

# Qualidade e Testes de Software

**Sarah de Oliveira Alcântara**

**sarah.alcantara01@etec.sp.gov.br**

# Conceitos Básicos de Testes

- ✓ **Erro:** é uma ação humana que produz um resultado incorreto.
- ✓ **Defeito:** A manifestação de um erro no software. Também conhecido como Bug
- ✓ **Falha:** quando o sistema se comporta de forma inesperada devido ao defeito.



# Conceitos Básicos de Teste

## Artefatos de Teste

- todo o conjunto de documentação gerado pelo processo de teste de software.

## Caso de Teste

- é composto por um conjunto de entradas, por passos de execução e um resultado esperado.

## Roteiro de Teste

- É composto por um conjunto de casos de teste definidos para uma determinada especificação.

# Conceitos Básicos de Teste

## Requisitos

- regras de negócio do sistema.

## Testar

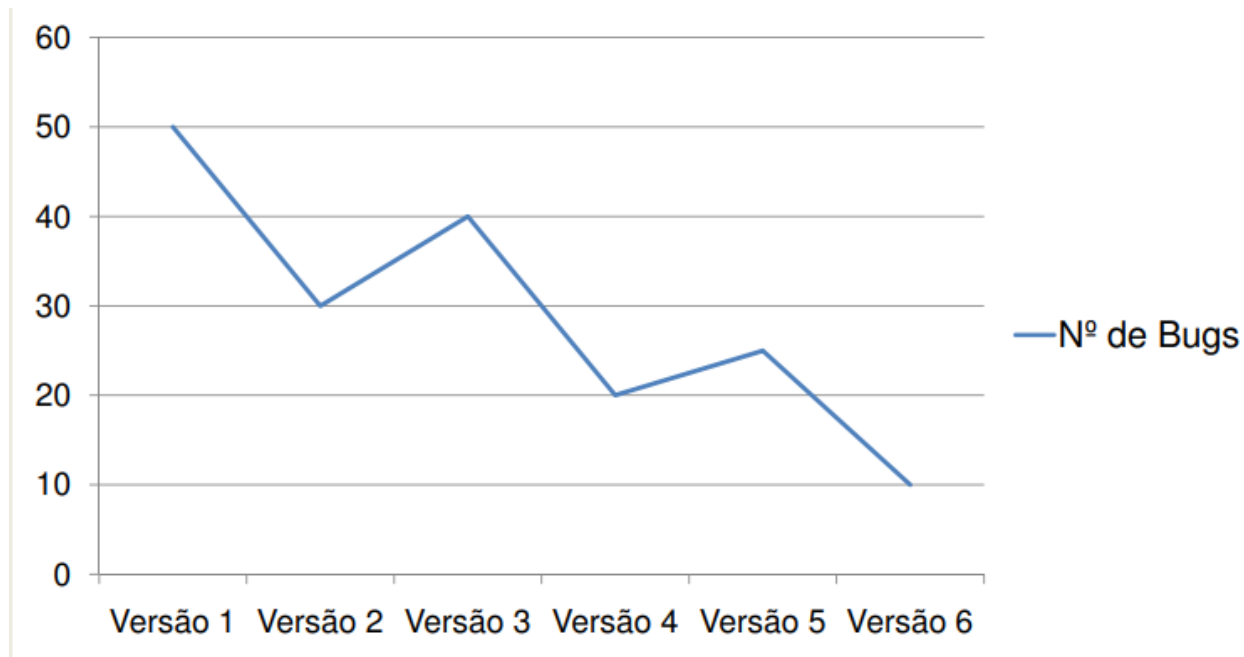
- descobrir falhas através da execução do sistema.

## Bug

- é um defeito encontrado no sistema em execução.

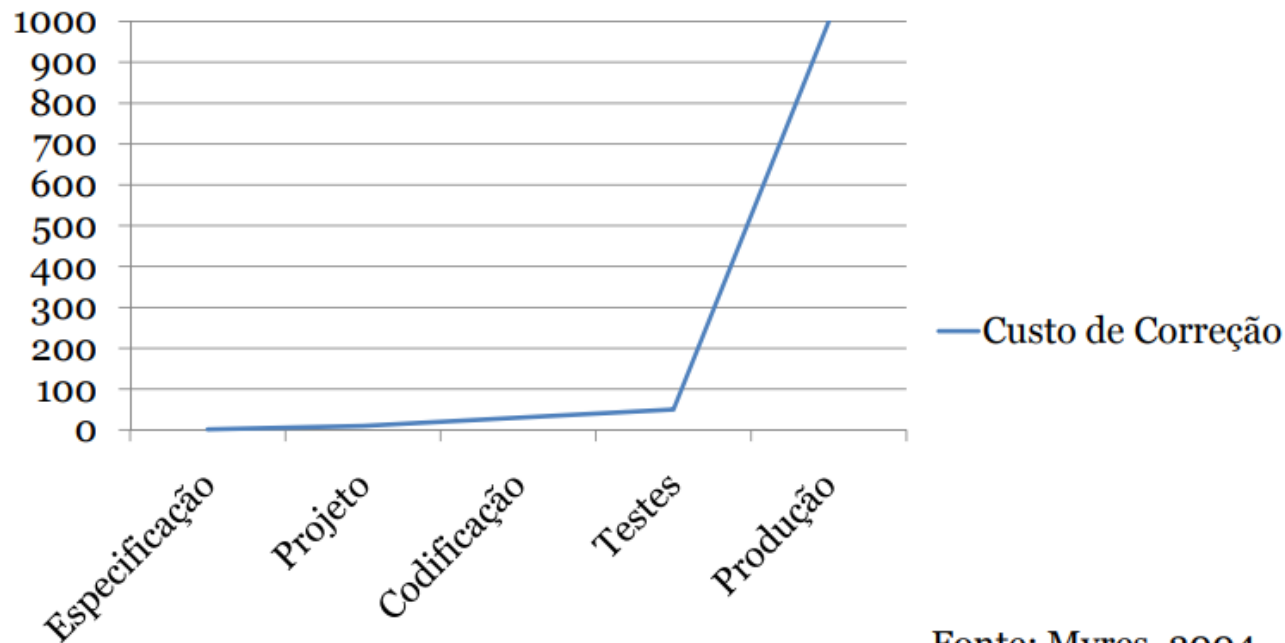
# Confiabilidade do Software

- ✓ Confiabilidade do Software é a probabilidade que o software não causará uma falha no sistema por um tempo especificado, sob condições determinadas.

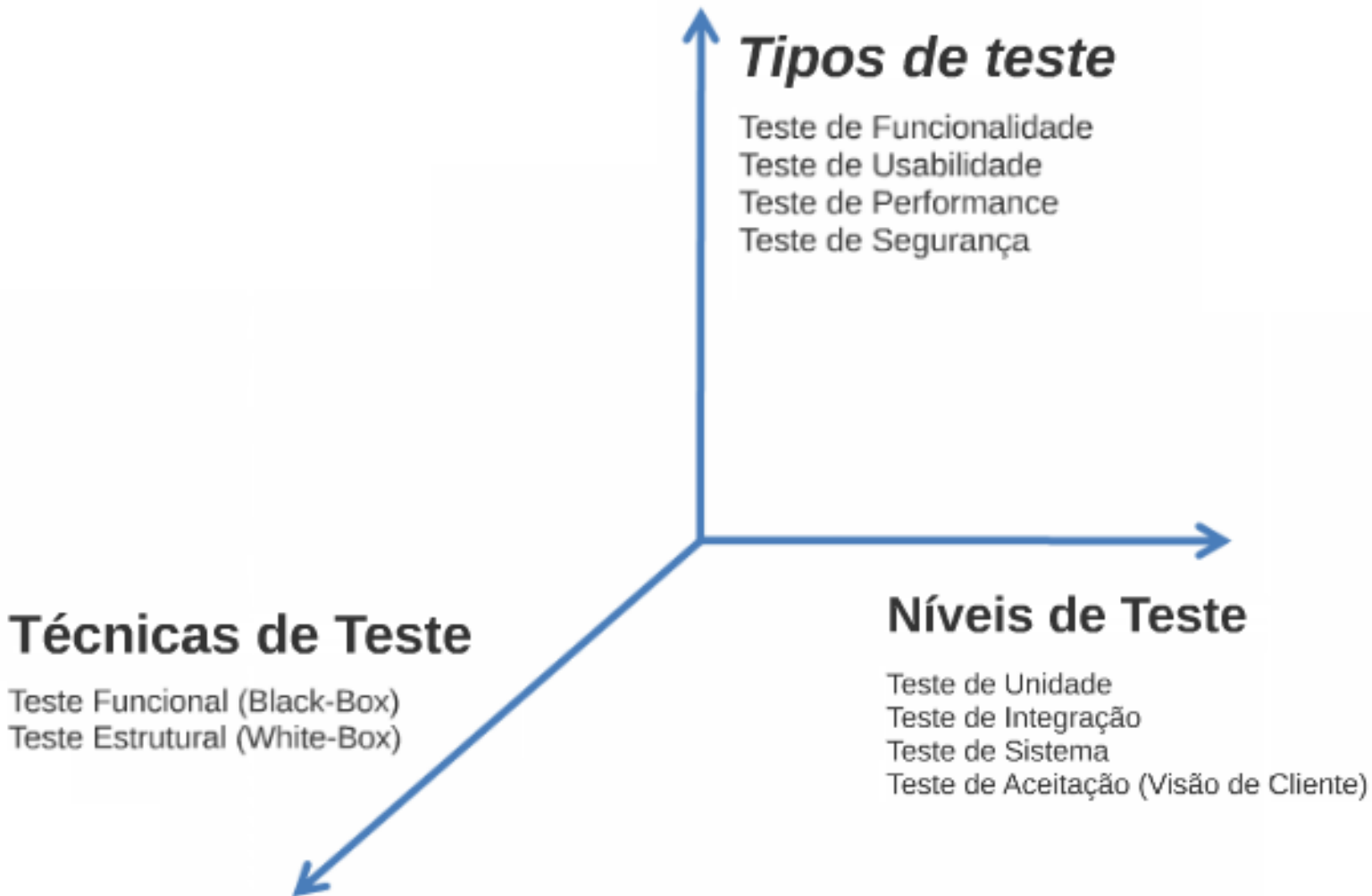


# O custo de um defeito

- ✓ O custo da correção de um defeito tende a ser cada vez maior quanto mais tarde ele for descoberto. [Myres, 2004]



Fonte: Myres, 2004



# Técnicas de Teste de Software

## ✓ Testes de Caixa-Branca (Estrutural)

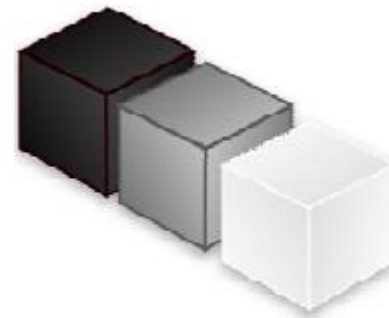
- Testes de Unidade
- Teste de Integração

## ✓ Testes de Caixa-Preta (Funcional)

- Testes Funcionais
- Testes de Aceitação
- Testes Exploratórios

## ✓ Testes de Caixa-Cinza

- Testes de Regressão
- Testes de Cobertura





# Níveis de Teste de Software

Teste Unitário



Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente

Teste de Integração



Visa provocar falhas associadas às interfaces entre os módulos quando esses são integrados para construir a estrutura do software que foi estabelecida na fase de projeto

Teste de Sistema



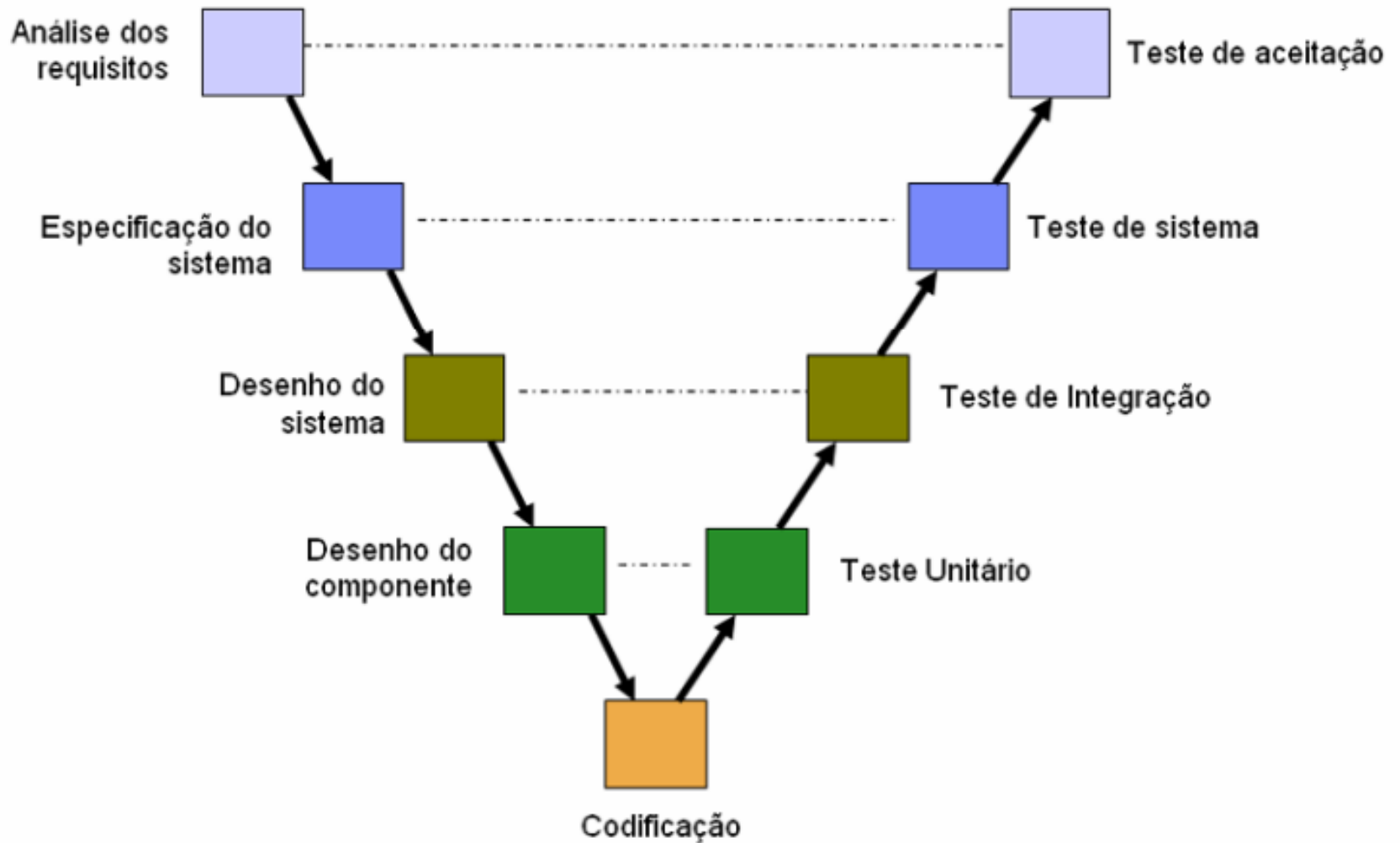
Avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia de manipulação do software

Teste de Aceitação



São realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado

# Níveis de Teste de Software



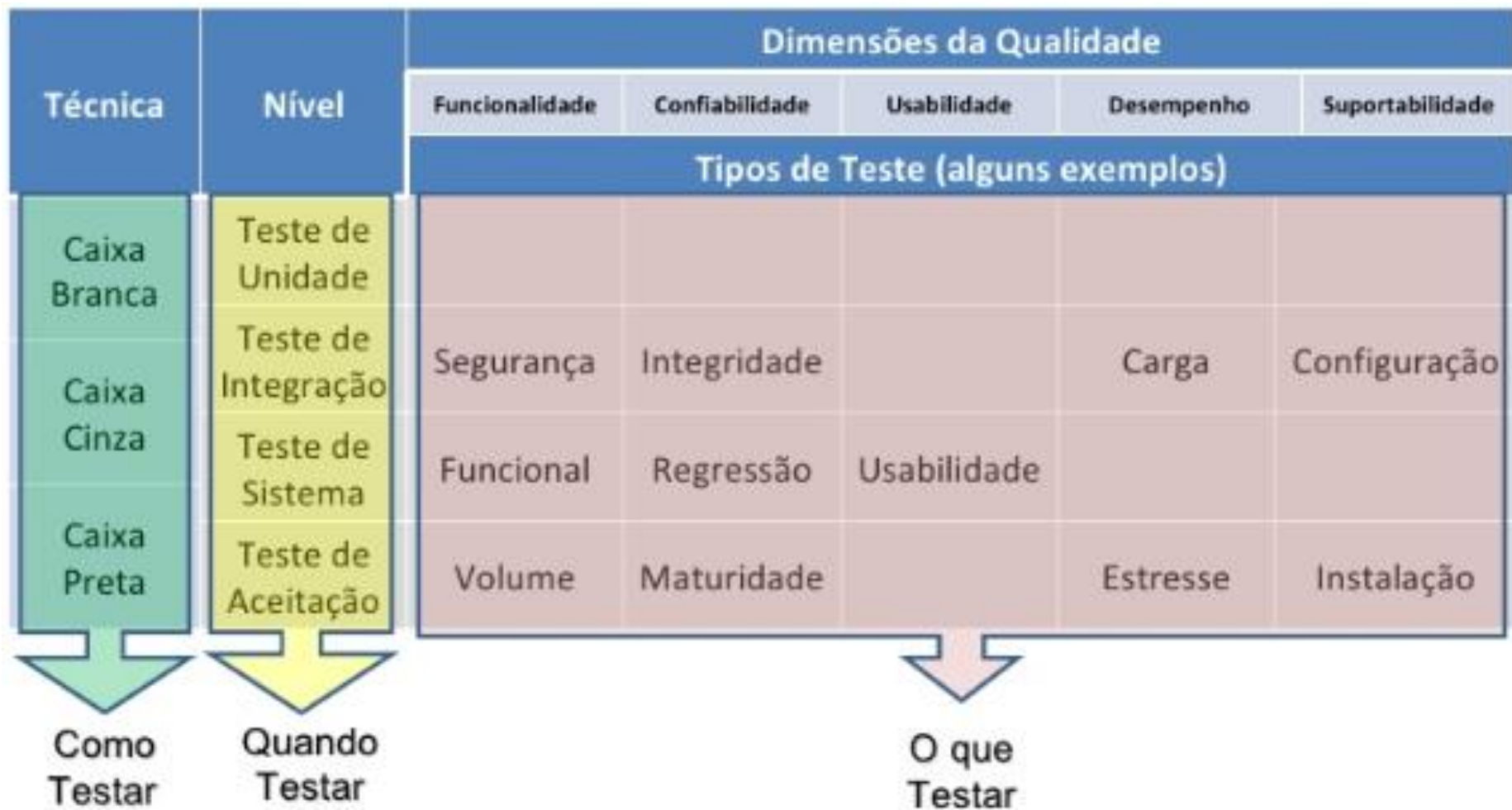
# Níveis de Teste de Software

Atributos	Nível dos Testes			
	Testes Unitários	Testes de Integração	Testes de Sistema	Testes de Aceitação
Escopo	<i>Unidades</i>	<i>Conjunto de unidades agrupadas</i>	<i>Sistema todo</i>	<i>Sistema todo</i>
Equipe	<i>Desenvolvedores</i>	<i>Desenvolvedores e Analistas de Sistema</i>	<i>Analista de Testes e Testadores</i>	<i>Analista de Testes, Testadores e Usuários</i>
Origem dos dados	<i>Criação manual</i>	<i>Criação manual</i>	<i>Criação automática / dados reais</i>	<i>Dados reais</i>
Volume dos dados	<i>Pequeno</i>	<i>Pequeno</i>	<i>Grande</i>	<i>Grande</i>
Interfaces	<i>Não existem</i>	<i>Não existem</i>	<i>Simuladas / Reais</i>	<i>Reais</i>
Ambientes	<i>Desenvolvimento</i>	<i>Desenvolvimento</i>	<i>Testes</i>	<i>Testes / Produção</i>

# Tipos de Teste de Software

<b>Características</b>	<b>Exemplos de testes</b>
Funcionalidade	Teste de funcionalidade
Confiabilidade	Teste de estresse
Usabilidade	Teste de usabilidade
Eficiência	Teste de desempenho
Manutenibilidade	Teste de caixa-branca etc.
Portabilidade	Teste de produção, teste alfa etc.

**Fonte: Bastos et al, Base de Conhecimento de testes em Software, 2012**



# A Equipe de Testes

## ✓ Gerente de teste

- Lidera a equipe de teste.
- Comunicação entre a equipe de teste e de desenvolvimento.
- Planeja os testes, define estratégias, etc.

## ✓ Arquiteto de teste

- Conhece os requisitos do sistema.
- Elabora os roteiros de teste.

## ✓ Testador

- É criativo ao executar os testes.
- Tem noções de programação.
- É objetivo ao descrever um erro.
- É perfeccionista





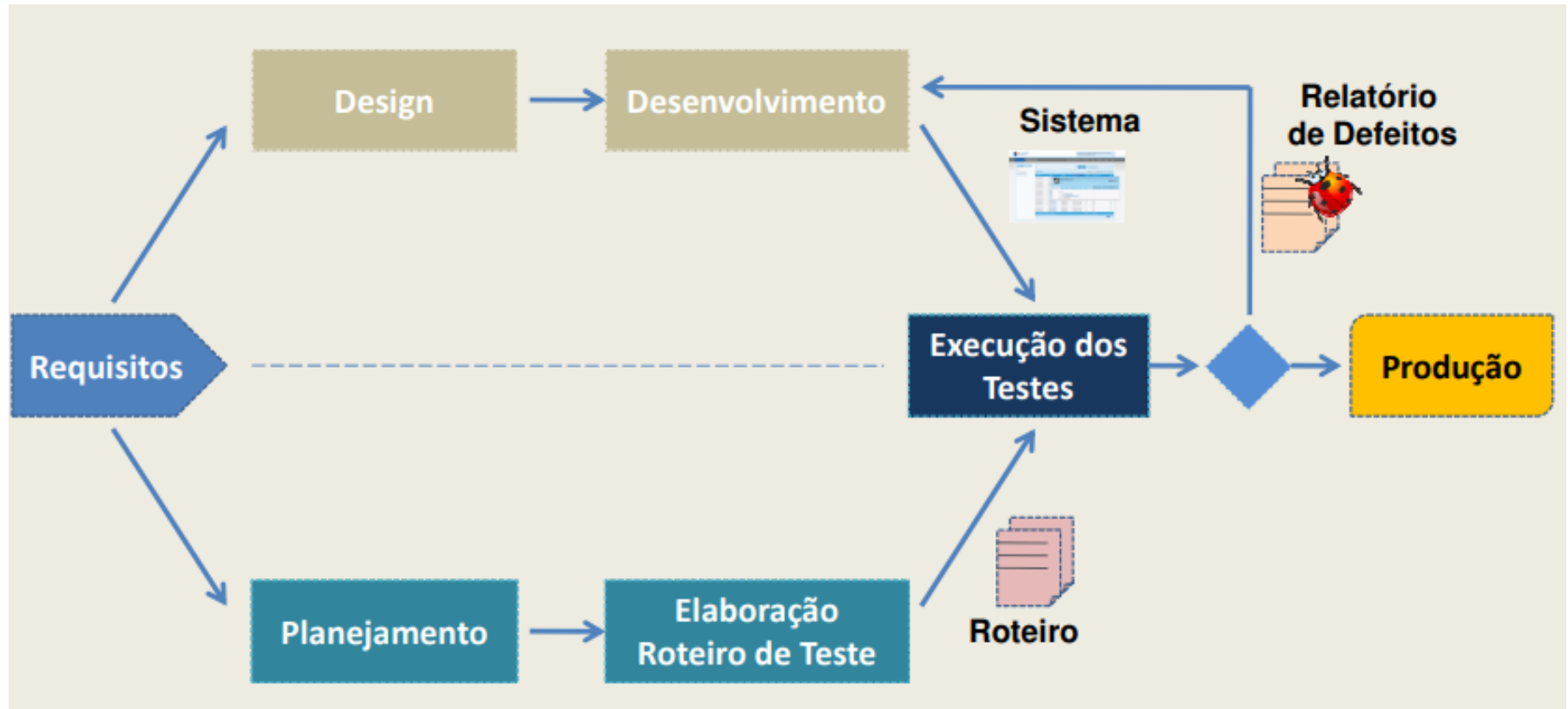
# A Equipe de Testes

## O Testador

- ✓ 1. Não deve testar seu próprio programa.
- ✓ 2. Não deve duvidar que um erro existe.
- ✓ 3. Deve ter cuidado para não reportar falsos bugs.
- ✓ 4. O testador não é inimigo do desenvolvedor.
- ✓ 5. O testador deve saber se comunicar com o desenvolvedor.
- ✓ 6. Os bugs descritos por ele devem ser baseados em fatos.
- ✓ 7. Um bom testador é aquele que encontra muitos bugs!



# Processo de Teste x Processo de Desenvolvimento





# Quando usar ferramentas de teste de software?

## ✓ Quando há apenas 1 testador para o projeto

- Deve realizar apenas testes manuais.
- Não há necessidade de ferramentas para criar os roteiros de teste.
- Ferramenta apenas para reportar os defeitos. (Redmine)

## ✓ Quando há uma equipe de teste

- Ferramenta para gerenciar a equipe. (Redmine)
- Ferramenta para criar os roteiros de teste. (TestLink)
- Ferramenta para reportar e gerenciar os defeitos. (Redmine)
- Automatizar os testes para auxiliar nos testes de regressão. (Selenium)

# Quando os Automatizar



- ✓ Frequência de execução dos testes.
- ✓ As funcionalidades são testadas mais de uma vez.
- ✓ Baixo esforço para automatizar (equipe experiente).
- ✓ Ferramentas de automação relevantes para a realidade do projeto.
- ✓ Dificuldade de executar os testes manualmente.

# HORA DE



# PRATICAR

# Bibliografia

- ✓ [Delamaro, 2007] Delamaro, M., Maldonado, J. C., Jino, M. "Introdução ao Teste de Software". Ed. Elsevier, Rio de Janeiro, 2007.
- ✓ [Dijkstra, 1972] Dijkstra, E. W. "The Humble Programmer". Communications of the ACM 15 (10): 859–866, 1972.
- ✓ [Molinari, 2008] Molinari, L. "Testes Funcionais de Software". Ed. Visual Books. Florianópolis, 2008.
- ✓ [Myres, 2004] Myres , G. F. "The Art of Software Testing". Ed. John Wiley & Sons, Inc. New Jersey, 2004.