Simulação e Teste de Software (CC8550)

Aula 02 – Ciclo de Vida de Teste de Software

Prof. Luciano Rossi

Ciência da Computação
Centro Universitário FEI

1º Semestre de 2025

Correção da Atividade 01

1) De acordo com a norma ISO/IEC 9126, quais são as principais características de um software de qualidade?

Características Técnicas

- **Correto:** O software deve atender aos requisitos e especificações.
- **Confiável:** Deve operar consistentemente sem falhas inesperadas.
- **Eficiente:** Utiliza os recursos computacionais (tempo e memória) de forma otimizada.
- **Íntegro:** Garantia de segurança e proteção contra acessos ou alterações indevidas.
- **Fácil de Usar:** Deve ser intuitivo e acessível para o usuário final.
- **Interoperável:** Pode se integrar e funcionar com outros sistemas.

centro universitário FEI

Correção da Atividade 01

1) De acordo com a norma ISO/IEC 9126, quais são as principais características de um software de qualidade?

Características de Manutenção

- **Manutenível:** Fácil de corrigir erros e atualizar conforme necessário.
- **Flexível:** Adaptável para mudanças futuras ou novos requisitos.
- **Testável:** Sujeito a testes para validação e verificação de funcionalidade.
- **Portável:** Funciona em diferentes ambientes ou plataformas.
- **Reutilizável:** Componentes podem ser aproveitados em outros sistemas.

centro universitário FEI

Correção da Atividade 01

2) Explique a diferença entre verificação e validação de software e forneça um exemplo prático para cada uma.

- **Verificação:** Processo que verifica se o software está sendo construído corretamente ("Are we building the product right?"). Exemplo: Revisão de código para garantir que as **funções** implementadas seguem os requisitos especificados.
- **Validação:** Processo que verifica se o software atende às necessidades do usuário ("Are we building the right product?"). Exemplo: Testes de aceitação para garantir que um **sistema** de login permite acesso apenas para usuários autorizados.

centro universitário FEI

Correção da Atividade 01

a) Liste os problemas encontrados no código e classifique-os como erro, defeito ou falha.

- **Erro:** A função divisao não trata a exceção corretamente, pois apenas imprime um erro sem interromper a execução adequadamente.
- **Defeito:** A função busca utiliza um loop ineficiente para percorrer a lista, o que pode impactar no desempenho.
- **Falha:** Caso o usuário tente dividir por zero, o programa retornará um erro inesperado em tempo de execução.

Correção da Atividade 01

b) Corrija o código acima para torná-lo mais eficiente e confiável.

```
1  def divisao(a, b):
2          try:
3                  return a / b
4          except ZeroDivisionError:
5                  return "Erro: divisão por zero"
6
7  def busca(lista, valor):
8          return lista.index(valor) if valor in lista else -1
```

Correção da Atividade 01

c) Implemente três casos de teste automatizados para validar a correção do código.

```
def test_divisao():
        assert divisao(10, 2) == 5
        assert divisao(10, 0) == "Erro: divisão por zero"

def test_busca():
        assert busca([1, 2, 3, 4, 5], 3) == 2
        assert busca([1, 2, 3, 4, 5], 6) == -1

def test_limite():
        assert divisao(1000, 10) == 100
        assert busca([10, 20, 30, 40, 50], 40) == 3
```

centro universitário FEI

STLC – Software Testing Life Cycle (Ciclo de Vida de Teste de Software)

Fases do STLC

O que é STLC?

O **Ciclo de Vida de Teste de Software (STLC)** é um conjunto de etapas sistemáticas realizadas para garantir a qualidade do software, identificando e corrigindo defeitos antes da entrega final.

Fases do STLC

Etapas principais

- **Análise de Requisitos** – Compreensão e validação dos requisitos.
- **Planejamento de Teste** – Definição da estratégia e escopo dos testes.
- **Design de Teste** – Criação dos casos e preparação dos dados de teste.
- **Configuração do Ambiente** – Preparação da infraestrutura de teste.
- **Execução de Testes** – Realização dos testes e registro de defeitos.
- **Encerramento do Teste** – Relatório final e análise das métricas.

centro universitário FEI

Análise de Requisitos

Compreensão e validação

Objetivo

Garantir que os requisitos do sistema estejam claros, compreendidos e testáveis.

- Revisão dos documentos de requisitos.
- Identificação de ambiguidades e inconsistências.
- Definição do escopo do teste.

centro universitário FEI

Planejamento de Teste

Definição da estratégia e escopo

Objetivo

Criar um plano detalhado para a execução dos testes.

- Definir cronograma e recursos necessários.
- Selecionar ferramentas de teste.
- Identificar riscos e estratégias de mitigação.

centro universitário FEI

Objetivo

Desenvolver os cenários e casos de teste para validar o sistema.

- Especificação dos casos de teste.
- Preparação dos dados de teste.
- Revisão dos casos de teste para qualidade.

Configuração do Ambiente

Preparação da infraestrutura

Objetivo

Garantir que o ambiente de teste esteja pronto para a execução.

- Configuração de servidores e banco de dados.
- Instalação de ferramentas de teste.
- Validação do ambiente de teste.

centro universitário FEI

Execução de Testes

Execução e registro de defeitos

Objetivo

Testar o sistema e identificar falhas.

- Execução dos casos de teste.
- Registro de defeitos encontrados.
- Reexecução dos testes após correções.

centro universitário FEI

Encerramento do Teste

Análise final e relatório

Objetivo

Documentar os resultados e encerrar formalmente os testes.

- Análise das métricas de teste.
- Preparação do relatório final.
- Identificação de melhorias no processo.

centro universitário FEI

Simulação e Teste de Software (CC8550)

Aula 02 – Ciclo de Vida de Teste de Software

Prof. Luciano Rossi

Ciência da Computação
Centro Universitário FEI

1º Semestre de 2025