ENGENHARIA DE SOFTWARE

Testes de Software

Ma. Vanessa Matias Leite

1

Unidade de Ensino: 03

Competência da Unidade:Saber identificar os tipos de testes

Resumo: Aplicar os diferentes tipos de testes de software para cada etapa de desenvolvimento

Palavras-chave: teste de software; verificação; validação; TDD; automação de teste;

Título da Teleaula: Testes de Software

Teleaula nº: 03

2

Na aula de hoje

- Verificação e Validação;
- Testes de Softwares;
- Tipos de Testes;
- Desenvolvimento orientado a testes (TDD);

3

Verificação, Validação e Teste

4

Verificação e Validação

- Processo de verificação e análise;
- Visa estabelecer a confiança que o sistema de software está adequado ao seu propósito.
- Ocorre em cada estágio do processo do software:
  - Revisões de requisitos;
  - Revisões no projeto;
  - Inspeções no código;
  - Teste do produto.

5

Verificação

- Verificar se o software está de acordo com suas especificações;
- Verificar se o software atende aos requisitos funcionais e não-funcionais não especificados.

Estamos produzindo o produto corretamente?

6

Validação

- Processo;
- Assegurar se o produto atende às expectativas do cliente;

Estamos produzindo o produto correto?

7

Teste de Software

- Consiste em uma sequência de ações executadas com o objetivo de encontrar problemas nos softwares;
- Busca encontrar defeitos e não garantir que o software é isento de problemas;

8

Teste de Software

O processo é separado em 4 grandes etapas:

- Planejamento;
- Projeto de casos de teste;
- Execução do programa com os casos de teste;
- Análise dos resultados;

9

Casos de Teste

- Entrada no programa e a saída correspondente;
- Deve-se escolher um bom caso de teste;
- Casos de teste com baixa qualidade não exercitam partes críticas do programa;

10

Planejamento dos Casos de Teste

- Definir o ambiente no qual o teste será realizado;
- Definir a entrada deste caso de teste;
- Definir a saída esperada para cada entrada;
- Definir os passos a serem realizados para executar os testes.

11

Resultado do Caso de Teste

- Passou: todos os passos do caso de teste foram executados com sucesso para todas as entradas;
- Falhou: nem todos os passos foram executados com sucesso para uma ou mais entradas;
- Bloqueado: o teste não pôde ser executado, pois o seu ambiente não pôde ser configurado.

12

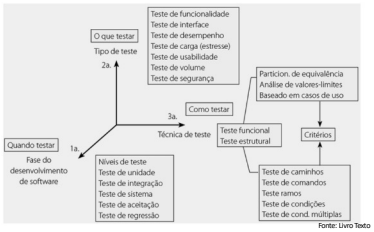
Defeito, Falha e Erro

- Defeito: trata-se de deficiência algorítmica que, se ativada, pode levar a uma falha.
- Falha: é tida como um não funcionamento do programa, provavelmente provocada por um defeito
- Erro: ocorre quando o resultado obtido em um processamento e o que se esperava dele não são coincidentes.

13

Testes- Fase do desenvolvimento de software

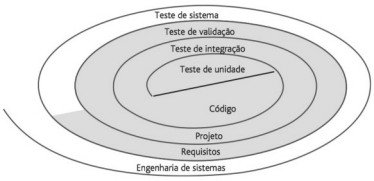
14



Fonte: Livro Texto

15

Tipos de Teste



Fonte: Pressman(2016)

16

Teste de Unidade

- Direcionado a uma rotina, classe ou pequena parte de um produto;
- Normalmente executada pelo próprio desenvolvedor
- Stub: é um trecho de código que substituirá as entradas, dependências e comunicações que a unidade deveria receber em uma execução do programa;

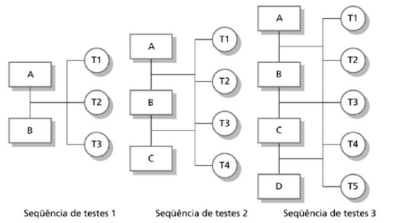
17

Teste de Integração

- Integração *top-down:* desenvolve-se primeiro o esqueleto do sistema e depois preenche com os componentes;
- Integração *bottom-up:* integrar os componentes de ifra-estrutura, em seguida adicionar os componentes funcionais.

18

Teste de Integração



Fonte: Sommerville (2007)

19

Teste de Validação

- Começa quando termina o teste de integração, quando os componentes individuais já foram exercitados;
- O teste focaliza ações visíveis ao usuário e saídas do sistema reconhecíveis pelo usuário;
- O software deve funcionar de uma maneira que pode ser razoavelmente esperada pelo cliente.

20

Teste de Sistema

- O software é apenas um elemento de um grande sistema de computador.
- Testes de integração de sistema e validação;

21

Técnicas de Testes

22

Teste funcional

- Teste caixa preta;
- O código fonte é ignorado;
- A preocupação é como funciona o software;
- Baseada nos requisitos básicos do software;

23

Teste funcional

Dois passos principais:

- Identificação das funções;
- Criação dos casos de testes;

24

Teste funcional

Utilizado para:

- Funções incorretas e omitidas;
- Erros de comportamento;
- Erros de desempenho;
- Erros na interface;
- Erros de iniciação e término.

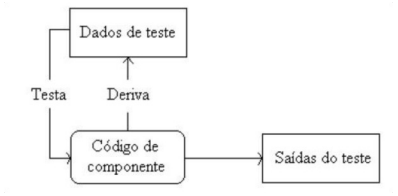
25

Teste estrutural

- Teste de caixa branca;
- Os testes são derivados do conhecimento da estrutura e da implementação do software;
- Conhecendo a estrutura de software, auxilia na identificação de partições e casos de testes adicionais;
- Grafo;

26

Teste estrutural

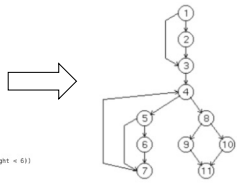


Fonte: Sommerville (2007)

27

Teste estrutural

```
/* 01 */ {
/* 01 */ char achar;
/* 01 */ int length, valid_id;
/* 01 */ lenght = 0;
/* 01 */ printf ("Indentificador");
/* 01 */ achar = fgetc(stdin);
/* 01 */ valid_id = valid_s(achar);
/* 01 */ if (valid_id)
/* 02 */     lenght = 1;
/* 03 */ achar = fgetc (stdin);
/* 04 */ while (achar != '\n')
/* 05 */ {
/* 05 */     if (!(valid_f(achar)))
/* 06 */         valid_id = 0;
/* 07 */     lenght++;
/* 07 */     achar = fgetc (stdin);
/* 07 */ }
/* 08 */ if (valid_id && (length >= 1) && (lenght < 6))
/* 09 */     printf ("Valido\n");
/* 10 */ else
/* 10 */     printf ("Invalido\n");
/* 11 */ }
```



Fonte: Livro texto

28

Desenvolvimento orientado a testes (TDD) e Testes específicos

29

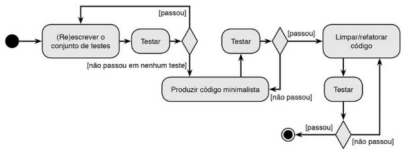
Desenvolvimento orientado a testes (TDD)

É uma técnica de programação que incorpora o teste ao processo de produção de código da seguinte forma:

1 – Escreva o Teste
2 – Veja o teste falhar
4 – Veja o teste passar
5 - Refatore o código
6 - Implemente o código

Fonte: Neto (2021)

30



Fonte: Wazlawick (2013)

31

Ferramentas para o TDD

- Normalmente é utilizado um framework que facilite o desenvolvimento e execução dos testes.
- Diversas plataformas de programação possuem frameworks de testes como o JUnit do Java, nUnit do Dot. Net e CPPUnit do C++.

32

Automação de Teste

- A fase de teste é trabalhosa, por isso, são necessárias ferramentas de automação;
- *Workbench* de teste: conjunto integrado de ferramentas para apoiar o processo de software.

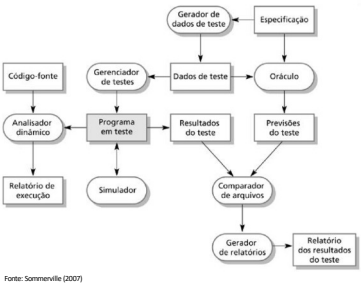
33

Automação de Teste

Ferramentas:

- Gerenciador de teste;
- Gerador de dados de teste;
- Oráculo;
- Comparador de arquivos;
- Gerador de relatórios ;
- Analisador de Ferramentas;
- Simulador

34



Fonte: Sommerville (2007)

35

Teste de Desempenho

Teste de desempenho é utilizado para:

- Avaliar a disponibilidade do aplicativo ou sistema;
- Analisar os critérios de desempenho;
- Comparar as características de desempenho de vários sistemas ou configurações de sistema;
- Procurar a fonte de problemas de desempenho;
- Encontrar os níveis de rendimento;

36

Teste de Estresse

Duas funções:

- Testar o comportamento de falha do sistema: testa se a sobrecarga acarreta uma "falha leve" em vez de um colapso no sistema.
- Estressar o sistema: analisar o surgimento de defeitos que não seriam normalmente descobertos.

37

Erros de um ambiente Web

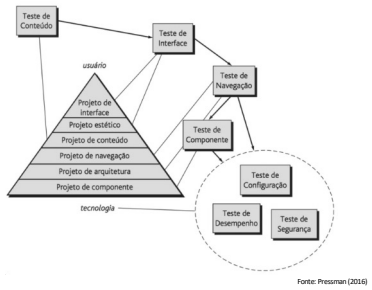
- A implementação em diferentes configurações e ambientes, pode dificultar a reprodução de um erro fora do ambiente no qual foi encontrado originalmente;
- Muitos erros podem ser atribuídos à configuração da WebApp;
- Devido às WebApps residirem em uma arquitetura cliente-servidor, os erros podem ser difíceis de localizar por meio das três camadas de arqui- tetura: o cliente, o servidor ou a própria rede

38

Processo de Teste

1) Testes que experimentam o conteúdo e a funcionalidade da interface;
2) Aspectos da arquitetura de projeto e da navegação da WebApp;
3) Examinam os recursos tecnológicos que nem sempre são aparentes para os usuários;

39



Fonte: Pressman (2016)

40

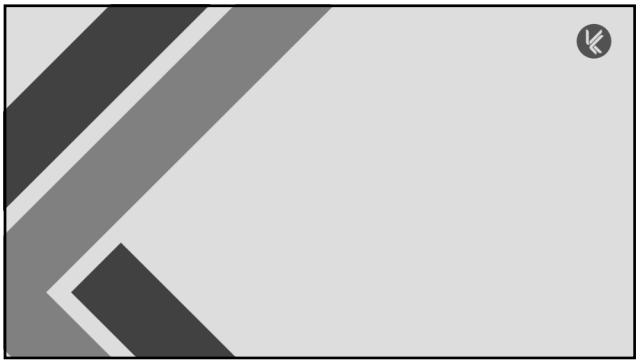
Recapitulando

41

Recapitulando

- Verificação e Validação;
- Testes de Softwares;
- Tipos de Testes;
- Desenvolvimento orientado a testes (TDD);

42



43