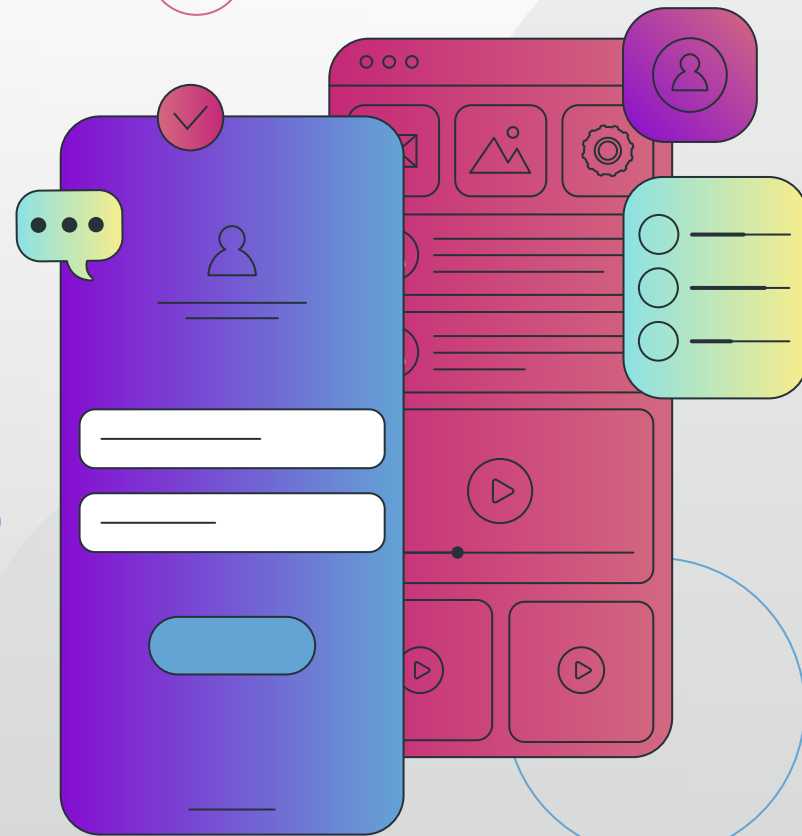


Engenharia de Qualidade

Aula 14 - Testes de Software

Prof. Douglas Moreno



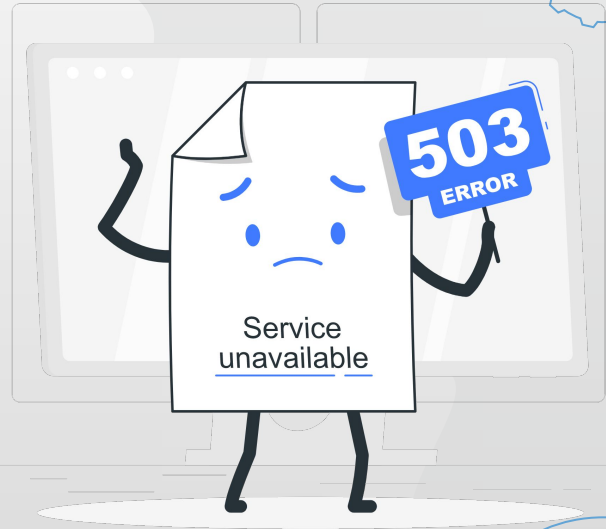


Testes de Software

Por mais que se planeje a construção de um software, erros são passíveis de ocorrer.

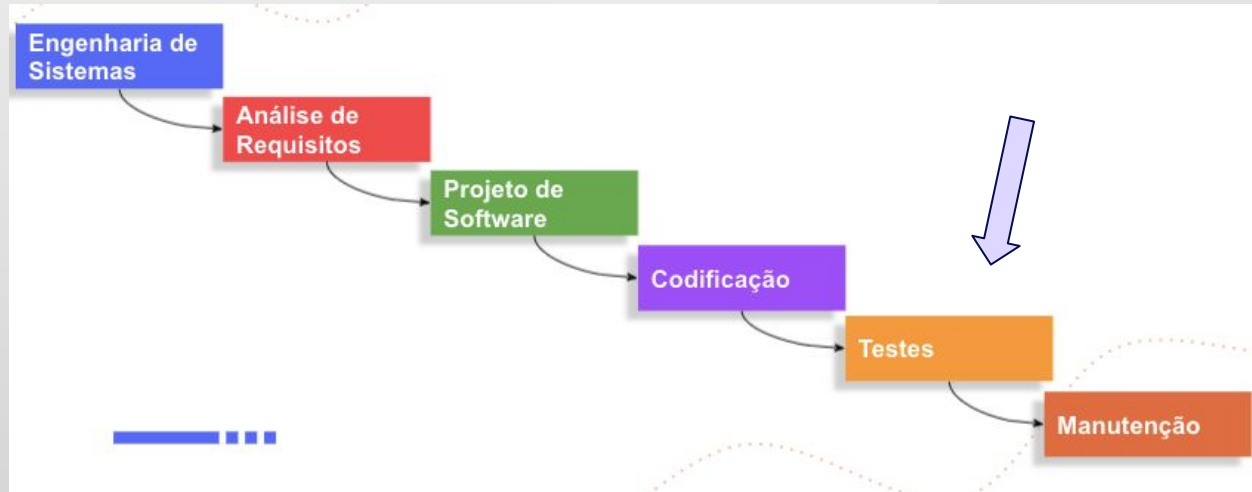
TESTES DE SOFTWARE

- Uma das etapas mais importantes no desenvolvimento de um software é a fase de **Testes**.
- A fase de testes é de fundamental importância, pois através dela é possível detectar e solucionar falhas/erros no software.



TESTES DE SOFTWARE

O conjunto de etapas realizadas durante os **testes** que são responsáveis por garantir o **funcionamento adequado** do software, assim como a sua confiabilidade e segurança.



ALGUMAS ETAPAS DE TESTE

Planejamento de Testes:

Envolve a definição dos objetivos dos testes, identificação dos requisitos a serem testados.

Preparação do Ambiente de Teste:

Configurar um ambiente de teste com os recursos necessários disponíveis para execução dos testes.

Execução dos Testes:

São registrados os resultados dos testes, incluindo quaisquer defeitos encontrados.



ALGUMAS ETAPAS DE TESTE

Reteste e Retorno ao Desenvolvimento:

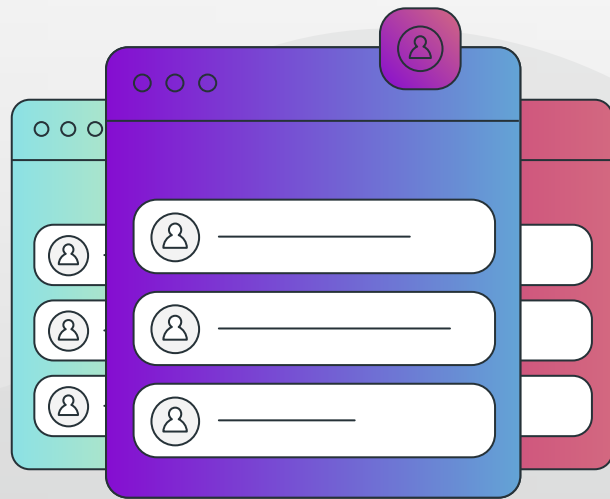
Caso sejam encontrados defeitos durante os testes, eles são corrigidos pela equipe de desenvolvimento e reexecutados para verificar se as correções foram eficazes.

Encerramento dos Testes:

Se todos os critérios de aceitação forem atendidos, os testes são encerrados e o software é liberado para o cliente/usuário.



"Os testes são realizados com a **inserção de dados reais** para que se possam obter os resultados esperados de acordo com os requisitos e especificações previamente estabelecidos".





Por que testes são importantes?



FOGUETE ARIANE 5

Uma comissão de inquérito investigou as causas da explosão e publicou um relatório duas semanas depois. A causa do acidente foi uma falha de software no quadro de referência inercial.



SOFTWARE COM ERROS FATAIS



Alguns dos mais famosos erros de softwares da história

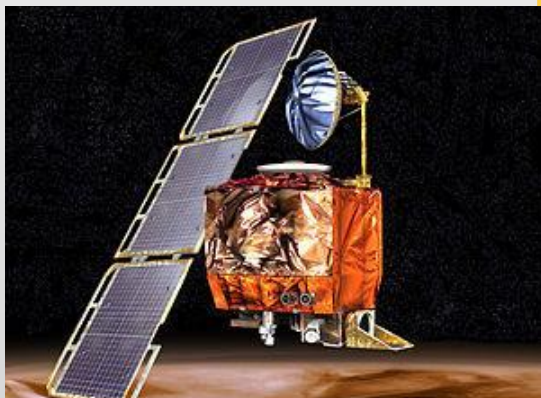


THERAC-25

Falta de testes adequados e validação contínua, é possível compreender a urgência de uma gestão de qualidade de software robusta

MARS CLIMATE ORBITER

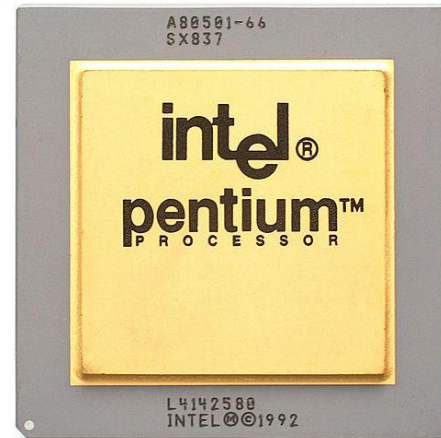
A causa principal do fracasso foi uma falha na conversão de unidades entre os sistemas de medição dos contratantes envolvidos.



SOFTWARE COM ERROS FATAIS



Alguns dos mais famosos erros de softwares da história



PENTIUM A8050

Conhecido como "Pentium FDIV Bug", é um famoso caso de falha de hardware em um processador.



BOEING 737 MAX

Dados incorretos de um sensor e forçava o nariz da aeronave para baixo repetidamente, até que os pilotos perdessem o controle.



SOFTWARE COM ERROS FATAIS



Alguns dos mais famosos erros de softwares da história



HOSPITAL DE TRILLIUM

Erro de software no sistema de controle da unidade de radioterapia resultou na administração de doses letais de radiação. Três pacientes morreram.



Principais Tipos de Testes

Quais testes você faria?



Quais testes você faria?

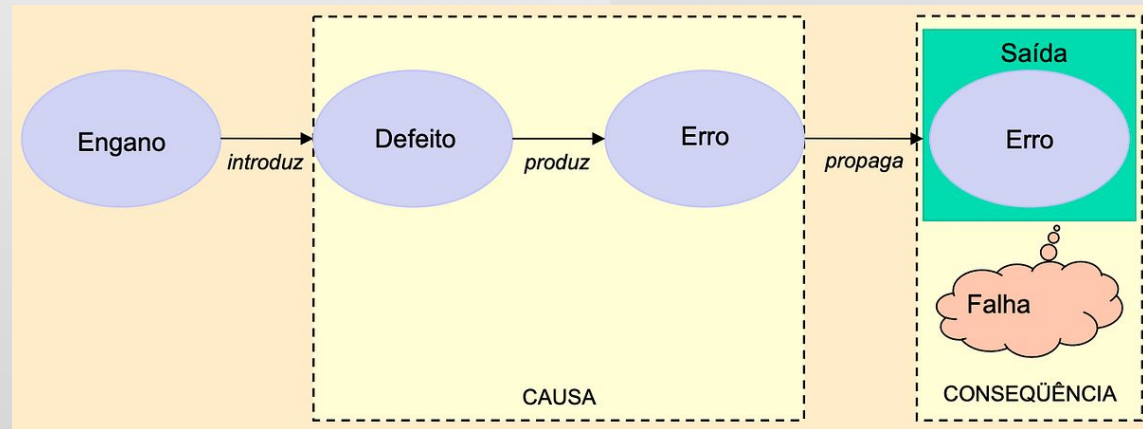


INTRODUÇÃO

- **Validação:** Assegurar que o produto final corresponda aos requisitos do usuário.
 - **Estamos construindo o produto certo?**
- **Verificação:** Assegurar consistência, completitude e corretitude do produto em cada fase e entre fases consecutivas do ciclo de vida do software
 - **Estamos construindo corretamente o produto?**
- **Teste:** Examina o comportamento do produto por meio de sua execução (análise dinâmica)

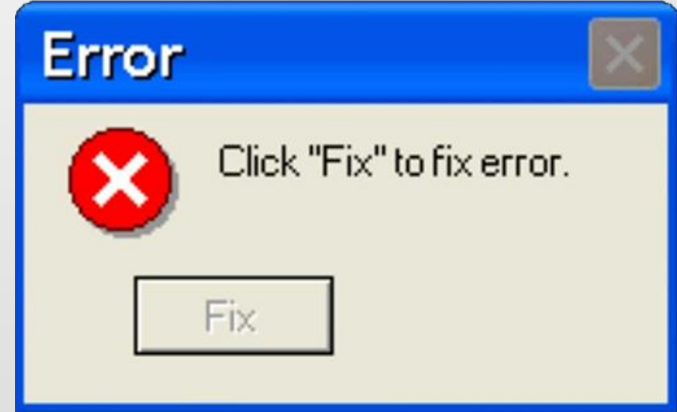
DEFEITO -> ERRO -> FALHA

- **Defeito:** deficiência mecânica ou algorítmica que, se ativada, pode levar a uma falha
 - Instrução ou comando incorreto
- **Erro:** item de informação ou estado de execução inconsistente
- **Falha:** evento notável em que o sistema viola suas especificações



DEFEITOS NO PROCESSO DE DESENVOLVIMENTO

- A maior parte é de origem humana.
- São gerados na comunicação e na transformação de informações.
- Continuam presentes nos diversos produtos de software produzidos e liberados (10 defeitos a cada 1000 linhas de código).
- A maioria encontra-se em partes do código raramente executadas .



DEFEITOS NO PROCESSO DE DESENVOLVIMENTO

- **Principal causa:** tradução incorreta de informações
- Quanto antes a presença dos defeitos for revelada, menor o custo de correção do defeito e maior a probabilidade de corrigi-lo corretamente
- **Solução:** introduzir atividades de VV&T ao longo de todo o ciclo de desenvolvimento

Ooops! ERROR 404



ALGUNS PROBLEMAS COMUNS...

Todos já testaram
algum produto de
software...

Quais foram os
maiores desafios?

ERRO NA LINHA 9
Eu olhando a linha 9
perfeita:



TESTES DE SOFTWARE

Testes progressivos

- São elaborados conforme a evolução do software
- Testa-se as inovações no software

Testes regresivos

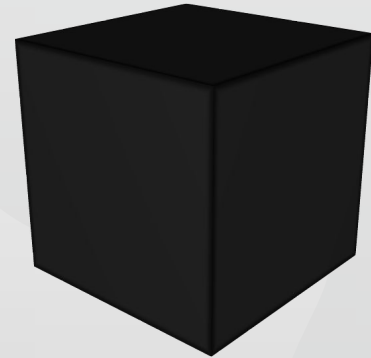
- Reexecutar um subconjunto de testes
- Deve assegurar que as alterações ou inserções de determinados segmentos do software não afetaram outras partes do software
- Toda nova versão do produto deve passar por uma nova sessão de testes

2.1 TESTE FUNCIONAL (TESTE DE CAIXA PRETA)

Testes de caixa preta ou comportamentais são testes baseados na especificação da funcionalidade, e não na sua estrutura.

- Testa os requisitos do sistema
- Não exige conhecimento da tecnologia
- Implantação simples

O foco, nesse caso, é nos requisitos da aplicação, ou seja, nas ações que ela deve desempenhar.



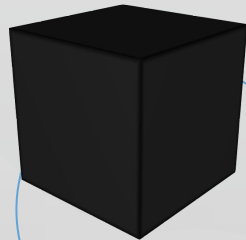
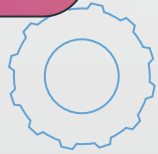
2.1 TESTE FUNCIONAL (TESTE DE CAIXA PRETA)

Pontos negativos:

- Não é possível testar segmentos complexos de códigos.
- Geralmente identifica os módulos defeituosos, mas não explica em detalhes se eles atendem ao requisito.
- Pode entregar resultados superestimados.

Pontos positivos:

- O testador não precisa ter conhecimento técnico e funcional do sistema.
- Não é necessário criar scripts.
- Os casos de testes podem ser projetados assim que as especificações funcionais estiverem completas.



2.2 TESTE ESTRUTURAL (TESTE DE CAIXA BRANCA)

Testes de caixa branca ou estruturais têm como objetivo avaliar a estrutura do código e verificar se todos os caminhos possíveis de execução do programa são testados e verificados.

- Testa a estrutura interna do software
- Exige conhecimento da tecnologia empregada
- Difíceis de serem projetados

Projetado em função da estrutura do componente e permite uma averiguação mais precisa do comportamento dessa estrutura.



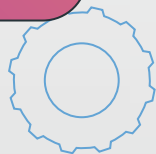
2.2 TESTE ESTRUTURAL (TESTE DE CAIXA BRANCA)

Pontos negativos:

- Custos mais altos.
- Se o código sofrer muitas alterações pode dificultar o processo de teste automatizado.
- Caso o código esteja parcialmente pronto, o teste de caixa branca poderá trazer resultados inconclusos.

Pontos positivos:

- Otimização ao encontrar bugs ocultos.
- Facilidade para criar testes automatizados.
- Meticulosidade, uma vez que o teste de caixa branca tem como objetivo cobrir ao máximo o código.



CATEGORIAS DE TESTES DE SW

- Os testes devem ser organizados em categorias para facilitar a priorização de tipos de erros a serem descobertos
- Dificulta a perda de foco nos testes
- Facilita a identificação de cenários

Funcionalidade

Usabilidade

Estresse

Volume

Configuração

Compatibilidade

Segurança

Performance

Instalação

Recuperação

Confiabilidade e
Disponibilidade

2.3 TESTE DE FUNCIONALIDADE

Tem por objetivo:
Simular os cenários de negócio
Garantir que todos os requisitos
funcionais foram implementados

Devem seguir os documentos de
especificação funcional Testa:
Pré e pós-condições, e cenários
primários, alternativos e de exceção



2.4 TESTE DE USABILIDADE

Tem por objetivo simular as condições de utilização do software sob a perspectiva do usuário final.

Testa:

**Facilidade de navegação
Clareza dos textos e mensagens
Acesso a mecanismos de ajuda
Volume reduzido de interações
Padronização visual**



Testes de Usabilidade

Esse processo ajuda a garantir que o produto final seja mais intuitivo, eficiente e agradável para os usuários finais.

2.5 TESTE DE CARGA (STRESS)

Baseado em alterações sucessivas de processamento.

Deve-se:

- **Elevar e reduzir o nº de transações simultâneas**
 - **Aumentar o nº de usuários**
- **Aumentar e reduzir o tráfego de rede**



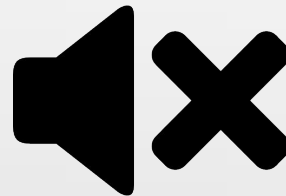
O objetivo principal é identificar o ponto em que o sistema começa a falhar, para que possam ser feitas melhorias na robustez e estabilidade.

2.6 TESTE DE VOLUME

Tem por objetivo determinar os limites de processamento. Não focaliza oscilações, mas sim o aumento contínuo da execução.

Deve-se:

- **Aumentar sucessivamente o volume de:**
 - **Transações**
 - **Consultas**
 - **Dos tamanhos dos arquivos processados**



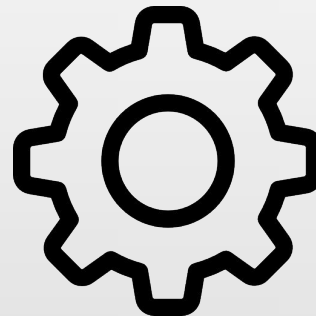
O teste de volume é uma técnica utilizada para avaliar o desempenho e a confiabilidade de um sistema, aplicativo ou banco de dados quando ele é submetido a grandes quantidades de dados.

2.6 TESTE DE CONFIGURAÇÃO

Tem por objetivo executar o software sobre diversas configurações de softwares e hardwares.

Deve-se:

- **Variar os sistemas operacionais**
- **Variar browsers**
- **Variar hardwares que irão interagir com a solução**



Ele ajuda a garantir que o software funcione de maneira confiável em qualquer ambiente em que for utilizado.

2.7 TESTE DE COMPATIBILIDADE

Tem por objetivo executar o software interagindo com versões anteriores de outras aplicações ou dispositivos físicos.

Deve-se:

- **Importar os dados gerados pela solução anterior**
- **Comunicar-se com todas as versões de layout anteriores**



Garante que o software seja acessível e utilizável por uma base ampla de usuários, independentemente de sua plataforma ou dispositivo.

2.8 TESTE DE SEGURANÇA

Tem por objetivo detectar as falhas na segurança que podem comprometer o sigilo e fidelidade das informações. Pode também provocar perdas de dados ou interrupções.

Deve-se:

- **Tentar acessar funcionalidades avançadas, invadir servidores, descobrir senhas, simular infecção por vírus.**



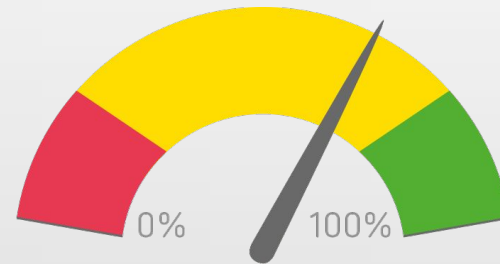
O teste de segurança é um tipo de teste de software que tem como objetivo identificar vulnerabilidades, falhas e riscos de segurança em um sistema, aplicativo ou rede

2.9 TESTE DE PERFORMANCE

Tem por objetivo verificar se o desempenho está consistente com os requisitos definidos.

Deve-se:

- **Para cada cenário, executá-lo e verificar se o desempenho foi o especificado**
- **Simular N usuários acessando a mesma informação ou processando a mesma transação simultaneamente**



Ele é essencial para garantir que o software funcione de maneira eficiente e confiável, mesmo quando submetido a diferentes níveis de demanda.

2.10 TESTE DE INSTALAÇÃO

Tem por objetivo validar os procedimentos de instalação, seguindo variações (normal, alternativas, ...).

Deve-se:

- **Realizar a 1ª instalação do software**
- **Realizar a instalação de um SW já instalado**
- **Realizar atualizações**
- **Realizar todas as alternativas de instalação**



Recomenda-se que o próprio usuário realize-a.

2.11 TESTE DE RECUPERAÇÃO

Tem por objetivo avaliar o comportamento do software após a ocorrência de um erro ou de determinadas condições anormais

Deve-se:

- **Interromper o acesso à rede por alguns instantes e por um longo período**
- **Realizar ações, cancelar o processamento e verificar se as ações foram ou não concluídas**



Esse tipo de teste é fundamental para assegurar a resiliência e a continuidade dos negócios em situações adversas.

2.12 TESTE DE CONFIABILIDADE E DISPONIBILIDADE

Confiabilidade

- Interrupção na execução por um problema

Disponibilidade

- Tempo para a resolução do problema

Deve-se

- Monitorar o ambiente de aceite
- Identificar as interrupções
- Cronometrar o tempo de correção



A confiabilidade está relacionada à precisão e consistência das funcionalidades do software ao longo do tempo.



FASES DOS TESTES DE VALIDAÇÃO

Teste de unidade

Teste de
integração

Teste de sistema

Teste de aceitação

TESTES DE UNIDADE

O teste unitário é feito apenas com o componente específico, para evitar que erros provenientes de outras partes do sistema possam se propagar e resultar em efeitos colaterais no componente testado.

- Estratégia caixa branca e preta
- Testa partes do software
- Requer conhecimento da estrutura interna
- Pode ser executado pelo desenvolvedor

Categorias

- Funcionalidade
- Usabilidade

Executar

- Todas as linhas de código
- Todos os desvios condicionais
- Todos os fluxos alternativos



TESTE DE INTEGRAÇÃO

Os testes de integração verificam se não há falhas operacionais dos componentes quando utilizados de forma conjunta.

- Estratégia caixa branca e preta
- Testa integração entre partes do software
- Requer conhecimento da estrutura interna
- Pode ser executado pelo desenvolvedor

Categorias

- Funcionalidade
- Usabilidade
- Segurança

Exercitar

- Todas as dependências entre componentes
- Todas as interfaces do Sistema

TESTES DE SISTEMA

Esse tipo de teste serve para verificar funcionalidades e características de desempenho e usabilidade do Sistema de modo geral.

- Estratégia de caixa-preta
- Testes aplicados no software como um todo
- Não requer conhecimento da estrutura interna
- Deve ser executado por grupo de teste independente

Categorias

- Performance / Instalação
- Recuperação / Carga



TESTES DE ACEITAÇÃO

Os testes de aceitação costumam ser executados com vários grupos distintos, em momentos diferentes do ciclo de desenvolvimento.

- Estratégia de caixa-preta
- Testes aplicados no software como um todo
- Não requer conhecimento da estrutura interna
- Deve ser executado pelos usuários finais

Categorias

- Funcionalidade
- Usabilidade
- Segurança





VALIDAÇÃO DO ACEITE

Aceite formal

Clientes planejam e realizam os testes de software

Alpha-teste

Clientes são convidados a operar o software na empresa de desenvolvimento

Beta-teste

Clientes selecionados recebem o software para operar em seu ambiente

Implantação total

Todos os clientes recebem o software devidamente testado



ALGUMA PERGUNTA?

MEUS CONTATOS:

(63) 999835068

OBRIGADO!