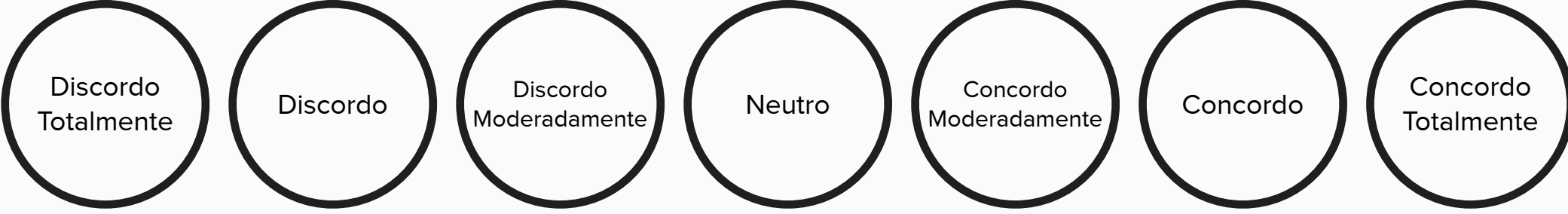


Influência da IC no tempo de revisão

O tempo total de revisão do código aumenta de forma proporcional ao tempo total de execução da IC.

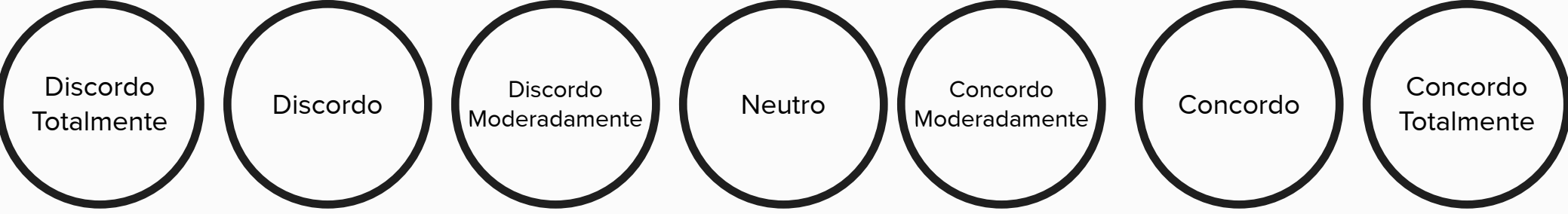


Pontos contrários

Pontos a favor

Influência da IC no tempo de revisão

Quanto mais pipelines de IC são executados mais o tempo de revisão aumenta.



Pontos contrários

Pontos a favor

Efeitos das más práticas de IC na revisão de código

Quais más práticas de IC podem afetar a revisão de código?

5 - A falta de testes de regressão afeta na garantia da qualidade do software, visto que a integração de novas funcionalidades podem afetar no que já estava funcionando antes.

2 - Branches divergentes são especialmente complicados quando os merges acontecem e equipe não possui experiência para resolver os merges adequadamente. Idealmente, os merges não deveriam gerar conflitos.

2 - Branches divergentes acontece muito com a demora dos desenvolvedores atualizarem a suas branches com a padrão.

2 - conflitos de código demandam tempo para resolução.

3 - Discordo. Acredito que o desenvolvedor precisa de fato ter um certo controle do ambiente, mas não precisaria ter o controle total do ambiente.

4 - Concordo que apenas o desenvolvedores que optaram por receber as notificações não seja o ideal. No entanto, acho que essas notificações podem ser enviadas somente para partes interessadas, uma equipe específica, um grupo de desenvolvedores.

5 - Uma pipeline sem testes é um pipeline incompleta.

6 - passos manuais no pipeline podem levar a atrasos, já que dependem de alguém (humano) executar algo.

6 - Passos manuais são passos feitos para serem esquecidos. Além disso, a integração fica parcialmente contínua.

7 - Outra má prática clássica. Um ambiente de testes similar ao de produção evita que alguns problemas ocorram somente em produção.

A má prática 9 é bem parecida com a 7.

12 - scripts longos ou com tarefas demoradas podem ser lentos e atrasar o processo.

10 - É uma falha comum manter as configurações padrões que nem sempre atendem as demandas do time.

12 - Quando o script impede a continuidade do trabalho, essa má prática se torna ainda mais graves.

13 e 10 são bem parecidas. É preciso estar

- 1. Feature branches are used instead of feature toggles:**
Esta prática envolve a criação de ramos separados (feature branches) para desenvolver novas funcionalidades em vez de usar toggles (interruptores) de funcionalidade. Os toggles permitem que as funcionalidades sejam desenvolvidas e ocultas até estarem prontas para serem ativadas, enquanto os ramos de funcionalidade podem levar a problemas de integração e dificuldades na entrega contínua.

2. Divergent Branches:
Isso ocorre quando os ramos do código se afastam demais uns dos outros e se tornam difíceis de serem mesclados novamente. Divergent branches podem causar conflitos de mesclagem significativos e tornar a manutenção do código mais difícil.

3. Developers do not have complete control of the environment:
Isso significa que os desenvolvedores não têm controle total sobre o ambiente em que estão trabalhando, o que pode levar a problemas de compatibilidade e dificultar a reprodução de problemas em ambientes de desenvolvimento.

4. Failures notifications are only sent to teams/developers that explicitly subscribed:
Se apenas as equipes ou desenvolvedores que se inscreveram para receber notificações são informados sobre falhas na integração contínua, isso pode resultar em atrasos na resolução de problemas, já que outros envolvidos podem não estar cientes dos problemas.

5. Missing tests on feature branches:
Desenvolver funcionalidades em ramos separados sem escrever testes adequados pode levar a problemas de qualidade, já que não há garantia de que as novas funcionalidades funcionarão corretamente e não afetarão negativamente o código existente.

6. Some pipeline's tasks are started manually:
Tarefas que são iniciadas manualmente em um pipeline de integração contínua podem levar a inconsistências e atrasos, além de aumentar o risco de erros humanos.

7. Lack of testing in a production-like environment:
Não realizar testes em um ambiente semelhante ao de produção pode resultar em problemas que não são detectados até que a funcionalidade seja implantada no ambiente de produção real.

8. Test cases are not organized in folders based on their purposes:
A organização inadequada dos casos de teste pode tornar difícil para os desenvolvedores e testadores encontrar e gerenciar os testes necessários, tornando o processo de teste menos eficiente.

9. The CI server hardware is used for different purposes other than running the CI framework:
Usar o hardware do servidor de integração contínua para outros fins pode afetar o desempenho e a confiabilidade do processo de integração contínua.

10. External tools are used with their default configurations:
Usar ferramentas externas com configurações padrão pode não atender às necessidades específicas do projeto, resultando em subutilização das ferramentas e funcionalidades não aproveitadas.

11. A build fails because of some flakiness in the execution, whereas it should not:
Flakiness refere-se a instabilidade nos testes ou na construção do código. Falhas intermitentes podem ser frustrantes e difíceis de depurar, prejudicando a confiabilidade do processo de integração contínua.

12. Lengthy build scripts:
Scripts de construção longos podem tornar o processo de compilação demorado e complexo, afetando o tempo de entrega e a manutenção.

13. Quality gates are defined without developers considering only what is dictated by the customer:
Estabelecer critérios de qualidade sem levar em consideração os requisitos de todas as partes envolvidas pode resultar em processos de integração contínua que não atendem às expectativas do cliente ou que são excessivamente rigorosos.

14. Developers and operators are kept as separate roles:
Manter desenvolvedores e operadores como papéis separados pode levar a lacunas de comunicação e atrasos na resolução de problemas, pois os desenvolvedores podem não ter uma compreensão completa das operações em produção e vice-versa.

Benefícios e desafios da IC na revisão de código

Desafios

Benefícios