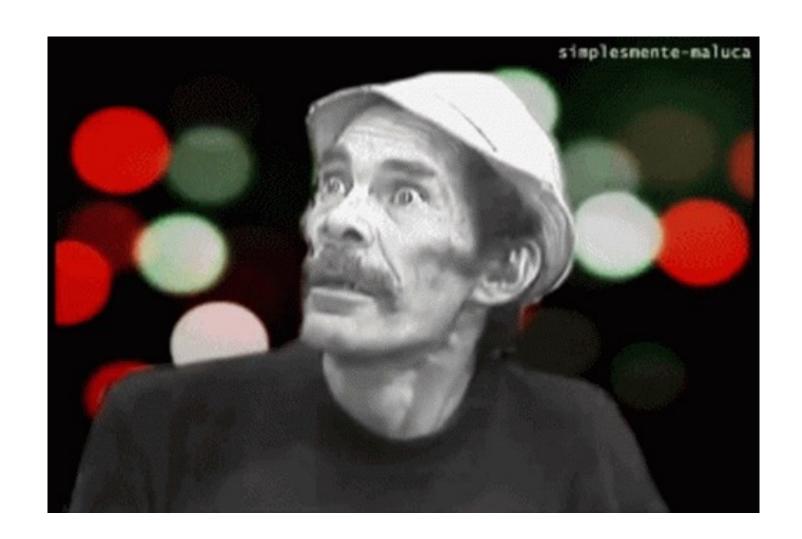
TESTE E QUALIDADE DE SOFTWARE

João Choma Neto

joao.choma@gmail.com

Unicesumar – Maringá – 2023/2

ANTERIORMENTE



ARTEFATOS

Os artefatos de teste são documentos e entregas produzidos durante o processo de teste de software

Esses artefatos têm o propósito de fornecer informações sobre o planejamento, execução e resultados dos testes

PLANO DE TESTE

É um documento que descreve a abordagem geral de teste:

- 1. Objetivos do teste
- 2. Escopo
- 3. Cronograma
- 4. Recursos necessários e
- 5. Critérios de aceitação

CASOS DE TESTE

São documentos que descrevem os cenários específicos que serão testados

- 1. Entrada de dados
- 2. Ações a serem realizadas
- 3. Resultados esperados

ROTEIRO DE TESTE

São documentos que detalham a sequência de etapas que os testadores devem seguir ao executar os casos de teste

RELATÓRIO DE EXECUÇÃO DE TESTES

São documentos que resumem os resultados dos testes executados

- 1. Mostram quais casos de teste foram bemsucedidos
- 2. Mostram quais casos de teste falharam
- 3. Medem o progresso do teste
- 4. Identificam quais áreas que exigem atenção do testador

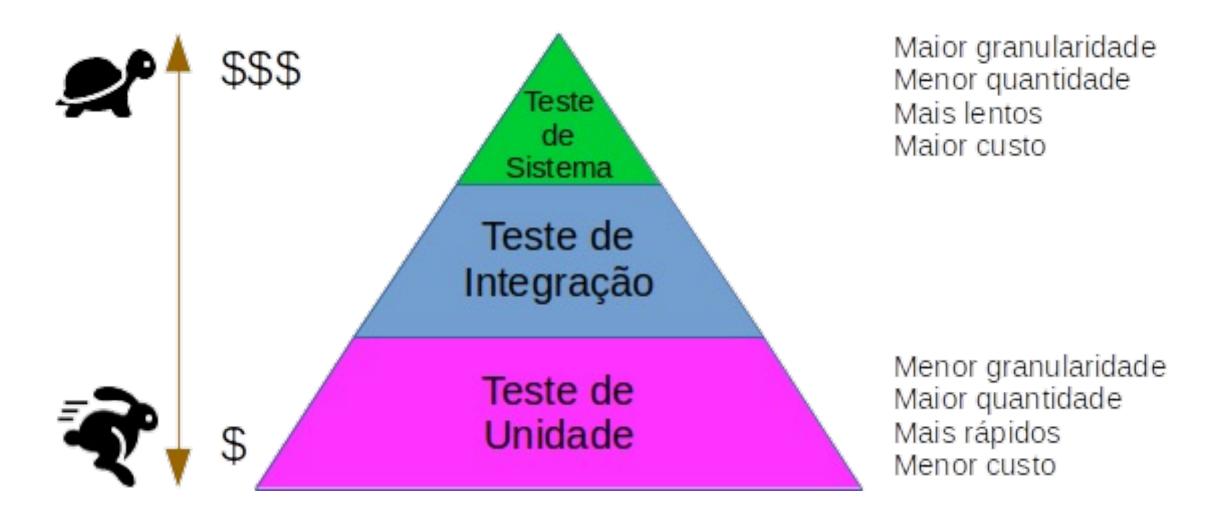
MATRIZ DE RASTREABILIDADE

É uma tabela que mostra a relação entre os requisitos do software e os casos de teste A matriz tem o objetivo de garantir que todos os requisitos foram cobertos por testes

TESTE DE UNIDADE

https://github.com/JoaoChoma/test eequalidadedesoftware/tree/main/ Semana04





 Teste unitário é uma prática de teste de software na qual cada componente individual ou "unidade"

- Uma unidade pode ser a menor parte testável de um aplicativo
- Função
- Método
- Classe
- Módulo

•O objetivo principal dos testes unitários é identificar defeitos nas unidades de código antes que elas sejam integradas ao restante do sistema

- Isolar as unidades de código e testá-las independentemente
- Os desenvolvedores podem verificar se cada unidade executa as tarefas designadas de acordo com suas especificações

- Identificar problemas cedo no processo de desenvolvimento
- Tornar mais fácil e mais econômico corrigir erros

 Cada teste unitário deve ser automatizado para que possa ser executado repetidamente e integrado a fluxos de trabalho de desenvolvimento contínuo

 Processo estruturado para garantir que cada componente individual do código seja testado de maneira isolada

UNIDADE

- · Identifique as Unidades de Código
- Determine as unidades de código que você deseja testar

FRAMEWORK

- Escolha um Framework de Testes
- Utilize um framework de testes apropriado para a linguagem de programação que você está usando

FRAMEWORK

- JUnit (Java)
- NUnit (.NET)
- pytest (Python)
- Jasmine (JavaScript)

CASOS DE TESTE

Crie Casos de Teste

- Para cada unidade de código, escreva casos de teste que cubram diferentes cenários
- Incluir casos em que a unidade deve produzir resultados corretos
- Incluir casos em que deve lidar com entradas inválidas ou inesperadas

TESTES

- Escreva os Testes
- Escreva o código dos testes usando o framework escolhido

TESTES

Dados de Teste

- Inserir dados de teste relevantes para os casos de teste
- Criação de objetos fictícios
- Valores de entrada simulados
- Cenários de teste específicos

EXECUTAR

- Execute os Testes
- Verifique os Resultados
- O framework de testes geralmente fornecerá relatórios indicando quais testes passaram e quais falharam

REPETIR

- Repetir o processo de teste para todas as unidades de código
- Cada nova modificação recursos ao código repetir os testes
- Cada adição novos recursos, criar novos testes de unidade e repetir os testes

REFERÊNCIAS

Ian Sommerville − Engenharia de Software. 10ª Edição. São Paulo: Pearson Education do Brasil, 2019.

Roger S. Pressman – Engenharia de software: uma abordagem profissional. 7º Edição. Porto Alegre: AMGH Editora Ltda, 2011.

Shari Lawrence Pfleeger – Engenharia de Software: teoria e prática. 2ª Edição. São Paulo: Pearson Education do Brasil, 2004.

TESTE E QUALIDADE DE SOFTWARE

João Choma Neto

joao.choma@gmail.com

Unicesumar – Maringá – 2023/2