

Sequência de **TESTE DE SOFTWARE**

1º Teste de Unidade:

Quando Aplicar: O teste de unidade deve ser aplicado durante a fase de desenvolvimento, logo após a implementação de uma unidade de código (geralmente uma função ou método).

Exemplo: Se um desenvolvedor implementar uma função de cálculo de imposto, um teste de unidade seria escrito para garantir que a função retorne os resultados corretos com base em cenários de entrada específicos.

2º Teste de Componente:

Quando Aplicar: O teste de componente é aplicado quando várias unidades de código (módulos ou classes) são desenvolvidas e precisam ser testadas juntas.

Exemplo: Se um aplicativo for dividido em módulos, como um módulo de autenticação e um módulo de gerenciamento de perfil, o teste de componente garante que esses módulos funcionem corretamente quando integrados.

3º Teste de Integração:

Quando Aplicar: O teste de integração é realizado à medida que os módulos e componentes individuais são integrados para verificar se a interação entre eles é livre de erros.

Exemplo: Quando um sistema de e-commerce é desenvolvido, o teste de integração verificará se o módulo de gerenciamento de estoque funciona corretamente com o módulo de processamento de pedidos.

4º Teste de Sistema:

Quando Aplicar: O teste de sistema é aplicado quando o software está pronto e pronto para ser testado como um todo para garantir que atenda aos requisitos definidos.

Exemplo: Em um aplicativo de reservas de passagens aéreas, o teste de sistema verifica se os clientes podem pesquisar voos, selecionar assentos e efetuar reservas de acordo com os requisitos do sistema.

5º Teste de Aceitação:

Quando Aplicar: O teste de aceitação é realizado após o teste de sistema e geralmente envolve os usuários finais ou stakeholders para validar se o software atende às suas expectativas.

Exemplo: No teste de aceitação de um site de comércio eletrônico, os clientes reais realizam compras e fornecem feedback sobre a usabilidade e a experiência geral de compra.

6º Teste de Regressão:

Quando Aplicar: O teste de regressão é aplicado sempre que são feitas alterações no software para garantir que essas alterações não causem a reintrodução de erros existentes.

Exemplo: Após a correção de um bug em um aplicativo de processamento de pagamentos, o teste de regressão é executado para confirmar que o bug não reapareceu e que outras partes do aplicativo não foram afetadas.

7º Teste de Desempenho:

Quando Aplicar: O teste de desempenho é aplicado para avaliar o desempenho do software, geralmente antes do lançamento, para verificar se ele pode lidar com cargas típicas e picos de uso.

Exemplo: Em um site de notícias, o teste de desempenho pode simular milhares de acessos simultâneos para garantir que o servidor possa suportar o tráfego durante um evento de notícias de grande alcance.

8º Teste de Segurança:

Quando Aplicar: O teste de segurança é aplicado para identificar vulnerabilidades de segurança no software. Deve ser incorporado ao longo do ciclo de desenvolvimento.

Exemplo: Um teste de segurança em um aplicativo bancário procura por vulnerabilidades, como injeções SQL ou falhas de autenticação, que poderiam ser exploradas por invasores.

9º Teste de Usabilidade:

Quando Aplicar: O teste de usabilidade é aplicado durante o desenvolvimento e após o lançamento para avaliar a facilidade de uso e a satisfação do usuário.

Exemplo: Em um aplicativo de saúde, os testes de usabilidade podem envolver usuários reais que fornecem feedback sobre a navegabilidade e a compreensão das informações fornecidas.

Lembrando que a ordem e a aplicação desses testes podem variar dependendo da metodologia de desenvolvimento, como desenvolvimento ágil, em que os testes podem ser realizados de forma iterativa e intercalada com o desenvolvimento.