

# Técnicas de revisão de software

WELLERSON RESENDE MONTEIRO | RA: 8222243349

PALOMA LOPES DE SOUSA | RA: 822167506

LUCAS VASCONCELLOS RAMOS DE SOUSA | RA: 8222242709

GABRIEL NEULES GOMES RODRIGUES SOARES | RA: 822167394

A revisão de software é uma prática fundamental no ciclo de desenvolvimento de software, que visa garantir a qualidade do produto final e a eficiência do processo de desenvolvimento. Ela envolve a análise crítica de artefatos do software, como código-fonte, documentação e requisitos, com o objetivo de identificar falhas, erros e oportunidades de melhoria. Essa atividade pode ser realizada de diversas formas, incluindo revisões informais entre pares, revisões formais com checklists e auditorias técnicas.

## **Importância da Revisão de Software**

A revisão de software é essencial para detectar defeitos de forma precoce no processo de desenvolvimento, o que reduz custos e aumenta a eficiência. Segundo *Brooks* (1975), "mais barato do que corrigir falhas no final do processo de desenvolvimento é preveni-las durante a fase de design e codificação". A identificação de problemas em etapas iniciais do desenvolvimento facilita sua correção e evita retrabalho dispendioso. Além disso, revisões ajudam a garantir que o software atenda aos requisitos de qualidade, desempenho e segurança, fundamentais para a satisfação dos usuários e a continuidade do sucesso do projeto.

## **Tipos de Revisão de Software**

Existem diferentes tipos de revisão de software que variam em termos de formalidade e estrutura. Dentre as mais comuns estão:

1. **Revisões informais:** São realizadas de maneira mais casual e geralmente envolvem a troca de feedback entre desenvolvedores. Essas revisões podem ser feitas de forma espontânea, com base na experiência e julgamento dos envolvidos (Fagan, 1976).
2. **Inspeções:** Esse é um processo formal, estruturado, no qual um time de revisores examina o artefato de software em busca de defeitos. As inspeções, conforme *Fagan* (1976), têm um grande poder na redução de defeitos, já que o foco é encontrar problemas de forma eficaz e com alta taxa de detecção.

3. **Walkthroughs:** São reuniões mais colaborativas onde um autor apresenta seu trabalho e o time discute possíveis melhorias. Embora não seja tão rigoroso quanto uma inspeção, o walkthrough permite o envolvimento direto de diferentes membros da equipe no processo de revisão.
4. **Auditorias técnicas:** Envolvem uma análise minuciosa e sistemática dos artefatos do software para garantir conformidade com padrões e regulamentos específicos, especialmente em contextos onde a segurança e a confiabilidade são prioritárias (Chulani et al., 2003).

## Benefícios da Revisão de Software

A revisão de software oferece diversos benefícios, como a melhoria da qualidade do código, a detecção precoce de erros e o aprimoramento contínuo do processo de desenvolvimento. *Gilb* (1988) argumenta que revisões são um meio eficaz de promover a melhoria contínua, pois permitem que os desenvolvedores aprendam com os erros passados e apliquem esse conhecimento em projetos futuros. Além disso, revisões regulares ajudam a manter a consistência do código, reduzem a complexidade e melhoram a manutenção do software.

## Desafios da Revisão de Software

Apesar dos benefícios, a revisão de software também apresenta desafios. Um dos principais problemas é o **custo de tempo**: as revisões podem ser demoradas e consumir recursos significativos, principalmente quando são realizadas de maneira formal. Além disso, há a questão da **resistência dos desenvolvedores**, que muitas vezes veem as revisões como um processo de julgamento ao invés de uma oportunidade de aprendizado (Baker, 2002). A eficácia de uma revisão também depende da **qualidade da comunicação** entre os envolvidos e da capacidade dos revisores de identificar problemas complexos.

## Conclusão

Em suma, a revisão de software é uma prática que contribui significativamente para a qualidade e sustentabilidade do software. Por meio da detecção precoce de erros e do aprimoramento contínuo dos processos, ela pode reduzir os custos e aumentar a confiabilidade do sistema. No entanto, deve ser bem planejada e realizada com foco na colaboração entre os membros da equipe, para que seus benefícios sejam plenamente alcançados.

## Referências

- BAKER, W. H. (2002). *Software Inspections and Reviews: What We Have Learned*. Journal of Software Maintenance and Evolution: Research and Practice, 14(5), 281-293.
- BROOKS, F. P. (1975). *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley.

- CHULANI, S., KAPOOR, M., & RUSSELL, J. (2003). *Software inspections and reviews: An empirical analysis*. IEEE Transactions on Software Engineering, 29(10), 1056-1069.
- FAGAN, M. E. (1976). *Design and Code Inspections to Reduce Errors in Program Development*. IBM Systems Journal, 15(3), 182-211.
- GILB, T. (1988). *Principles of Software Engineering Management*. Addison-Wesley.