ISBN 978-85-61091-69-9

ESTUDO E DEFINIÇÃO DA APLICAÇÃO PARA CONTROLE DE VERSÕES DOS ARTEFATOS GERENCIADOS PELA FERRAMENTA S.A.Do.M (SOFTWARE ARTIFACTS DOCUMENTATION AND MANAGEMENT)

Julio Cezar Fialho Freire de Carvalho [1], **Aline Maria Malachini Miotto Amaral**[2]

RESUMO: Um aspecto essencial para a qualidade de um sistema de software é a sua documentação. Artefatos de software são peças fundamentais em um processo de desenvolvimento. Eles são produzidos e consumidos por atividades do processo e, assim, estão presentes em todo o ciclo de vida de desenvolvimento de software. Os projetos de desenvolvimento de software contemporâneos têm progressivamente aumentado de tamanho e complexidade, sendo cada vez mais comum sua realização por equipes trabalhando concorrentemente e localizadas geograficamente distantes, tornando a documentação e acompanhamento das modificações um trabalho delicado e de fundamental importância para o sucesso do projeto. Com base nos estudos e resultados alcançados com o projeto de iniciação cientifica "Estudo e Implementação de Um Sistema para Controle de Versões[1]", formou-se uma sólida base de conhecimentos sobre a abordagem do gerenciamento de versões e sua essencialidade para um bom gerenciamento de configuração de software, permitindo assim atender de forma efetiva as necessidades do modelo de dados demandado pela ferramenta S.A.Do.M. O presente projeto propõe o desenvolvimento do mecanismo de controle de versões dos artefatos de software que a ferramenta S.A.Do.M irá armazenar. Para tanto foram usados os estudos providos pelo projeto[3] de pesquisa já realizado e também no modelo de dados e levantamento dos requisitos feitos para a ferramenta S.A.Do.M que tem seu estudo e desenvolvimento executado pelo Grupo de Pesquisas em Sistemas de Informação (GPSI).

PALAVRAS-CHAVE: Artefatos; Software; Versões.

1 INTRODUÇÃO

A existência de mecanismos eficientes de comunicação não é suficiente para solucionar os problemas de desenvolvimento concorrente de projetos de software. Ao contrário, o novo cenário traz novos problemas para o processo de desenvolvimento. Por exemplo, o trabalho concorrente de equipes geograficamente distribuídas dificulta o controle de alterações nos componentes de um projeto em desenvolvimento. Muitas vezes, a documentação é negligenciada em decorrência de fatores como falta de uma política organizacional que valorize essa tarefa, falta de ferramentas para apoiar a documentação e prazos exíguos, o que faz com que desenvolvedores optem por deixar a documentação em segundo plano.

Ferramentas provendo facilidades de documentação permitem agilizar o desenvolvimento de software, realizando algumas tarefas automaticamente, e permitem que modificações sejam mais facilmente realizadas, simplificando a tarefa de manutenção. (PETERS; PEDRYCZ, 2001). Entretanto, é difícil encontrar ferramentas nas

[1] Acadêmico do Curso de Sistemas de Informação do Centro Universitário de Maringá – CESUMAR, Maringá – PR. Programa de Bolsas de Iniciação Científica do PIBIC/CNPq-Cesumar. juliocffc@gmail.com
[2] Orientadora e docente do CESUMAR. Departamento de Sistemas de Informação do Centro Universitário de Maringá – CESUMAR, Maringá – PR. amiotto@cesumar.br

quais a documentação possa ser feita de forma integrada, customizada e consistente.

Atualmente, a grande maioria das abordagens para solucionar esse tipo de problema é focada em código fonte. Contudo, os sistemas mais complexos demandam um esforço adicional para a sua compreensão, tornando necessário o uso de artefatos que descrevem aspectos não representados pelo código-fonte, agregando informação em um nível de abstração mais elevado (PRESSMAN, 1996).

O atual projeto propõe a implementação de um flexível gerenciador de versões integrado à ferramenta S.A.Do.M a fim de gerenciar com efetividade o processo de armazenamento e documentação dos mais diversos artefatos de software conforme propõe as metodologias hoje existentes e permitindo ao gerente de projeto adaptar novas metodologias conforme sua necessidade. Para tanto, a implementação desenvolvida foi baseada no código fonte do conceituado CVS (*Concurrent Version System*).

Um bom gerenciamento de artefatos e documentação durante a produção de um software torna-se um fator de extrema atenção e importância para que se chegue a um produto final de qualidade e sem desperdício de recursos pessoais e financeiros (BRUEGGE et. al, 2006). O alcance dos objetivos propostos pela ferramenta S.A.Do.M está envolvido diretamente com o seu sistema de controle de versões que deverá ser desenvolvido e integrado à ferramenta em questão.

Dentro deste contexto e apoiado pelos estudos realizados pelo projeto de pesquisa[1] já realizado, o objetivo deste trabalho foi desenvolver uma aplicação que irá gerenciar as versões dos artefatos armazenados pela ferramenta S.A.Do.M, permitindo um controle efetivo dos mesmos.

Este trabalho está organizado da seguinte forma: a seção 02 apresenta uma introdução ao gerenciamento de versões, enquanto que na seção 03 são apresentados algumas das características que devem ser disponibilizados na ferramenta S.A.Do.M com base nos recursos disponibilizados pela Subversion/TortoiseSVN. Finalmente, na seção 04 são apresentadas as considerações finais desta pesquisa.

2 FERRAMENTAS DE GERENCIAMENTO DE VERSÕES

Atualmente existem diversas ferramentas que apóiam o desenvolvedor de software no processo de gerenciamento de versões de seus artefatos produzidos, trazendo agilidade ao desenvolvimento e conseqüentemente possibilitando uma melhoria do produto final; no entanto nem todas atendem de forma flexível projetos que possuam variados tipos de artefatos e metodologias especificas. De todas as ferramentas encontradas e estudadas, todas focam em um determinado tipo de desenvolvimento ou artefato.

Dentre as ferramentas mencionadas, pode-se citar a Borland TeamSource, Intersolv PVCS Version Manager, Microsoft Visual SourceSafe, Rational Clear Case e Concurrent Version System entre muitas outras. Para esta pesquisa foram levantadas as características de uma ferramenta predecessora e tomada como referência por muitas outras, o CVS, o qual possui seu código fonte aberto e pouca restrição quanto ao seu modo de utilização. Porém, para foco deste estudo adotou-se o SubVersion, resultado de inúmeras melhorias e considerado sucessor do CVS.

2.1 SUBVERSION

Dentre as características do SubVersion que mais contribuíram para a sua escolha, pode-se citar a forma com que esta registra o histórico de mudanças de árvores de diretório inteiras ao longo do tempo, ou seja, o SubVersion controla não só as versões dos

arquivos, como também dos diretórios. Outra função relevante do SubVersion é a possibilidade de adicionar, excluir, copiar e renomear tanto arquivos quanto diretórios. Cada novo item adicionado tem um histórico próprio, limpo e novo. Ao contrário do que acontece no CVS, o SubVersion suporta que quando enviadas conjuntamente modificações (que pode envolver diversos arquivos e diretórios), as mesmas sejam registradas no repositório de forma atômica.

3 PROPOSTA DE GERENCIAMENTO DE VERSÕES DA FERRAMENTA S.A.DO.M

Como já mencionado anteriormente, optamos por utilizar e personalizar os recursos disponibilizados pela ferramenta de gerenciamento de versões Subversion, uma vez que a mesma possui código livre (open source) e atende a todas as necessidades de gerenciamento de versões identificadas para a ferramenta S.A.DO.M.

A seguir serão apresentados alguns recursos da ferramenta Subversion, bem como a seqüência exemplificada de ações para configurar e utilizar um repositório de documentos com diferentes versões a serem gerenciadas.

3.1 PROCESSO DE CHECKOUT

Para este processo deve-se criar uma nova pasta para se realizar o Checkout dos documentos salvos.

Após solicitar a ação de checkout, informações sobre: o repositório, a pasta de checkout e o tipo de revisão que será feita devem ser preenchidas. Desta forma, recupera-se todos os artefatos de um projeto (presentes em um repositório) para futura revisão.

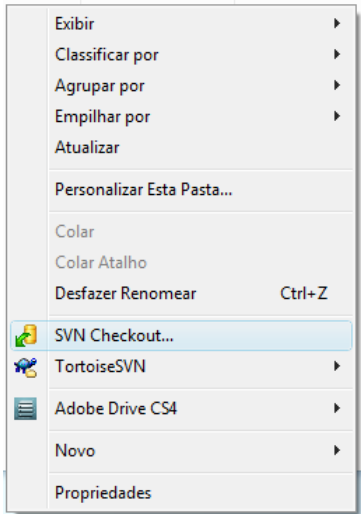

Figura 1. Recorte da tela do Windows (aplicação TortoiseSVN) para inicio do processo de Checkout

3.2 PROCESSO DE *LOCK* (TRAVAMENTO) E *RELEASE LOCK* (DESTRAVAMENTO)

O *Lock* se torna necessário para impedir que outras pessoas recuperem o documento enquanto este está sendo alterado.

Deve-se Clicar no botão direito sobre o documento a ser travado e escolher a opção TortoiseSVN>Get Lock:

Logo após este processo é solicitada identificação do responsável pelo bloqueio do artefato.

A partir deste momento o arquivo estará bloqueado para alteração de outros usuários.

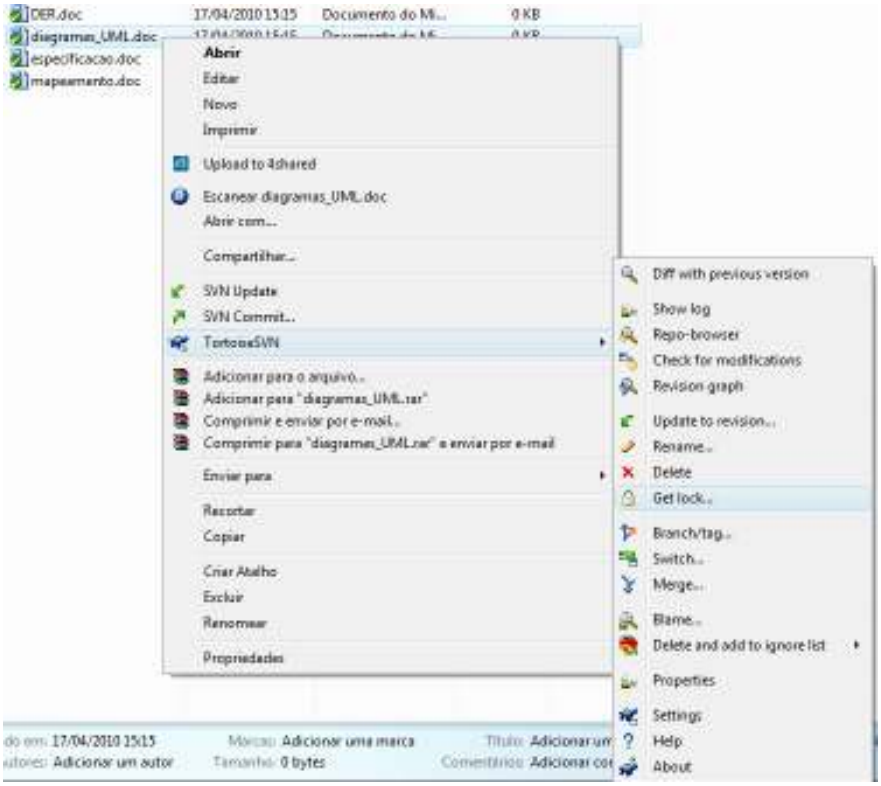

Figura 2. Recorte da tela do Windows (aplicação TortoiseSVN) para escolha do processo de Lock

3.3 PROCESSO DE COMMIT (ALTERANDO E SALVANDO DOC. NO REPOSITÓRIO)

O artefato “especificação.doc”, foi alterado, e com isto ferramenta TortoiseSVN automaticamente marca o documento alterando o ícone de verde com check para um vermelho com exclamação.

Para que as alterações sejam salvas no repositório deve-se aplicar o comando Commit.

Para isto deve-se clicar com o botão direito no arquivo a ser sincronizado e depois na opção SVN *Commit.*

Após isso, algumas confirmações são solicitadas, e o artefato tem sua versão atualizada no repositório.

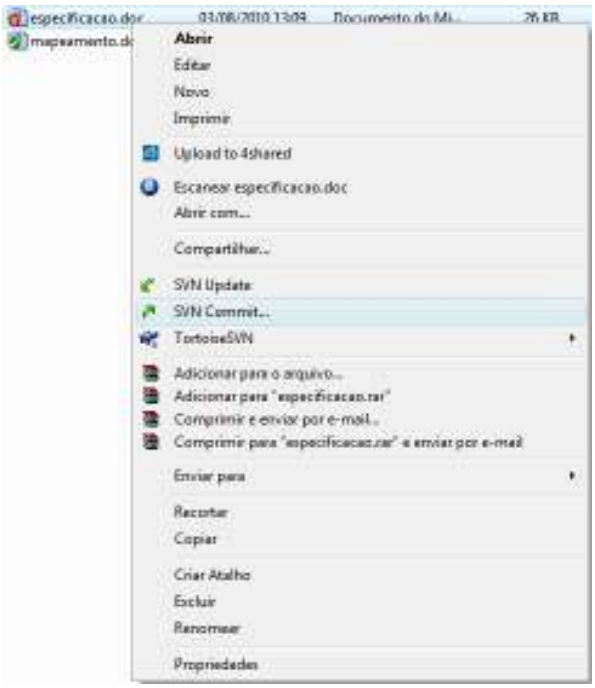

Figura 3. Recorte da tela do Windows (aplicação TortoiseSVN) para realizar o *commit* no artefato alterado

3.4 REALIZAÇÃO DE *REVERT* PARA RECUPERAR UM ORIGINAL E DESPREZAR ALTERAÇÕES

Para recuperar um artefato original executa-se o processo parecido com o descrito na seção 3.4., no entanto fazendo o caminho inverso, ou seja, ao invés de confirmar as alterações no repositório, descartamos as alterações e mantemos o artefato original no repositório.

Para isto deve-se clicar com o botão direito no arquivo a ser recuperado e depois na opção SVN *Revert.*

Após isso, algumas confirmações são solicitadas, e o artefato tem sua versão recuperada no repositório.

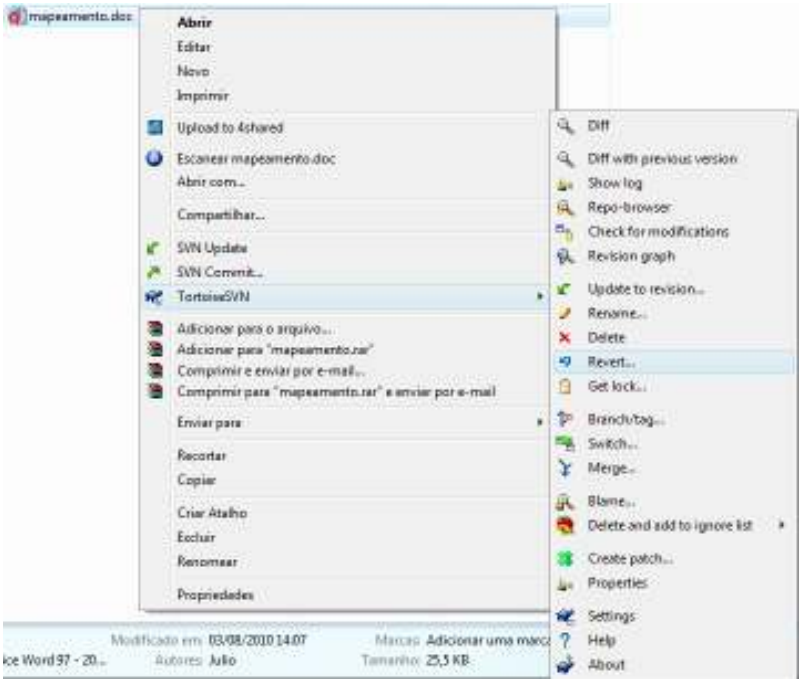

Figura 4. Recorte da tela do Windows (aplicação TortoiseSVN) para realizar o *revert* no artefato alterado

4 CONSIDERAÇÕES FINAIS

Dentro do contexto de produção de software, as atividades de GCS e, mais especificamente as atividades de controle de versões, orientam os desenvolvedores durante a evolução do software e permitem um controle e recuperação de inúmeras informações sobre os artefatos controlados, sendo parte fundamental no processo de desenvolvimento.

A documentação de um software é extremante fundamental para assegurarmos sua qualidade. Para termos uma documentação completa e bem definida temos que garantir que durante o processo de software teremos ferramentas que nos auxiliarão na tarefa de gerenciar, documentar e armazenar tal documentação de todas as mudanças ocorridas pelos mais diferentes tipos de artefatos de software.

Diante de tais fatores e baseado nos estudos realizados foram apresentados principais recursos da ferramenta Subversion/TortoiseSVN e como os mesmos implementam os recursos de gerenciamento de versões que devem ser suportados pela ferramenta S.A.Do.M. Deve-se ressaltar que as ferramentas estudadas cumprem suas funções com méritos sendo fáceis de instalar e de utilizar, e também que as mesmas são de código aberto o que permite sua integração a ferramenta S.A.Do.M, bem como sua personalização quando necessário.

REFERÊNCIAS

BRUEGGE, Bernd et. al. Supporting Distributed Software Development with fine-grained Artefact Management. In: **IEEE International Conference on Global Software Engineering**. 2006, IEEE Computer Society, Washington, DC.

JUNQUEIRA, D. C. **Um Controle de Versões Refinado e Flexível**, 2007. 106p. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional – Universidade de São Paulo), São Carlos, São Paulo.

PETERS, J.; PEDRYCZ, W. **Engenharia de Software**: Teoria e Prática. Rio de Janeiro: Campus, 2001.

PRESSMAN, R.S. **Engenharia de Software.** São Paulo: Makron Books, 1995.