

Teste de Software

# Introdução ao Teste de Software e sua Terminologia



**PUC Minas**

Instituto de Ciências Exatas  
e Informática

Prof. Lesandro Ponciano

Departamento de Engenharia de Software  
e Sistemas de Informação (DES)

# Objetivos da Aula

---

- Introduzir a abordagem de VV&T
  - Estática e dinâmica
- Introduzir as técnicas de teste de software
  - Caixa branca, caixa preta, caixa cinza
- Introduzir os níveis de teste de software
  - Testes durante o desenvolvimento
  - Testes da *release*
  - Testes de usuário

# Verificação, Validação e Teste (VV&T)

- Verificação

- "Estamos construindo o produto da maneira certa?"
- Verificar se o software atende os requisitos funcionais e não funcionais

- Validação

- "Estamos construindo o produto certo?"
- Validar se o software atende às expectativas do cliente
- É mais geral, pois vai além do especificado

# **VV&T Dinâmicas e Estáticas**

---

- Estáticas
  - Não requerem a existência de um programa ou modelo executável
  - Não requerem a execução do código
- Dinâmicas
  - Se baseiam na execução de um programa
  - Teste de Software é uma atividade dinâmica

# Alguns Termos do Jargão

---

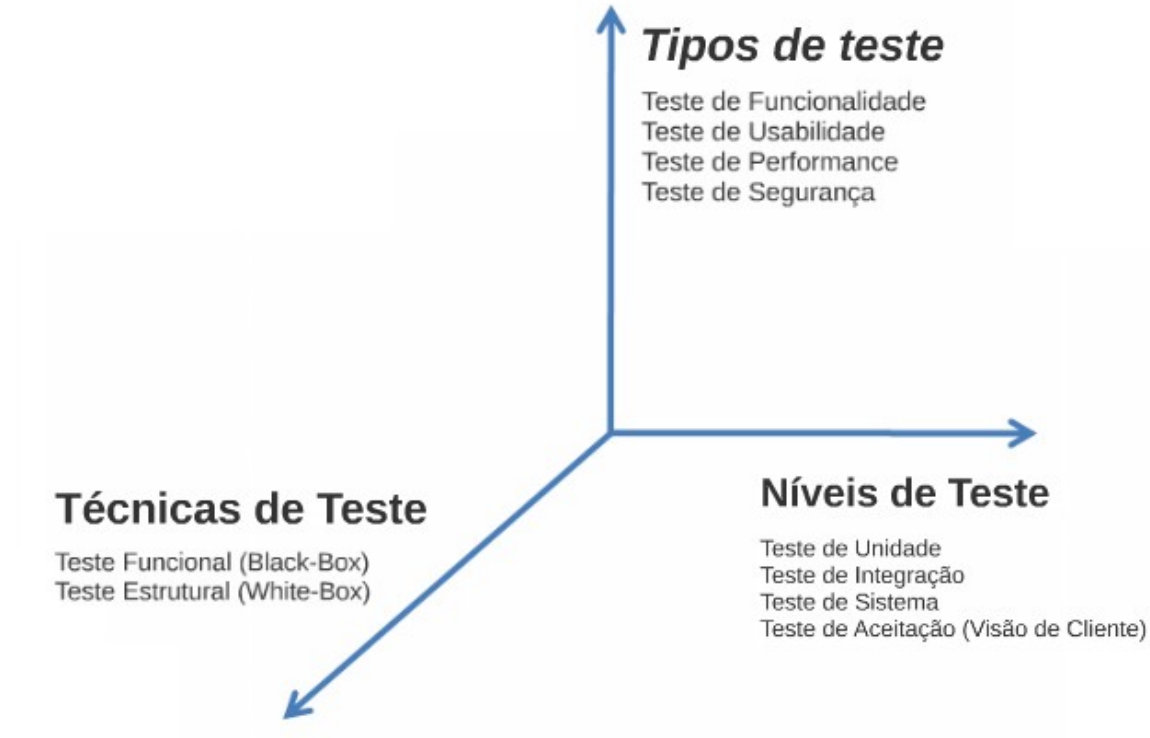
- **Defeito** (*fault*)
  - é uma deficiência algorítmica, de definição de dados ou processo que, se ativada, pode levar a uma falha. *Bug!*
- **Erro** (*error*)
  - é um estado de execução inconsistente ou inesperado
- **Falha** (*failure*)
  - é um evento notável em que o sistema viola suas especificações
- Essas definições ajudam a distinguir diferentes elementos envolvidos com o problema, mas elas
  - não são unanimidade na Engenharia de Software
  - não são seguidas consistentemente o tempo todo

# Teste e *Debugging*

---

- Teste de Software
  - Objetivo: revelar a presença de defeitos no software
  - Falhando nesse objetivo, busca aumentar a confiança sobre o software
  - "Os testes podem mostrar apenas a presença de erros, e não sua ausência" Edsger Dijkstra
- *Debugging* (depuração)
  - Consequência não previsível do teste
  - Após revelada a presença do defeito, ele deve ser encontrado e corrigido

# Categorização e Alguns Exemplos



# Técnicas de Teste

---

- Técnica Estrutural - Teste **Caixa Branca**
  - O testador conhece a estrutura interna do software
  - Testa-se fluxos da estrutura e a lógica interna do software
- Técnica Funcional - Teste **Caixa Preta**
  - O testador não conhece a estrutura interna do software
  - É orientado a entrada e saída de dados do software
- Teste **Caixa Cinza**
  - Combina elementos de teste de caixa branca para otimizar a geração de casos de teste de caixa preta



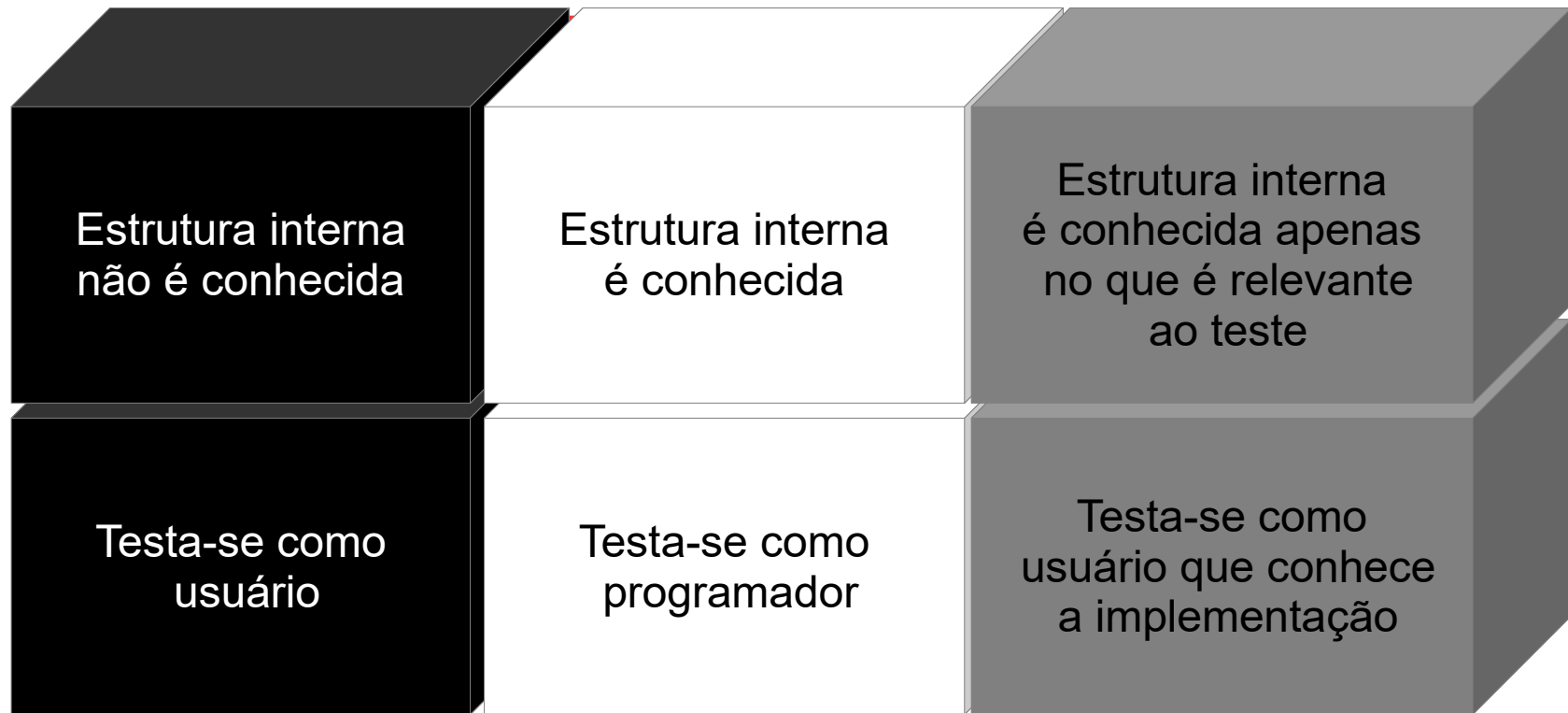


Figura baseada em "Differences between Black box, White box, Gray box testing" por Jaiganesh Periyasamy

# Níveis de Teste de Desenvolvimento

- Teste Unitário
  - Teste de Componente (ou Teste de Integração)
  - Teste de Sistema
  - Teste de Requisitos
  - Teste de Cenário
  - Teste de Desempenho
  - Teste de Usuário
- 
- Desenvolvimento
- Release*

# **Teste no Desenvolvimento**

---

- Geralmente é de caixa branca e visa encontrar *bugs*
- **Teste Unitário** (ou teste de unidade)
  - Unidade é a menor parte do software, exemplo: uma função
  - Testar independentemente cada unidade
- **Teste de Componente (ou Teste de Integração)**
  - Componentes são compostos por diversos objetos que interagem e fornecem uma interface padrão
  - Testar a interface padrão dos componentes
- **Teste de Sistema**
  - O sistema é testado com os componentes integrados
  - Testar a compatibilidade na interação entre os componentes

# Teste de Regressão

---

- Consiste em executar um conjunto de casos de teste após uma manutenção do software para testar se uma alteração adicionou novos defeitos
  - não é realizado durante o processo "normal" de desenvolvimento, apenas nas manutenções
- Quando faz-se uma manutenção, é necessário garantir que
  - As novas funcionalidades funcionam como esperado
  - As funcionalidades que já estavam implementadas continuam funcionando como esperado

# Teste de *Release*

---

- Geralmente é de caixa preta
  - O foco é validar que o software atende o desejado
  - Se o software é bom o suficiente para uso externo
- Teste de **Requisitos**
  - Casos de teste derivados dos requisitos
  - Mostrar que cada requisito está como esperado
- Teste de **Cenário**
  - Testa-se o software percorrendo um cenário de uso
  - Testam-se vários requisitos em um mesmo cenário
  - Testam-se combinação de requisitos
- Teste de **Desempenho**

# Teste de Usuário

---

- **Teste Alfa**

- Usuários trabalham com a equipe de desenvolvimento para testar o software no local do desenvolvedor

- **Teste Beta**

- Uma *release* do software é disponibilizada aos usuários para que eles usem e levantem problemas que identificaram

- **Teste de Aceitação**

- Clientes testam o sistema para decidir se ele está pronto para ser aceito e implantado no ambiente do cliente

# Atividade de Fixação

---

- 1) Qual a diferença entre VV&T estática e dinâmica? Em qual teste de software se inclui? Por quê?
- 2) Qual a diferença entre teste de unidade, de integração e de sistema?
- 3) Estabeleça semelhanças e diferenças entre teste funcional e teste estrutural.
- 4) Quais abordagens podem ser seguidas para se definir os casos de teste de *release*?
- 5) O que são teste alfa, beta, e de aceitação?

# Referências

---

SOMMERVILLE, Ian. Engenharia de Software - 9a edição. Pearson ISBN 9788579361081. (Capítulo 8)

PRESSMAN, Roger. Engenharia de software. 8. Porto Alegre ISBN 9788580555349. (Capítulo 17)

DELAMARO, Márcio; MALDONADO, José; JINO, Mario. Introdução ao teste de software. Elsevier Brasil, 2016. (Capítulo 1)