TESTES

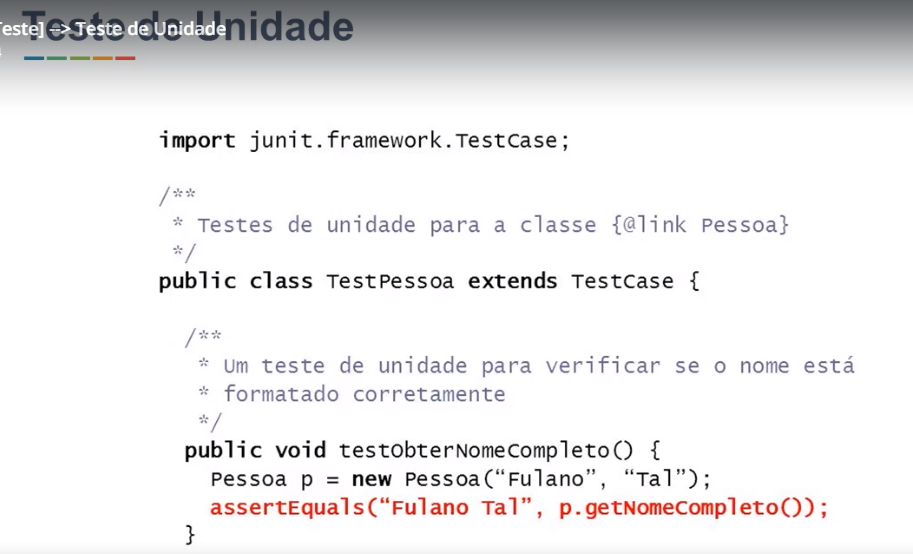
**Quando testar?**

NÍVEIS DE TESTES

1. Teste de Unidade

É feito pelo desenvolvedor, pois é o nível mais baixo de testes, onde testa uma única funcionalidade independente das demais.

Ex: Teste da classe pessoa – onde cria o nome e sobrenome de uma pessoa.



1. Testes de integração

São testes que envolvem um conjunto de componentes, onde é verificada a comunicação entre eles, onde a saída de um componente pode ser a entrada de outro componente.

Esses testes também são realizados pelos desenvolvedores.

Exemplo: Cálculo de dias de atraso, que vai me retornar um inteiro e cálculo de juros que vai me retornar um valor decimal, onde será calculado os juros a partir da quantidade de dias de atraso, calculado pelo componente de dias em atraso.

1. Testes de sistema

Testes o sistema sob o ponto de vista do cliente final, verificando as funcionalidades em busca de falhassem relação aos objetivos originais. Realizado pela equipe de testes, após a codificação está concluída. Utiliza a interface gráfica como browser, celular...

1. Testes de aceitação

Semelhante ao teste de sistema com a diferença que é realizado pelo usuário final, onde ele irá testar e realizar o aceite ou não da funcionalidade entregue.

1. Testes alfa

Semelhante ao teste de aceitação, porém realizado por um grupo maior de clientes finais, executando o sistema de forma não planejada. Geralmente o grupo de pessoas é da própria organização e também do cliente.

Visa identificar falhas ainda não detectadas até o momento, enquanto mais clientes finais utilizam o sistema de forma natural, não planejada.

A equipe de programadores fica a disposição para solucionar possíveis problemas encontrados nos testes.

1. Testes Beta

Testes realizados de forma não planejada, por um grupo desconhecido de clientes, como por exemplo apenas clientes cuja língua inglesa e a versão disponível é Beta, onde o cliente pode ao encontrar falhas, aponta para que seja verificado e corrigido antes de ser lançada a versão final.

Dessa forma se tiver um erro, terá reclamações reduzidas, evitando assim grandes impactos de reclamações.

1. Testes de regressão

Reexecutar testes, sempre que necessário, após alteração serem realizadas no sistema, para conferir se tudo continua funcionando corretamente (detectar efeitos colaterais).

Nesse momento os testes automáticos fazem diferença.

**Como testar?**

Teste de caixa branca: Aquele que se conhece o código, são os testes unitários, testes estáticos.

Testes de caixa preta: Testes baseado em entradas e saídas de cenários macro. São realizados os testes de integração, sistema, aceitação, alfa e beta.

Ferramenta usada em testes estáticos, onde é necessário verificar visualmente a qualidade do código como: se está documentado, se variáveis e constantes possuem boa nomenclatura, se o código está organizado e com boa legibilidade, as consultas de banco de dados estão sendo fechadas após utilizadas. Todos esses pontos são verificados por uma ferramenta a mais famosa **Findbugs.**

Existe uma planilha de checklist que pode ser usada pelo testador quando for realizar os testes estáticos, onde são definidos os itens a serem verificados, status, artefato com erro e as correções a serem realizadas.

Quanto mais cedo encontramos uma falha, mais barato será corrigi-lo, quanto mais tarde esse custo será maior.

**O que testar?**

1. Testes de funcionalidade

Validar se as funcionalidades do sistema estão funcionando corretamente. Para esse tipo de teste são usados todos os níveis de testes, que foram citados acima.

1. Testes de desempenho

Valida o desempenho do sistema com relação ao seu tempo de resposta para determinadas operações.

Ferramenta: Jmeter (jmeter.apache.org)

1. Testes de usabilidade

Validam aspectos que envolvem a experiência do usuário ao utilizar o sistema.

O quanto está fácil usar o sistema, as cores estão agradáveis, os menus estão intuitivos de serem entendidos, o usuário consegue encontrar o que ele procura sem maiores dificuldades, o quanto está eficiente.

Uma das formas de medir é observar um usuário e verificar quais as suas expressões ao utilizar o sistema. Para se observar o quanto é prazeroso para o usuário usar o sistema.

1. Testes de segurança

Validam a proteção do sistema contra invasões ou acesso não autorizado a informações.

Verifica se as informações trafegadas estão criptografadas, verificar se os perfis de acesso estão configurados corretamente

1. Testes de portabilidade

Validam o funcionamento do sistema em diferentes plataformas e dispositivos, nas quais o sistema está proposto a funcionar.

1. Testes de stress

Validam o comportamento do sistema em condições estremas.

**Planilhas para os testes**

1. Testes de sistema

Nessa planilha deve ser criado os senários mais detalhados possíveis, que serão realizados pela equipe de testes de TI

1. Testes de aceitação

Nessa planilha são descritos os senários de forma mais funcional, onde a equipe da área de negócio que irá realizar esses testes. Essa planilha é bem semelhante a utilizada nos testes de sistema.

1. Testes alfa e beta

Nessa planilha de testes deixa de existir a aba de senários, pois esse teste é feito de forma livre, onde usuários da área de negócio realiza testes livres na ferramenta. Nessa planilha existe a aba itens reportados com as colunas: título do item, status, descrição, quem reportou, observação, resolvido.

**Boas Práticas Não Técnicas**

Registar todas as falhas encontradas e acompanhar até que estejam resolvidas.

**Boas Práticas Técnicas**

Ao realizar os testes, se for necessário, crie novos cenários de testes e acrescente na planilha de testes esse novo cenário.

Se encontrar uma falha, detalhar ao máximo, incluindo a versão do browser, dispositivo e prints.

Após a falha ser corrigida, realizar reteste. Teste de regressão.

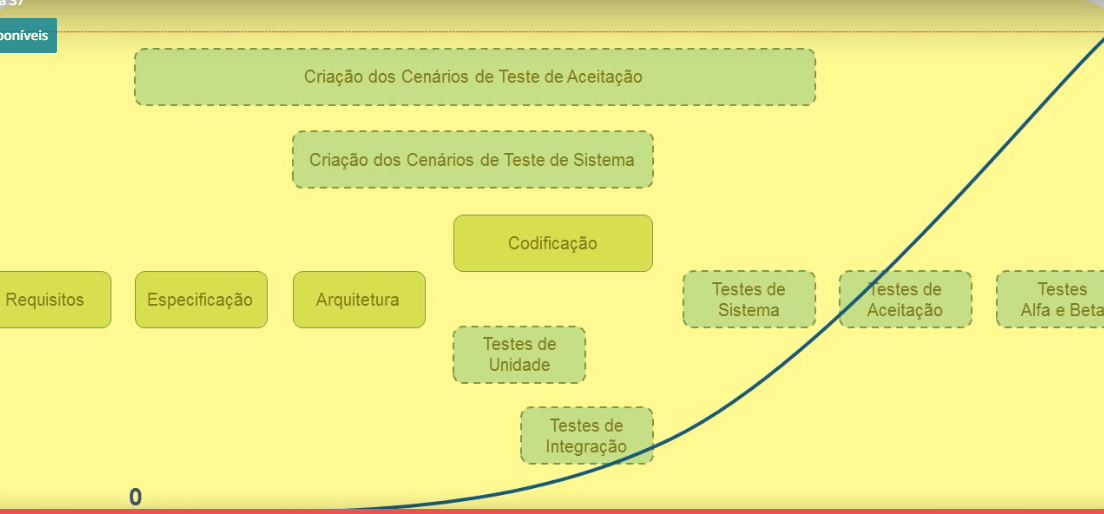
Buscar padrões ao reportar uma falha. Verificar se o erro ocorrido com um produto está associado apenas a tal produto ou se envolve todos os produtos da mesma categoria.

Tentar valores limites

Formulários: compôs em branco, campo com valor inválido, quantidade de caracteres, e-mail invalido, mensagens de erro coerentes.

Cálculos: valores positivos, valores negativos, zero, intervalor fechados e abertos

**Custo da Falha**



**Certificações**

