

Prof. Ms. Claudio Souza Nunes

Teste de Software

(introdução)

Engenharia de Software II

Bacharelado em Sistemas de Informação





Teste - Definição

É considerado teste qualquer atividade que tenha como objetivo avaliar um atributo ou capacidade de um programa ou sistema e determinar se este atende aos resultados esperados.

Princípios do Teste

1. O Teste demonstra a presença de defeitos

O Teste é executado apenas para demonstrar que existem erros no software, mas não pode provar que erros não existem. Outra certeza é que o teste reduz a probabilidade de permanência de defeitos no software, porém, se nenhum defeito for encontrado durante todo o processo de teste, isso não prova que o software esteja perfeito.

2. Não é possível fazer um teste "100% completo"

É inviável testar tudo – todas as combinações de entradas em um software. Para que seja realizada uma cobertura satisfatória de testes, então, é necessário levar em consideração os riscos e as prioridades do projeto.

3. Teste Antecipado

As atividades de teste além de ter objetivos bem definidos, devem iniciar o quanto antes dentro do ciclo de desenvolvimento do software.

4. Agrupamento de Defeitos

A maioria dos defeitos em um software estão em um número pequeno de módulos do software.

5. Paradoxo do Pesticida

Os casos de testes precisam ser atualizados e/ou reescritos diversas vezes dentro do clico de testes, para evitar que um mesmo caso seja executado várias vezes e nenhum defeito seja encontrado — o paradoxo do pesticida.

6. O Teste depende do contexto

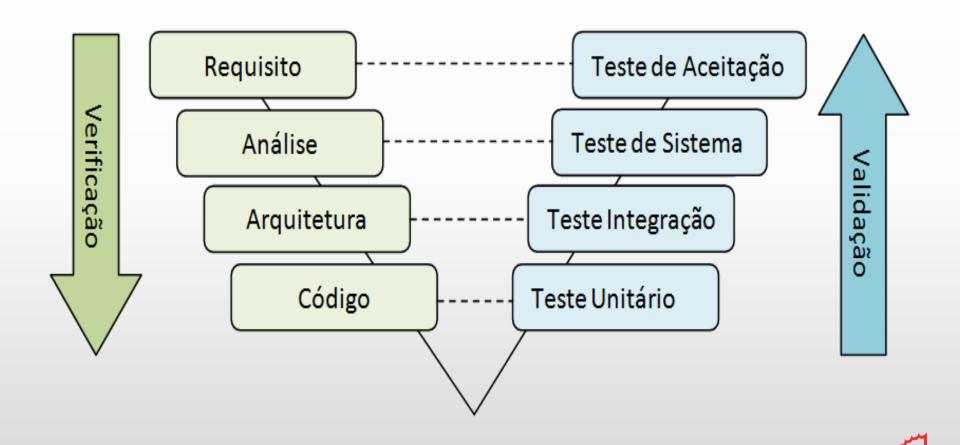
Os testes são realizados de forma diferente dependendo do contexto, ou seja, dependendo do tipo de produto de software a ser testado.

7. A ilusão da ausência de erros

O software deve atender às necessidades, expectativas e solicitação do cliente/usuário. Se isso não for satisfatório, então, não adianta construir o software, testes, e corrigir os defeitos encontrados.



Ciclo de vida do teste de software



Tipos de Teste

- **1. Teste de Configuração ou de Instalação** Verifica o comportamento do sistema em diferentes configurações de software e hardware. Testa se o software funciona no hardware a ser instalado e sob diferentes condições, como pouco espaço de memória, interrupções de rede, interrupções na instalação etc.
- **2. Teste de Integridade** Testa a resistência do software às falhas (robustez). Verifica se os componentes envolvidos vão permanecer íntegros mesmo com um alto volume de dados. Por exemplo, verificar o comportamento de uma tabela com milhões de registros.
- **3. Teste de Segurança** Testa se o sistema e os dados são acessados de maneira segura, apenas pelo autor das ações. Identifica falhas de segurança de um software ou ambiente em que está sendo executado. Avalia as vulnerabilidades em aplicações e serviços frente a diferentes tipos de ataques de segurança.
- **4. Teste Funcional** Testa os requisitos funcionais, as funções e os casos de uso. Verifica se a aplicação faz o que deveria fazer e da maneira correta.
- **5. Teste de Unidade** Testa um componente isolado ou classe do sistema. Basicamente se fornece valores válidos ou inválidos e verificando se o retorno foi de acordo com o esperado.
- **6. Teste de Integração** Testa se um ou mais componentes combinados funcionam de maneira satisfatória. Há quem diga que o teste de integração é composto por vários testes de unidade.
- 7. Teste de Performance O teste de performance se divide em 3 tipos: Teste de carga (testa o software sob as condições normais de uso. Ex.: tempo de resposta, número de transações por minuto, usuários simultâneos etc.), Teste de stress (testa o software sob condições extremas de uso. Grande volume de transações e usuários simultâneos. Picos excessivos de carga em curtos períodos de tempo.) e teste de estabilidade (testa se o sistema se mantém funcionando de maneira satisfatória após um período de uso.)
- **8. Teste de Usabilidade** Teste focado na experiência do usuário, consistência da interface, layout, acesso às funcionalidades entre outros.
- **9. Teste de Acessibilidade** Subconjunto de testes de usabilidade, onde se leva em consideração pessoas com todas as habilidades e deficiências. O significado deste teste é verificar a capacidade de uso e acessibilidade. Acessibilidade visa atender pessoas de diferentes habilidades, tais como: deficiência visual, deficiência física, deficiência auditiva, comprometimento cognitivo, dificuldade de aprendizagem entre outros.
- **10. Teste de Regressão** Reteste de um sistema ou componente para verificar se alguma modificação recente causou algum efeito indesejado, além de, certificar se o sistema ainda atende os requisitos.

Caixa Branca

X

Caixa Preta



Teste de caixa branca

 O analista de testes tem acesso ao código fonte, conhece a estrutura interna do produto sendo analisado e possibilita que sejam escolhidas partes específicas de um componente para serem avaliadas. Esse tipo de teste, também conhecido como teste estrutural, é projetado em função da estrutura do componente e permite uma averiguação mais precisa do comportamento dessa estrutura. Perceba que o acesso ao código facilita o isolamento de uma função ou ação, o que ajuda na análise comportamental das mesmas.

Teste de caixa preta

- O analista não tem acesso ao código fonte e desconhece a estrutura interna do sistema. É também conhecido como teste funcional, pois é baseado nos requisitos funcionais do software. O foco, nesse caso, é nos requisitos da aplicação, ou seja, nas ações que ela deve desempenhar.
- Para mostrar quais problemas que esse tipo de teste rastreia, podemos citar alguns exemplos:
- Data de nascimento preenchida com data futura;
- Campos de preenchimento obrigatório que não são validados;
- Utilizar números negativos em campos tipo valor a pagar;
- Botões que não executam as ações devidas;
- Enfim, todo tipo de falha funcional, ou seja, falhas que contrariam os requisitos da aplicação.
- Há que se destacar, contudo, que existe um elemento comum aos dois tipos de teste. Tanto no teste de caixa branca quanto no teste de caixa preta, o analista não sabe qual será o comportamento da aplicação ou do alvo de teste em uma determinada situação. A imprevisibilidade de resultados é comum aos dois casos.