

Descrição da Atividade:

Cada aluno (ou grupo, conforme alinhado com o professor) deverá realizar uma pesquisa sobre **Testes de Software**, explorando os seguintes tópicos:

1. O que é teste de software?

Definição.

Teste de software é a execução do sistema com intenção de encontrar defeito, para garantir que o software atenda os requisitos esperados, para garantir qualidade e segurança no produto final.

Importância no Ciclo de Desenvolvimento

A importância dos testes no ciclo de desenvolvimento é imensa. Eles não são apenas uma "última etapa" antes do lançamento, mas sim uma atividade contínua que deve ser integrada em todas as fases do desenvolvimento, desde a concepção até a manutenção. Os testes ajudam:

Garantir a qualidade: Asseguram que o software funcione conforme o esperado e atenda às necessidades do usuário.

Reduzir custos: Descobrir e corrigir defeitos no início do ciclo de desenvolvimento é muito mais barato do que fazê-lo após o lançamento.

Aumentar a confiança: Um software bem testado gera mais confiança nos usuários e na equipe de desenvolvimento.

Melhorar a experiência do usuário: Um software sem falhas e com bom desempenho oferece uma experiência superior ao usuário.

Validar requisitos: Confirmam que o software foi construído de acordo com as especificações e requisitos definidos.

2. **Objetivos dos testes de software:**

Por que testar?

Identificar e corrigir defeitos, Garantir a qualidade do produto, Reduzir riscos, Melhorar a experiência do usuário, Validar funcionalidades, Evitar custos elevados de manutenção

Quais problemas os testes ajudam a prevenir ou resolver?

bugs, falhas de segurança, problemas de desempenho e gargalos de desenvolvimento garantindo que o software funcione conforme o esperado

3. **Características dos testes:**

Quais são elas?

Algumas características do teste de software são:

Verifica se o sistema foi desenvolvido corretamente (verificação) e se atende às necessidades do usuário (validação).

É baseado em regras bem definidas, como os requisitos do sistema.

Pode ser executado várias vezes e deve produzir sempre o mesmo resultado.

Preferencialmente, os testes devem ser realizados por alguém que não participou do desenvolvimento do sistema.

Todo o processo deve ser documentado, incluindo os casos de teste e os erros encontrados.

Deve-se testar o maior número possível de partes e funcionalidades do sistema.

Os testes devem ser atualizados sempre que o sistema sofrer alterações.

Rastreabilidade cada teste deve ser vinculado a requisitos e especificações

Repetibilidade nenhum teste tem número limite de vezes

Objetividade é testar diretamente o alvo e obter resultados desse teste específicos

4. **Tipos de teste de software:**

Teste de Unidade.

São um tipo de teste que verifica o funcionamento correto de unidades individuais de código, como funções, métodos ou classes

Teste de Integração.

Testes de integração de software são um tipo de teste que verifica se diferentes módulos ou componentes de um sistema funcionam corretamente quando são combinados.

Teste de Sistema.

É o teste que verifica se todo o sistema funciona corretamente, como se fosse um usuário usando o sistema do começo ao fim.

Ele testa tanto as funções principais (como botões, formulários, etc.) quanto pontos não funcionais (como velocidade e segurança).

É feito num ambiente parecido com o real e ajuda a decidir se o sistema está pronto para ser usado.

Teste de Condição.

técnica que se concentra em examinar o código-fonte de um software para verificar se as expressões condicionais estão funcionando corretamente, garantindo que todas as condições possíveis sejam testadas. Isso inclui garantir que as condições "verdadeiras" e "falsas" sejam testadas para cada declaração condicional, como if e while.

Fluxo de Dados

Em computação e análise de dados, fluxo de dados refere-se ao movimento de informações dentro de um sistema ou processo. Ele descreve como os dados entram, são processados e saem, podendo ser representado visualmente através de diagramas de fluxo de dados (DFDs). Esses diagramas ajudam a entender a sequência de informações, atores e etapas envolvidas em um processo, seja ele de desenvolvimento de software ou de um processo de negócio.

Teste de caminho

Teste de caminho, também conhecido como teste estrutural ou teste de caminho básico, é uma técnica de teste de software que visa garantir a cobertura de todos os caminhos possíveis em um programa. Essa técnica, frequentemente utilizada em testes de caixa branca, analisa o fluxo de controle do código para identificar caminhos linearmente independentes que precisam ser testados.

Teste de loop

Teste de loop, no contexto de desenvolvimento de software, refere-se à execução repetida de um trecho de código ou um conjunto de testes para verificar seu comportamento e garantir que funcione corretamente. Este processo é fundamental para identificar erros e assegurar a qualidade do software.

Teste de Aceitação.

verifica se o sistema atende aos requisitos e expectativas do cliente ou usuário final. Em outras palavras, eles respondem a pergunta “o sistema faz o que o usuário espera?”

Testes Funcionais e Não Funcionais.

Testes funcionais: O teste funcional serve para conferir se o sistema faz tudo o que deveria fazer. Ele verifica cada função do sistema para ter certeza que está funcionando corretamente, sem

erros. Para isso, o teste usa como base uma lista com tudo que o sistema precisa fazer, chamada de documento de requisitos. Alguns exemplos de teste funcional são: testar partes pequenas do código (teste de unidade), testar como o sistema mostra as informações para o usuário (teste de interface) e testar se funções antigas continuam funcionando após mudanças (teste de regressão).

Testes não funcionais: Os testes não funcionais verificam como o sistema se comporta, e não o que ele faz. Eles avaliam aspectos como:

Desempenho: se o sistema é rápido e responde bem.

Segurança: se os dados estão protegidos contra invasões.

Usabilidade: se o sistema é fácil de usar.

Estabilidade: se o sistema não trava ou falha.

Capacidade: se suportar muitos usuários ao mesmo tempo.

Exploratório, Caixa Branca, Caixa Preta, etc:

EXPLORATÓRIO:

O que é: Teste feito sem um roteiro fixo. O testador usa a aplicação como um usuário real, explorando e buscando falhas de forma livre.

Quem faz: Geralmente QA experiente ou desenvolvedor.

Vantagem: Pode encontrar erros que testes automatizados ou roteirizados não encontram.

Exemplo: Abrir várias abas, digitar dados errados, tentar ações fora do esperado.

CAIXA BRANCA:

técnica onde o testador tem acesso ao código-fonte e estrutura interna do sistema, permitindo analisar e testar o software com base nesse conhecimento

CAIXA PRETA:

Técnica onde o testador não tem conhecimento da estrutura interna do código ou implementação. o foco é apenas na funcionalidade e comportamento externo do sistema, tratando-o como uma “caixa preta” onde se observa a entrada e a saída verificando se ele se comporta conforme o esperado, sem se preocupar com o que acontece dentro.

CAIXA CINZA:

esse teste combina elementos da caixa branca e caixa preta. isso significa que os testadores tem conhecimento da estrutura interna do sistema (mas não conhecimento completo, como no caso da caixa branca), e usam esse conhecimento para guiar seus testes, combinando com a perspectivas do usuário final (como o caso da caixa preta).

StakeHolders: indivíduo ou grupo de pessoas que afetam o software no projeto completo.