ISSN 1415-8825

Empresa Brasileira de Pesquisa Agropecuária
Centro Nacional de Pesquisa Tecnológica em Informática para a Agricultura
Ministério da Agricultura e do Abastecimento
Rua Dr. André Tosello, s/nº Caixa Postal 6041 - Barão Geraldo
13083-970 - Campinas, SP
Fone (019) 289-9800 Fax (019) 289-9594

COMUNICADO TÉCNICO

Nº. 8 dezembro/98, p.1-7

IMPLANTAÇÃO DE CONTROLE DE VERSÃO DE SOFTWARE NO SUBPROJETO INFORMATIZAÇÃO DO SIGER

Paulo César de Oliveira [1]
Evandro Bacarin [2]

1. Introdução

Os processos de desenvolvimento e de manutenção de software são extremamente dinâmicos. É inevitável deparar-se com situações de mudança neste contexto. Para que tais processos sejam realizados com disciplina e qualidade, gerência de configuração de software (GCS) é um aspecto fundamental. Desta forma, instituições de grande relevância no campo de engenharia de software, tais como o *"Institute of Electrical and Electronics Engineers"* (IEEE) e o *"Software Engineering Institute"* (SEI), têm concentrado esforços para a formalização e a padronização na área de GCS.

Embora o termo gerência de configuração de software seja amplamente utilizado não há uma definição de consenso para ele. A definição adotada neste Comunicado Técnico (apresentada na próxima seção) reflete uma combinação das visões do IEEE e do SEI a respeito desta área.

A gerência de configuração se estende por todo o ciclo de vida de um sistema de software e é indispensável para sistemas que tenham características tais como: complexidade, grande porte e equipe de trabalho numerosa. O Sistema de Informação Gerencial da Embrapa (Siger), em implementação na Embrapa Informática Agropecuária, se enquadra nestas características. Entretanto, procedimentos rudimentares de GCS eram executados pela sua equipe de desenvolvimento.

Este documento apresenta inicialmente conceitos básicos relativos à GCS e introduz o aspecto de controle de versão no contexto de gerência de configuração de software. Também aborda os avanços obtidos com a implantação de controle de versão no escopo do subprojeto Informatização do Siger, através do uso da ferramenta CS-RCS e da adoção de procedimentos de trabalho pela equipe do referido subprojeto.

2. Gerência de Configuração de Software

A configuração de um determinado software é uma coleção organizada de seus componentes, também chamados de itens de configuração, em um ponto específico (discreto) de tempo. Exemplos de itens de configuração incluem códigos fonte e binário, documentos de especificação, compiladores e bases de dados. Configurações refletem a estrutura de um software no tempo.

[1] Mestre em Engenharia Elétrica, Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo, 13083-970 - Campinas, SP.
[2] Mestre em Ciência da Computação, Embrapa Informática Agropecuária.

Tiragem: 90 exemplares

Não há uma definição de consenso para GCS. A definição adotada neste trabalho é baseada nos padrões propostos pelo IEEE (IEEE, 1994) e pelo SEI (Software ..., 1998), e estabelece GCS como *o* processo, no contexto de ciclo de vida de sistemas, que engloba os seguintes aspectos:

- *identificação da configuração de software* em pontos específicos de tempo: a identificação se refere tanto ao software como um todo, quanto a cada um de seus componentes. Informações de identificação englobam: nome, versão e documentação das características funcionais e físicas;
- *controle sistemático sobre mudanças na configuração:* este controle contempla a inserção e remoção de componentes, bem como alterações nas características de componentes;
- *registro e disponibilização de informações* relativas ao processo de mudança e ao estado de implementação das mudanças;
- *auditoria e revisão de configurações*: para verificar se um software está com sua estrutura (configuração) correta e completa, bem como se requisitos de mudanças foram atendidos pela implementação;
- *manutenção da integridade e da rastreabilidade de configurações:* a manutenção da integridade é realizada através de trabalho em equipe coordenado. Para isto, mecanismos como controle de concorrência devem ser utilizados. A manutenção da rastreabilidade é alcançada através do controle sistemático sobre configurações, aliado ao registro e à disponibilização de informações. Desta forma, pode-se obter um histórico de configurações de componentes e do software completo em pontos específicos de tempo, bem como das mudanças que originaram diferentes configurações.

Outras definições de GCS existentes na literatura (Appleton, 1998) agregam os aspectos a seguir ao processo de gerência de configuração:

- *gerência de construção* (*"building"*) de um produto de software: a construção de um produto de software está atrelada ao ambiente de implementação utilizado para o seu desenvolvimento. A configuração do software deve conter informações como a identificação do ambiente de implementação e das características necessárias para a sua operacionalização. Desta forma pode-se responder a questões tais como: "Que versões de arquivos e de ferramentas foram usadas para gerar esta última versão do produto?";
- *gerência de processo*: para assegurar que os procedimentos, as políticas e o modelo de ciclo de vida adotados em uma corporação estejam sendo executados corretamente pela equipe de desenvolvimento. Por exemplo, na empresa fictícia ACME mudanças na configuração de um determinado software são discutidas por um comitê. Desta forma, as mudanças sobre itens de configuração por desenvolvedores somente pode ser iniciada após a sua aprovação por este comitê. A incorporação das mudanças ao software ocorre de forma similar, isto é, somente após a aprovação da implementação das mudanças pelo comitê é que arquivos alterados podem ser armazenados num repositório de GCS;
- *trabalho em equipe*: para a coordenação e o controle das atividades realizadas pelos membros de uma equipe e das interações entre múltiplos desenvolvedores de um produto.

No contexto de GCS, o controle de versão de componentes é um aspecto de extrema relevância. Controle de versão combina procedimentos e ferramentas para gerenciar diferentes versões de um item de configuração. Para isto, um conjunto de atributos pode ser associado a uma versão (por exemplo, um rótulo que identifica o número da versão). Uma configuração pode ser especificada e construída através do estabelecimento de um conjunto de atributos de versões de componentes que reflita as características desejadas para tal configuração.

Correntemente há um número muito grande de ferramentas para suporte a controle de versão disponíveis. Funcionalidades destas ferramentas incluem:

- manipulação de versões de qualquer tipo de arquivo (por exemplo: código fonte, arquivos executáveis e documentos);
- avanço/recuo de versões;
- suporte ao desenvolvimento em paralelo (*"branching"*);
- rótulos para versões, visando identificação de atributos;
- visualização gráfica de diferenças;
- junção (*"merging"*) gráfica e dirigida de versões diferentes de arquivos para integração de esforços em paralelo;

- suporte a projetos (agregação de itens de configuração) e não somente a arquivos individuais;
- rastreamento de mudanças na estrutura de projetos.

A utilização de uma ferramenta para suporte ao controle de versão, adequada às necessidades de um projeto de desenvolvimento de software, é um passo inicial significativo para se estabelecer uma disciplina de gerência de configuração no escopo do mesmo. A seção seguinte trata do caso do subprojeto Informatização do Siger, caracterizando a situação de gerência de configuração antes de se implantar uma ferramenta para suporte ao controle de versão. Além disso, apresenta a ferramenta escolhida.

3 . Gerência de Configuração de Software no Subprojeto Informatização do Siger

O Sistema de Informação Gerencial da Embrapa (Siger) é um sistema corporativo de grande porte, que visa prestar suporte ao processo de gestão de pesquisa no Sistema Nacional de Pesquisa Agropecuária. Tal sistema engloba as fases de planejamento, acompanhamento, controle e avaliação de pesquisa. Estas fases são aplicadas sobre figuras programáticas definidas pelo Sistema Embrapa de Planejamento (SEP), que são: subprojeto, projeto, programa, PAT, PDU e PDE. O subprojeto de Informatização do Siger (Embrapa, 1997) está sendo executado na Embrapa Informática Agropecuária.

A primeira implantação do sistema computacional Siger ocorrerá no final do segundo semestre de 1998, sendo que várias versões do mesmo serão lançadas a partir de então. O conhecimento embutido naquele sistema é bastante amplo e complexo, tanto ao que diz respeito ao seu domínio de aplicação (gestão da pesquisa) quanto ao seu aspecto de informatização.

A equipe envolvida no desenvolvimento do sistema é relativamente numerosa, contando com mais de dez membros. Arquivos representando documentos, base de dados e código fonte são compartilhados por eles.

O quadro apresentado nos parágrafos anteriores mostra que procedimentos e ferramentas associados à gerência de configuração são prementes no subprojeto de informatização do Siger. Entretanto, durante uma fase inicial de sua existência, procedimentos rudimentares de GCS foram adotados. Na seqüência, estes procedimentos serão descritos.

Um diretório raiz no sistema de arquivos em uma estação de trabalho Sun foi estabelecido para conter arquivos compartilhados pela equipe. Tal diretório é visível pelos microcomputadores da equipe do Siger através do software SAMBA (Samba, 1998). Isto garante uma maior proteção sobre o acesso a arquivos (através dos mecanismos de segurança para um grupo de usuários de nome siger presentes no ambiente UNIX) e uma maior tolerância a falhas (devido à robustez das estações de trabalho).

Quando era necessário realizar a alteração de um dado arquivo (por exemplo, em um código fonte compartilhado), o membro da equipe responsável pela mudança copiava manualmente o arquivo da estação de trabalho para seu microcomputador local. O arquivo na estação de trabalho não ficava bloqueado para uso por outros membros. Desta forma, inconsistências poderiam ocorrer. Ao terminar a mudança o processo inverso era realizado: o mesmo membro copiava manualmente o arquivo de sua máquina local para a estação de trabalho. Mensagens eletrônicas eram enviadas para os demais membros notificando os dois eventos descritos anteriormente, a fim de tentar evitar inconsistências.

Um membro responsável foi definido para cada módulo do Siger. Antes de se realizar a alteração de arquivos compartilhados, inicialmente o membro responsável tinha que ser consultado e possíveis conflitos eram por ele resolvidos. Na estação de trabalho, apenas a versão mais recente de arquivos era armazenada. Portanto, versões intermediárias dos mesmos não poderiam ser recuperadas posteriormente. Somente grandes marcos de desenvolvimento, como por exemplo, a versão de todo o sistema Siger implantada para validação no final de 1997, eram armazenados em arquivos compactados.

Recentemente esta situação foi modificada através da implantação da ferramenta para suporte ao controle de versão denominada CS-RCS 2.0 (ComponentSoftware, 1998a, 1998b) e da adoção de procedimentos associados a GCS pela equipe de desenvolvimento do Siger. Na seqüência, a ferramenta implantada é apresentada.

O CS-RCS é baseado no RCS da Free Software Foundation (Free ..., 1998a, 1998b, 1998c) executado sob ambiente de rede Windows 95 ou Windows NT. A versão correntemente disponível é a 2.0.118. Funcionalidades básicas de controle de versão requeridas pela equipe do Siger são suportadas pela ferramenta, tais como:

- retirada de arquivos do repositório de controle de versão para alteração (*"check out"*) com controle de concorrência;
- adição de novas versões de arquivos ao repositório de controle de versão (*"check in"*) com controle de concorrência;
- navegação pelo histórico de versões de arquivos e visualização de versões;
- verificação do estado atual de arquivos (por exemplo, armazenado no repositório, em alteração, etc.);
- manipulação de marcos: associação de rótulos que identificam marcos de desenvolvimento a arquivos, visando recuperação futura;
- diferenciação de arquivos tipo texto e documentos Word/RTF, para comparação entre versões diferentes.

Outras capacidades da ferramenta incluem manipulação de arquivos texto e binário, suporte a projetos, controle de acesso simultâneo e integração com o ambiente de desenvolvimento interativo de aplicações Delphi 3.0. Finalmente, é importante ressaltar o suporte a ambiente de rede heterogêneo, através da integração com o software SAMBA (Samba, 1998). As características aqui mencionadas, aliadas a um preço competitivo, tornaram o CS-RCS a opção de ferramenta para suporte ao controle de versão adotada pelo subprojeto de Informatização do Siger.

A Fig. 1 ilustra a integração entre o CS-RCS e o Delphi 3.0, ferramenta correntemente utilizada em vários esforços de desenvolvimento de software na Embrapa Informática Agropecuária, incluindo o Siger. Através da opção Tools do menu principal do Delphi, várias operações de controle de versão disponíveis no CS-RCS podem ser executadas. Dentre elas pode-se observar: *"check in/out"* de arquivos, histórico de revisões e manipulação de projetos. Cabe ressaltar ainda que a integração com o Delphi 3.0 contempla apenas parte da funcionalidade global do CS-RCS e que ambos podem ser usados de forma complementar por equipes de desenvolvimento sem nenhum tipo de dificuldade.

Maiores detalhes sobre a utilização do CS-RCS no Siger podem ser obtidos em (Embrapa, 1998a). A próxima seção trata de aspectos de implantação do CS-RCS 2.0 e de procedimentos de trabalho adotados no subprojeto de Informatização do Siger a partir do uso desta ferramenta.

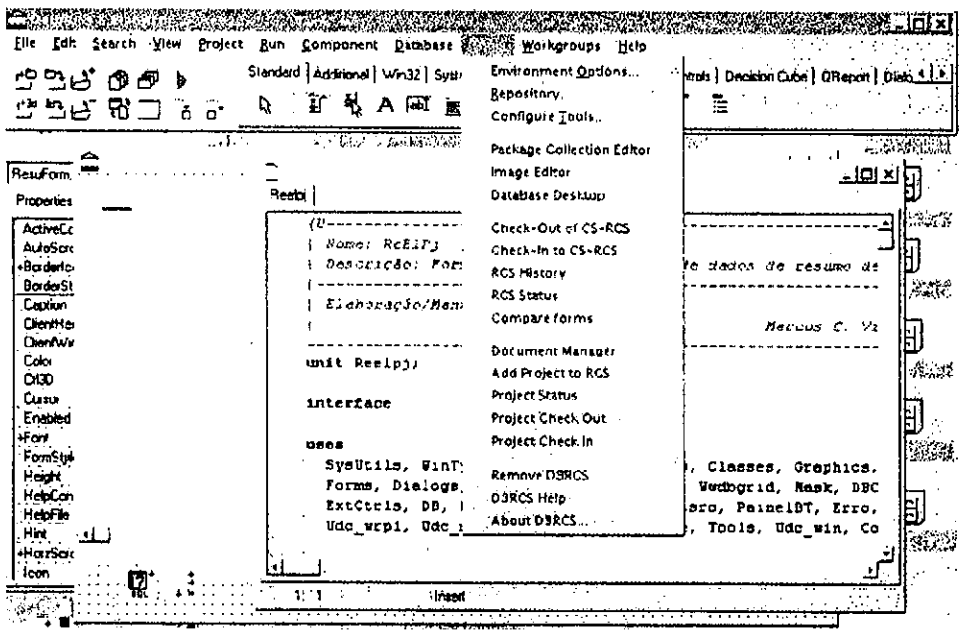

FIG. 1. Integração do CS-RCS ao Delphi 3.0.

4. Implantação de Controle de Versão no Subprojeto Informatização do Siger

Para que se entenda como é feito o controle de versão no subprojeto de Informatização do Siger é necessário que seu processo de implementação seja compreendido.

O Siger está sendo implementado em Delphi (versão 3.0) e é composto por vários módulos, tais como: módulo de elaboração de projeto, módulo de elaboração de subprojeto, módulo de acompanhamento de projeto, entre outros. Cada módulo é de responsabilidade de um único programador e organizado em diretórios específicos. Entretanto, existem arquivos que são comuns a todos os módulos. Podem ser units (código fonte Delphi, colocadas em um diretório denominado *comum*) ou componentes Delphi desenvolvidos para o Siger (armazenados em um diretório denominado *compsiger*).

Quando o módulo é compilado, seu arquivo executável (ou DLL) é colocado em um diretório denominado *bin*, que contém todos os arquivos necessários para a execução do Siger. Eventualmente o módulo é testado por uma equipe específica e, como resultado, um relatório de testes apontando os erros encontrados pode ser produzido. Baseando-se no processo de desenvolvimento até aqui descrito, foram criados projetos CS-RCS para cada um dos módulos do Siger como, por exemplo:

- S32FElProj : para o módulo de elaboração de projeto;
- S32FAcProj : para o módulo de acompanhamento de projeto;
- S32FComum : para os arquivos comuns;
- S32FCompSig : para os componentes do Siger;
- S32Bin : para os arquivos executáveis;
- DocSTeste : para os relatórios de testes.

Os projetos listados são suficientes para que seja descrita na seqüência a política de controle de versão do Siger. Suponha que o programador responsável pelo módulo de elaboração de projeto (que corresponde ao projeto S32FElProj do CS-RCS) tenha anteriormente enviado para equipe de testes uma versão de sua DLL (cujo nome é elabproj.dll) e tenha recebido um relatório apontando os erros encontrados. Para corrigir um erro, o programador deve retirar cópias bloqueadas (*"check out locked"*) dos arquivos que serão alterados. O bloqueio serve para assegurar que nenhum outro programador possa alterar simultaneamente os arquivos em questão.

Feitas as alterações, novas revisões dos arquivos podem ser armazenadas no repositório de controle de versão (*"check in"*) do projeto S32FElProj do CS-RCS. Eventualmente, é necessário fazer correções em algum arquivo de um projeto comum aos módulos (por exemplo, S32FComum). Neste caso, o programador retira uma cópia bloqueada do arquivo, faz as alterações necessárias, acrescenta ao repositório de versões (S32FComum, no exemplo) a nova revisão do arquivo e avisa ao demais integrantes da equipe de desenvolvimento sobre esta nova revisão. Note que, como o arquivo é usado por toda a equipe de desenvolvimento, esta alteração deve ser feita com bastante critério.

Em algum momento o programador julga que fez as correções suficientes em seu módulo e deseja enviar a nova versão de elabproj.dll para a equipe de testes. O programador deve, então, acrescentar ao repositório as novas revisões dos arquivos que tenha alterado e marcar as revisões mais recentes dos arquivos do projeto S32FElProj com o rótulo **VT***ddmmaa* (*ddmmaa* é a data em que o módulo foi enviado para a equipe de testes, por exemplo, VT210698). A seguir, o arquivo elabproj.dll (componente do projeto S32Bin do CS-RCS) deve ser retirado com opção de bloqueio (*"check out locked"*). O módulo é compilado, produzindo a nova revisão de elabproj.dll. Esta nova revisão é, então, devolvida ao repositório (*"check in"*) do projeto S32Bin do CS-RCS e marcada com o mesmo rótulo dos arquivos fontes (VT210698, no exemplo). Por fim, o programador avisa por e-mail a equipe de testes sobre a existência da nova versão do módulo, informando o seu rótulo (VT210698, no exemplo) para permitir a recuperação desta versão.

Assim sendo, a equipe de testes obtém (*"check out"*) uma cópia não bloqueada de elabproj.dll e uma cópia bloqueada do relatório de testes do módulo (projeto DocSTeste do CS-RCS). Novos testes são executados a fim de verificar se os erros foram corrigidos e se não foram introduzidos novos erros. O relatório de testes é alterado para indicar os erros novos e os persistentes. Finalizados os testes, a nova revisão do relatório é acrescentada ao repositório de controle de versão e marcada com o mesmo rótulo dos módulos (VT210698, no exemplo). O programador é avisado, retira a nova versão do relatório de testes e o processo é retomado até que os erros sejam definitivamente eliminados.

Note que, se desejar, o programador pode retirar uma cópia bloqueada do relatório de testes e, conforme corrige os erros apontados, anota como o erro foi corrigido ou informa que não conseguiu reproduzi-lo e, antes de enviar para equipe de testes a nova versão da DLL, devolve o relatório de testes ao repositório (*"check in"*).

Eventualmente, deseja-se produzir uma versão de demonstração do Siger. Para isto, escolhe-se uma versão de teste (marcada com um rótulo **VT***ddmmaa*) de cada um de seus módulos e marca-se tais versões com o rótulo **DEMO_***xxxx* (*xxxx* é qualquer cadeia de caracteres, por exemplo, DEMO_validacao). Para versões de distribuição o processo é o mesmo. Os módulos são marcados com um rótulo escolhido pelo gerente do projeto. Desta forma, rótulos são utilizados para identificar marcos e para facilitar a futura recuperação de versões de arquivos associados aos marcos.

Vale observar que os arquivos executáveis e DLL não precisariam estar armazenados no repositório de versões (S32Bin). Isto foi feito para facilitar o processo de testes, uma vez que a equipe de testes não participa da implementação do Siger. Seria pouco produtivo se tal equipe necessitasse compilar os módulos que fosse testar.

O processo de controle de versão do subprojeto de Informatização do Siger envolve outros aspectos de gerência de configuração do software, tais como versão do Delphi e das bibliotecas comerciais usadas para produzir uma determinada versão do Siger. Estes aspectos estão abordados em um documento interno do subprojeto (Embrapa, 1998b).

5. Conclusões

Embora gerência de configuração de software seja um processo amplo, no subprojeto de Informatização do Siger o foco corrente está no aspecto de controle de versão. Muitos passos devem ainda ser trilhados neste contexto para o emprego de GCS de uma forma completa e automatizada no referido subprojeto, o que será realizado de forma gradual.

No que diz respeito ao controle de versão, após um período inicial de implantação no Siger, uma revisão dos procedimentos internos de trabalho e do uso da ferramenta selecionada foi conduzida. Nesta revisão, vários membros da equipe aprovaram a escolha da ferramenta CS-RCS e validaram os procedimentos de trabalho empregados.

O CS-RCS contempla funcionalidades básicas desejadas de controle de versão e atende aos requisitos de desempenho esperados. Tais requisitos englobam tempo médio de *"check in"* e *"check out"* satisfatórios. O fato do CS-RCS realizar corretamente a manutenção e a recuperação de versões de arquivos e de projetos traz uma grande confiabilidade àquela ferramenta. Além disso, a automatização das operações de controle de versão aliada ao controle de acesso concorrente garantem a consistência dos arquivos e das versões manipuladas.

Este conjunto de fatores trouxe um avanço sobre os procedimentos de trabalho relacionados a GCS empregados anteriormente, onde não havia manutenção de versões intermediárias nem suporte automatizado ao controle de versões. Adicionando-se a este quadro, existia um fator agravante: a falta de controle de acesso concorrente, que poderia gerar grandes inconsistências sobre documentação e código do Siger.

Desta forma um processo de desenvolvimento mais sistematizado e confiável se consolidou no subprojeto de Informatização do Siger, implicando em melhoria na qualidade do sistema Siger (produto deste processo).

Por fim, cabe ressaltar que a ferramenta CS-RCS pode ser empregada por outros esforços de desenvolvimento de sistemas baseados em ambiente Windows na Embrapa Informática Agropecuária e que não necessariamente usem o Delphi. Espera-se, entretanto, que outras capacidades relevantes possam ser incorporadas a esta ferramenta (tais como *"branching"* e *"merging"* gráfico) rapidamente para que ela se torne uma solução mais completa.

6. Referências Bibliográficas

APPLETON, B. SCM definitions page. Disponível: *Brad Appleton* site. URL: http://www.enteract.com/~bradapp/acme/scm-defs.html Consultado em 10 nov. 1998.

COMPONENTSOFTWARE. *CS-RCS ComponentSoftware RCS, version 2.0:* user's guide. [S.l.], 1998a.

COMPONENTSOFTWARE. What is CS-RCS?? Disponível: *ComponentSoftware* site (20 Dec. 1998). URL: http://www.componentsoftware.com/csrcs Consultado em 22 jun. 1998b.

EMBRAPA. Centro Nacional de Pesquisa Tecnológica em Informática para a Agricultura (Campinas, SP). *Guia de utilização do CS-RCS 2.0.115 no Projeto Siger.* Campinas, 1998a. Documento interno elaborado para o Subprojeto 14.0.97.910.03.

EMBRAPA. Centro Nacional de Pesquisa Tecnológica em Informática para a Agricultura (Campinas, SP). *Informatização do SIGER Programa, PDU e PDE.* Campinas, 1997. 19p. (EMBRAPA. Programa 14 Intercâmbio e Produção de Informação em Apoio às Ações de Pesquisa e Desenvolvimento. Subprojeto 14.0.97.910-03). Projeto em andamento.

EMBRAPA. Centro Nacional de Pesquisa Tecnológica em Informática para a Agricultura (Campinas, SP). *Proposta para gerência de configuração no Projeto Siger.* Campinas, 1998b. Documento interno elaborado para o Subprojeto 14.0.97.910.03.

FREE SOFTWARE FOUNDATION. GNU's Not Unix! Disponível: *Free Software Foundation* site. URL: http://www.fsf.org Consultado em 22 jun. 1998a.

FREE SOFTWARE FOUNDATION. Overview of the GNU project. Disponível: *GNU's not Unix!* site. URL: http://www.fsf.org/gnu/gnu-history.html Consultado em 22 jun. 1998b.

FREE SOFTWARE FOUNDATION. What is the Free Software Foundation? Disponível: *GNU's not Unix!* site URL: http://www.fsf.org/fsf/fsf.html Consultado em 22 jun. 1998c.

IEEE (New York, Estados Unidos). *Software engineering standards, 1994 edition.* Los Alamitos: IEEE Computer Society, 1994.

SAMBA. Welcome to the Samba web pages. Disponível: *Samba* site. URL: http://samba.anu.edu.au/samba Consultado em 10 nov. 1998.

SOFTWARE ENGINEERING INSTITUTE. Software configuration management: a key process area for level 2 repeatable. Disponível: *The SEI Capability Maturity Model* site http://rbse.jsc.nasa.gov:80/process_maturity/CMM/TR25/tr25_l2f.html Consultado em 10 nov. 1998.

7. Palavras-chave

Gerência de configuração de software; controle de versão; CS-RCS.

IMPRESSO

Empresa Brasileira de Pesquisa Agropecuária
Centro Nacional de Pesquisa Tecnológica em Informática para a Agricultura
Ministério da Agricultura e do Abastecimento
Rua Dr. André Tosello, s/nº Caixa Postal 6041 - Barão Geraldo
13083-970 - Campinas, SP
Fone (019) 289-9800 Fax (019) 289-9594