

Atividade 1: Técnicas de Revisão de Software

Integrantes:

- RA: 8222243147 - Nome: Bruno Rodrigues Reis - Turma/Curso: Ciência da computação

Revisão de Software - O que é e como funciona?

As técnicas de revisão de software são práticas que ajudam a encontrar e corrigir problemas no código ou em documentos do projeto antes que eles virem dor de cabeça. A ideia é pegar os erros o quanto antes, pra deixar o software o mais redondo possível.

Tipos de Revisão:

Inspeções:

É tipo uma auditoria mesmo, bem formal e detalhada. A equipe se prepara, faz reuniões pra analisar o software, discute os problemas encontrados e depois gera relatórios pra corrigir o que estiver errado.

Revisões Técnicas:

Um pouco menos formal, mas ainda organizada. O objetivo é garantir que o software segue as regras e padrões definidos. Geralmente envolve análise de documentos e testes.

Revisões por Pares (Peer Reviews):

É quando um colega de equipe revisa o trabalho de outro. É uma abordagem colaborativa pra encontrar erros lógicos e garantir que tudo tá seguindo o padrão.

Walkthroughs (Passeios Guiados):

O próprio autor do material vai apresentando o que fez enquanto os outros dão feedback. É mais pra entender e sugerir melhorias do que pra corrigir erros.

Ferramentas:

Para a realização das revisões são é comumente utilizado ferramentas como:

- Github / Gitlab : Ferramentas de controle de versão que permitem Revisões por Pares (Pull Requests / Merge Requests) onde os desenvolvedores revisam o código dos colegas antes de aceitar as mudanças.

- Crucible : Ferramenta dedicada para revisões de código colaborativas e detalhadas.
- Review Board : Ferramenta open-source focada em revisões colaborativas de código. Ajuda a identificar erros e melhorar a qualidade antes da integração.

Vantagens de Revisar o Software:

- Encontrar problemas cedo e evitar gastos futuros;
- Melhorar a qualidade do software final;
- Aprendizado e troca de conhecimento entre o time.

Conclusão:

Cada técnica tem seu jeito de funcionar e se encaixa melhor dependendo do projeto. Mas, no fim das contas, todas ajudam a melhorar o software antes que os problemas se tornem mais difíceis (e caros) de corrigir.