

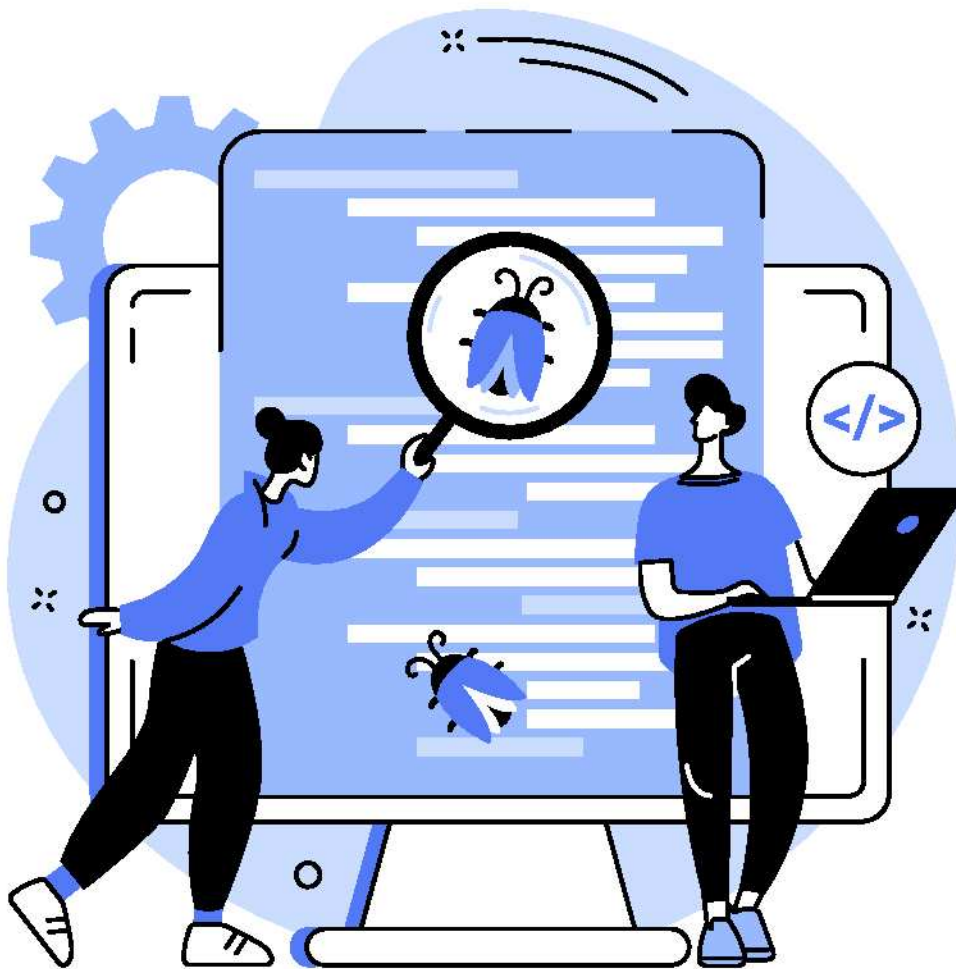


TESTE E MANUTENÇÃO DE SOFTWARE

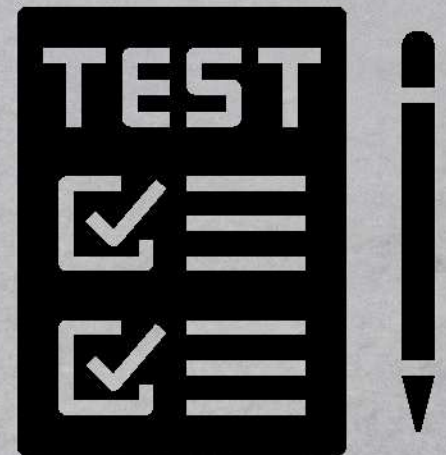




SOFTWARE TESTING



TESTE DE SOFTWARE



POR QUE TESTAR?

TESTES SERVEM PARA MOSTRAR QUE UM PROGRAMA FAZ O QUE É PROPOSTO A FAZER E PARA DESCOBRIR OS DEFEITOS DO PROGRAMA ANTES DO USO

1

Verificar se o sistema está FAZENDO o que foi SOLICITADO que ele fizesse no requisito.

TESTE DE SOFTWARE é o processo que visa executar o sistema de forma controlada, com o objetivo de AVALIAR O SEU COMPORTAMENTO, baseado no que foi ESPECIFICADO.

2

Garantir que o negócio não vai CORRER RISCOS provocados por DEFEITOS em produção.

Será que o TESTE prova que tudo está bem e funcionando adequadamente?

3

Assegurar a QUALIDADE do sistema.



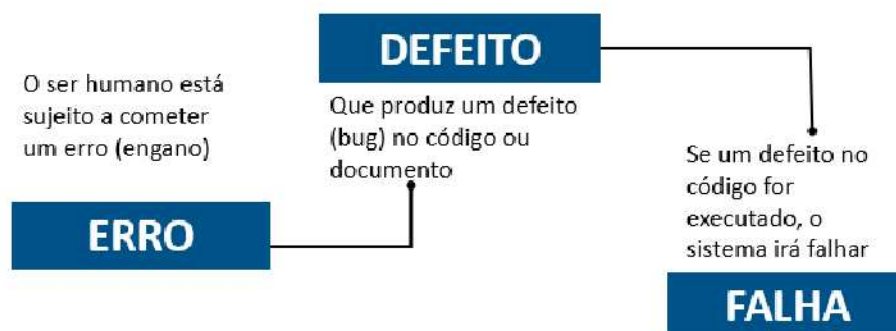
#TESTEDESFTWARE

Teste de Software

ERRO – Engano cometido por seres humanos

FALHA – Resultado ou manifestação de um ou mais defeitos

DEFEITO – Resultado de um erro encontrado num código ou num documento (bug)



Técnicas de Testes de software

#TESTES

CAIXA BRANCA -

Dentro desta categoria de teste o desenvolvedor tem acesso ao código fonte da aplicação.

CAIXA CINZA um ponto de equilíbrio virtual entre o teste de caixa-branca e o caixa-preta.



CAIXA PRETA – O testador não possui acesso algum ao código fonte do programa.

#CAIXAPRETA

Testes de CAIXA PRETA.

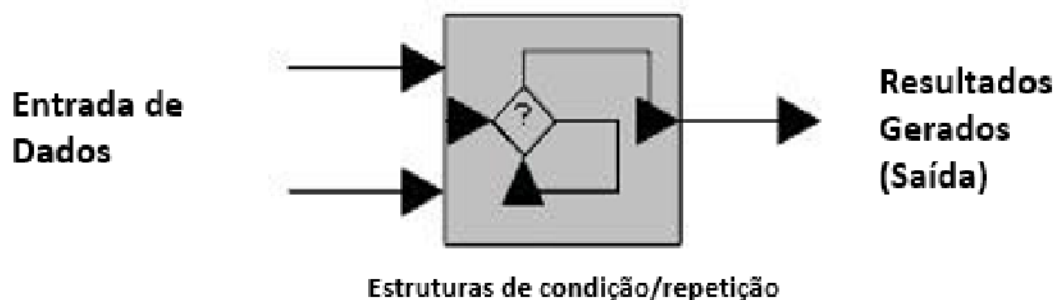


Se o tipo de saída não ocorre, então houve uma situação de DEFEITO.

Técnicas de Testes de software

#CAIXABRANCA

Testes de **CAIXA BRANCA**.



#RESUMINDO

Caixa Preta: garantem que os requisitos foram atendidos, ou seja, que os requisitos estão corretamente codificados.

Caixa Branca: garantem que os sistemas sejam estruturalmente sólidos e que funcionem no contexto técnico onde serão instalados.



ALGUNS TIPOS DE TESTE DE SOFTWARE



TIPO DE TESTE	DESCRIÇÃO
Teste de Unidade	Teste em um nível de componente ou classe. É o teste cujo objetivo é um "pedaço do código".
Teste de Integração	Garante que um ou mais componentes combinados (ou unidades) funcionam. Podemos dizer que um teste de integração é composto por diversos testes de unidade.
Teste Positivo-Negativo	Garante que a aplicação vai funcionar no "caminho feliz" de sua execução e funcionará no seu fluxo de exceção.
Teste de Interface	Verifica se a navegabilidade e os objetivos da tela funcionam assim como foram especificados e se atendem da melhor forma ao usuário.
Teste de Aceitação do Usuário	Testa se a solução será bem vista pelo usuário. Ex: caso exista um botão pequeno demais para executar uma função, isso deve ser criticado em fase de testes (aqui, cabem quesitos fora da interface também).
Teste de Volume	Testa a quantidade de dados envolvidos (pode ser pouca, normal, grande ou além de grande).
Testes de Configuração	Testa se a aplicação funciona corretamente em diferentes ambientes de hardware ou de software.
Testes de Instalação	Testa se a instalação da aplicação foi bem sucedida.
Teste de Sistemas	Testa a execução do sistema como um todo, a fim de validar a exatidão e perfeição na execução de suas funções.
Teste de Usabilidade	Testa e simula as condições de utilização do software sob a perspectiva do usuário final. Esses testes focalizam a facilidade de navegação entre as telas, clareza dos textos e as mensagens que são apresentadas aos usuários, dentre outros aspectos da interface do sistema.
Testes de Progressão	Testa apenas as funcionalidades (ou requisitos não funcionais) especificadas para a versão.
Teste de Fumaça	Teste o qual acontece rapidamente, executando as principais funcionalidades do sistema sem se preocupar com as condições de erro. É o mesmo que o teste do "caminho feliz".



A MANUTENÇÃO SEMPRE VAI EXISTIR!

Estágios de DESENVOLVIMENTO e MANUTENÇÃO têm
sido considerados integrados e contínuos

MANUTENÇÃO DE SOFTWARE

A manutenção **SEMPRE VAI EXISTIR** e **CONSUMIRÁ** bastante **TEMPO** por parte da equipe de desenvolvimento.



1.

Os **CUSTOS** de manutenção podem ser **MAIORES** do que os custos de desenvolvimento inicial.

2.

TROCA DAS PESSOAS que compõem as equipes de desenvolvimento.

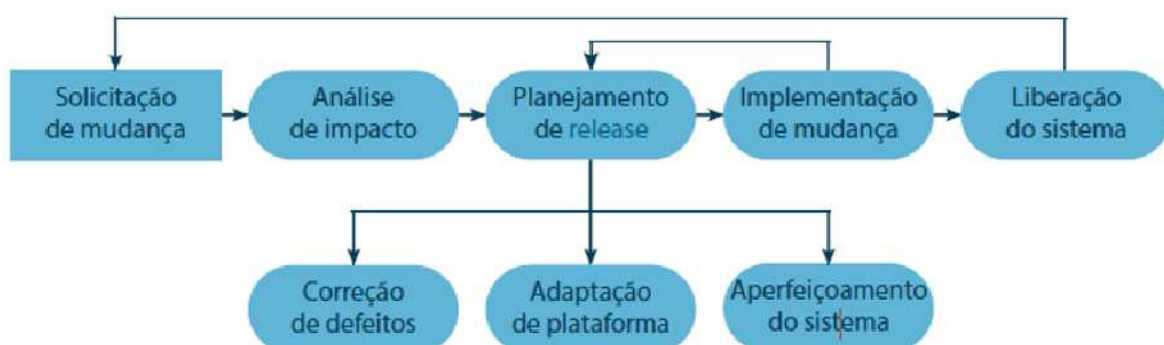
3.

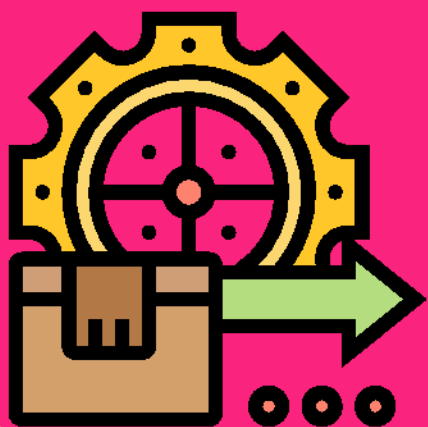
FALTA de **DOCUMENTAÇÃO** ou incompleta ou desatualizada.

4.

Depois que o software é implantado, com certeza ocorrerão **MUDANÇAS**

- Estágios de **DESENVOLVIMENTO** e **MANUTENÇÃO** têm sido considerados **integrados** e **contínuos**:





TIPOS DE MANUTENÇÃO

- ✿ **ADAPTATIVAS:** São alterações que visam adaptar o software a uma nova regra de negócio. Finalidade: adequar o sistema ao ambiente no qual está inserido.
- ☀ **CORRETIVAS:** Tem o objetivo de solucionar defeitos encontrados no software. Problemas de funcionalidade são comuns e em alguns casos devem ser corrigidos de forma emergencial. Mesmo que equipe de desenvolvimento utilize as melhores técnicas e etapas de testes, os problemas poderão acontecer.
- ✿ **EVOLUTIVAS:** São alterações que visam agregar novas funcionalidades e melhorias para o software. Num mercado cada vez mais dinâmico e concorrido, é natural que sejam exigidas **NOVAS FUNCIONALIDADES** e mudanças para um produto se manter competitivo.



Por que a MANUTENÇÃO DO SOFTWARE é importante?



Ajustes pós-implantação

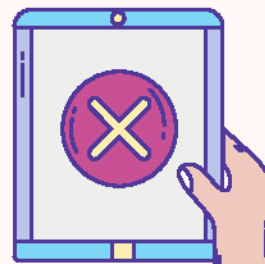


Melhorias substanciais

Inevitável que ocorram MUDANÇAS:pequenos ajustes após a implantação para melhorias, por força da legislação, para atender novos requisitos dos usuários, ou por estar com erros.



Atualização na legislação e atender novos requisitos



Erros no software

Para as empresas os processos de MANUTENÇÃO são considerados MENOS DESAFIADORES do que o desenvolvimento de um software original, ainda que tenha um CUSTO MAIS ELEVADO e riscos.

