



**GESTÃO E QUALIDADE DE SOFTWARE - ATIVIDADE 01 -
AULA 03**

Integrantes do grupo:

Gabriel Viegas Capecchi – 82213442

Gabriel Mariotti Higa – 822141216

Gabriela Dardis Rodrigues – 822141330

Igor Britto - 822141647

Maria Fernanda Mendes Tobias – 822137255

Renato Peduto Filho - 822126254

**São Paulo
2025**

Conceito de Teste

Os testes de software são essenciais para garantir que um sistema funcione corretamente, prevenindo bugs, reduzindo custos de desenvolvimento e melhorando o desempenho.

A abordagem contínua, iniciada na fase de design e mantida ao longo do desenvolvimento e produção, permite identificar e corrigir falhas mais cedo, sem esperar pela conclusão de todas as partes do software.

Entre os principais tipos de testes, destacam-se:

- Testes unitários: verificam componentes individuais.
- Testes de integração: analisam a comunicação entre módulos.
- Testes funcionais: validam se os requisitos são atendidos.
- Testes de desempenho: avaliam velocidade e estabilidade.
- Testes de segurança: detectam vulnerabilidades.
- Testes de usabilidade: garantem uma experiência intuitiva ao usuário.

A aplicação eficaz desses testes é fundamental para desenvolver softwares confiáveis e de alta qualidade.

Estratégias de Teste de Software

As estratégias de teste de software consistem em práticas e abordagens adotadas para detectar falhas, validar funcionalidades e garantir que o software atinja os requisitos especificados.

Uma boa estratégia de teste contém diversos tipos de testes como:

- Teste funcionais: Verificam se cada função do software opera conforme os requisitos (Teste de Unidade, Teste de Integração, Teste de Sistema);
- Teste não funcionais: Examinam os requisitos não funcionais do software, os seus aspectos (Teste de Desempenho, Teste de Usabilidade, Teste de Segurança);

- Teste Automatizados: Permitem a execução repetitiva e eficiente de casos de teste, automatizando os testes. (Teste de Regressão, Teste de Interface de Usuário);
- Teste de Integração contínua: São testes automatizados realizados em um novo código desenvolvido com o código já existente, garantindo sua funcionalidade conforme novos componentes são adicionados;
- Teste de aceitação pelo usuário: Realizados pelos próprios usuários finais para verificar se o software atende às suas expectativas;

As estratégias de testes são importantes para mitigar riscos e falhas no produto final, assegurando que o software seja de alta qualidade e se enquadre com as expectativas e necessidades do cliente.

Conceitos de Verificação e Validação

A verificação trata de garantir que o produto está sendo desenvolvido conforme o planejado, ou seja, ela busca responder se estamos construindo o produto da maneira correta. Esse processo foca na conformidade com as especificações e requisitos técnicos definidos no início do projeto. Ele inclui atividades como revisão de código, análise de artefatos de desenvolvimento (documentos, diagramas, etc.), e testes internos para verificar se o software está sendo construído corretamente e de maneira consistente, sem falhas ou omissões.

A Validação tem um alcance mais amplo e busca responder se estamos construindo o produto correto. Se concentra em avaliar se o software, além de seguir as especificações, realmente atende às necessidades dos usuários e partes interessadas. Em muitos casos, isso implica não apenas em verificar se o software faz o que foi solicitado, mas se ele resolve os problemas reais e atende às expectativas das partes interessadas de forma eficaz. Esse processo pode envolver interação direta com os clientes, testes de usabilidade e até mesmo ajustes nas funcionalidades conforme surgem novas necessidades. A validação frequentemente exige uma análise mais profunda e consultoria com o cliente, pois muitas vezes as necessidades reais não estão completamente explícitas nas especificações iniciais.

Embora esses dois processos se complementam, a verificação é mais focada na qualidade interna do software e se está sendo construído corretamente conforme as diretrizes, enquanto

a validação foca na adaptação do produto ao mundo real, ou seja, se ele cumpre o propósito para o qual foi criado, atende às expectativas do usuário final e as demandas do mercado. Ambas as práticas são fundamentais para o sucesso do software e devem ser aplicadas de forma complementar durante todo o ciclo de desenvolvimento, a fim de garantir que o produto final seja correto, confiável e eficaz.

Teste de Software, Teste Unitário, Teste de Integração

Todo software desenvolvido passa por diversas etapas, uma delas é o teste de software, essa etapa é essencial no desenvolvimento de sistemas, ela é a responsável por identificar e corrigir possíveis falhas, garantindo assim que o software opere da forma desejada e atenda a necessidade do cliente. Para isso possuímos alguns tipos de teste, por exemplo, o teste unitário e o teste de integração.

O teste unitário vai se concentrar na verificação de partes individuais do código, podendo ser alguns métodos, funções, funcionalidades ou classes, mas testando tudo de forma isolada.

Dessa forma, esses diversos testes isolados garantem que cada parte do código pode operar de maneira independente. Já o Teste de Integração, verifica a interação entre diferentes componentes do sistema. Tendo como objetivo identificar as falhas na comunicação ou na combinação das partes, para verificar se funcionam de forma harmônica.

Teste de Validação, Teste de Sistema, e Depuração

1. Teste de Validação

O teste de validação verifica se o software atende às expectativas e necessidades do usuário final.

- 1) Objetivo: Confirmar que o produto atende aos requisitos funcionais e não funcionais.
- 2) Técnicas utilizadas:
 - a) Teste de Aceitação do Usuário (UAT - User Acceptance Testing);

- b) Teste de Requisitos;
 - c) Teste Funcional.
- 3) Resultado esperado: Um software aprovado na validação está pronto para ser entregue ao cliente ou usuário final. Caso falhe, os desenvolvedores devem corrigir os problemas identificados.

2. Teste de Sistema

O teste de sistema avalia o software como um todo, verificando se todos os módulos funcionam corretamente quando integrados.

- 1) Objetivo: Testar a interação entre os componentes do sistema e validar requisitos técnicos e funcionais.
 - a) Técnicas utilizadas:
 - b) Testes de Integração
 - c) Testes de Performance (carga, estresse)
 - d) Testes de Segurança
- 2) Resultado esperado: Se o sistema passar nos testes, ele está apto para a validação final. Caso contrário, os problemas encontrados precisam ser corrigidos antes da liberação para os usuários.

3. Depuração (Debugging)

A depuração não é um teste propriamente dito, mas sim o processo de identificação, análise e correção de defeitos no software.

- 1) Objetivo: Encontrar e corrigir erros no código.
- 2) Ferramentas utilizadas:
 - a) Depuradores (Debuggers) como GDB, Visual Studio Debugger, Chrome DevTools
 - b) Logs e rastreamento (logging)
 - c) Testes unitários para isolar erros

- 3) Resultado esperado: O erro deve ser identificado e corrigido, garantindo que o software funcione corretamente.

Referências

IBM. Teste de software. Disponível em:

<<https://www.ibm.com/br-pt/topics/software-testing>>.

CRISTINA, J. Estratégias de testes: qualidade e confiabilidade. Disponível em:

<<https://www.objective.com.br/insights/estrategia-testes/>>. Acesso em: 1 abr. 2025.

PROF GILLEANES GUEDES ENGENHARIA DE SOFTWARE E UML. Introdução a Verificação e Validação de Software. Disponível em:

<<https://www.youtube.com/watch?v=XNS1zq0jHcg>>. Acesso em: 1 abr. 2025.

Testes Funcionais de Software. Disponível em:

<<https://www.devmedia.com.br/testes-funcionais-de-software/23565>>.

ATLASSIAN. Os diferentes tipos de testes em software. Disponível em:

<<https://www.atlassian.com/br/continuous-delivery/software-testing/types-of-software-testing>>.

MINDS, L. Saiba a diferença entre Teste, Validação e Verificação de Software. Disponível em:

<<https://www.logicalminds.com.br/saiba-a-diferenca-entre-teste-validacao-e-verificacao-de-software/>>.

Engenharia de Software e Sistemas - IF682 2011.1. Teste de software. Teste de sistema e componentes. Disponível em:

<https://cin.ufpe.br/~if682/20111/slidesAulas/18_TestSoftware.pdf>. Acesso em: 1 abr. 2025.

LEE, G. Tipos de testes de software: diferenças e exemplos. Disponível em:

<<https://www.loadview-testing.com/pt-br/blog/tipos-de-testes-de-software-diferencas-e-exemplos/>>.

IBM. Teste de software. Disponível em:

<<https://www.ibm.com/br-pt/topics/software-testing>>.

VIEIRA, H. Testes de integração: o que é, quais os tipos e como automatizar? Disponível em:

<<https://www.objective.com.br/insights/teste-de-integracao/>>.

Testes Unitários: Tutorial Sobre O Que é, Como Fazer + Ferramentas A Utilizar. Disponível em: <<https://www.nimblework.com/pt-br/agile/testes-unitarios/>>.

