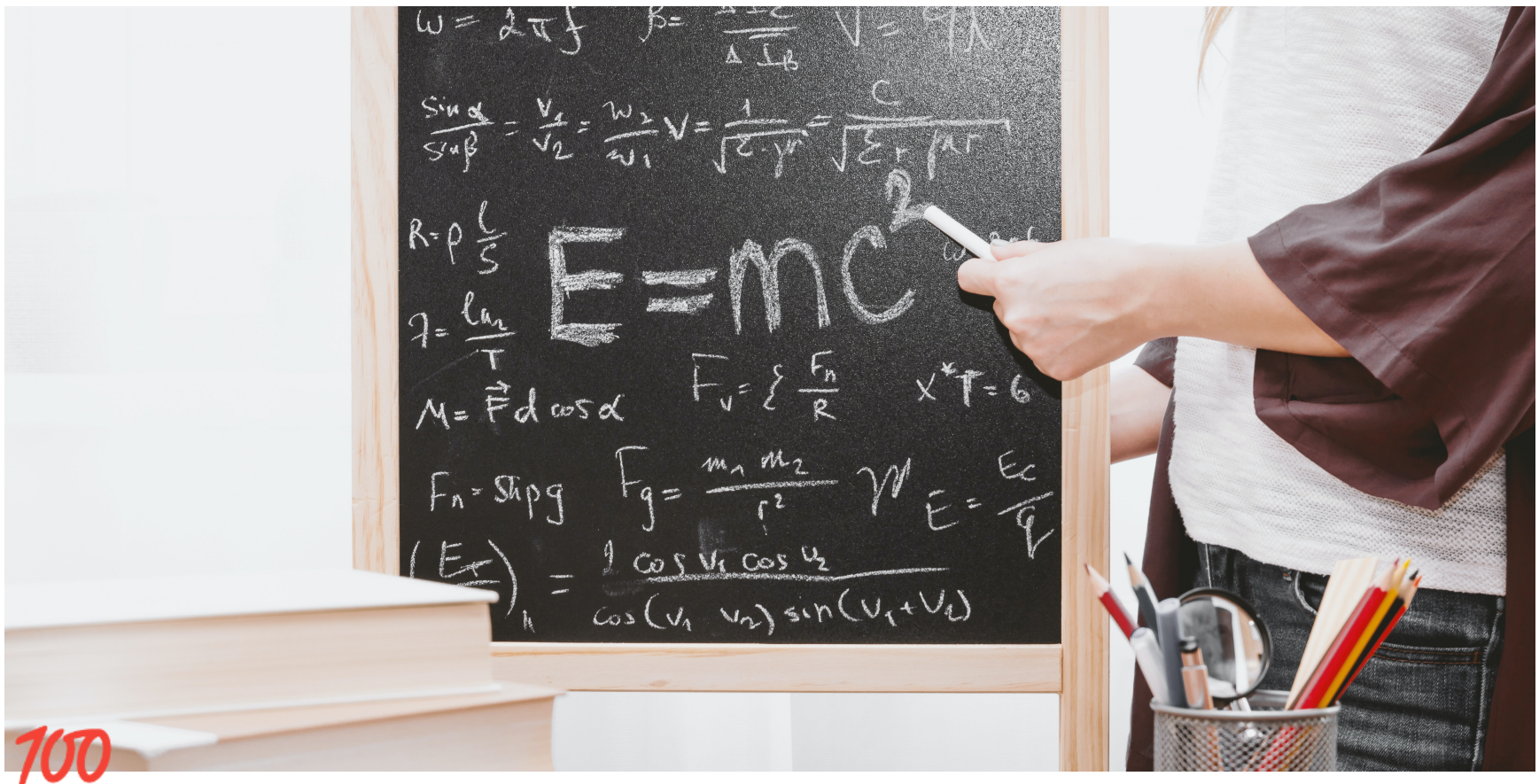

TESTE DE SOFTWARE

Introdução:

Teste de software é avaliar se algo se comporta como deveria, se algo se comporta como deveria. Se quando estamos escrevendo uma e-mail e apertamos o botão enviar esperamos que o comportamento seja que o meu e-mail seja enviado ao destinatário, esse e o nosso comportamento esperado.

Mas vale ressaltar que teste não testa só quando todas as condições são favoráveis, mas também quando não são e mesmo assim e algo esperado, no exemplo anterior se são tentarmos enviar o e-mail, mas não percebemos que estamos sem internet, e esperado que aprece alguma mensagem dizendo que não há conexão para enviar o e-mail, e mesmo esse erro e 'um comportamento esperado. Se colocarmos num software que quando o usuário realizar determinada ação de forma erronia retorne um erro especifico, e necessário testar também se nessas condições esse erro realmente e retornado ao usuário.

Além do mais teste não e só rodar as coisas de certo e esperada, e também gerar erros refatorar e se preparar para o que puder dar de errado

verdades importantes

- **Um teste não validam as completamente as saídas de um software**

  Pois as entradas são infinitas. Pois em muitos software e impossível testar todos os tipos de saídas,; se tentar testar um tabuleiro de xadrez e entendermos que as nossas saídas são cada movimento de cada, a derrubada de peças, pontuação, jogadas se houver inteligência artificial, só para testar a movimentação de cada peça do xadrez no teríamos 16 peças de uma cor vezes 2 por serem duas cores (pretas e brancas) x 64 casas, daria 2048 possibilidades, isso considerando que vamos testar cada peça por vez, pois poderíamos testar diferentes arranjos como saída, que daria muito mais possibilidades.

- **Teste não garante qualidade**

  Qualidade e um contexto subjetivo para cada produto. a muitos produtos na indústria da moda que suas funcionalidades são questionáveis com sapatos, calças e etc. é tem qualidade pois tem matéria-prima de qualidade a um processo de qualidade para garantir beleza estética e etc., entretanto quando testado se confortável. passando

ou não indique que um produto tem ou não qualidade Poias qualidade é um conceito muito mais amplo e depende das percepções dos clientes, do criador e etc.

▼ **testes tem uma custo em media de 20% a 30% do total da construção de um software**

E nesse custo nos temos a elaboração, a realização do teste, quem testa e desenvolve com que corrige os erros

▼ **Tester são Siths ou Daleks:**

O perfil de um testador e uma de uma pessoa pessimista sobre um software, que procura e tem como objetivo encontrar no software, detalhes destrutivos os programadores geralmente tem um visão positivista, otimista,, O tester tem como principal tarefa ir em rumo ao contrario é ser o cara chato que aponta os erros. que desconfia de tudo, detalhista, calculista e que observa todos os detalhes, paciente, focado, motivado, perspicaz, criterioso,. Precisa ter uma visão critica sobre o que o software deve fazer, o que o cliente.

Casos de sucesso só que não:

- **O primeiro Jogo da Disney multimidia**, desenvolvido para crianças, e no 26 de dezembro daquele ano o departamento de atendimento recebeu varias ligações relatando o ocorrido do problema devido a não rodar em todos as PCs existentes no mercado (jogo rei leão)
- **Ariane 5:** foguete perdeu controle de altitude e ativou a autodestruição, um problema de converter um numero real para um inteiro de 16 bits, por o numero estar fora da faixa recebeu um erro out of range que desnorteou o sistema

  O foguete demorou 10 anos, custo 8 bilhões e carregava uma carga de 500 milhões.
- **Na Rússia, 1983 um Bug na união soviética**, fez com que um software mostrasse que os estados unidos tinha enviado 5 misseis, por sorte o coronel Stanislav Petrov , considerou que a informação estava errada e não ordenou o contra ataque

Fundamentos:

Quando Testar?

Quando falamos em testar, nos falamos de níveis de teste diz muito a respeito da maturidade pois os tipos de teste devem ser implementados com forme o sistema ou software for ganhando maturidade

Teste unitário

Isolar é testar uma pequena parte do sistema independente do todo, com o objetivo de encontrar falhas de funcionamento

- feitos por programadores
- Sub-rotinas , métodos, classes entre outras pequenas umidades do software são o alvo.
- E preciso esta sempre atualizado, sendo necessário revisitar caso mudança nas regras de negocio ou atualização de versão

Teste de Integração:

tem como objetivo validar a comunicação entre os componentes de um sistema

- Feitos pelo programador
- Os alvos são funcionalidades que envolvem a integração de componentes,
- devem ser feitos como os teste unitários antes do sistema esta concluído

- acontece entre classes, funções, sistemas e serviços que se comunicam entre si

exemplo Adriane 5: componente A, entretanto um método de um componente B e espera um valor inteiro, porém, vem um valor decimal, causando falha no componente A.

Teste de Sistema

Executa o sistema sob a perspectiva do cliente\usuário final do software, buscando falhas em relação ao objetivo original. enquanto percorrer o sistema

- Planejados e executados pela equipe de teste
- Cenários de teste coerentes com os requisitos especificados para o sistema
- são realizados após codificação do software

Testes Aceitação

Executar o sistema sob o ponto de vista do usuário, usando das mesmas tecnologias que o usuário

- Planejados e executados por um grupo de usuários finais, geralmente uma ou duas pessoas, que simularão operações rotinas, de modo a verificar se seu comportamento esta de acordo com o solicitado
- E tem como objetivo o cliente dizer se ele aceita ou não aquele sistemas

Teste Alfa

Tem como objetivo executar o software de forma não planejada, sob um ponto de viste do usuários finais

- Realizado por um grupo pequeno de usuários, que tem conhecimento do desenvolvimento do software ou sistema, que não participaram do teste de aceitação, grupo de (50 a 100 pessoas)
- Que testaram o software ou sistema de forma não planejada, utilizando o sistema de forma mais Natural, para identificação de erros não detectados até o momento ,
- O time de teste acompanha de perto esses testes para recolher as falhas a serem corrigida e possíveis melhorias

Teste Beta

Entre o teste Alfa e o Beta tem algumas particulares, entretanto tem o objetivo de testar o sistema de forma natural e não planejada da perspectiva do usuário, porem:

- Realizado por um grupo grande de pessoas, geralmente desconhecidas é que atendem a determinados requisitos e critérios definidos pelo fornecedor do sistema. Exemplo: um App de investimento e natural que o grupo de pessoas que seja selecionada para realizar seja pessoas que já investiram pelo menos um vez na vida
- Não a acompanhamento da equipe de teste, geralmente o que se encontra é no sistema um local para feedback do usuário

Teste de regressão

Não tem uma ordem especifica para execução e pode ser realizado a qualquer momento, e tem como objetivo realizar teste já feitos, após uma melhoria ou correção de erro, com a finalidade de identificar se a solução proposta não causa um erro em outra parte do sistemas.

E uma coisa natural que alterando ou proposto um nova solução para determinado problemas a mesma possa causar um erro em uma outra parte do sistema de forma inesperada os testes de regressão nos ajudam a garantir que todo sistema ou software esta funcionando de forma esperada

Entretanto os testes de regressão podem ser custoso, caso não automatizados, pois quando manuais tem dependência de que alguém os rode.

Como testar?

Caixa Branca

São texto que olham direto para o código no momento que estamos testando, vamos observar estruturas condicionais loop

Analise Estáticas ou Teste Estático:

Que é um tipo de teste que não a execução, O tester olha o código e o analise em busca de possíveis falhas e melhorias para o código

Alguns dos principais problemas que são identificados aqui é:

- Falta de documentação ou documentação não muito bem estruturada
- falta de tratamento de exceções
- smell codes e algumas outras boas praticas não aplicadas.
- Existe algum loop sem condição de parada
- Conexões abertas estão sendo fechadas
- código organizado e com Legibilidade
- Variáveis, Constantes e afins tem uma boa nomenclatura
- findbugs ferramentais para esses tipos de testes

- Teste de cobertura

Teste de caminhos:

Tem como objetivo de calcular a complexidade lógica do software, e descobrir os caminhos básicos do software de forma que todos os caminhos senha efetuados.

Teste de estrutura de controle

E um complemento para o teste do caminho básico e assim garantir uma alta qualidade para técnica de caixa branca

Teste de condições

São teste realizados, nas condições booleana simples ou composto para analisar os desvios possíveis, onde o teste examina os lados verdadeiros ou falsos da condição.

Uma condição encontra se errada das seguintes formas:

Erros de operador booleano,

Erros de variáveis booleana,

Erros de parêntese booleano,

Erros de operador relacional

Erros de expressão aritmética

Caixa Preta

São os teste que não olham para o código como executar um código e ver se ele roda sem erros até final

Alguns teste são:

- Teste baseado em cenários
- Teste baseado em Casos de Uso
- Analises de valores Limites

Testes Manuais vs Testes Automatizados:

Testes manais, são aqueles ao serão executados com a intervenção humana, de forma que o tester tenha que realizado, como adicionar um novo usuário a um banco de dados de um sistema já pronto, e você faz isso na mão sem usado de uma script

E**ssas características são:**

- Velocidade de execução - baixa
- Repetitivo e cansativo
- não exige tecnologias
- alto custo a cada execução
- possuem limitações onde envolvem situações de grande paralelismos
- podem explorar além do cenário de teste, quando necessário
- Podem avaliar questões visuais com cores, ícones, e formas
- podem avaliar questões de usabilidade

Teste Automatizados, são teste desenvolvidos com um script e podem ser rodados simplesmente com click de botão ou run em alguém ambiente, podem levar mais tempo para serem desenvolvidos. Além de que a empresas que utilizam de teste automatizados de forma agendada para rodar de madrugada, o que é conhecido como Little Build onde caso encontrado um erro pelos teste o sistema dispara um e-mail informando o erro

características são

- Executa muito rápido
- não cansam
- Exigem domínio de uma tecnologia especifica
- alto custo apenas na criação
- permite testar situações impossíveis de testar manualmente
- faz apenas o programado para fazer

- não avaliam questões visuais
- não avaliam questões de usabilidade

O que Testar?

teste de funcionalidade

Valida que as funcionalidades do sistema estão funcionando corretamente como esperados

e todos os testes que ja vimos checam isso

Teste de desemprenho

valida o desempenho do sistema no que diz respeito ao seu tempo de resposta para determinadas ações. Em muitos sistema é esperado que o tempo de seja mais curto, e que as ações sejam rápidas

Ferramenta: Jmeter

Teste de usabilidade

Verifica aspectos de experiencia do usuário . Como:

elementos da interface como cores, ícones, tipo, e etc.

Teste de segurança

Valida a Proteção de sistemas contra invasões ou acessos indevidos por agentes não autorizados. E são muito necessários quanto mais as informações e dados trafegados sejam sigilosos

Teste de portabilidade

verifica o funcionamento de um sistema em diferentes dispositivos e plataformas. Atualmente nos temos bastante dispositivos para testas seja em APP com isso ou Android, e isso é mais recorrente com website que temos vários tipos de navegadores disponíveis e é necessário testar em cada um deles

Teste de stress

Leva o comportamento a condições extrema,, e validar se o sistema ainda responde de forma aceitável. A ferramenta Jmeter também pode ajudar soque ao invés de como nos teste de desempenho simular alguns usuários no de stress vamos testar com um grande grupo com o objetivo de levar o sistema ao limite

Quando e como os testes entram no desenvolvimento

Planejando cenários

O primeiro cenário que podemos criar são os **cenários de Teste de aceitação**, que apartir dos requisitos do sistema, possam ser pensados, são eventualmente formas de uso, que o cliente usaria no seu dia a dia. efetivamente se simplifica no que o vai realizar com o sistema?

O segundo cenário que podemos criar e logo depois o a coleta de requisitos e pôs refinamento, onde são construídas as especificações, onde estarão contidas todas as definições técnicas do sistemas, delas e apartir dali podemos já trabalhar em cima da criação dos **cenários de teste do sistema**

Boas Praticas

Não técnico

- A pessoas que especifica os cenários de teste de sistema não deve estar participando do provocação do sistemas
  - De preferencia seja alguém que não trabalho no projeto

- Faça um seleção dos cenários, avaliando:
  - O tempo disponível para os mesmo
  - A maturidade do testador(es)
  - A relevância das funcionalidades a serem testados

Técnicos

Sempre que encontrar uma falha, detalhe a bem

- Em que dispositivo acontece
- Realizando o que, o que esta fazendo no sistema quando deu erro
- prints de tela

Analise se existe um padrão na falha antes de reportar. se essa falha se estende a outros produtos associados, a uma categoria.

Em teste, teste os valores limites, um campo que aceita entre 5 a 10, deste o com valores 12, 4 ,191 100, e etc. E muito comum haver erro no controle desses limites

Em formulários teste deixar campos em branco, valores inválidos, quantidades de caracteres

Em cálculos, teste valores positivos, negativos., Zero, intervalos fechados e abertos

🧤 Indo Mais Fundo em Teste

🌚 O Universo dos Testes unitários

🏘️ Bons e Maus Testes

Dublês de Testes

🏝️ Estilos