Testes ao Longo do Ciclo de Vida de Desenvolvimento de Software

Introdução:

O SDLC descreve as etapas de um projeto de desenvolvimento de software, organizadas logicamente e cronologicamente:

- Análise
- Design
- Desenvolvimento
- Testes
- Entrega
- Manutenção

Modelos de Ciclo de Vida:

- Sequenciais: Modelo em Cascata, Modelo em V
- Iterativos e Incrementais: Espiral, Métodos Ágeis (Scrum, Kanban etc.)

Testes no Contexto do SDLC

Princípios Gerais

- Os testes devem começar desde os estágios iniciais (princípio do teste antecipado ou abordagem *shift-left*).
- Para cada atividade de desenvolvimento, há uma atividade de teste correspondente.
- Diferentes níveis de teste têm objetivos específicos, evitando redundância.

Impacto do Modelo de SDLC nos Testes

Modelos Sequenciais

- Testes dinâmicos só ocorrem após o desenvolvimento.
- Envolvimento dos testadores nas revisões de requisitos e projeto de testes.

Modelos Iterativos e Incrementais

- Permite testes estáticos e dinâmicos em cada iteração.
- Necessidade de feedback rápido e testes de regressão extensivos.

Desenvolvimento Ágil

- Mudanças são constantes.
- Menos documentação e maior uso de testes baseados em experiência.
- Extensa automação de testes.

Abordagens Orientadas a Testes

TDD (Test Driven Development)

- Testes escritos antes do código.
- Código é implementado para satisfazer os testes.
- Envolve refatoração contínua.

ATDD (Acceptance Test Driven Development)

- Baseia-se nos critérios de aceite.
- Testes escritos antes do desenvolvimento do recurso correspondente.

BDD (Behavior Driven Development)

- Usa linguagem natural (Dado/Quando/Então).
- Foco no comportamento esperado do sistema.

DevOps e Testes

Visão Geral

DevOps promove sinergia entre desenvolvimento (incluindo testes) e operações.

Práticas técnicas:

- Integração Contínua (CI)
- Entrega Contínua (CD)

Benefícios do DevOps para os Testes

- Feedback rápido.
- Encoraja envio de código com testes.
- Ambientes de teste estáveis.
- Alta automação reduz testes manuais repetitivos.
- Aumenta a visibilidade sobre qualidades não funcionais.

Desafios e Riscos

- Exige automação robusta e bem mantida.
- Ferramentas de CI/CD precisam ser definidas.
- Necessita recursos e investimento inicial.
- Importante: Testes manuais continuam necessários, especialmente com foco na experiência do usuário.

Abordagem Shift-Left

- Testes iniciam nas fases iniciais do SDLC (à esquerda do ciclo).
- Não substitui testes nas fases finais, mas os complementa.
- Detecta defeitos mais cedo, otimizando o processo geral.

Boas práticas:

- Revisar especificações com foco em testes.
- Escrever testes antes do código (TDD, ATDD, BDD).
- Uso de CI/CD com testes automatizados.
- Incluir análise estática e testes não funcionais precoces.

Retrospectivas e Melhorias Contínuas

- Reuniões pós-projeto ou por iteração para revisar o que funcionou ou não.
- Envolvem toda a equipe (testadores, devs, PO, arquitetos etc.).
- Resultados documentados e integrados à melhoria contínua.

Benefícios:

- Maior eficácia dos testes.
- Melhoria na documentação e comunicação.
- Maior cooperação entre áreas.

Níveis de Teste

Teste de Componente

- Unidade
- **Realizado por:** Desenvolvedor

- **Características:**
  - Isolado de outros componentes
  - Automatizado na maioria das vezes
  - Requer acesso ao código-fonte
  - Verifica se a unidade faz exatamente o que deveria

Teste de Integração

- **Realizado por:** Desenvolvedor ou testador
- **Foco:** Verificar a **interação entre componentes** ou **entre sistemas**
- **Características:**
  - Requer acesso ao código-fonte
  - Pode ser feito de forma incremental (adicionando componentes aos poucos)
  - Detecta falhas na comunicação entre diferentes partes do sistema

---

Teste de Sistema

- **Realizado por:** Equipe de QA ou testadores especializados
- **Foco:** Validar o sistema **como um todo**, com todos os módulos integrados
- **Características:**
  - Testa **fluxos completos**, de ponta a ponta
  - Verifica se os requisitos funcionais e não funcionais são atendidos
  - Pode incluir validações de conformidade legal e regulatória

---

Teste de Aceite

- **Foco:** Garantir que o sistema esteja funcionando **como esperado pelo usuário final**
- **Realizado por:** Cliente, usuário final ou testadores com foco no negócio
- **Características:**
  - Baseado em critérios de aceitação definidos nos requisitos
  - Testa o sistema em **ambientes reais ou simulados**
  - Normalmente feito antes da entrega final do software

Aceite de Usuário (UAT)

- Validar o uso do sistema por usuários em um ambiente de produção ou simulado.
- Foco em validar se o sistema atende às **necessidades reais do usuário**

Aceite Operacional (OAT)

- Verifica se o sistema funciona corretamente em **ambientes de produção**
- Foca em aspectos como desempenho, segurança e confiabilidade

---

Aceite Contratual e Normativo

- Garante que o software esteja em conformidade com:
  - **Contratos firmados**

- **Normas legais ou regulamentares específicas**

Alfa

- Realizado **dentro da organização desenvolvedora** (no local)
- Pode ser feito por:
  - Clientes
  - Operadores
  - Testadores
- Ambiente controlado, com suporte técnico próximo

Beta

- Realizado **fora da organização desenvolvedora** (em ambiente do cliente ou em campo)
- Pode ser feito por:
  - Clientes em potencial
  - Operadores reais
- Serve como prévia do uso real antes do lançamento oficial

Tipos de Teste

Teste Funcional

- Avalia o que o sistema faz.
- Realizado em todos os níveis.
- Baseado nas regras de negócio.

Teste Não Funcional

- Avalia como o sistema se comporta.
- Ex: desempenho, segurança, usabilidade.

Teste de Caixa Preta

- Baseado em especificações.
- Não exige conhecimento do código.
- Verifica o comportamento esperado.

Teste de Caixa Branca

- Baseado na estrutura interna (código).
- Requer conhecimento técnico.
- Avalia a cobertura estrutural.

Teste de Confirmação

- Retestar testes que falharam devido a um defeito
- Verifica se o defeito foi resolvido.

Teste de Regressão

- Garante que funcionalidades existentes continuam funcionando.
- Detecta efeitos colaterais.
- Foco em automação contínua.

Teste de Manutenção

Aplicado após a implantação para:

- Corrigir defeitos
- Adicionar, alterar ou remover funcionalidades
- Manter performance e confiabilidade

Escopo depende de:

- Risco da mudança
- Tamanho do sistema

Tipos de teste de manutenção:

- Modificação: atualizações, correções, melhorias
- Atualização/Migração: mudanças de plataforma
- Aposentadoria: fim da vida útil do sistema

Inclui:

- Avaliação da mudança
- Testes de regressão