



ESTUDO E IMPLEMENTAÇÃO DE UM SISTEMA PARA CONTROLE DE VERSÕES

Vilson Raphael Vello de Andrade¹; Renan Rodrigues Cuoghui²; Aline Maria Malachini Miotto Amaral³

RESUMO: A demanda por qualidade tem levado a comunidade de software ao desenvolvimento de inúmeras técnicas e ferramentas que auxiliem o desenvolvimento perante o grande número de itens de configuração envolvidos no decorrer de um desenvolvimento. Para assegurar tal qualidade é necessário o uso de procedimentos e ferramentas que auxiliem o processo de desenvolvimento de software. Dentre tais processos destaca-se o gerenciamento e controle de versões, que foi o objetivo deste projeto. Para tal, foi realizado um estudo baseado na ferramenta S.A.Do.M (*Software Artifacts Document and Management*), desenvolvida pelo Grupo de Pesquisa em Sistemas de Informação (GPSI), que tem como principal objetivo o armazenamento e gerenciamento da documentação de artefatos de software, no seu efetivo controle e gerenciamento das diferentes versões dos artefatos armazenados. Buscou-se com este projeto, a obtenção de conhecimento necessário sobre o tema abordado para a implementação de um flexível gerenciador de controle de versões integrado à S.A.Do.M.

PALAVRAS-CHAVE: Artefatos de Software, Controle de Versões, Gerenciamento de Configuração de Software.

INTRODUÇÃO

Os níveis de complexidade e abstração exigidos pelos atuais sistemas impactam diretamente nos esforços empenhados para realizar a sua documentação (BRUEGGE et al, 2006). Considerando as constantes mudanças que afetam as regras do mercado e também as necessidades dos usuários, torna-se difícil prever como um software evoluirá com o passar do tempo (PRESSMAN, 2005). Nesse contexto de modificação constante e alta complexidade dos artefatos de software produzidos, a Gerência de Configuração de Software (GCS) é a área da Engenharia de Software cujo principal objetivo é evitar a perda de controle do projeto do ponto de vista da gerência de artefatos. A GCS também pode ser definida como o aspecto do processo de gerência que foca exclusivamente em controlar as mudanças que ocorrem durante o processo (ISO, 1995). Um dos procedimentos que se destaca durante o desenvolvimento de software é o controle de versões, sendo este uma das principais funcionalidades do Gerenciamento de

¹ Acadêmico do Curso de Sistemas de Informação. Centro Universitário de Maringá – CESUMAR, Maringá – PR. Bolsista do Programa de Bolsas de Iniciação Científica do PIBIC/CNPq-Cesumar (PIBIC-Cesumar). raphael@raphaelvello.com.br

² Acadêmico do Curso de Sistemas de Informação. Centro Universitário de Maringá – CESUMAR, Maringá – PR. renan2844@hotmail.com

³ Docente do CESUMAR. Departamento de Sistemas de Informação do Centro Universitário de Maringá – CESUMAR, Maringá – PR. amiotto@cesumar.br

Configuração de Software. A falta de um correto controle de versões dos artefatos de software gerados impacta diretamente na correta elaboração da documentação do software produzido.

O controle de versões combina procedimentos e ferramentas para gerenciar diferentes versões que são criadas durante o processo de engenharia de software, preocupando-se com a manutenção da rastreabilidade das versões de um sistema, coordenando o trabalho paralelo de desenvolvedores por meio das mais diferentes funcionalidades.

De acordo com o relatório sobre Qualidade e Produtividade no Setor de Software Brasileiro (MCT, 2002) referente a 2001, somente 10,4% das empresas entrevistadas (45 empresas de um espaço amostral de 431) praticavam gestão de mudanças. Além disso, somente 20,1% das empresas entrevistadas (85 empresas de um total de 423) faziam uso de ferramentas de apoio ao GCS.

Em função da necessidade de ferramentas que gerenciem a produção de software de forma flexível, adequando-se à necessidade de cada desenvolvimento e controlando seus artefatos de software, o Grupo e Pesquisas em Sistemas de Informação⁴ (GPSI) desenvolvem, através de pesquisas científicas, estudos que visam à implementação de tal ferramenta, contribuindo efetivamente para a melhora da produtividade e também para a aplicação de técnicas de reuso de componentes. O sucesso de tal ferramenta, chamada inicialmente de S.A.Do.M (*Software Artifacts Documentation Manager*), relaciona-se diretamente a um eficaz controle e gerenciamento de versões, sincronização de mudanças, armazenamento de histórico de mudanças, comunicação com o repositório e recuperação de revisões para os mais variados tipos de artefatos.

Dentro deste contexto este trabalho teve como objetivo realizar um estudo detalhado das características necessárias para um suporte efetivo ao gerenciamento das versões dos artefatos que serão armazenados pela S.A.Do.M.

O presente artigo está organizado da seguinte forma: na seção é detalhada a metodologia aplicada na pesquisa; na seção 03 são apresentados os resultados obtidos até o momento e finalmente na seção 04 temos as conclusões do trabalho realizado.

2. MATERIAL E METODOS

Foi realizado, em uma primeira etapa, o levantamento e estudo de todas as características e terminologias envolvidas durante o gerenciamento de versões e revisionamento de artefatos de software. Formou-se uma sólida base de conhecimento sobre o tema abordado nesta pesquisa possibilitando em uma segunda etapa realizar o levantamento bibliográfico no que diz respeito ao conceito de gerenciamento de configuração de software e controle de versões. Buscou-se as melhores práticas e modelos com uma análise primária das ferramentas hoje existentes assim como a evolução de diversas outras. Após tais estudos, foram levantadas as necessidades da ferramenta S.A.Do.M assim como seu modelo que foi objeto de outro projeto de pesquisa executado pelo GPSI; com base neste levantamento foi elaborado o modelo para implementação do gerenciador de versões suportado pela ferramenta S.A.Do.M.

3. RESULTADOS E DISCUSSÕES

3.1 Ferramentas de gerenciamentos de versões estudadas

Existem hoje diversas ferramentas que implementam o controle de versões, automatizando o processo de forma a agilizar e facilitar o trabalho do desenvolvedor de

⁴ Grupo de Pesquisas em Sistemas de Informação (GPSI) – Grupo destinado a pesquisas na área de sistemas de informação do Centro Universitario de Maringá - CESUMAR

software. 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.

Dentre as características do SubVersion que mais contribuíram para a sua escolha como ferramenta base para o desenvolvimento do gerenciador de versões da S.A.Do.M 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.

O SubVersion possui um mecanismo através do qual é possível atribuir informações diversas (nomeadas de *properties*) aos artefatos, sem que seja necessário armazená-las no interior do repositório. Cada artefato pode possuir uma ou mais propriedades. Mesmo havendo propriedades pré-definidas pelo SubVersion, é possível criar e armazenar propriedades personalizadas, não havendo um limite para a quantidade de propriedades por artefato.

3.2 Modelo da Ferramenta S.A.Do.M proposto pelo GPSI

Durante diversas reuniões realizadas pelo GPSI, debateu-se acerca dos diversos recursos que a ferramenta S.A.Do.M deveria suportar chegando à concepção do modelo de dados representado pela figura 1.

Após a elaboração do modelo de dados, definiu-se que as entidades Projeto, Artefato e Artefato Vinculado seriam alvos diretos do sistema de gerenciamento de versões e, a fim de apoiar outras necessidades da ferramenta S.A.Do.M, identificou-se alguns requisitos primários, tais como:

- A estrutura do armazenamento dos artefatos a serem documentados: esta será feita em repositório centralizado;
- O método de controle e acesso a versões de artefatos: que deverá ser do tipo Lock-Modify-Unlock evitando conflitos em alterações por mais de um desenvolvedor;
- O layout de repositórios: que deverá ser do tipo compartilhado, permitindo que determinado artefato possa compor mais de um projeto.

Com esta base formada e tendo em vista os propósitos e necessidades expostos nos estudos realizados pelo GPSI no decorrer da concepção da ferramenta S.A.Do.M apresenta-se como trabalho futuro o desenvolvimento, de forma integrada à ferramenta S.A.Do.M, da ferramenta de controle de versões que apóie a documentação de software.

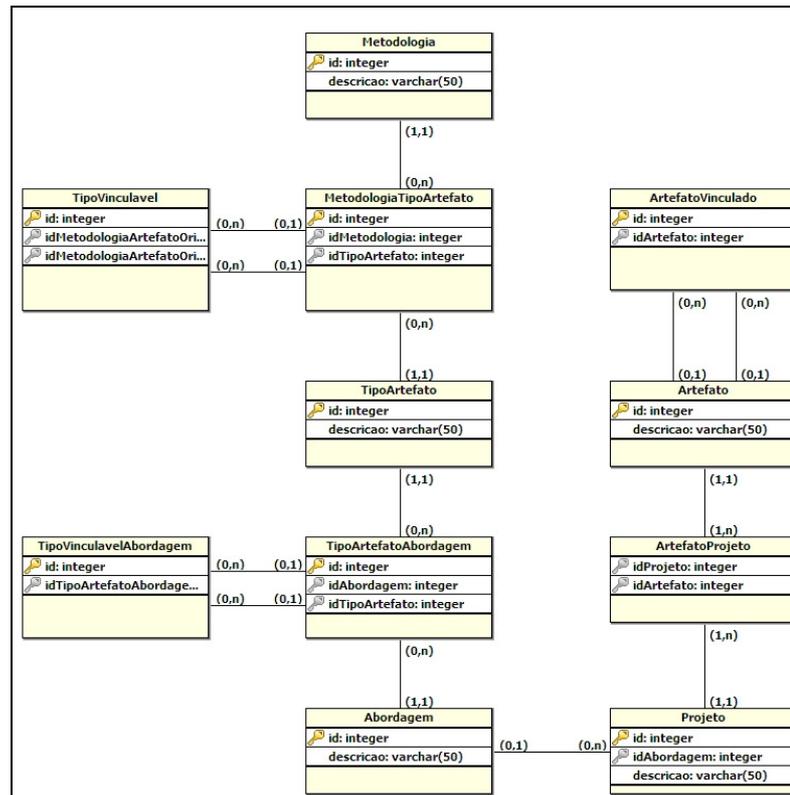


Figura 1. Modelo lógico de dados da ferramenta S.A.Do.M

3.3 Características de Gerenciamentos de Versões a serem Suportados na S.A.Do.M

Apoiado pelos estudos realizados e pelo modelo de dados apresentado na figura 1 pode-se destacar as principais características do Gerenciador de Versões da ferramenta S.A.Do.M.

3.3.1 Relacionamento das versões dos artefatos e estado dos objetos

As versões de um mesmo artefato estão ligadas através de um relacionamento de derivação linear. Cada versão é criada como sucessora de outra, e esta é considerada sua antecessora.

3.3.2 Estado dos artefatos

Para cada versão também é estabelecido um atributo de status identificador de quais operações poderão ser realizadas sobre este artefato. Este status poderá ter os seguintes valores: *estável*, *em trabalho* ou *out of version*. Uma versão possui status “em trabalho” enquanto estiver sendo modificada. Quando esta versão estiver pronta, ou seja, enquanto quem efetuou o Check - Out deste artefato não efetuar um Check-in. Após tal processo de Check-in ela é definida como estável e a versão anterior passa ter seu status “out of version”.

A versão “estável” é a versão corrente, mais atual, do artefato, não podendo ser alterada, até que ela mude novamente de status. Pode-se remover versões que possuam status “em trabalho” ou “estáveis”, desde que essas não possuam sucessoras, porém estas não terão versões anteriores e rastreáveis.

3.3.3 Armazenamento do histórico de alterações

Quando um desenvolvedor efetua alterações em um artefato, o repositório armazena em um banco de dados as informações sobre estas atualizações. Estes dados propiciam um importante histórico para a solução de problemas decorrentes da manutenção de artefatos. Através deste histórico obtêm-se informações pertinentes às alterações

efetuadas agilizando o acesso a dados como: desenvolvedor que efetuou a manutenção, data/hora em que ocorreram e a descrição informada.

Certas premissas também foram levantadas e deverão ser seguidas pela S.A.Do.M para um eficiente gerenciamento pelo controle de versões.

3.3.4 Restrições de acesso

O desenvolvedor ligado a um projeto específico pode executar as tarefas de get, lock, unlock, check-out, check-in, adição de artefato e voltar versão. Já o administrador do repositório pode efetuar todas as operações do desenvolvedor, além de remover artefatos, voltar qualquer versão e criar projetos deliberando responsabilidades aos desenvolvedores.

4. CONCLUSÕES

A qualidade de software esta intrinsecamente ligada à qualidade dos processos utilizados para o desenvolvimento, com base neste fato a melhoria da qualidade de software é uma constante obtida com a melhoria da qualidade dos processos.

A documentação de um software é extremamente 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 definidas as principais características que deverão ser suportadas no gerenciamento de versões da ferramenta S.A.Do.M.

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.

CONRADI, R.; WESTFECHTEL, B. **Version Models for software configuration management**. ACM Computer SWR., ACM, New York, NY, USA.

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.

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

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

SOMMERVILLE, I.; ANDRADE, M. **Engenharia de Software**. São Paulo: Addison Wesley, 2003.

ISO/IEC12207, **Information Technology – Software Life Cycle Processes**. International Standard Organization. 1995.