

### Aula 5: As atividades de teste no processo de desenvolvimento de software



## Apresentação

Nesta aula, iremos definir o conceito de teste para o processo de desenvolvimento de *software*.

A fase de teste tem como objetivo detectar possíveis defeitos ou erros que possam surgir na fase de implementação. Nessa fase, de testes, deve-se coletar os resultados e analisá-los e consertá-los antes de sua implantação.

Essa fase é essencial para aumentar a qualidade do produto ou sistema em que será implantado.

---

## Objetivos

- Conhecer as atividades de teste no processo de desenvolvimento de *software*;
- Entender as necessidades da etapa de teste na melhora da qualidade do sistema;
- Analisar os diversos tipos de teste para sistema ou produto.

# Testes de Software



📷 Teste de software. (Fonte: alphaspirt / Shutterstock)

## Conceito de teste para o processo de desenvolvimento de software

Teste de Software é uma ação que faz parte do processo de desenvolvimento de software, e tem como principal objetivo o de revelar falhas para que sejam corrigidas até que o produto final atinja a qualidade desejada ou acordada.

Os analistas de testes, técnicos de testes ou homologadores, são responsáveis para realizar uma bateria de testes de diferentes naturezas e propósitos, envolvendo não apenas os testes funcionais da aplicação, mas diversas outras atividades como:

- Avaliação da especificação de requisitos,
- Avaliação de projeto técnico,
- Verificações em outros documentos,
- Testes de performance e capacidade,
- Avaliação de interface,
- Dentre outros.



**“Quanto mais cedo, defeitos forem encontrados antes da implementação do sistema, o custo de correção é menor em relação ao encontrado na fase de produção”**

---

(Regra de Myers)

De acordo com o padrão IEEE número 610.12-1990, podemos classificar:

**Defeito** – passo, processo ou definição de dados incorreto, por exemplo, uma instrução ou comando incorreto.

**Engano** – ação humana, por exemplo, tomada pelo programador, que produz um resultado incorreto.

**Erro** – diferença entre o valor obtido e o valor esperado. Qualquer resultado incorreto ou inesperado na execução do programa.

**Falha** – produção de uma saída incorreta com relação à especificação.



---

## **CAUSA**

Engano

Defeito

Erro



---

## CONSEQUÊNCIA

Falha

**Teste de Software** é um processo que faz parte do desenvolvimento de software, e tem como principal objetivo revelar falhas/bugs para que sejam corrigidas até que o produto final atinja a qualidade desejada.



1

### Teste

Processo definido com intenção de encontrar um erro.



2

### Objetivo

Encontrar um erro que ainda não foi descoberto. Um teste bem sucedido corresponde à descoberta de um erro não previsto.



3

### Critério

Definição de uma métrica que, após análise do comportamento do sistema, atenda o critério.



4

## Procedimento

Conjunto de instruções para a realização de testes.



5

## "Script" de teste

É uma representação definida de um procedimento teste.

# Processo de Teste de Software

O Processo de Teste de Software divide-se em três partes principais e fundamentais:

- **Planejamento dos testes:** "Garantir que os testes sejam preparados antes do fim da implementação do produto".
- **Execução dos Testes:** "Executar os casos e procedimentos de teste especificados e comparar os resultados esperados e obtidos, registrando esses resultados".
- **Controle dos testes:** "Garantir que os testes planejados sejam executados corretamente e seus resultados possam ser registrados através da sua monitoração constante".

Vejamos alguns tipos de testes:



## **Testes de sistemas**

Neste teste, o sistema é analisado como um todo, ou seja todos os seus componentes, para validar a execução das suas funções acompanhando cenários chamados de casos de teste e verificar se estão em conformidade com os requisitos anteriormente definidos. Este teste deve ser feito por uma equipe independente e diferente daquela que desenvolveu o sistema.



## **Teste caixa preta**

Neste teste o objetivo é testar todas as entradas e saídas desejadas, mediante uma determinada entrada definida de dados. Aqui não se está preocupado com o código, cada saída indesejada é vista como um erro. Os mecanismos internos do sistema não são levados em conta.



### **Teste caixa branca**

Neste caso o objetivo principal é testar o código dos componentes do sistema, quanto a sua estrutura e construção. Os mecanismos internos do sistema serão analisados e suas representações lógicas também. Este teste não exclui a necessidade do Teste Caixa Preta, uma vez que o funcionamento interno do sistema ou produto pode corresponder logicamente, podendo produzir uma saída diferente da esperada. É comum se encontrar partes do código que nunca foram testadas.



### **Testes de Segurança**

Dada as circunstâncias atuais, onde cada vez mais a segurança dos sistemas é posta à prova, o Teste de Segurança tem como meta garantir que o funcionamento da aplicação esteja exatamente como especificado. Verifica também se o software se comporta adequadamente mediante as mais diversas tentativas ilegais de acesso, visando possíveis vulnerabilidades. Para isso, testa se todos os mecanismos de proteção embutidos na aplicação de fato a proteção de acessos indevidos. Deve-se utilizar os diversos papéis, perfis, permissões, para navegar no sistema.

# Modalidade dos testes

Quanto à utilização do código:

## Testes estáticos

São testes realizados pela análise estática e visual do código fonte, podendo haver um questionário ou alguma ferramenta para acompanhar os testes. Nesse caso, os componentes do software são verificados sem que o produto seja executado. Seja por meio de uma ferramenta automatizada ou dos testes manuais, o principal objetivo dessa técnica é identificar erros de programação, tais como:

- Prática ruins de programação
- Práticas ruins;
- Erros de sintaxe;
- Falhas de segurança.

A análise estática auxilia gestores de TI a identificar todas as linhas de código que foram mal escritas durante a criação de um software. Todos os caminhos de execução, processamento e exibição de valores são examinados. Como consequência, erros mais comuns são descobertos mais rapidamente.

## Testes Dinâmicos

São testes baseados na execução do código do programa, podendo ser encarado como de forma complementar a análise estática. Trabalha, principalmente, com as informações que são inseridas nas rotinas de entrada e saída de dados. Além disso, são verificados itens como:

- O tempo de resposta;
- A performance da aplicação;
- A capacidade do software se adaptar a diferentes ambientes;
- O comportamento funcional.



A análise dinâmica permite que problemas mais sutis sejam identificados. Sem considerarmos o grau de complexidade, as chances de um “bug” passar por uma análise estática e uma análise dinâmica, sem ser rastreado é consideravelmente baixa. Dessa forma, o teste dinâmico consegue dar mais segurança e confiabilidade ao produto final.

E quanto ao objetivo na busca pelo erro:

### Testes de unidade

Teste em nível de módulo, componente ou classe. É o teste cujo objetivo é um “pedaço do código”, ou em alguns módulos definidos que representam uma única unidade. A documentação do projeto define a quantidade de módulos a serem testados.

### Testes de integração

Teste utilizado para garantir que um ou mais componentes combinados, ou unidades, não contenham erros. Podemos dizer que um teste de integração é composto por diversos testes de unidade.

### Testes de validação

Teste realizado após a integração de todos os módulos ou unidades do sistema.

## Referências

---

GUSTAFSON, Davis A. **Engenharia de software**. 8. ed. São Paulo: Pearson Education, 2007. cap. 8 e 13.

PAULA FILHO, Wilson de. **Engenharia de software**: fundamentos, métodos e padrões. 3. ed. São Paulo: LTC, 2009. cap. 1, 5 e 21.

SOMMERVILLE, Ian. **Engenharia de software**. 1. ed. Porto Alegre: Artmed, 2003. cap. 10.

## **Próximos Passos**

---

Etapa de codificação onde será apresentada a parte de desenvolvimento do código baseado nos modelos e desenhos apresentados.

## **Explore Mais**

---

Pesquise na internet sites, vídeos e artigos relacionados ao conteúdo visto.

Em caso de dúvidas, converse com seu professor online por meio dos recursos disponíveis no ambiente de aprendizagem.