

VERSIONANDO APLICAÇÕES COM SUBVERSION

¹FERNANDO DA SILVA GODOI,

²KÉSSIA RITA DA COSTA MARCHI

¹Acadêmico do curso de Sistemas de Informação

²Professor titular da UNIPAR - Mestrado em Ciências da Computação

Introdução: Iniciado no começo do ano 2000 pela *CollabNet*, o *Subversion* (SVN) foi criado para ser um substituto ao *Concurrent Version System*(CVS), e seu foco era correção de bugs e falhas existentes no CVS. Atualmente há outras ferramentas para controle de versão. Entretanto, observa-se que o SVN está sendo utilizado no gerenciamento de muitos projetos de portes variados. Tal ferramenta combinada com ferramentas de gerenciamento de projetos e metodologias de desenvolvimento ágil proporcionam hoje as melhores práticas para desenvolvimento de softwares.

Objetivo: Observar os benefícios do controle de versão utilizado no desenvolvimento de aplicações em Java.

Metodologia: Para este trabalho foi utilizado o SVN no desenvolvimento de uma aplicação Java, denominada Sistema de Gerenciamento Industrial, em que foi designado um *user* para cada desenvolvedor, permitindo-os atuarem no mesmo projeto simultaneamente.

Resultados: O resultado foi bastante satisfatório, pois, tal ferramenta, permite designar várias equipes de trabalho para o mesmo projeto, atribuindo a cada equipe uma tarefa simultaneamente. Com isso obteve-se um ganho na produtividade, resultando em um produto final com a qualidade esperada. Também observa-se que esta ferramenta vem ao encontro de métodos de desenvolvimentos ágeis.

Discussão: O Subversion é um sistema de controle de versão *free/open-source* que permite o gerenciamento de arquivos, diretórios e alterações. Para tal controle, sua arquitetura é semelhante a um Cliente/Servidor (RAPCINSKI, 2009; FERNANDES, 2007). Possui duas soluções para controle de versão, sendo: *lock-modify-unlock* e *copy-modify-merge*, sendo esta última utilizada no desenvolvimento deste trabalho, pois, quando cada *user* efetua o *checkout*, é feita uma cópia do arquivo para tal *user*, permitindo a eles efetuar edição em suas cópias simultaneamente. Quando o arquivo é remetido ao servidor, o SVN efetua a fusão, sendo necessária intervenção humana em alguns casos para garantir que a fusão completa do arquivo esteja correta (COLLINS-SUSSMAN, 2002; FERNANDES, 2007). Seguindo a recomendação de Kitamura (2009) e Collins-Sussman (2002) o layout do repositório do protótipo está dividido em 3 (três) diretórios, sendo eles: *Trunk*, *Branch* e *Tag*. No *Trunk* localizam-se arquivos da linha de trabalho, no *Branch* arquivos concluídos, disponíveis para teste e no *Tag* a versão final já testada.

Conclusão: Com a utilização do SVN, é possível proporcionar aos desenvolvedores uma maior flexibilidade, permitindo que se criem equipes distribuídas geograficamente. Também é observado que a utilização do SVN possibilita ao gerente de projeto manter o controle sobre as tarefas pertinentes ao desenvolvimento, uma vez que, a cada integrante da equipe cabe o dever de remeter ao repositório suas alterações, e tais arquivos ficam disponíveis em um repositório *online*.

Referências: RAPCINSKI, H. “Utilizando Subversion com Controle de Versão” Disponível em: <http://www.guj.com.br/content/articles/svn/SubVersion_GUJ.pdf> Acessado em: 30/07/2011.

COLLINS-SUSSMAN, B. et al. “Controle de Versão com Subversion”. (2002)
Disponível em: <<http://svnbook-pt-br.googlecode.com/files/svn-book-pdf.zip>>
Acessado em: 05/08/2011.
FERNANDES, T. C. (2007) “Introdução ao Subversion”. Disponível em:
<<http://renesp.com.br/files/so/diversos/tutorialsvn.pdf>> Acessado em: 01/08/2011.
KITAMURA, L. A. (2009) “Controlando Projetos com NetBeans e Subversion”.
Disponível em <<http://www.slideshare.net/lekitamura/controlando-projetos-com-netbeans-e-subversion>> Acessado em: 10/08/2011.