FACULDADE



GESTÃO EM TECNOLOGIA DA INFORMAÇÃO

###### Lauda

“Controle de Versão de Software”

Alexsander Bitencourt de Lima

Guilherme Caetano da Silva

Goiânia, setembro de 2019

INTRODUTAÇÃO

Controle de versão é uma importante atividade relacionada a muitas fases do ciclo de vida do desenvolvimento de software. Muitos sistemas de controle de versão têm sido desenvolvidos para gerenciar o histórico das versões de software e atividades humanas associadas com a intenção de produzir software de maior qualidade e suportar trabalho em equipe.

Tipos de Controle de versão

SUBVERSION

Você precisa entender os conceitos básicos de controle de versão: arquivo, diretório de trabalho, revisão, changeset, ramos e etiquetas (tags). Também é necessário conhecer as particularidades de implementação do Subversion em relação à ramificação através de diretórios, e à convenção da estrutura de diretórios trunk/branches/tags.

O Subversion é simples e direto. Seu tempo de aprendizado é curto, e a chance de desastres causados por imperícia é pequeno.

MERCURIAL

É um pouco mais complicado que o Subversion porque além dos conceitos de controle de versão centralizado, também é necessário entender conceitos básicos do controle de versão distribuído, tais como sincronização de repositórios e o registro do histórico em um grafo acíclico direcionado.

O Mercurial não possui nenhuma particularidade de implementação que fique exposta na estrutura do projeto ou nos comandos. Mesmo operações avançadas como rebase, por exemplo, exigem apenas conceitos básicos de controle de versão distribuído.

GIT

Tem várias qualidades: é popular principalmente entre projetos open source, é rápido (até um certo tamanho de projeto) e tem um bom design interno. Porém, simplicidade não está entre essas qualidades. Pelo contrário, o Git é bastante complexo e sua interface é inconsistente e pouco intuitiva.

Há vários lugares onde os arquivos podem estar: stash, diretório de trabalho (working tree na terminologia do Git), index (também chamado de stage area ou cache), repositório local e remoto. A área stash é opcional, mas o index não pode ser evitado completamente. Trata-se de uma área intermediária de preparação para a geração da revisão.

Se o Git fosse um carro, para aprender a dirigi-lo você precisaria antes entender como funciona o motor de combustão interna. Para acelerar, não bastaria apenas pisar no pedal do acelerador; no carro Git seria necessário entender o funcionamento da sonda lambda e acionar diretamente a válvula-borboleta para liberar mais ar para a mistura ar-combustível e, assim, dar mais potência ao motor.

CONCLUSÃO

Entretanto, a vasta informação desses sistemas de controle de versão não é bem apresentada e cria-se uma barreira para entender e explorar as informações disponíveis. Essa barreira degrada a colaboração dos membros da equipe e a habilidade de se entender o histórico do software.

Um dos aspectos importantes do sistema de controle de versão é suportar trabalho em equipe, que tem sido o estilo de desenvolvimento de software predominante. Normalmente, um projeto de software é dividido em pequenos componentes e cada um dos membros da equipe toma conta de componentes diferentes. Entretanto, esses componentes podem ser modificados por pessoas diferentes nos processos de desenvolvimento e manutenção.