



Análise de falhas

Conceitos Principais:

- Ponte: Falha → Causa → Prevenção
- Por que estudar: Reduzir custos, reputação, segurança, impacto social
- Custo relativo: Análise (0,1x), Design (0,5x), Implementação (2x), Testes (5x), Validação (20x), Manutenção (100x)

Classificação das Falhas:

- Requisitos
- Design/Projeto/Arquitetura
- Codificação
- Integração
- Manutenção



Gestão e qualidade de software



Qualidade - ISO 9126

Seis Características Principais:

- Funcionalidade: Adequação, Precisão, Interoperabilidade, Segurança
- Confiabilidade: Maturidade, Tolerância a falhas, Recuperabilidade
- Usabilidade: ,Apreensibilidade, Operacionalidade, Atratividade
- Eficiência: Tempo de resposta, Throughput, Recursos utilizados
- Manutenibilidade: Analisabilidade, Modificabilidade, Estabilidade
- Portabilidade: Adaptabilidade, Instalabilidade, Conformidade

Exemplos de Métricas:

- Taxa de falhas por mil horas
- Tempo médio de recuperação (MTTR)
- % de usuários que entendem interface
- Tempo médio de execução
- % de cobertura de testes



Testes unitários

Características:

- Rápido: executa em milissegundos
- Automático: sem intervenção manual
- Repetível: mesmo resultado sempre
- Independente: não depende de outros testes

Estrutura (4 Passos):

- 1.Preparar dados de entrada
- 2.Executar função/método
- 3.Verificar resultado esperado vs obtido
- 4.Limpar se necessário

Benefícios:

- Reduz custo de correção
- Facilita manutenção e refatoração
- Documenta comportamento esperado



TDD

Ciclo RED-GREEN-REFACTOR:

- 1.RED: Escrever teste que falha (código não existe ainda)
- 2.GREEN: Código mínimo para passar no teste
- 3.REFACTOR: Melhorar código mantendo testes verdes

Benefícios TDD:

- Qualidade: código testado desde o início
- Segurança: mudanças validadas por testes
- Redução de retrabalho: erros detectados cedo
- Entrega mais ágil: feedback imediato



Versionamento

Conceitos Git Fundamentais:

- Repositório: base de código versionada
- Commit: registro de alteração com autor, data, mensagem
- Branch: linha paralela de desenvolvimento
- Merge: união de branches

Fluxos de Trabalho (Workflows):

- Git Flow: feature, develop, release, hotfix branches
- GitHub Flow: main + pull requests
- Trunk-Based: commits pequenos direto na main

Branching Strategies:

- Feature Branch: nova funcionalidade isolada
- Release Branch: preparação para entrega
- Hotfix Branch: correção rápida em produção