|  |  |  |  |
| --- | --- | --- | --- |
| RM: | 96562 | Nome: | LEONARDO PAGANINI |

**INSTRUÇÕES:**

* Esta avaliação vale **10 pontos** e baseia-se no conteúdo sobre **Introdução à Qualidade de Software** e **Introdução aos Testes de Software**.
* Cada questão vale 1,0 ponto.
* A avaliação pode ser realizada com consulta ao material.
* Questões de múltipla escolha sem resposta ou com mais de uma resposta assinalada serão desconsideradas.

**Entrega:**

* Você pode entregar o documento Word com as respostas no Teams (Tarefas).
* Entregas feitas após o prazo (**01/09/2023 às 23:59**) terão desconto de **2 pontos** da nota atingida.

**ATENÇÃO E SUCESSO!**

1. De acordo com o Modelo V do SDLC (*Software Development Lifecycle*), quais são os níveis de teste?

a) Testes Unitários, de Integração, Funcionais e Não-Funcionais.

b) Testes Unitários, de Integração, de Sistema e de Aceitação.

c) Testes de Desempenho, de Segurança, de Usabilidade e de Acessibilidade.

d) Testes Funcionais, Não-Funcionais, de Regressão e de Aceitação.

2. Das opções abaixo, qual apresenta apenas tipos de testes não-funcionais?

a) Testes de Sistema, de Caixa-Preta, de Segurança e de Performance.

b) Testes de Desempenho, de Segurança, Exploratórios e de Acessibilidade.

c) Testes de Desempenho, de Segurança, de Usabilidade e de Escalabilidade.

d) Testes de Integração, de Sistema, de Aceitação e de Regressão.

3. O que é teste de confirmação?

a) Uma técnica de teste que se concentra na verificação interna do código-fonte.

b) Uma técnica de teste que verifica se os defeitos reportados foram corrigidos.

c) Uma técnica de teste que verifica se as alterações no software não afetaram funcionalidades existentes.

d) Uma técnica que repete apenas testes que falharam anteriormente.

4. O custo da correção de um defeito:

a) Não é importante.

b) Aumenta quanto mais tarde ele for detectado.

c) É maior para requisitos funcionais do que não-funcionais.

d) Não há como determinar.

5. É verdade sobre o SQA (*Software Quality Assurance*):

a) É uma parte do SQC (*Software Quality Control*).

b) Engloba apenas análise de requisitos e inspeção de código.

c) É sinônimo de teste de software.

d) Permeia todo o SDLC (*Software Development Lifecycle*).

6. O que diz o princípio 5 do teste de software (Paradoxo do Pesticida)?

a) Casos de teste se tornam obsoletos e precisam ser atualizados.

b) Assim como o pesticida, testes de software são a melhor arma contra *bugs* (insetos).

c) Um pequeno número de módulos geralmente contém a maioria dos defeitos.

d) O teste deve feito de forma diferente em diferentes contextos.

7. De acordo com o que foi apresentado em aula, testar é:

a) o processo de demonstrar que não há mais erros presentes no software.

b) mostrar que o software atende todos os requisitos explícitos.

c) o processo de executar um software com a intenção de encontrar erros.

d) executar todas as linhas de código ao menos uma vez.

8. É verdade sobre testes unitários:

a) Não podem ser automatizados.

b) São executados uma única vez.

c) Garantem a integração entre módulos.

d) Geralmente são escritos e executados pelos desenvolvedores.

9. Podem ser considerados subgrupos dos Testes de Regressão:

a) Smoke Tests e Sanity Tests.

b) Smoke Tests e Testes de Confirmação.

c) Sanity Tests e Testes de Confirmação.

d) Testes Exploratórios e Baseados em Estrutura.

10. Abordagem de teste que visa garantir que o sistema não faz o que não deveria fazer:

a) Suposição de Erro.

b) Cobertura de Decisão.

c) Teste Negativo.

d) Teste Positivo.