

PESQUISA – BOAS PRÁTICAS DE VERSIONAMENTO DE CÓDIGOS

OBJETO: Obter o conhecimento sobre boas práticas de desenvolvimento de software que buscam assegurar a qualidade, permitindo a seleção de ferramentas e a aplicação de conceitos de segregação de ambientes, homologação e produção, enfatizando os benefícios de sua utilização.

1. Introdução

A seleção do GIT apresenta-se como alternativa viável e prática para a utilização de conceitos de versionamento, pois seu uso garante a criação de versões da aplicação e o trabalho em equipe, permitindo que um mesmo arquivo seja alterado por mais de uma pessoa ao mesmo tempo. Os ramos (branches) se assemelham a pequenos ambientes, onde o código é replicado e pode ser livremente alterado, sem afetar o código principal, permitindo que futuramente esses ramos sejam mesclados com um ramo principal (main).

2. Vantagens

Rapidez: como os processos são operados localmente, deixa de ser preciso contatar o servidor central para proceder a operações como *commit*, *log* ou *diff*.

Autonomia: permite trabalhar em modo desconectado, em qualquer local, só sendo necessária uma conexão com a rede para troca de revisões com outros repositórios.

Ramos individuais: contrariamente aos chamados sistemas centralizados, combinar ramos não é uma obrigação para cada *commit* e antes uma decisão que depende do programador; o trabalho local faz-se num ramo individual que não tem interferência com os demais ramos, mesmo em processos de sincronização entre repositórios.

Facilidade na Fusão: os DVCS usam o rastreamento automático, o que facilita de forma significativa o processo de fusão.

3. Sugestões de boas práticas

3.1 - Utilizar sempre os repositórios online, evitando a perda de dados;

3.2 - Jamais começar a trabalhar em arquivos locais. Primeiramente devemos sincronizar os arquivos locais com os do repositório remoto, garantindo que estamos atualizados com a equipe;

3.3 - Atualizar com frequência diária (no mínimo) o servidor remoto;

3.4 - Commits

- . Devem ser realizados sempre com mensagens, preferencialmente em inglês,
- . Objetividade e clareza para o entendimento das razões das mudanças aplicadas ao código,
- . Evitar fazer muitos commits sem necessidade;

3.5 - Assegurar compilações sem quebra: Subir apenas o que está funcionando (código completo e funcional, ou um documento revisado);

3.6 - Manter o controle e backup das versões (releases) lançadas, através do uso de TAGS;

4. FONTES:

blog.myscrumhalf.com/algumas-boas-praticas-no-controle-de-versao/

<https://pt.linkedin.com/pulse/boas-pr%C3%A1ticas-de-git-para-beneficiar-o-software-silveira-siqueira>