UNIVERSIDADE SÃO JUDAS TADEU

BACHARELADO EM SISTEMA DE INFORMAÇÃO

Geovane Augusto Costa dos Santos 824124157
Natan Fernandes Araujo Ibiapina 821225920
Gabriela Alves Rodrigues 82311687
Olivia Frankiw de Carvalho 824118841
João Luiz Santana Borean 82415181

CONCEITOS DE TÉCNICAS DE REVISÃO DE SOFTWARE

São Paulo

2025

Sumario

Técnicas de Software: Uma Visão Geral

As técnicas de software são métodos, práticas e ferramentas utilizadas no desenvolvimento de sistemas para garantir que o produto final seja eficiente, de qualidade e bem organizado. Segundo **Ian Sommerville** e **Roger Pressman**, essas técnicas são essenciais para transformar requisitos abstratos em um produto funcional e confiável.

1. Técnicas de Processo (Sommerville)

Sommerville destaca que a escolha de **modelos de processo** influencia diretamente a forma como o software será desenvolvido. Alguns exemplos são:

- **Modelo em Cascata**: Um processo sequencial (com fases como levantamento de requisitos, design, codificação e testes), que exige documentação detalhada (SOMMERVILLE, 2015, p. 52).
- **Desenvolvimento Ágil**: Um processo baseado em ciclos curtos (chamados de sprints), com ênfase em feedback constante do usuário, como no exemplo do **Scrum** (SOMMERVILLE, 2015, p. 68).
- **Prototipação**: Consiste na criação de versões simplificadas do sistema para testar e validar ideias antes de seu desenvolvimento completo.

2. Técnicas de Desenvolvimento (Pressman)

Pressman enfatiza técnicas que auxiliam na codificação e na gestão de projetos:

- **Engenharia de Requisitos**: Utiliza entrevistas, workshops e diagramas de casos de uso para entender e documentar as necessidades do cliente (PRESSMAN, 2016, p. 122).
- **Orientação a Objetos**: Foca na organização do código através do uso de classes, herança e encapsulamento (PRESSMAN, 2016, p. 210).
- **Padrões de Projeto (Design Patterns)**: São soluções reutilizáveis para problemas comuns encontrados durante o desenvolvimento de software (por exemplo, **Singleton** e **Observer**).

3. Técnicas de Qualidade (Ambos os Autores)

A qualidade do software é garantida através de várias práticas:

- **Revisões de Código**: Análises em grupo do código para identificar erros e falhas antes dos testes formais (PRESSMAN, 2016, p. 381).
- **Testes Automatizados**: Ferramentas como JUnit e Selenium são utilizadas para testar automaticamente as funcionalidades do sistema (SOMMERVILLE, 2015, p. 220).

- **Refatoração**: Processo de reestruturar o código para melhorar sua legibilidade e organização, sem alterar seu comportamento ou funcionalidade (PRESSMAN, 2016, p. 450).

Referências:

auditeste.com.br
Aspectos negativos gerados pela falta de qualidade de software
1 de maio de 2024 — Exploração dos custos ocultos da baixa qualidade de software: retrabalho, suporte técnico e insatisfação do cliente. · Retrabalho: a necessidade ...

Codurance
Por que o gerenciamento de riscos de software é importante?
5 de abril de 2023 — Os riscos comumente associados podem ser conformidade com regulamentos, certificações, problemas de produção, e proteção de dados, os quais ...
https://d3.works/boas-praticas-de-desenvolvimento-de-software-e-seguranca/

https://www.splunk.com/en_us/blog/learn/software-liability.html

Pressman Bruce R. Maxim. Engenharia de Software. 9ª edição

Sommerville, Ian (2015). Engenharia de Software (10ª edição).
- Capítulo 4: "Requisitos de Software" (p. 98-120): Discussão sobre priorização de requisitos essenciais.
- Capítulo 24: "Gerenciamento de Qualidade" (p. 654-670): Conceito de equilíbrio entre qualidade e restrições práticas.

Pressman, Roger S. (2016). Engenharia de Software: Uma Abordagem Profissional (8ª edição).
- Capítulo 1: "A Natureza do Software" (p. 20-22): Crítica ao over-engineering.
- Capítulo 14: "Garantia de Qualidade de Software" (p. 381-400): Importância de testes focados no essencial.