

Testes automatizados

11/08/2025

Na Aula de hoje

- Compreender o processo de teste de software conforme a norma ISO/IEC/IEEE 29119-2.
- Refletir sobre o momento adequado para testar ("Quando testar?").

🔍 Quando testar?

Quando vocês acham que o teste de software deve começar? No fim do desenvolvimento? Durante? No início?

Por que testar cedo?

- Erros são mais baratos de corrigir quando descobertos cedo
- Defeitos encontrados tardiamente impactam custos, prazos e qualidade
- Testar desde o início ajuda a **prevenir falhas** ao invés de apenas **corrigir**

💡 Custo de correção ↑ quanto mais tarde o defeito é detectado

ISO/IEC/IEEE 29119

É uma **norma internacional** que define um **modelo genérico e padronizado** para o processo de teste de software.

- As **etapas formais** do processo de teste (planejamento, análise, execução, encerramento);
- Os **documentos e artefatos** envolvidos (como planos e relatórios de teste);
- Uma **visão mais tradicional e controlada**, geralmente aplicada em ambientes com processos bem definidos (por exemplo, empresas que seguem modelos como CMMI ou que trabalham com sistemas críticos).

Sigla	Representa	Área
ISO	Organização Internacional de Padronização	Normas globais (geral)
IEC	Comissão Eletrotécnica Internacional	Normas técnicas (eletrônica e TI)
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos	Padrões técnicos (principalmente TI)

A norma **ISO/IEC/IEEE 29119** foi criada para ser um **referencial global em teste de software**.

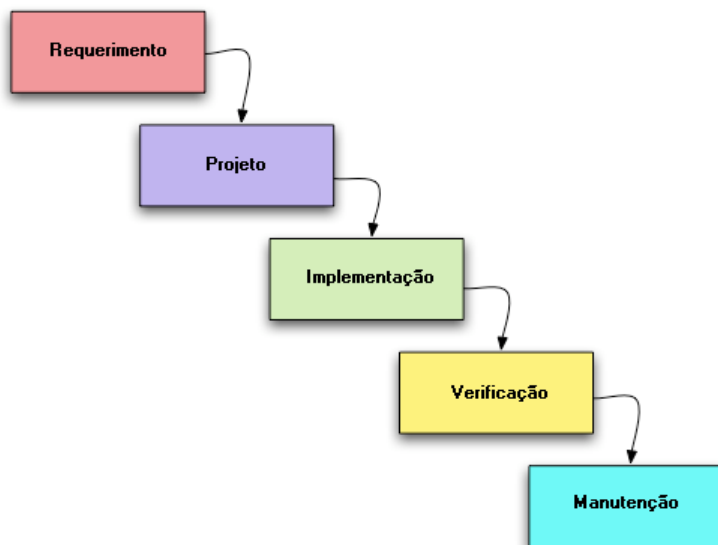
Processo de Teste

de Software (ISO 29119-2)

A norma ISO/IEC/IEEE 29119-2 define um processo padronizado que inclui (modelo cascata):

1. Planejamento de Teste
 2. Controle de Teste
 3. Análise e Projeto de Teste
 4. Implementação e Execução
 5. Avaliação dos Critérios de Saída
 6. Encerramento de Teste
-

Modelo Cascata



Distribuição dos Testes

Nível	Quando ocorre?	Quem realiza?
Unidade	Durante o desenvolvimento	Desenvolvedores
Integração	Após testes de unidade	Devs/testadores
Sistema	Próximo à entrega	QA/Testadores

Revisitando a Pergunta:

Quando Testar?

A resposta certa: **Sempre que possível.**

- Desde o início do projeto
 - Durante o desenvolvimento
 - Após cada integração
 - Antes da entrega ao cliente
-

Modelo Ágil

Refere-se a uma **filosofia de desenvolvimento** (ex: Scrum, XP, Kanban), com foco em:

- **Iterações curtas**, entregas contínuas;
 - **Colaboração** entre equipes;
 - **Adaptação rápida** a mudanças.
No contexto ágil, os testes:
 - São realizados **desde o início e continuamente**;
 - Frequentemente **automatizados**;
 - Com pouco ou nenhum uso de documentação formal tradicional;
 - Focados em práticas como **TDD**, **BDD**, **integração contínua**.-
-

Modelo Ágil

- Surge no começo dos anos 2000 com a insatisfação com os modelos formais
 - Os modelos existentes não levavam em consideração mudanças constante nos requisitos
 - Eram muito burocráticos e não tinham flexibilidade
 - Os testes só eram feitos os testes ao final do processo de desenvolvimento.
-

Modelo Ágil

Em 2001 foi criado o Manifesto ágil, objetivo: Criar uma abordagem comum para desenvolvimento de software que fosse **mais leve, adaptável e centrada nas pessoas**.

Os 4 valores do Manifesto Ágil

1. **Indivíduos e interações** mais que processos e ferramentas
 2. **Software funcionando** mais que documentação abrangente
 3. **Colaboração com o cliente** mais que negociação de contratos
 4. **Responder a mudanças** mais que seguir um plano
-

Impacto do Modelo Ágil

- Reduziu burocracia
 - Acelerou entregas
 - Melhorou a comunicação com clientes
 - Tornou o processo **mais iterativo, incremental e adaptável**
-

Como o Ágil afeta os teste?

- Os testes passam a ser feitos **desde o início**
 - Foco em **automação** e **feedback rápido**
 - Práticas como **TDD** (Test Driven Development) ganham espaço
 - Os papéis de “dev” e “QA” se aproximam
-

Testes no desenvolvimento ágil:

- **Testes automatizados** desde o início
- **Test Driven Development (TDD)**
- **Integração contínua** com testes automáticos

- Desenvolvedores e testadores trabalham juntos

Inovações no Teste Ágil

- Testes como parte do ciclo de entrega contínua
 - Uso de ferramentas de **CI/CD**
 - **Feedback rápido** sobre qualidade
 - Redução de retrabalho
-

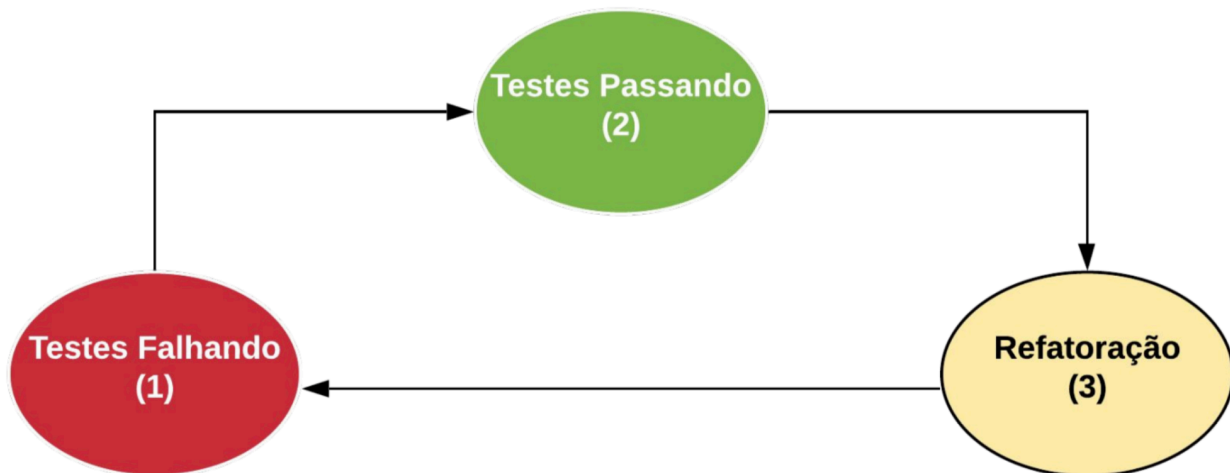
Test Driven Development

TDD (Desenvolvimento Dirigido por Testes): Escrever os testes antes de escrever o código, proposto pelo Extreme Programming.

Benefícios TDD

- Evita que os devs esqueçam de escrever os testes
 - Incentiva a escrita de código com testabilidade
 - Melhora o design do código, pois o dev passa a ser o primeiro cliente do seu código
 - Cobertura pode chegar a 90%
-

Test Driven Development



Resumo final

- Testes devem ser feitos sempre que possível e o quanto antes
 - No desenvolvimento de cada feature
 - Quando algo específico quebra
 - Os testes ocorrem em diferentes níveis ao longo do ciclo de desenvolvimento
 - Métodos ágeis integram o teste ao fluxo contínuo de entrega
-

Atividade

- Pesquise sobre as diferenças entre testes ágeis e testes da padronização ISO/IEEE e levante 5 motivos pelos quais se usa testes ágeis hoje em dia.
- Pesquise sobre outras filosofias de Testes (assim como o TDD existem outras?)