

Gerência Cooperativa de Configuração de Software

Ricardo Luiz Schneider
schneide@dcc.ufrj.br

Orientador: Marcos Roberto da Silva Borges
mborges@nce.ufrj.br

Departamento de Ciência da Computação
Instituto de Matemática
Universidade Federal do Rio de Janeiro
RJ

Abstract – The problems due to the frequent and strong changes in the software process and product components are presented. A good Software Configuration Management – SCM (or GCS in portuguese) seems to be a solution to those problems. Some commercial and academic SCM tools were searched and analyzed. Almost all of those tools do not incorporate a workgroup approach, wich is a very important characteristic, due to the increasing complexity and geographical dispersion of the development teams. As a result a Cooperative SCM - Software Configuration Management tool is proposed.

Palavras-chave – gerência de configuração, controle de versões, qualidade de software, software cooperativo, trabalho em grupo

1. Introdução

Um dos maiores problemas da Engenharia de Software é a dificuldade de lidar com as mudanças que ocorrem durante todo o ciclo de vida dos sistemas. A busca de solução tem gerado muitas pesquisas e modelos alternativos. O ciclo de vida clássico (modelo em *cascata*), herdado das *engenharias* mais convencionais, tem se mostrado pouco eficaz na administração das mudanças que ocorrem durante o desenvolvimento e manutenção de produtos de software. Vários outros modelos de ciclo de vida (como a prototipação, ciclo de vida em espiral, desenvolvimento incremental e linguagens de quarta geração) são propostos na literatura com o objetivo de minimizar o impacto das mudanças no resultado final do processo de desenvolvimento de sistemas, tentando diminuir o risco de insucesso nesses projetos.

Neste contexto de incertezas e mudanças, conseguir a Garantia de Qualidade dos Produtos de Software passa a ser um dos objetivos permanentes e fundamentais da Engenharia de Software [PRE97]. Pressman, na referência citada, apresenta seis áreas de atividades diretamente ligadas à Qualidade e que devem merecer a atenção continuada do Engenheiro de Software:

- Métodos de Engenharia de Software;
- Padrões e Procedimentos;
- Revisões Técnicas Formais;
- Medições e Métricas de Software;
- Gerência de Configuração de Software e Garantia de Qualidade de Software; e,
- Testes.

Destas atividades, a Gerência de Configuração de Software – GCS - representa um papel central em relação às demais, agindo como uma atividade *guarda-chuva*. Revisões de Software e Testes podem estar vinculadas diretamente ao processo de Gerência de Configuração. Os Padrões e Procedimentos serão automaticamente reforçados em presença dos controles previstos na Gerência de Configuração. Medições e Métricas, principalmente as relativas aos componentes de tamanho e complexidade, podem ser incorporadas facilmente à Gerência de Configuração. E as demais atividades podem se beneficiar de um efetivo controle do processo de Gerência de Configuração.

O modelo CMM – *Capability Maturity Model*, do SEI – Software Engineering Institute da Carnegie Mellon University, de avaliação da capacitação das organizações em desenvolver software com qualidade, coloca já no nível 2 - *Repetível*, dos cinco níveis de maturidade previstos, a exigência da aplicação de técnicas de GCS. Isto demonstra a importância que este modelo atribui ao gerenciamento de configuração de software.

Por outro lado, a tendência universal de horizontalizar e diminuir o tamanho das organizações administrativas, em geral, e das equipes de desenvolvedores de software, em particular, tem levado à formação de equipes de desenvolvimento com maior poder de decisão (*empowered teams*) [KHO95]. Estas equipes se colocam cada vez mais dispersas geograficamente, formando equipes virtuais em que a palavra chave de atuação é Cooperação.

Pela análise das principais ferramentas existentes de GCS, constatamos uma carência de características de cooperação ou colaboração na dinâmica operacional das mesmas, geralmente centradas na atuação coordenadora da gerência de projeto. O Code Co-op é um exemplo de software em que estão incluídos alguns aspectos de cooperação, mas basicamente, em relação à distribuição dos documentos. Já experiências mais avançadas de colaboração entre os desenvolvedores, foram propostas e utilizadas em grupos distribuídos de desenvolvimento, como no projeto Apache [FIE99]. Neste caso, no entanto, a obtenção do nível adequado de cooperação foi obtido por mecanismos organizacionais convencionais, sem

o apoio de ferramentas de *enforcement*.

2. Gerência de Configuração de Software

Segundo Babich [BAB86], "... Gerência de Configuração de Software é a arte de identificar, organizar e controlar as modificações de um software sendo desenvolvido por uma equipe de projeto. O objetivo é maximizar a produtividade pela minimização das falhas."

Gerência de Configuração de Software (GCS) é um conjunto de atividades aplicadas durante todo o processo de software, tendo em vista que as mudanças ocorrem a qualquer tempo, que inclui em especial as seguintes atividades:

1. Identificar as mudanças;
2. Controlar as mudanças;
3. Garantir que cada mudança será implementada adequadamente; e,
4. Reportar as mudanças às pessoas que delas devam tomar conhecimento.

O resultado do processo de software é um conjunto de informações que constituem os itens de gerência de configuração. Uma *baseline* (estrutura base) é "uma especificação ou produto que foi formalmente revisto e aceito, que a partir de então serve como base para outros desenvolvimentos, e que só pode ser mudado através de um procedimento formal de controle de mudança". A *baseline* é formada pelo conjunto de itens de gerência de configuração que formam uma determinada *Versão* do produto em desenvolvimento ou já entregue ao uso.

3. A Cooperação na Gerência de Configuração

O desenvolvimento de software é um processo social, de um grupo de pessoas. Mesmo numa situação, bastante atípica, em que uma única pessoa resolva construir um produto de software, sem nenhuma interação com outras pessoas, ela deverá exercer dois papéis sociais diferentes: o de programador e o de usuário. Deverá usar o *chapéu* de programador e o de usuário, em diversas decisões de projeto que deva tomar, e *interagir* consigo mesmo para avaliar, sob enfoques diferentes, essas decisões.

A situação típica, por outro lado, é a de um grupo de pessoas, muitas vezes dispersas geograficamente, exercendo os papéis de clientes da contratação do desenvolvimento, usuários dos produtos gerados, analistas, projetistas, programadores, avaliadores e certificadores dos objetos de software sendo desenvolvidos. Khoshafian [KHO96] afirma que "uma das mais convincentes visões para o próximo milênio é a da colaboração e do trabalho em equipe. Cooperação, colaboração, democratização, *empowerment* e trabalho em equipe estão se tornando essenciais para a sobrevivência de qualquer unidade de negócio viável, desde o fornecedor de serviços individual, grandes companhias virtuais ou até mesmo nações".

Os métodos e processos da Engenharia de Software em geral, e da GCS em particular, devem se ajustar a esta perspectiva de cooperação, colaboração e democratização cada vez maiores do trabalho das equipes de desenvolvimento.

Uma ferramenta de GCS pode ser concebida, então, como um sistema de Groupware, em que o fluxo das ações do processo seja organizado e ditado pela atuação do time de desenvolvimento.

A cooperação entre os membros das equipes deve ser conduzida (*enforcement*) através do registro e administração dos diálogos e decisões tomadas ao longo do processo.

Em diversas situações é necessário registrar e recuperar decisões de análise, projeto, implementação, avaliação e teste, tomadas individualmente pelos membros da equipe de desenvolvimento, ou tomadas em conjunto por mais de uma pessoa, mas que não precisam

estar reunidas para tomar a decisão. Isto se dá em função do grau de importância ou urgência da decisão. Temos então uma situação típica de **Comunicação Assíncrona** entre os membros da equipe. Sendo importante consolidar, registrar e divulgar o resultado da decisão.

Em outras situações, serão necessários contatos diretos e pessoais entre os membros da equipe. Isso será necessário quando decisões importantes devam ser tomadas. Para este tipo, podem ser necessárias reuniões presenciais onde a comunicação se dá de forma muito mais completa, em que gestos e expressões assumem um contexto mais realístico da comunicação pretendida. No entanto, cada vez mais, a tecnologia permite que reuniões entre pessoas possam ser realizadas de forma *virtual*, transmitindo de forma interativa e online, tanto as imagens dos protagonistas da reunião, quanto os textos, imagens e sons dos objetos em análise - diagramas, documentos, etc. A **Comunicação Síncrona**, tanto a presencial quanto a virtual, deve ser registrada e recuperada pela ferramenta de GCS, garantindo a documentação adequada das decisões tomadas.

Outro ponto a ser explorado na arquitetura das ferramentas de GCS é o seu potencial integrador das diversas atividades da Engenharia de Software, em especial aquelas ligadas diretamente à Garantia da Qualidade. Em todas elas a COOPERAÇÃO será reforçada com mecanismos automáticos. Potencialmente, pretendemos explorar as seguintes atividades:

- Inspeção e Avaliação dos produtos de software
- Testes dos componentes e produtos
- Reutilização de componentes e produtos
- Medições de tamanho e complexidade do software
- Gerência de mudanças
- Administração do processo de desenvolvimento

4. Conclusão

Este trabalho está em andamento e deverá ser enriquecido com uma pesquisa mais ampla de outras ferramentas de GCS. O principal resultado pretendido é a construção de uma ferramenta de Gerência Cooperativa de Configuração de Software - GCCS. A previsão otimista é que seja concluído até março de 2001.

Além disto, o autor está conduzindo e orientando um trabalho de pesquisa nos assuntos correlatos à Gerência de Configuração com três grupos de alunos, para a elaboração de seus trabalhos de final de curso do Curso de Bacharelado em Matemática, modalidade Informática, do Instituto de Matemática da UFRJ.

5. Bibliografia

- [BAB86] BABICH, W.A. Software Configuration Management. Addison Wesley, 1986
- [BUC92] BUCKLE, K.K. et alii. Software Configuration Management. MacMillan Education Ltd. 1982, 152pp.
- [FIE99] FIELDING, Roy T. Shared Leadership in the Apache Project, in Communications of ACM, abril 1999/Vol 42, N° 4., 42-43pp.
- [KHO95] KHOSHAFIAN, Setrag & BUCKIEWICZ, Marek. Introduction to Goupware, Workflow, and Workgroup Computing. John Wiley & Sons, Inc. 1995, 376pp
- [PRE97] PRESSMAN, Roger S. Software Engineering - A practitioner's Approach. The McGraw-Hill Companies, Inc. 1997, 852pp