

Resumo: Revisões de código: um guia completo sobre como conduzir uma revisão de código

Uma revisão de código é um ato de verificar conscientemente partes de código em busca de erros e bugs. As revisões de código podem ser executadas por ferramentas/software de revisão de código e por humanos. Quando um desenvolvedor humano executa a revisão de código, é importante que a pessoa que verifica e testa o código não seja a mesma pessoa que escreveu o código em primeiro lugar.

As ferramentas de revisão de código podem dar uma grande ajuda, mas - como acontece em qualquer outro contexto - os humanos sempre podem identificar coisas que as máquinas não percebem. Além disso, ter a chance de ver o trabalho de um colega lhe dá a oportunidade de melhorar suas próprias habilidades e aprender alguns truques. Por outro lado, quando você recebe suas revisões de código de um colega, pode obter feedback valioso e dicas de melhoria.

Todos os desenvolvedores que executam revisões de código usam listas de verificação para verificar e testar todos os aspectos que precisam ser verificados e testados.

Principais etapas da lista de verificação: depurar (depurar e remover todos os problemas é sua prioridade), segurança (isso também é considerado uma prioridade), legibilidade do código (se o código é autoexplicativo, se é claro e conciso e se todas as convenções de linguagem e projeto são seguidas), duplicação de código (força você a revisar o código enquanto se concentra em um único aspecto de cada vez), nomeação (revisar variáveis, constantes, campos de classe, nomes de propriedades, procurar oportunidades para melhorá-los tornando-os mais descritivos), testes (automatizados), documentação (certifique-se de atualizar a documentação), possibilidade de melhoria, acompanhar alterações e fornecer feedback.

Resumo: Code Review – Revisão de código pode ser automatizada!

Travis: utilizando-se de scripts podemos construir no nosso pipeline de entrega contínua, ou de integração contínua.

[ESLint](#), [TSLint](#), [Codelizer](#), [Checkstyle](#), [SonarQube](#): verifica se o código está aderente a convenção.

<https://github.com/marketplace/restyled-io>: Com o auxílio de um `.prettier.json` ou `.brittany.yaml` ele promete abrir PR formatando o seu código.

<https://github.com/marketplace/commit-message-lint> : verifica e formata a mensagem de commit.

[Codefactor](#): certifica que seu código atenda aos padrões de qualidade.

[Codacy](#): é uma ferramenta automatizada de análise / qualidade de código.

[Codebeat](#): ferramenta automatizada de análise de código estático.

[coveralls](#): verifica se a cobertura de teste automatizado mudou.

[Codecov](#): fornece ferramentas altamente integradas para agrupar, mesclar, arquivar e comparar relatórios de cobertura.

[ImgBot](#): verifica se uma imagem pode ser menor.

[Dependabot](#): ferramenta que visa manter as dependências atualizadas.

[Snyk](#): ajuda a encontrar e corrigir vulnerabilidades conhecidas.

[Sonatype DepShield](#) : é capaz de identificar e criar uma issue para resolver problemas de segurança em seu código ou dependências.

[archunit](#) e [sonar](#)qube: revisa a arquitetura de software.