

AULA 2 – TESTES DE SOFTWARE

OBJETIVO DA AULA

Conhecer a conceituação de Testes de Software, abordando seus tipos, características e aplicabilidade.

APRESENTAÇÃO

Os testes de software são necessários para garantir que os requisitos estão funcionando da forma como foi planejado.

Existem vários tipos de testes, e estes abrangem diversos tipos de características, envolvendo questões como funcionalidade e usabilidade.

Nesta aula você aprenderá um pouco sobre essa fase tão importante no processo de construção e entrega de softwares.

1. CONCEITOS DE TESTE DE SOFTWARE

Testes de Software constitui-se como uma disciplina da Engenharia de Software que objetiva verificar se o produto funciona da maneira como foi estabelecido inicialmente.

O ofício de testar Software é tão importante que teve um incrível crescimento ao longo dos anos, tanto que hoje temos uma função específica para quem atua nesta área, que é a de Analista de Testes.

Muito embora esse conceito aparente simples, há muita confusão em relação ao que é realizado na fase de testes de software. Para início de conversa, é importante saber que testar software não significa apenas descobrir erros no processo de execução, mas também validar outros aspectos, como a validação dos requisitos, a análise de adequação no tempo de resposta, e também questões de usabilidade, pois o software precisa ser fácil de usar.

Quando um software é verificado e validado, busca-se identificar se há erros nos requisitos do produto, a validação diz respeito se o software está correto em termos de funcionamento, ou seja, se os resultados produzidos são os esperados.

Vale destacar que a realização do teste busca a identificação de erros, mas não exatamente a ausência de erros, pois um teste não tem como identificar a falta de um requisito, por exemplo.

2. CICLO DE VIDA

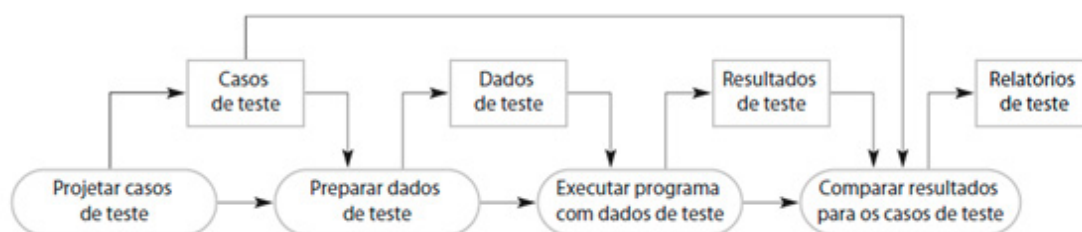
Para realizar um teste de software é importante que haja um planejamento e que um ciclo de vida seja implementado e respeitado. Embora haja outras abordagens, podemos definir um ciclo de vida de software da seguinte forma.

- 1) Análise de Requisitos;
- 2) Planejamento de Testes;
- 3) Integração do Caso de Teste;
- 4) Configuração do Ambiente de Teste;
- 5) Implementação;
- 6) Encerramento.

Na análise de requisitos os profissionais identificam todos os tipos de testes que devem ser executados, há a comunicação com as partes interessadas no projeto, são definidas prioridades e o foco das validações. Também é analisada a necessidade de automatizar os testes.

A Figura 1 mostra um modelo completo de processo de software.

FIGURA 1 | **Modelo de Processo de Software**



Fonte: Sommerville (2011, p. 147).

Na fase de planejamento são elaborados os planos de teste, são definidas ferramentas para a validação, estimados o tempo de trabalho e o custo aproximado para o projeto. Nessa etapa também são definidas as funções e responsabilidades de cada profissional.

Na integração os casos de teste e scripts são elaborados, analisados e aplicados, para que posteriormente os dados sejam avaliados.

Na configuração do ambiente de teste é necessário verificar a arquitetura utilizada, preparar e configurar o ambiente, ou seja, requisitos de hardware e software.

Na implementação os profissionais de teste vão realizar as devidas validações considerando o que foi estabelecido nos casos de teste. Nessa fase recomenda-se a documentação dos resultados dos testes, com o registro dos erros, e deve-se reportar os problemas para a equipe de desenvolvimento. Após as devidas correções as validações devem ser executadas novamente.

Por fim, no encerramento os resultados obtidos são discutidos, para reduzir falhas e custos, possibilitando também a otimização dos processos e cumprimento dos objetivos do negócio. Outra boa prática é a elaboração de um relatório de qualidade abordando todo o processo dos testes, incluindo os detalhes do projeto.

Toda a atividade no processo de testes de software vai procurar analisar o software sob algum ponto de vista técnico. Além dos requisitos funcionais, os testes de software também vão avaliar os requisitos não funcionais como:

- Operabilidade – funcionamento do sistema;
- Observabilidade – visualização dos resultados e coerência na geração das saídas, sinalizando erros;
- Controlabilidade – capacidade de automação e controle do processo;
- Decomposabilidade – capacidade de os módulos de software serem testados independentemente;
- Simplicidade – o quão fácil e intuitivo é o software;
- Estabilidade – indicador de alterações durante os testes;
- Compreensão – o quão clara é a documentação do componente.

3. TÉCNICAS DE TESTES

Os testes podem ser classificados sob diversos pontos de vista, com base em suas técnicas.

Nesse quesito temos as seguintes técnicas:

- Técnica Estrutural (Caixa Branca);
- Testes Funcionais (Caixa Branca);
- Técnica Baseada em Erros.

3.1. TÉCNICA ESTRUTURAIS (CAIXA BRANCA)

As técnicas estruturais, também conhecidas como Teste da Caixa Branca, é a técnica aplicada para descobrir falhas de construção interna das partes do software, ou seja, são verificadas estruturas condicionais e de repetições, se elas são desnecessárias ou foram mal especificadas.

Essa abordagem visa verificar se os caminhos no código foram executados ao menos uma vez. Quanto às estruturas lógicas, são apuradas todas as decisões de verdadeiro ou falso. As estruturas de repetição também são averiguadas para garantir que todos os caminhos possíveis tenham sido percorridos.

3.2. TÉCNICA FUNCIONAL (CAIXA PRETA)

Essa técnica também é conhecida como Teste da Caixa Preta, e utilizada para a validação dos requisitos sem que sejam observadas a elaboração dos códigos. Neste teste são submetidas as entradas e observado o resultado das saídas, que devem corresponder com o que foi planejado.

3.3. TÉCNICA BASEADAS EM ERROS

Essa técnica tem o objetivo buscar erros cometidos durante o desenvolvimento de sistemas. É uma técnica mais imediatista adotada pelos programadores durante o período de desenvolvimento. Ainda assim, ela deve estar contida na elaboração do plano completo de testes, considerando a teoria de que o próprio autor sempre busca testar o correto.

4. TIPOS DE TESTES

Há diversos tipos de testes utilizados dentro do processo de testagem de software. A seguir destacamos os tipos mais utilizados, com o objetivo de cada um:

- Teste de funcionalidade – busca de erros funcionais, ou seja, que tem relação direta com o funcionamento da funcionalidade testada;
- Teste de interface – verifica a usabilidade e a intuitividade na interface do software;
- Teste de conteúdo de informação – verifica as estruturas de dados definidas, criadas e persistidas no banco de dados;
- Teste de desempenho – verifica o comportamento do sistema relacionado ao desempenho quando colocado em uso;
- Teste de carga – também conhecido como teste de stress, verifica o comportamento do sistema diante de situações anormais de recursos em quantidade, frequência ou volumetria;
- Teste de regressão – consiste em repetir testes já realizados a partir de ajustes efetivados, assim quando o software tem alguma configuração mudada, deve-se garantir que essa alteração funcionará e não impactará em nenhuma outra funcionalidade que já esteja rodando;
- Teste de aceitação – este é o teste realizado pelo usuário final como treinamento, validação de todos os requisitos e obtenção da aprovação final para a obtenção da homologação do software;
- Teste de recuperação – realizado para verificar se a recuperação é executada adequadamente após uma falha operacional;
- Teste de segurança – testa questões relacionadas à segurança do sistema quanto aos dados e às invasões.

Existem ainda dois tipos de testes muito comuns, que dizem respeito ao local em que os testes são realizados:

- Teste alfa – é realizado pelo usuário nas instalações do desenvolvedor, para que qualquer ajuste percebido como necessário;
- Teste beta – realizado pelo usuário em seu próprio ambiente, este teste acontece quando o produto já está pronto para o uso, mas ainda requer um monitoramento do usuário, que **relata os comportamentos fora do esperado à equipe de desenvolvedores**

5. NÍVEL DO TESTE

Os testes também podem ser classificados pelo seu nível, isso é, eles podem ser classificados conforme os seguintes níveis:

- Componente ou unidade;
- Integração;
- Validação;
- Sistema.

Os testes ao nível de Componente ou Unidade são focados em pequenas partes do software, segundo o que foi definido na codificação do software. Este tipo de teste serve para validar pequenas partes do software com base em suas entradas possíveis e saídas esperadas.

Os testes de integração têm por objetivo unir os diversos módulos do sistema e testá-los em conjunto, são realizados assim que são finalizados os testes de cada uma das partes.

Os testes de validação, como o próprio nome sugere, validam os requisitos definidos pelo cliente. Este teste ocorre após o sistema ter sido completado e terem sido realizados os outros tipos de testes.

Nos testes de sistema são realizados os testes de segurança e recuperação, o que visa garantir a segurança do sistema e a proteção dos dados nele armazenados.

Para fins de resumo, seguem abaixo os testes abordados nesta aula e suas classificações por nível.

Testes de Componentes ou Unidade:

- Teste de Funcionalidade;
- Teste de Interface;
- Teste de Conteúdo da Informação;
- Teste de Desempenho;
- Teste de Carga.

Testes de Integração:

- Teste de Regressão.

Testes de Validação:

- Teste Alfa;

- Teste Beta

Testes de Sistema:

- Teste de Desempenho;
- Teste de Carga;
- Teste de Recuperação;
- Teste de Segurança.

Embora tenhamos abordado muitos testes e suas classificações nesta aula, cabe dizer que esse estudo não esgota os muitos tipos de testes que existem, a literatura está repleta de outras formas de se testar software.

CONSIDERAÇÕES FINAIS

Nesta aula você viu a importância da fase de teste de software como disciplina de Engenharia de Software. Esta fase resultará na aprovação do software produzido, autorizando assim a implantação do mesmo.

Testar software não significa apenas encontrar erros, mas validar requisitos e verificar questões como segurança, desempenho e usabilidade, questões não-funcionais fundamentais em um produto.

Abordamos o ciclo de vida do teste de software, apresentando o que deve ser feito antes e durante o processo de testagem de software.

Também foram discutidas, amplamente, as classificações dos testes, que podem ser de acordo com técnicas, tipos e níveis, apresentando os objetivos destas classificações e dos próprios testes.

MATERIAIS COMPLEMENTARES

Link: *Técnicas e Fundamentos de Testes de Software?* Disponível em: <https://www.devmedia.com.br/guia/tecnicas-e-fundamentos-de-testes-de-software/34403>. Acesso em: 05 dez. 2022.

Vídeo: *Entenda os 7 princípios do Teste de Software que todo engenheiro de software deve saber.* Disponível em: <https://www.youtube.com/watch?v=FCMYM01Wflc>. Acesso em: 05 dez. 2022.

Vídeo: *Níveis e Técnicas de Teste. Entenda unidade, integração, sistema, regressão, funcional e estrutural.* Disponível em: <https://www.youtube.com/watch?v=5V012tGZot4>. Acesso em: 05 dez. 2022.

REFERÊNCIAS

PRESSMAN, R.G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison – Wesley, 2019.

O conteúdo desta obra é eletrônico e não pode ser reproduzido ou transmitido por qualquer meio, sem a expressa autorização da Gran Faculdade, sob pena de responsabilização civil e criminal.