Teste de Software

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

2.3 Teste de Aceitação

- Semelhante ao teste de sistema;
- Verifica a aceitação do público em relação ao funcionamento do software;
- Esse tipo de teste leva em consideração o usuário final;
- O teste de aceitação pode ser dividido em:
  - **Teste de aceitação formal**
  - **Teste de aceitação informal**
  - **Teste beta**

2.3 Teste de Aceitação

- **Teste de aceitação formal:**

. O que será testado? Determinado módulo; tela por tela...

. Como será testado? Manualmente pelo usuário; de forma automática...

. É necessário que haja um planejamento prévio;

. É possível medir a evolução do teste (caso de uso);

. Geralmente, os defeitos encontrados são previsíveis.

2.3 Teste de Aceitação

- **Teste de aceitação informal:**

. Não há planejamento rígido;

. É um teste subjetivo;

. Geralmente, apenas um grupo seleto de usuários finais que irá realizar esse tipo de teste (contratante), portanto, não envolve a equipe técnica de analistas de sistema e desenvolvedores (contratada);

. Não é possível controlar como o sitema será testado;

. Risco de usuário final gostar de uma versão inacabada do sistema, perdendo o foco do teste.

2.3 Teste de Aceitação

- **Teste beta**

. Os testes são realizados por qualquer pessoa que venha a usar o sistema no futuro (não há seleção prévia);

. É o menos controlável;

. Defeitos muito subjetivos são revelados;

. O nível de conhecimento e de interesse dos usuários finais pode ser desnivelado e partes do sistema podem não ser testadas;

. Serve para testar uma versão “quase pronta” do sistema antes do lançamento final.

2.4 Teste de Integridade de Dados

- Consiste no processo de validação da confiabilidade e da integridade dos dados de um sistema;
- Visa assegurar a robustez do programa e sua capacidade de resistir a falhas;
- Exemplo do caixa eletrônico:
  - Se um cliente sacar R$100,00, é preciso que essa quantia seja realmente debitada da sua conta depois que a máquina entregar o dinheiro;
  - Se um valor diferente for debitado, então há falha de integridade de dados.

2.5 Testes de Configuração e de Instalação

- Garantem a o suporte do sistema;
- O sistema deve funcionar adequadamente na plataforma que foi programado. Mas, é necessário realizar testes em outras plataformas também:
  - Exemplo de teste de configuração: testar sistemas web no Chrome, Firefox e Edge. Além disso, variar o tamanho da tela.
- Casos de teste de instalação:
  - O que acontece quando a instalação é interrompida abruptamente?
  - Como o sistema se comporta com a internet desligada?

2.6 Testes de Performance

- Verificar se o sistema é rápido e estável. São três tipos:
  - **Teste de Estresse**
  - **Teste de Carga**
  - **Teste de Estabilidade**

2.6 Testes de Performance

- Submetem o software a situações extremas;
- Avaliam o comportamento do software quando ele trabalha no limite de recursos, exemplo:
  - Espaço de memória reduzido;
  - Espaço em disco rígido reduzido;
  - Poder de processamento pequeno;
  - Latência de internet alta (atraso);

2.6 Testes de Performance

- Esse tipo de teste deve ser feito em aplicações críticas. Exemplos:
  - Servidores de arquivos ou web com grande quantidade de acessos simultâneos;
  - Aplicações industriais;
  - Jogos de computador.
- Encontrar falhas em condições de operação extremas, como registrar operações incorretas, devem ser detectadas e evitadas antes do lançamento do sistema;
- Um bom teste de estresse pode revelar essas informações aos analistas de sistemas.

2.6 Testes de Performance

- Realizar testes de estresse é desafiador, porque é preciso configurar adequadamente a plataforma de execução;
  - Aplicação desktop: remover hardware manualmente do computador? Muito trabalhoso...
  - Aplicativo Android: possuir aparelhos de várias marcas, tipos, e especificações? Muito custoso...
  - Aplicação web: simular uma rede instável? Pode ser trabalhoso...
- Existem ferramentas que ajudam a emular um ambiente para teste de estresse.

2.6 Testes de Performance

- WinStress (da Ultra-X):
  - permite reduzir artificialmente o desempenho de um computador, de acordo com a configuração desejada pelo testador;
  - sendo possível variar parâmetros como carga de CPU, memória disponível, espaço em disco disponível e carga de rede.
- DieselTest, OpenSTA e DBMonster
  - permitem testar um software simulando um número arbitrário de conexões.
- JMeter
  - possibilita configurar cenários para testar aplicações web.

Teste de Software

Exercício 1:

- Considerando o JMeter, teste a tela de login de um sistema web.
- Site do JMeter: https://jmeter.apache.org/
- Baixe o JMeter (*Download Releases - Binaries*)
- Leia o manual de instruções (*User Manual*)

Teste de Software

Exercício 2:

- Visite o site http://www.opensourcetesting.org/
- Responda:
  a) Do que o site trata?
  b) Cite três ferramentas de teste descritas no site e que não foram mostradas em sala.
  c) Explique brevemente sobre as ferramentas citadas no item anterior.

Teste de Software

Prof. Dr. Alan Souza

alan.souza@unama.br

2020

3. Abordagens de Teste

Há duas abordagens principais:

- Caixa preta
- Caixa branca

Existe uma terceira abordagem:

- Caixa cinza

3.1 Teste de caixa branca

- Possui acesso ao código-fonte do sistema;
- Conhece as classes, os métodos que as compõem, etc;
- Permite uma busca precisa do comportamento de determinada estrutura;
- Exemplo: teste unitário.

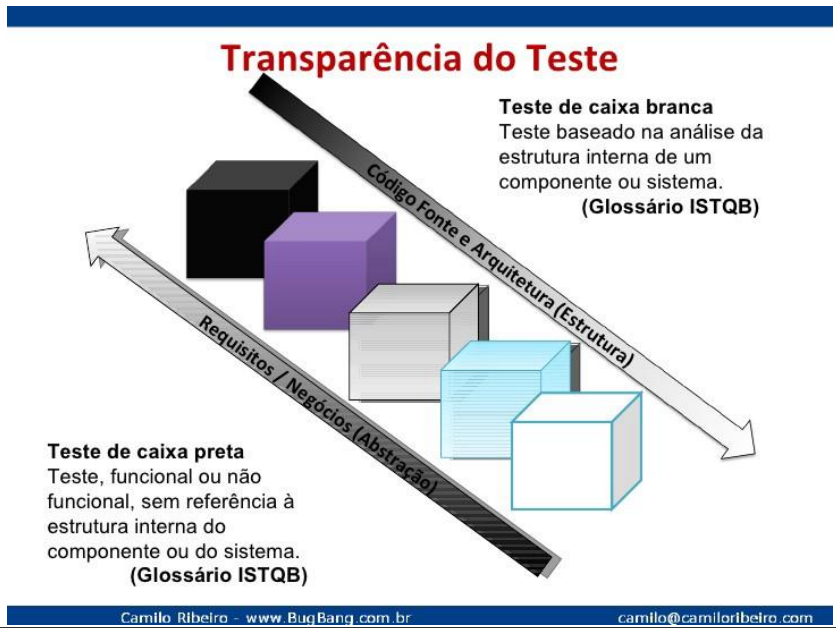
3.2 Teste de caixa preta

- Tem como base os requisitos do software;
- O foco é nas ações que o software vai desempenhar;
- São resumidos em entradas e saídas geradas pelo sistema;
- Exemplos: teste de sistema, de aceitação (formal, informal, beta), de estresse.

3.2 Teste de caixa cinza

- É uma mistura entre a caixa branca e caixa cinza;
- Não se possui acesso completo ao código-fonte do sistema;
- Conhece os requisitos;
- Pode acessar o banco de dados para verficar diretamente se determinada funcionalidade processou os dados corretamente;
- Exemplo: teste de integridade.

3. Abordagens de Teste



Teste de Software

Exercício 1:

- Em dupla, crie duas questões objetivas e uma discursiva a respeito dos tópicos abordados até esta aula;
- As questões objetivas deverão serguir os modelos do próximo slide;
- Depois que finalizar a criação, envie as questões e o gabarito para o email do professor: **alan.souza@unama**

Teste de Software

Exercício 1:

Modelo 1:

Julgue os itens a seguir:

I. $(010)_2$ é igual a $(1)_{10}$.

II. A memória RAM é permanente.

III. A raiz quadrada de 100 é 10.

São verdadeiras as afirmações:

a) I, II, III.

b) I apenas.

c) II apenas.

d) III apenas.

e) II e III apenas.

Modelo 2:

Avalie as asserções a seguir e a relação proposta entre elas.

I. O número três é ímpar.

PORQUE

II. O valor do resto da divisão de três por dois é um.

A respeito dessas asserções, assinale a opção correta.

a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.

b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.

c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.

d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.

e) As asserções I e II são proposições falsas.