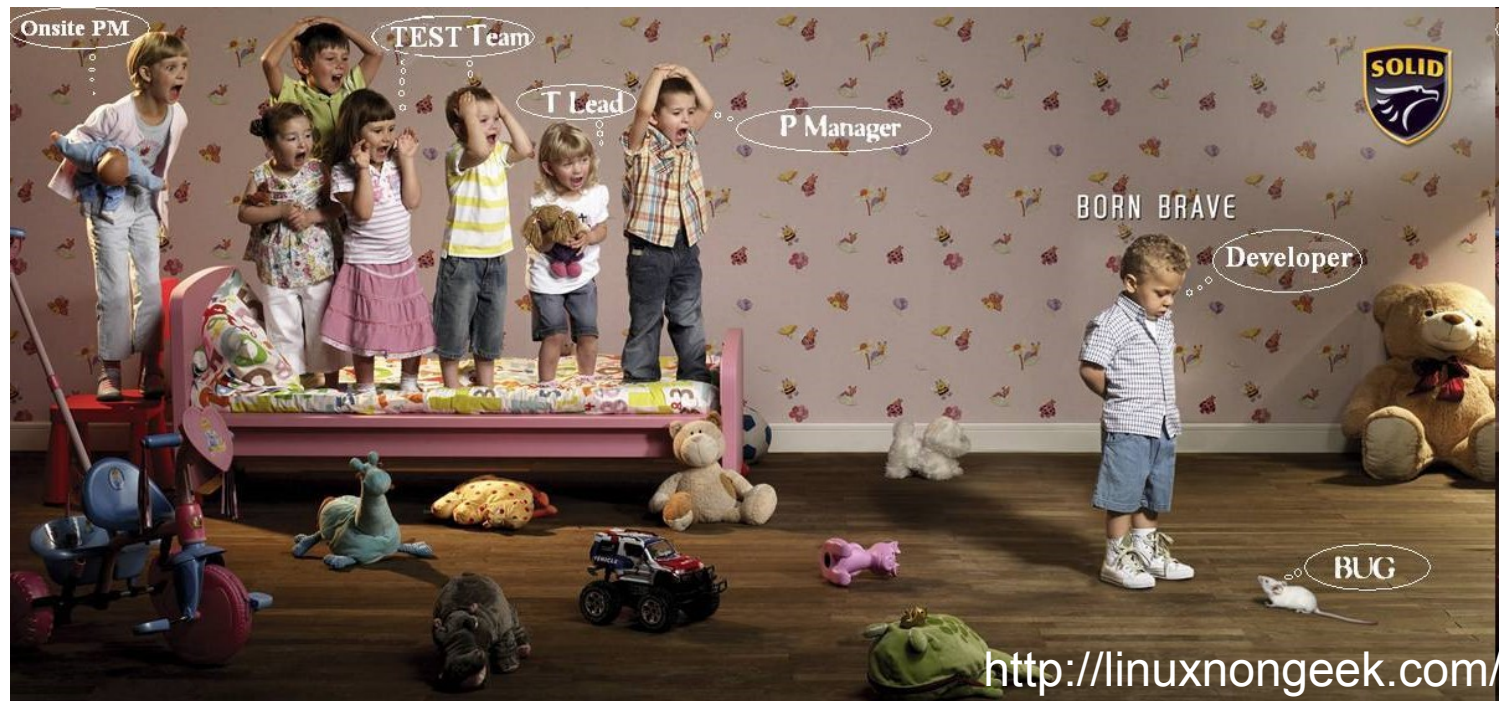
- Curiosidade;
- Pessimismo profissional;
- Olho crítico;
- Atenção aos detalhes;
- Boa comunicação;
- Conhecimento diversificado.



Identificar falhas pode ser visto como uma **crítica** ao **produto** ou ao **autor** do produto;

Erros Clássicos em Testes

1. O Papel da Equipe de Testes.



Responsável por **garantir** a **qualidade** do produto?

Erros Clássicos em Testes

1. O Papel da Equipe de Testes.

A equipe de testes deve **contribuir** com a **qualidade** do produto

- Frequentemente são os únicos na organização que utilizam a ferramenta como um expert;
- Conhecem o sistema como um todo;
- Priorizam o que é realmente importante para o usuário

Erros Clássicos em Testes

2. Planejando o esforço de testes



- Realizar **apenas** testes funcionais;
- **Desconsiderar** testes de configuração, instalação, documentação;
- Deixar os testes de carga e stress para o **último momento**;
- Confiar **apenas** no Beta testing;
- **Não identificar** corretamente as **áreas de risco**.

Erros Clássicos em Testes

3. O trabalho da equipe de testes



- Dar **mais atenção** a **execução** do que o **design** dos testes;
- **Excesso** de **detalhes** nos procedimentos;
- Não perceber e/ou explorar aspectos **“irrelevantes”**;
- Conjuntos de testes compreensíveis **apenas** por seus “donos”;
- **Reportagem** de defeitos **deficiente**;

Ferramentas

- **Ferramentas de Gerenciamento**
 - Testes
 - Requisitos
 - Incidentes
 - Configuração
- **Ferramentas de Suporte a Testes Estáticos**
 - Processo de Revisão
 - Análise Estática (D)
 - Modelagem (D)
- **Ferramentas de Suporte a Especificação**
 - Design de Testes
 - Preparação de dados de teste

Ferramentas

Potenciais Benefícios

- Redução de trabalho repetitivo
- Consistência
- Repetitividade
- Facilidade de acesso à informações geradas

Riscos

- Expectativas não realistas
- Subestimar o tempo / custo para a introdução da ferramenta
- Subestimar o tempo para gerenciar os testes produzidos
- Super confiança na ferramenta

O Mercado

- Mercado em expansão contínua;
- Algumas licitações já exigem testadores habilitados;
- Valorização por parte de:
 - Desenvolvedores
 - Compradores de serviços
 - Governo
- A Índia hoje lidera o mercado mundial de offshore

Resumo

<- Contexto ->

Softwares de baixa qualidade

<- Problema ->

Usuários insatisfeitos

Software não atende os requisitos esperados

<- Solução ->

Testar do início ao fim do projeto

<- Avaliação da solução ->

Fundamental para alcançar os níveis de qualidade esperados.

Dúvidas



Referências

- Black, Rex. Managing the Testing Process. Canadá – Wiley, 2002
- Beizer, Boris, 1995, Black Box Testing, Wiley
- Black, Rex, 2007. Foundations of Software Testing, Thompson Learning
- Kaner, Cem; Bach, James; Pettichord Bret. Lessons Learned in Software Testing – Wiley, 2002
- Sommerville, Ian. Engenharia de Software. 6a. Edição. São Paulo - Addison Wesley, 2003

Links Interessantes

- www.stickyminds.com
- www.satisfice.com
- www.kaner.com
- www.testingeducation.org
- www.pettichord.com
- www.amland.nohome.c2i.net/schaefer/
- www.whatistesting.com
- www.onestoptesting.com
- www.developsense.com/blog.html
- www.softwareqatest.com
- www.softwaretestinghelp.com

Sites

Wiki do LabPS

<http://twiki.cin.ufpe.br/twiki/bin/view/LabPS/WebHome>

Twitter do LabPS

http://twitter.com/Rede_LabPS

INES

<http://www.ines.org.br>

Amigos

<http://amigos.ines.org.br>

Cin/UFPE

<http://www.cin.ufpe.br>



Instituto Nacional de Ciência
e Tecnologia para Engenharia
de Software

Rede de Laboratórios de Produtividade de Software