

Utilização de Sistemas de Controle de Versão para facilitar o desenvolvimento colaborativo de software

Alunos: *André Silveira e Wesley Dal'Col Von Doelinger*

Orientadora: *Sonia A. Schuetze*

Resumo

De modo a atender às diversas demandas do mercado, é cada vez mais comum que o desenvolvimento de software seja realizado 24 horas por dia, por equipes distintas trabalhando de forma descentralizada ao redor do globo. Por isso, são cada vez mais necessárias ferramentas que propiciem o rastreamento e a integração do código fonte. Alguns sistemas de controle de versão que foram desenvolvidos para auxiliar na primeira tarefa, evoluíram, tornando-se também boas ferramentas de programação integrado-colaborativa. O foco deste trabalho é extensão do uso dessas ferramentas para auxiliar o desenvolvimento colaborativo entre equipes geograficamente distantes.

Palavras-chave: Controle de Versão, Colaboração, Integração, Desenvolvimento de Software.

1. Introdução e Justificativa

Nos primórdios da computação os computadores eram manipulados apenas por um pequeno grupo de técnicos altamente especializados, que precisavam plugar uma grande quantidade de cabos nos enormes painéis que compunham os computadores da época, num processo lento e muito suscetível a erros.

Com o desenvolvimento da tecnologia e o advento das linguagens e técnicas de programação modernas, a tarefa de criar programas de computador pode ser desempenhada por um número maior de pessoas, muitas vezes, cada uma sendo responsável por apenas uma fração de todo o sistema. Unir todas essas partes e garantir que elas funcionem juntas não é uma tarefa simples. Analogamente a correção manual de erros em grandes sistemas de computação é frequentemente uma tarefa tão dispendiosa quanto criá-lo e mantê-lo funcionando.

Para resolver esses problemas de forma automatizada, com o passar dos anos algumas ferramentas foram sendo desenvolvidas, como os programas diff e patch que facilitam processo de aplicar uma correção em uma determinada parte do software, assim como os Sistemas de Controle de Versão dos quais podemos citar o CVS, o Subversion e Git, que são utilizados para rastrear e controlar a evolução do código fonte, o que torna o trabalho das equipes de desenvolvimento um pouco menos complicado.

Entretanto, a dinâmica atual do mercado de tecnologia gera demandas cada vez mais agressivas, fazendo com que o desenvolvimento de software seja feito quase que 24 horas por dia, muitas vezes por equipes geograficamente distantes ou em fuso-horários muito diferentes, resurgindo o desafio de integrar vários componentes de software agora produzidos não somente por pessoas distintas mas por empresas e instituições das mais diversas naturezas e culturas, com o máximo de qualidade possível minimizando o tempo de desenvolvimento e os custos.

Com projetos compostos por milhões de linhas de código, torna-se impraticável que a integração entre as partes seja realizada manualmente, pois basta somente um equívoco para provocar erros muito difíceis de corrigir.

Para tentar resolver esse e outros problemas novos Sistemas de Controle de Versão foram criados, melhorando os mecanismos de mesclagem de código e buscando uma arquitetura descentralizada, de forma que as equipes possam trabalhar com um mínimo de interferência entre si, enquanto a etapa de integração é feita cada vez mais de forma automática.

2. Objetivos e Metodologia

O objetivo principal deste trabalho é demonstrar como os Sistemas de Controle de Versão modernos podem ser utilizados para facilitar o desenvolvimento de software colaborativo, explorando duas de suas principais características, a arquitetura descentralizada e a facilidade com que integram peças de código.

Será feito um breve levantamento histórico contextualizando alguns dos principais Sistemas de Controle de Versão em uso atualmente.

Em seguida demonstrar-se-á como a arquitetura descentralizada facilita o desenvolvimento de ramificações independentes do código fonte com um mínimo de conflitos entre elas, confrontando esse cenário com situações onde não há ferramenta de controle de versão ou quando a ferramenta utilizada não permite essa abordagem.

Também será mostrado como as novas técnicas de rastreamento da genealogia de um projeto de software permitem uma integração muito mais suave e automática do que quando são utilizadas ferramentas menos sofisticadas ou nenhuma ferramenta.