EQUIPE: FOXTROT

Rebeca; Sara; Samuel; Tiago.

TESTE DE SOFTWARE

**Objetivo do teste**

Um dos primeiros objetivos do teste é demonstrar tanto aos desenvolvedores como aos clientes que esse software atenderá os requisitos. No caso de softwares customizados deve existir um teste para cada requisito. Para softwares genéricos devem ser desenvolvidos teste para todas as características do sistema e das combinações que serão incorporadas.

Esse objetivo primário requisita testes de validação, ou seja, testes em que se é esperado que o sistema execute corretamente as tarefas usando determinado conjunto de testes classificados em casos que possibilitam demonstrar o uso esperado do sistema.

Um segundo objetivo dos testes de software é descobrir e identificar as situações em que o software se comporta de maneira incorreta, ou seja, uma forma que não atende satisfatoriamente os requisitos e especificações. Essas situações são consecutivas de defeitos no software.

Os casos de teste de testes de defeito, são projetados com o intuito de expor os comportamentos inadequados e defeitos do software, além se voltar para eliminação desses, comportamentos como panes, interações indesejáveis, processamentos incorretos e corrução de dados são apontados e corrigidos. Esses testes podem ser deliberadamente obscuros, não necessitam de que seja refletida a maneira precisa como o sistema é usado.

Nos testes de validação são encontrados defeitos no sistema e alguns testes de defeito podem interpretar que o programa corresponde aos requisitos.

A prioridade no teste de defeitos é encontrar entradas que causam comportamentos anômalos estimulando o sistema a gerar corretamente saídas.

**TDD**

**Teste driven development:**

TDD é uma pratica muito comum entre programadores, ela consiste em escrever o teste antes mesmo de ter começado a produzir o código.

Com isso o programador garante que uma parte significativa do seu sistema esteja com o funcionamento adequado, pois todas as funcionalidades produzidas terão seu próprio teste desenvolvido.

Além do mais, com os testes feitos antes da produção do código, é possível que os desenvolvedores possam ser guiados pelos seus próprios testes.

**As praticas fundamentais para a produção de testes são:**

-Escrever testes que tenham uma rápida execução, e que testem pequenos blocos de códigos lembre-se a ideia não é produzir testes enormes, e sim testes pequenos rápidos que consigam lhe auxiliar a desenvolver sua aplicação.

-Escrever os testes antes de começar a produzir o code core.

**O ciclo de vida TDD:**

O ciclo de vida TDD consiste em criar o teste, executar todos os possíveis testes para que a aplicação venha a falhar, desenvolver a aplicação que será testada e assim executar os testes para ver se a aplicação será aprovada, refatorar a funcionalidade criada e testar novamente, só assim a aplicação será finalmente aprovada.

No TDD temos os Tipos de Teste, são eles os Teste Unitários, Testes de Integração e Teste de Aceitação, são alguns tipos deles, existem vários outros.

Teste unitário: Teste apenas de componentes do sistema, pode-se usar e ferramentas como Junit, Jmock.

Teste de integração: Testa a integração entre componentes que envolvem dois ou mais, como classes junto do SGBD, pode ser usado de ferramentas tais como, Junit, DBUnit.

Teste de aceitação: Testa uma funcionalidade, ou caso de uso e engloba vários componentes do software, pode-se usar de ferramentas, como Junit e Selenium.

Sendo uma ferramenta baseada em teste ela traz benefícios, tais como os testes serem feitos na IDE, não ter a necessidade de fazer uma “instalação” do software para executar o teste, bugs são achados com maior simplicidade e corrigidos com maior velocidade, bugs certificados por testes unitários, o código é possibilita um melhor design, simplifica a técnica de refactoring. Tais como benefícios tem vantagens, sendo elas a de feedback mais rápido sobre novas funções e outras já existentes, código mais limpo, já que códigos devem ser mais simples para o teste, segurança na refatoração pois é possível ver o que está ou não afetando, segurança na correção de bugs, produtividade aumentado, vendo que o programados encontra menos bugs, não desperdiçando tempo em depuração, o código tende a ser mais flexível, já que para escrever testes deve-se separar em pequenos fragmentos o nosso código, para que sejam testáveis, sendo assim, o código estará menos acoplado.

**QUESTIONÁRIO**

**1) Dois benefícios do teste de software:**

Os testes de software podem ajudar a prevenir comportamentos indesejados, funcionalidades que não estão funcionando como foi previsto, elimina qualquer execução indesejada por parte do software, pode ajudar a expor defeitos para que possam ser corrigidos o quanto antes, assim, economizando tempo e trabalho do programador.

**2) Três vantagens do TDD:**

Escrever os testes antes do código, garante que quase todo, ou todo o sistema terá um teste que poderá avaliar o sistema para que ele funcione corretamente, os testes podem guiar o desenvolvedor durante a montagem de seu projeto, garante que todo o sistema esteja correto.

**3) Explique o teste de regressão:**

É quando o programador testa novamente as partes que não foram alteradas em todo o sistema, para ter a certeza de que tudo esta funcionando corretamente.

**4) Explique o teste unitário:**

O objetivo principal dos testes unitários é mostrar que cada unidade atende de forma correta o que foi especificado para ela.

**5) Quais os tipos de teste TDD?**

Testes unitários, testes de integração, testes de aceitação e testes de regressão.

**6) Quais os benefícios do teste unitário? Explique!**

Com os testes unitários é possível encontrar qualquer problema o quanto antes, facilitam a mudança da unidade, simplificam a integração e ainda podem melhorar a documentação do sistema.