

Teste de Software

Atividade Avaliativa I

Formação: 2 ou individual

Data de entrega: 11/3/2024

Questões:

1. Qual é o principal objetivo dos testes unitários?

- a) Verificar a integração entre diferentes componentes do sistema.
- b) Identificar falhas de usabilidade na interface do usuário.
- c) Testar unidades isoladas de código para garantir seu funcionamento correto.
- d) Avaliar a performance do sistema em condições extremas.
- e) Validar a arquitetura de rede do sistema.

2. O que é TDD (Test-Driven Development)?

- a) Uma técnica para desenvolver software onde os testes são escritos após a implementação do código.
- b) Um método de desenvolvimento que prioriza a documentação sobre os testes.
- c) Uma abordagem que utiliza testes automatizados para direcionar o processo de desenvolvimento.
- d) Uma metodologia que exclui completamente a necessidade de testes durante o desenvolvimento.
- e) Uma prática que se concentra exclusivamente em testes de aceitação do cliente.

3. Qual é a principal vantagem de escrever testes antes de implementar o código (TDD)?

- a) Redução dos custos de desenvolvimento.
- b) Aumento da complexidade do código.
- c) Melhoria na qualidade do código.
- d) Atraso no lançamento do produto.
- e) Diminuição da produtividade da equipe.

4. O que é um "teste unitário" no contexto do desenvolvimento de software?

- a) Um teste que verifica a funcionalidade de um sistema inteiro.

- b) Um teste que verifica a interação entre diferentes componentes do sistema.
- c) Um teste que verifica o comportamento de um módulo ou função isoladamente.
- d) Um teste que verifica a performance do sistema em condições reais.
- e) Um teste que valida a experiência do usuário.

5. Quais são as características de um bom teste unitário?

- a) Deve ser dependente de outros testes.
- b) Deve ser lento para executar.
- c) Deve ser repetível e automatizado.
- d) Deve verificar apenas uma pequena parte do código.
- e) Deve ser escrito após a implementação do código.

6. Qual é o principal propósito dos testes unitários dentro de um projeto de desenvolvimento de software?

- a) Garantir que todas as funcionalidades estejam documentadas.
- b) Identificar falhas no design de interfaces de usuário.
- c) Assegurar que cada parte do código funcione conforme o esperado.
- d) Validar a integração entre diferentes sistemas externos.
- e) Testar a segurança do sistema contra ataques externos.

7. Quais são os possíveis benefícios de adotar TDD (Test-Driven Development) em um projeto de software?

- a) Redução do tempo de desenvolvimento.
- b) Aumento da complexidade do código.
- c) Melhoria na qualidade do código.
- d) Aumento dos custos de manutenção.
- e) Diminuição da participação da equipe de desenvolvimento.

8. Qual é a sequência de passos típica no ciclo do TDD (Test-Driven Development)?

- a) Implementar o código, escrever testes, fazer refatorações.
- b) Escrever testes, implementar o código, fazer refatorações.
- c) Fazer refatorações, escrever testes, implementar o código.

- d) Implementar o código, fazer refatorações, escrever testes.
- e) Fazer refatorações, implementar o código, escrever testes.

9. Qual é o principal objetivo de refatorar o código durante o processo de TDD?

- a) Introduzir bugs intencionais no código.
- b) Reduzir a legibilidade do código.
- c) Melhorar a estrutura e a organização do código.
- d) Adicionar funcionalidades não solicitadas.
- e) Atrasar o lançamento do produto.

10. Qual é a diferença entre testes unitários e testes de integração?

- a) Testes unitários verificam unidades isoladas de código, enquanto testes de integração verificam a interação entre diferentes partes do sistema.
- b) Testes unitários são escritos após a implementação do código, enquanto testes de integração são escritos antes.
- c) Testes unitários são mais lentos para executar do que testes de integração.
- d) Testes unitários são mais propensos a encontrar erros de lógica do que testes de integração.
- e) Testes de integração são mais específicos do que testes unitários.

Exercício de Implementação:

Caso 1) Imagine um sistema usado em uma clínica de emagrecimento. A função que calcula o IMC não está retornando valores satisfatórios. Implemente uma classe em Java para calcular o IMC e também para realizar o Teste Unitário.

Caso 2) Em um sistema de frente de caixa, os itens vendidos devem passar por uma rigorosa verificação, para que as vendas não gerem prejuízos. Dentre as verificações, não se pode praticar descontos maiores que 3%, nem vender produtos com preços unitários iguais a zero. Crie uma implementação em Java para representar a classe dos itens vendidos assim como as classes para aplicação de teste unitário.