

Apoio à Gerência de Configuração de Artefatos de Software integrado a Execução de Processos de Software

Ernani Sales, Carla Lima Reis, Rodrigo Quites Reis

Programa de Pós-Graduação em Ciência da Computação - Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará (UFPA) – Belém – PA – Brasil

ernani@webapsee.com, clima@ufpa.br, quites@computer.org

Resumo. *Controle de Versões, Controle de Modificações, Apoio à Workspace, Gerenciamento da Construção, Auditoria e Contabilidade são conceitos importantes para a Gerência de Configuração de Software (GCS). Este artigo descreve uma infra-estrutura que implementa tais conceitos de forma integrada a um Ambiente de Desenvolvimento de Software Centrado em Processo (ADS). A principal contribuição desta proposta é integrar processos e técnicas de GCS à execução do processo de software, facilitando sua implantação em organizações de desenvolvimento de software.*

Abstract. *Version Control, Change Control, Workspace Control, Development Management, and Auditing and Accounting are important concepts for Software Configuration Management (SCM). This paper describes an infrastructure to deploy these concepts, focusing on integration to a Process-centered Software Engineering Environment (PSEE). The main contribution of this work concerns on integrating the SCM processes and techniques to the software process enactment in order to make SCM deployment easier at software development organizations.*

1. Introdução

O aumento da complexidade do software desenvolvido nos dias atuais reflete na complexidade da documentação produzida no desenvolvimento de software [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 projeto [ISO 1995]. Dessa forma, o processo de gerência de configuração é transversal ao processo de desenvolvimento de software, podendo ser executado durante todas as suas etapas [Pressman 2005].

Diversas normas e modelos de maturidade de processo consideram a GCS como um processo chave para o aumento de qualidade [ISO 1995; Chrissis *et al.* 2003; SOFTEX, 2008]. Além disso, normas e diretrizes foram definidas para melhorar os processos específicos de GCS [IEEE 1987; IEEE 2005; Asklund & Bendix 2003].

Esses modelos definem atividades e procedimentos necessários durante o processo de software para prover GCS de maneira eficiente. Contudo, 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 GCS.

Estublier *et al.* (2005) aponta a área de Apoio ao Processo em GCS como sendo a mais desejada e criticada característica presente em ferramentas de GCS. Essa constatação indica que é preciso melhorar as tecnologias de apoio ao processo em GCS. Estublier afirma, ainda, que há uma necessidade de se definir formalismos específicos e de alto nível em que processos de GCS possam ser modelados, executados, adaptados, medidos e otimizados. Tal abordagem visa integrar o modelo orientado a atividades, peculiar a processos de software, com o modelo orientado a produto, intrínseco à GCS.

Além disso, Estublier *et al.* (2005) enfatiza que é necessário definir conceitos e interfaces mais simples para a tecnologia de apoio a processo em GCS para que a mesma seja utilizada na prática. Também ressalta que o progresso na área de apoio a processo em GCS está entre outros pontos numa melhor integração com outras ferramentas, tais como gerenciadores de projetos, *call tracking*, entre outras. Portanto, confronta a idéia de que os sistemas de GCS são aplicações independentes dentro do contexto de desenvolvimento de software.

Vale ressaltar ainda os benefícios da evolução na área de Apoio ao Processo em GCS, além do controle das mudanças do software: redução dos custos, melhoria da acurácia dos prazos e estimativas, melhoria da qualidade do software, e melhoria da repetibilidade de sucesso bem como diminuição das possibilidades de falhas no desenvolvimento de software [Paulk *et al* 1993].

Do ponto de vista de automação existem diversos tipos de tecnologias voltadas especificamente à GCS. Entretanto, a maior parte é voltada para o gerenciamento de código-fonte e/ou é integrada a ferramentas CASE (*Computer Aided Software Engineering*) de baixo nível, isto é, que apóiam somente as fases de codificação e testes [Bruegge *et al.* 2006]. Portanto, existe um campo de pesquisa em aberto referente a tecnologias de GCS com apoio a níveis de abstração mais altos no desenvolvimento de software e que consigam fornecer mecanismos para representar os artefatos de software mais complexos de maneira efetiva.

Por outro lado, a tecnologia de processos de software também tem produzido coleções de ferramentas [Ben-Shaul *et al* 1997] [Yan *et al* 2003] [Lima Reis 2003] [Montoni *et al* 2006] integradas que facilitam as atividades de Engenharia de Software, durante todo o ciclo de vida do software ou, pelo menos, em porções significativas dele [Harrison *et al.* 2000], categorizadas como Ambientes de Desenvolvimento de Software (ADS). Tais ferramentas fornecem apoio gerencial ao modelo de processo de software, o qual descreve detalhadamente todos os aspectos que fazem parte de um processo de desenvolvimento, como: procedimentos, recursos, pessoas, prazos, artefatos e políticas organizacionais. Dependendo do nível de detalhe e do formalismo utilizado é possível executar tal processo, através da coordenação das tarefas modeladas, alocação dos recursos e pessoas necessárias, fornecimento de artefatos e monitoração do andamento das tarefas.

Segundo Abdala (2006) e Murta (2006), o desenvolvimento de tecnologias de apoio à GCS integradas a tecnologias de processos de software configura-se como uma contribuição para a melhoria da área de apoio ao processo de GCS. A idéia é reutilizar os conceitos existentes dentro de ambientes de processos de software e integrar com serviços que dêem apoio à GCS, contemplando controle de versões, controle de mudanças, gerenciamento da construção, entre outros.

Assim, este artigo apresenta uma infra-estrutura para apoio a gerência de configuração dos artefatos de software integrada a um ambiente de desenvolvimento de software centrado em processos denominado WebAPSEE [Lima *et al.* 2006]. Essa infra-estrutura possui um modelo de representação dos artefatos de software própria que provê apoio a modelagem e execução de processos de GCS de forma paralela a execução de processos de software, além de possibilitar o controle de versões, apoio a *workspace*, gerenciamento da construção, entre outras características.

O texto é organizado como segue. Na seção 2, alguns conceitos importantes em GCS para a descrição deste trabalho são apresentados. Na seção 3, o modelo proposto para apoio da gerência de configuração dos artefatos de software integrada ao ambiente WebAPSEE é descrito de maneira detalhada. Na seção 4, um estudo de caso com a utilização do modelo proposto em contexto real é apresentado. Na seção 5, são discutidos os trabalhos relacionados. Por fim, na seção 6, são discorridas as considerações finais.

2. Gerência de Configuração de Software

Segundo Estublier (2000), Gerência de Configuração (GC) é a disciplina responsável pela gerência das modificações ocorridas em sistemas complexos e de grande porte. Portanto, seu objetivo principal é gerenciar e controlar as numerosas correções, extensões e adaptações que são aplicadas a um sistema durante o seu ciclo de vida. Por sua vez, Gerência de Configuração de Software (GCS) trata especificamente o software, tendo como objetivo garantir um sistemático e rastreado processo de desenvolvimento em que todas as mudanças são precisamente gerenciadas para manter a estabilidade do software.

Para atender esses objetivos, diversos estudos foram realizados e difundidos por influência tanto do meio acadêmico quanto industrial. As subseções seguintes descrevem os requisitos para a automação de software e as perspectivas de GCS, enfatizando a relação entre GCS com ambientes de desenvolvimento de software.

2.1. Requisitos para automação de GCS

Toda tecnologia que objetiva fornecer apoio à GCS deve prover um vasto conjunto de funcionalidades de alto nível gerencial e tecnológico. Dart (1991) classificou essas funcionalidades em oito categorias relacionadas, sendo que cada uma refere-se a uma distinta necessidade de usuário dentro das perspectivas de GCS. A seguir serão descritas as categorias de funcionalidades de sistemas de GCS, segundo Dart (1991) e estudos empíricos de Estublier *et al* (2005):

- **Componentes:** apoio a identificação, classificação, armazenamento e acesso dos componentes que compõem o software. Tais atribuições envolvem o

gerenciamento de múltiplas versões, estabelecimento de linhas base¹ e configurações, e a manutenção de todas as informações relativas ao software e ao seu projeto de desenvolvimento;

- **Estrutura:** apoio a representação e utilização da arquitetura do software, identificando o relacionamento entre todas as suas partes em termos, por exemplo, de interfaces;
- **Construção:** apoio a construção de um software executável a partir de arquivos fonte versionados, de maneira eficiente, bem como dos artefatos de software envolvidos durante esse desenvolvimento. Além disso, deve ser possível retornar para versões antigas do software e de seus artefatos;
- **Auditoria:** apoio ao registro de todas as informações relativas às mudanças ocorridas no software e seus artefatos durante o seu ciclo de vida, tais como: o quê, quem, quando e por que houve a mudança;
- **Contabilidade:** apoio a obtenção de estatísticas relativas ao software e do processo seguido em seu desenvolvimento;
- **Controle:** apoio a análise do impacto de mudanças, permitindo especificar quais artefatos de software devem ser modificados de forma rastreada desde os requisitos funcionais até o código. Além de ferramental para a implementação do processo de requisição de mudanças e de localização de defeitos;
- **Processo:** apoio ao gerenciamento de como o software evolui, selecionando tarefas a serem executadas e executando-as dentro do contexto do processo de desenvolvimento de software;
- **Equipe:** apoio a condições necessárias para o trabalho colaborativo, possibilitando a identificação e tratamento de conflitos, além de *workspaces* individuais e de grupo.

2.2. Perspectivas de GCS

Segundo Murta (2006), a GCS pode ser tratada sob diferentes perspectivas em função do papel exercido pelo participante do processo de desenvolvimento de software: a perspectiva gerencial, a perspectiva de desenvolvimento de software e a perspectiva de integração.

Na **perspectiva gerencial**, a GCS é dividida em cinco funções definidas em [IEEE 2005], que são: identificação da configuração, controle da configuração, contabilização da situação da configuração, avaliação e revisão da configuração e gerenciamento de liberação e entrega.

Na **perspectiva de processo de desenvolvimento de software**, a GCS é dividida em três sistemas principais, que são: controle de modificações, controle de versões e gerenciamento de construção.

Na **perspectiva de integração**, existe a integração dos espaços de trabalho das duas perspectivas anteriores para abordar a GCS com o objetivo de se sobreporem. Nessa integração, as cinco funções descritas na perspectiva gerencial podem ser

¹ O termo linha base (*baseline*) representa um conjunto de ICs formalmente aprovados que serve de base para as etapas seguintes de desenvolvimento [IEEE 1990].

implementadas pelos três sistemas descritos na perspectiva de desenvolvimento, acrescidos de alguns procedimentos manuais.

Portanto, para que esses recursos sejam efetivamente utilizados, alguns serviços especializados devem ser definidos na integração dos espaços de trabalho de GCS e Ambientes de Desenvolvimento de Software (ADS). A Figura 1 ilustra a interação entre as perspectivas apresentadas.

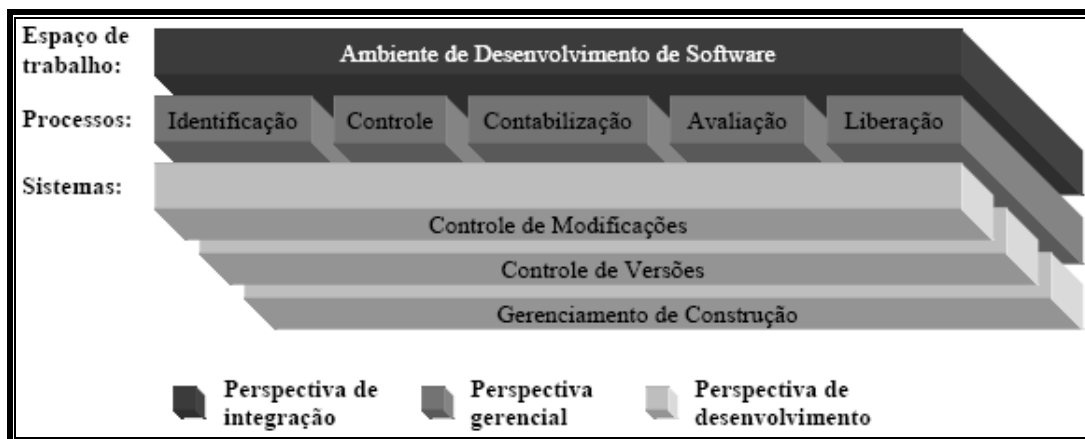


Figura 1 - Perspectivas para GCS [Murta 2006]

3. A Infra-Estrutura Proposta

Nessa seção é apresentada de forma detalhada a infra-estrutura proposta para apoio à GCS integrada ao ambiente WebAPSEE. Primeiramente, uma breve descrição do Ambiente WebAPSEE e seus artefatos é explicitada. Em seguida, uma visão geral da infra-estrutura e o detalhamento dos seus aspectos essenciais são apresentados.

3.1. Ambiente WebAPSEE

O ambiente WebAPSEE [Lima Reis e Reis 2007] é um ADS Centrado em Processos baseado em tecnologias de distribuição que provê serviços via internet para seus clientes para gerenciar e desenvolver processos de software. Nesse ambiente, destaca-se o compromisso de flexibilizar a execução do processo dinamicamente, isto é, em tempo de execução é possível que se altere o processo, excluindo atividades, deslocando pessoas e recursos de atividades e inserindo em outras, por exemplo.

Além de fornecer apoio automatizado para a gestão de processos de software, o ambiente WebAPSEE foi projetado para permitir a integração de vários serviços relacionados com uma visão bastante ampla do meta-processo de software. Isto é, a ferramenta apóia desde a modelagem inicial e abstrata de um processo até a sua análise *post-mortem*, passando pelo controle da execução dos processos - implementando um modelo de execução descrito por Lima Reis (2003).

3.2. Artefatos

Uma das principais características de ambientes de desenvolvimento de software (ADS) é armazenar, estruturar e controlar as informações, produtos de software e os objetos produzidos durante o processo de desenvolvimento de software. Esses objetos, denominados de artefatos de software, são por natureza, mais complexos do que os objetos tratados por sistemas de banco de dados tradicionais, pois são mais estruturados

e requerem operações complexas. Além disso, são fortemente inter-relacionados e sensíveis ao tempo (alguns objetos possuem versões) espaço (as versões podem estar distribuídas) [Lima Reis 2003].

Dessa forma, um ADS deve prover um módulo responsável por controlar o acesso e a evolução de artefatos de software. Geralmente este módulo é conhecido como repositório do ambiente e é implementado através de um sistema de gerência de banco de dados (SGBD). As versões dos documentos e produtos de software devem ser gerenciadas para permitir cooperação e consistência.

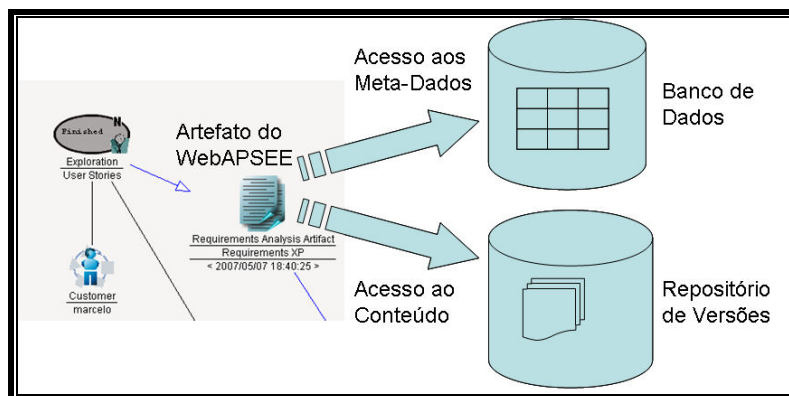


Figura 2 - Representação de Artefatos no WebAPSEE

Um artefato do WebAPSEE é representado por uma conexão de artefato que permite ligação com atividades (entrada e saída). As propriedades (meta-dados) do artefato são armazenadas no banco de dados, enquanto seu conteúdo (arquivo ou artefato propriamente dito) é armazenado em um repositório de artefatos. A Figura 2 mostra como estão distribuídas as informações sobre artefatos no ambiente WebAPSEE.

3.3. Visão Geral da Infra-estrutura

A infra-estrutura proposta neste trabalho visa fornecer processos e ferramental de apoio à GCS aplicadas aos artefatos de software dentro do contexto do ambiente WebAPSEE. Para tal foram definidas as seguintes macro-funcionalidades:

- **Controle de Versões**, a partir de uma interface de comunicação que acessa serviços de controle de versões de ferramentas externas através dos clientes do WebAPSEE (Console do Gerente e Agenda do Desenvolvedor) de maneira transparente ao usuário. Os principais serviços fornecidos são check-in, check-out (versão atual e anteriores) e log de versões, implementadas para CVS [CVS 2008] e Subversion [SVN 2008];
- **Controle de Modificações**, com a modelagem e execução de processos de controle de mudanças baseados no padrão IEEE Std 828 para gerenciar as solicitações de mudança relacionadas aos artefatos de um processo em execução no ambiente WebAPSEE. Dessa forma, o gerente defini um Plano de GCS e o acompanha via editor de processos e interferindo no fluxo de execução do processo;
- **Apoio a Workspace**, através de uma representação dos artefatos de software isolada do repositório de artefatos do ambiente e que permite a interação do desenvolvedor. Na Agenda do Desenvolvedor, o usuário pode verificar quais os

artefatos que ele precisa manipular em uma dada tarefa, e percebe a evolução desses artefatos no repositório do ambiente através de notificações (versão local X versão remota);

- **Gerenciamento da Construção**, a partir do gerenciamento dos artefatos de software, que permite a visualização e edição de rastros (conceitos de dependência e composição) entre artefatos, configurações, linha base, dentre outros implementados no modelo de dados do ambiente WebAPSEE;
- **Auditoria e Contabilidade da GCS**, a partir do registro e visualização das informações relativas à evolução dos artefatos do ambiente WebAPSEE, permitindo a geração de relatórios gerenciais, tais como: estado atual e histórico dos artefatos, versões aprovadas, estado das solicitações de mudança, estado de implementação da mudança, impacto da mudança, etc.

Vale ressaltar que essa infra-estrutura proposta é uma evolução do modelo apresentado em [Sales 2007]. O controle de modificações foi alterado para prover o apoio automatizado a processos de mudança baseados em normas/padrões atuais. O modelo de representação de artefatos foi redefinido para tratar os conceitos relativos à definição e acompanhamento de processos de GCS pelo ambiente WebAPSEE. Além disso, a interface definida para o controle de versões foi implementada para a ferramenta SubVersion [SVN 2008], além da já existente para o CVS [CVS 2008]. O apoio a *workspace* permaneceu da mesma forma.

3.4. Controle de Modificações

A implantação do controle de mudanças em uma organização permite ao gerente acompanhar e gerenciar todas as modificações que são realizadas nos artefatos de software durante o seu ciclo de vida. Para isso, é necessário que seja executado um processo de realização da mudança e que as pessoas envolvidas sejam informadas dessa mudança.

A proposta atual fornece um *template* de processo [Sales 2006] pré-definido para gerenciar mudanças, baseado no padrão IEEE Std 828, que é executado toda vez que uma solicitação de mudança em um artefato aprovado (pertencente à linha base do processo de desenvolvimento) é criada. Assim, o processo de mudança de um artefato ocorre paralelamente ao processo principal. Um detalhe importante desse controle de modificações é o planejamento do processo de GCS para cada um dos artefatos de software de um dado processo. Essa funcionalidade permite que o gerente de configuração, através de um formulário, defina os itens que estarão sendo controlados, de que forma, e atribuindo responsabilidades e papéis aos envolvidos no processo de acordo com o controle de modificações estabelecido. Ao final desse planejamento é gerado um Plano de GCS para o processo.

O *template* fornecido contém as seguintes atividades: (1) solicitação de modificação, iniciando um ciclo da função de controle para uma dada manutenção; (2) classificação da modificação, que estabelece a prioridade da solicitação em relação às demais solicitações efetuadas anteriormente; (3) análise de impacto, que visa relatar os impactos em esforço, cronograma e custo e definir uma proposta de implementação da

manutenção; (4) avaliação da modificação pelo Comitê de Controle da Configuração² (CCC), que estabelece se a modificação será implementada, rejeitada ou postergada, em função do laudo fornecido pela análise de impacto da modificação; (5) implementação da modificação, caso a solicitação tenha sido aprovada pela avaliação da modificação; (6) verificação da modificação com relação à proposta de implementação levantada na análise de impacto; e (7) atualização da linha base, que pode ou não ser liberada para o cliente em função da sua importância e questões de marketing associadas.

Para possibilitar a execução desse processo de mudança foi necessária a criação de uma máquina de estados para os artefatos e a extensão do conjunto de regras de execução do ambiente WebAPSEE. A Figura 3 mostra os possíveis estados que um artefato pode assumir durante o seu ciclo de vida e a Figura 4 apresenta uma das regras de execução definidas para tratar a interação do processo de mudança de um artefato com a execução do processo de desenvolvimento principal.

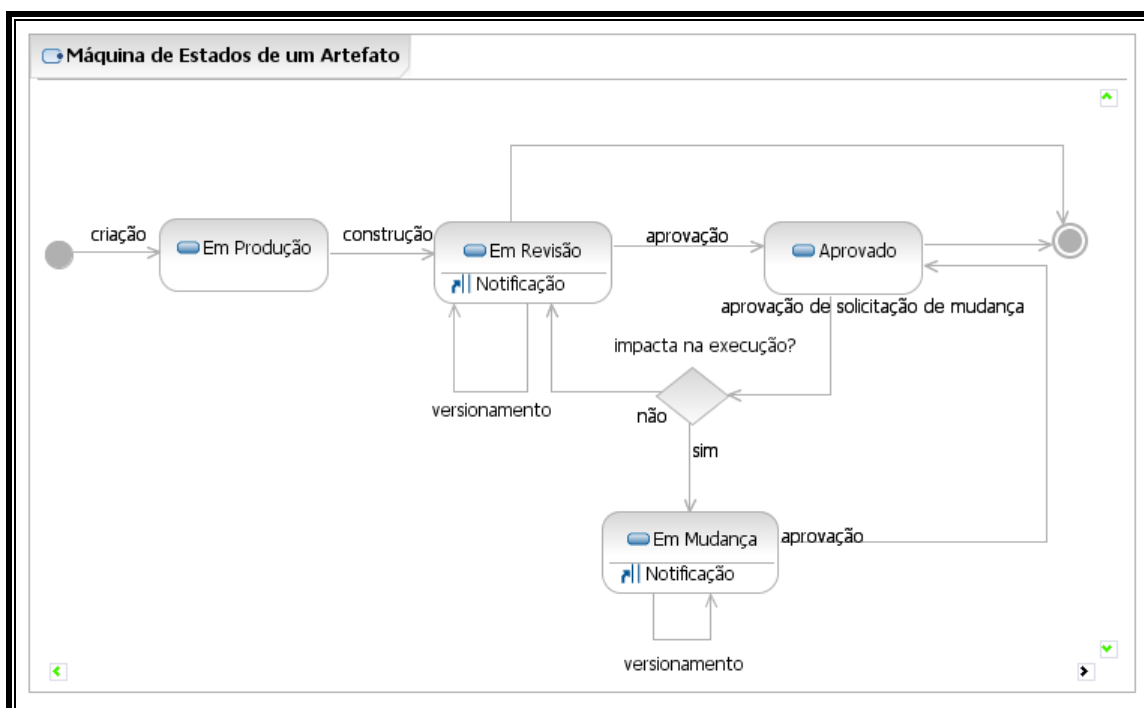


Figura 3 - Máquina de Estados de um artefato

Inicialmente, quando um artefato é criado no ambiente WebAPSEE (definição dos meta-dados do artefato) este entra no estado em produção visto que o seu conteúdo ainda será produzido durante a execução de um processo de software. Após a construção e envio do artefato para o repositório do WebAPSEE, o artefato passa para o estado em revisão, sendo que todas as vezes que entra nesse estado é feita a notificação para os interessados da mudança, através da visualização do artefato nos clientes do ambiente WebAPSEE. A partir desse estado o artefato está sob controle de versões, sendo atualizado toda vez que uma nova versão do seu conteúdo for desenvolvida. Após a aprovação explícita de um artefato e submissão do mesmo para a linha base do processo em execução, o estado vai para entregue e aprovado.

² O termo comitê de controle da configuração (*configuration control board*) representa o grupo de pessoas responsável por avaliar e aprovar ou reprovar modificações propostas para ICs e por assegurar a implementação das modificações aprovadas [IEEE 2005].

A partir desse momento, solicitações de mudança podem ser feitas ao artefato, iniciando um processo de controle de mudança. Caso a modificação seja aprovada, deve ser indicado se esse processo de mudança vai ou não impactar na execução do processo de software principal. Essa indicação é feita através do interpretador de condições lógicas do ambiente WebAPSEE que solicita explicitamente ao gerente uma tomada de decisão em um ponto do fluxo de execução do processo. Se a resposta for sim, o estado assumido é em mudança, impactando na execução do processo de software principal de acordo com as regras de interação definidas. Se a resposta for não, volta para o estado em revisão, podendo ser atualizado até uma nova aprovação e entrega para a linha base.

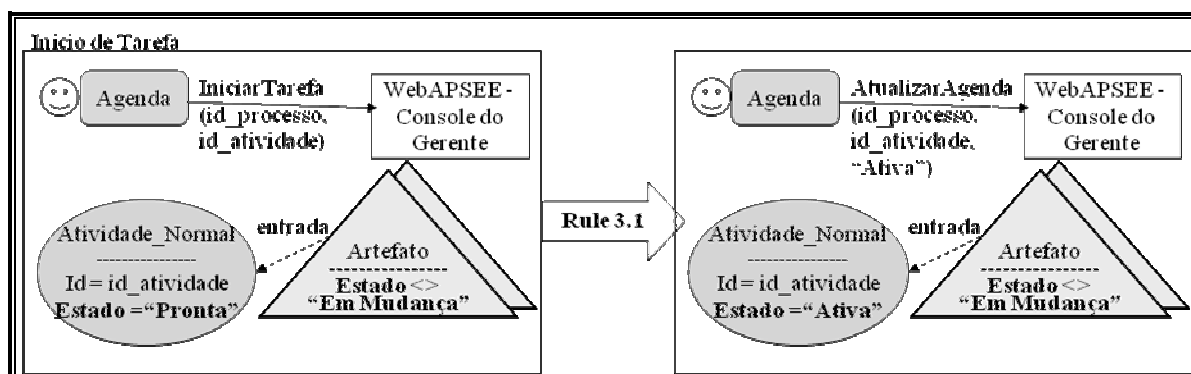


Figura 4 – Exemplo de regra de execução para controle de mudanças

As regras de execução foram criadas de acordo com o formalismo de Gramática de Grafos definido em Lima Reis (2003). No exemplo mostrado (Figura 4), temos uma restrição imposta à ação de iniciar de uma atividade do processo de software em execução. Nesse caso, para iniciar a tarefa é necessário que todos os artefatos de entrada para essa atividade não estejam no estado em mudança. Dessa forma, se o gerente decidir que uma mudança impacta na execução do processo de software principal, as atividades que envolvem os artefatos em mudança ficam esperando a finalização do processo de mudança.

3.5. Gerenciamento da Construção

O gerenciamento da construção envolve o processo de construção do software, produzindo itens de configuração derivados a partir de itens de configuração fonte; liberação, identificando as versões particulares de cada item de configuração que são disponibilizadas; e entrega, implantando os produtos de software no ambiente real.

Na abordagem proposta é fornecido um apoio em alto nível de abstração. Dessa forma, utiliza-se os relacionamentos de composição e dependência de artefatos de software presentes no modelo de dados do WebAPSEE para possibilitar a construção, liberação e empacotamento de artefatos de software compostos de forma automática. Além disso, é permitida a gerência de linhas base para manter a consistência da gama de artefatos de software envolvidos em um processo.

Outro aspecto importante é o mecanismo de definição e manutenção de rastros entre os artefatos, que auxilia na evolução dos artefatos de software de forma controlada, além de informar ao usuário o impacto de uma mudança dentro do contexto de controle de modificações. O modelo utilizado para implementar essas funcionalidades é apresentado na Figura 5.

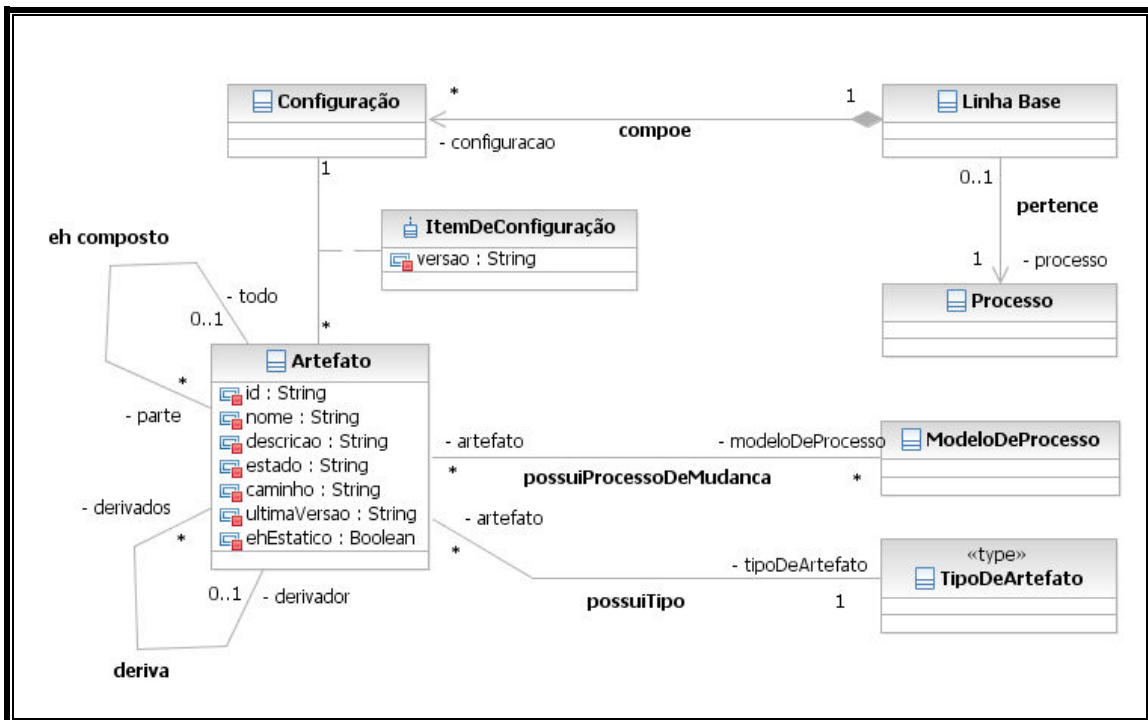


Figura 5 - Meta-dados de um artefato do WebAPSEE

As propriedades do artefato e seus relacionamentos são detalhados a seguir:

- Um **identificador** único, uma **descrição**, uma **denominação** mais completa, sua **localização** (caminho) no repositório de versões, a **última versão** do artefato e uma **flag** (ehEstatico) utilizada pelo mecanismo de definição de políticas estáticas do ambiente WebAPSEE;
- Um **tipo** na hierarquia de tipos de artefatos (mais detalhes em [Lima reis 2003]), que representa a categoria à qual o artefato pertence;
- Uma **relação de derivação** entre artefatos, indicando qual artefato serviu de base para a construção deste. Utilizado para representar rastreabilidade;
- Uma **relação de composição** entre artefatos, indicando que um artefato é composto de outros. Utilizado para auxiliar a construção de releases.
- Um conjunto de **modelos de processo**, que possibilita a definição e execução de processos de mudança ao artefato.
- Um artefato também pode pertencer a uma **configuração** sendo registrada a versão do **item** dessa configuração. Além disso, temos a representação da **linha base** que é composta de um conjunto de configurações e pertence a um **processo** de software.

3.6. Auditoria e Contabilidade da GCS

A auditoria e a contabilidade da GCS visam armazenar as informações geradas por todas as manipulações feitas com o software e seus artefatos, permitindo que essas informações possam ser acessadas em função de necessidades específicas. Essas necessidades específicas abrangem o uso de medições para a melhoria do processo, a estimativa de custos futuros e a geração de relatórios gerenciais.

O modelo proposto visa estender o módulo de registro de eventos do ambiente WebAPSEE, apresentado em [Paxiúba *et al* 2005], para capturar eventos relativos ao planejamento, modificações, versionamento, e outras operações realizadas sobre os artefatos para permitir a auditoria da GCS durante o ciclo de vida de um processo de desenvolvimento de software.

Além disso, o editor de processos do ambiente WebAPSEE possibilita visualizar o estado dos artefatos pela utilização de ícones gráficos (Figura 6), os processos de mudança em execução, e relatórios gerenciais obtidos automaticamente para auxiliar na melhoria dos processos da organização.

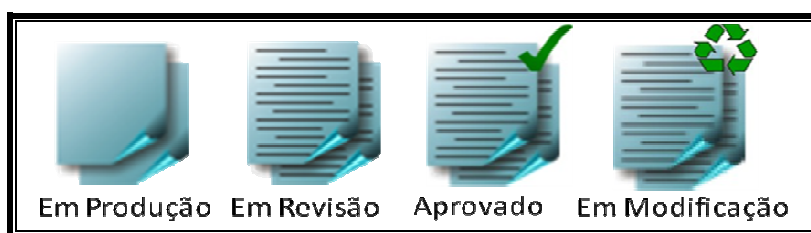


Figura 6 - Ícones Gráficos para representar estados do artefato

4. Exemplo de Utilização

A infra-estrutura proposta foi utilizada para gerenciar os processos de mudança dos artefatos envolvidos durante a execução de um processo de desenvolvimento de um sistema. Esse processo foi realizado durante a disciplina de Laboratório de Engenharia de Software do Curso de Ciência da Computação da Universidade Federal do Pará no segundo semestre de 2007.

Inicialmente, um modelo de processo para o desenvolvimento do sistema foi modelado no editor de processos do ambiente WebAPSEE. Além disso, o gerente definiu o Plano de Gerência de Configuração para o processo, acessando o formulário de planejamento de GCS. Nesse formulário são selecionados os artefatos pertencentes ao processo a terem sua evolução controlada (definição da linha base para o processo) e a seleção dos recursos (humanos e materiais) envolvidos no processo de controle de mudanças (Gerente de Configuração, Comitê de Controle de Configuração, etc). No final desse planejamento é gerado o Plano de Gerência de Configuração de Artefatos desse processo pelo ambiente.

O subprocesso descrito na Figura 7 está em execução com a atividade *Planning Tests* ativa e a atividade *Coding Use Cases* pronta para ser iniciada a qualquer momento pelo grupo de desenvolvedores alocado. Nesse momento, é feita a solicitação de mudança do artefato *Requirements Specification*, que está incluso na linha base do processo, tem como componente o artefato *Use Case Diagram* e deriva os artefatos *Test Cases* e *System Design* (relacionamentos de composição e derivação não são mostrados visualmente).

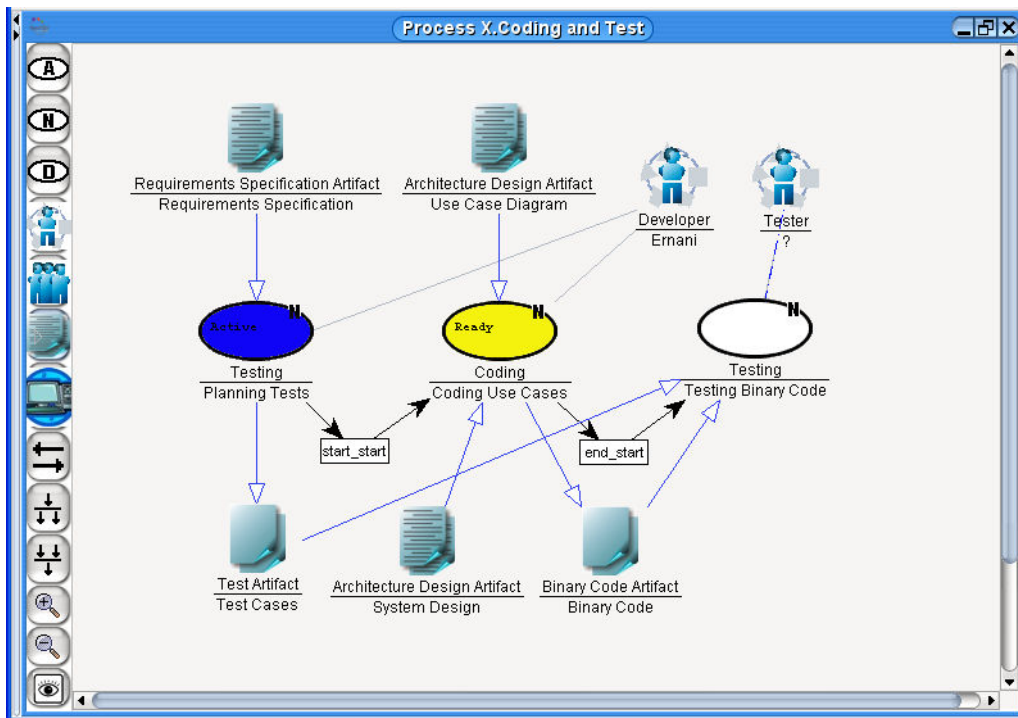


Figura 7 - Subprocesso de codificação em execução no ambiente WebAPSEE

Dada essa solicitação um processo de mudança é instanciado, de acordo com o plano definido para a gerência de configuração do processo, e entra em execução para viabilizar a mudança solicitada.

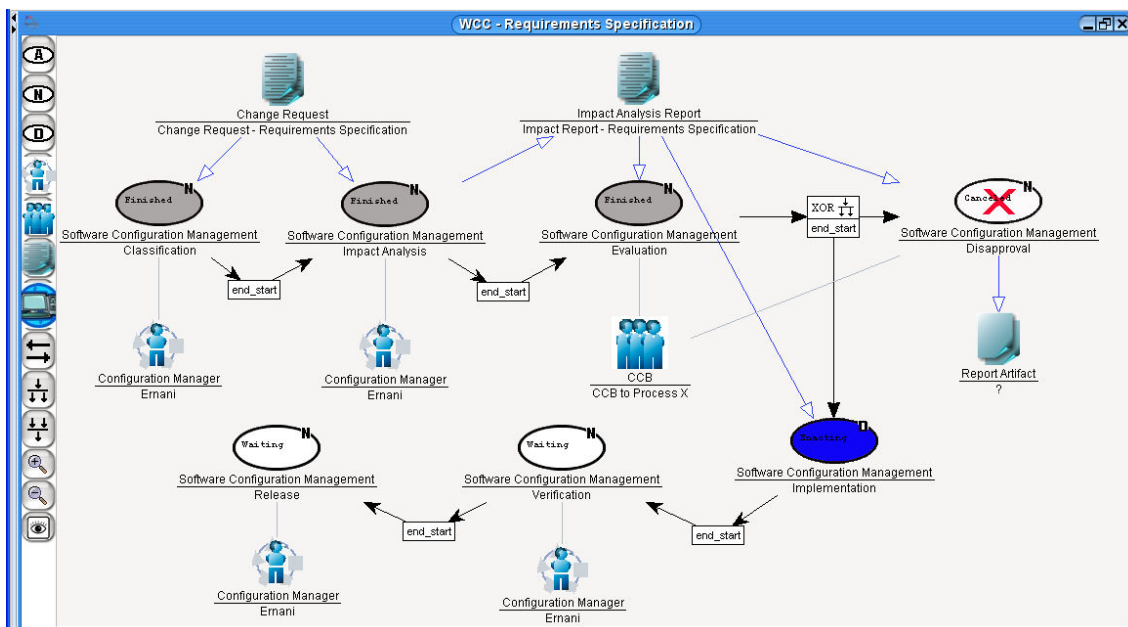


Figura 8 - Processo de Mudança associada ao artefato *Requirements Specification* em execução

A Figura 8 mostra o processo de mudança em execução que se encontra na implementação da mudança, subprocesso definido pelo usuário (isto é, o ambiente define o conjunto de artefatos que vão ser modificados de acordo com a rastreabilidade

definida pelo artefato – nesse caso, *Especificação de Requisitos, Diagrama de Casos de Uso, Plano de Testes e Projeto do Sistema* – porém as atividades, o fluxo de execução e as pessoas envolvidas são modeladas *on-the-fly* pelo gerente no editor de processos).

Como consequência da aprovação da implementação dessa mudança, a atividade *Planejar Testes* será pausada e a atividade *Codificar Casos de Uso* não poderá ser iniciada, tendo em vista as regras de execução que impedem a execução das mesmas por envolverem artefatos em processo de modificação.

5. Trabalhos Relacionados

Os principais trabalhos relacionados ao descrito nessa proposta, que são o componente OSCAR [Nutter *et al.* 2002], o ADELE-CM [Estublier 1994], e o Odyssey-SCM [Murta 2006], serão tratados a seguir.

O sistema OSCAR (*Open Source Component Artefact Repository*) é um componente da plataforma GENESIS [Aversano *et al.* 2002] projetado para interoperar de forma não intrusiva com sistemas de gerenciamento de *workflow*, ferramentas de desenvolvimento de software e sistemas com repositório próprio de informações com o objetivo de apoiar o trabalho colaborativo de equipes de desenvolvimento de software distribuídas.

De maneira diferente ao proposto nesse trabalho, OSCAR pode ser utilizado tanto de forma isolada quanto integrada com outra aplicação. Além disso, o OSCAR só fornece Controle de Versões e um sistema para Notificação de Eventos relativos aos artefatos de software, não atendendo as demais funcionalidades propostas neste trabalho.

Para fornecer o Controle de Versões, OSCAR também define uma interface abstrata contendo os serviços de gerência de configuração necessários para essa tarefa. Entretanto, possui uma implementação para o CVS [CVS 2008] e outra para o *Perforce Version Control System*, enquanto que o controle de versões no trabalho aqui proposto possui implementações para o CVS [CVS 2008] e para o Subversion [SVN 2008]. Já o sistema de notificação de eventos do OSCAR foi desenvolvido utilizando o serviço JMS (*Java Messaging Service*) diferente do proposto neste trabalho que utiliza um serviço próprio de ambiente WebAPSEE para troca de mensagens.

O sistema ADELE [Estublier 1994], inicialmente um sistema de GCS, possui três subsistemas: o *Object Manager* (OM), responsável pela modelagem e apoio de modelos de produto complexos; o *Activity Manager* (AM), responsável pela modelagem e apoio a processos de software; e o *Configuration Manager* (CM), responsável pela modelagem e apoio da evolução dos artefatos de software.

Para atender os seus objetivos, tanto o ADELE-CM quanto a proposta deste trabalho definem modelos peculiares para tratar a gerência de construção e a gerência da mudança. Ambos utilizam um modelo conceitual orientado a objetos com a representação de rastros, relação de dependência, derivação, entre outros. Contudo, o ADELE-CM possui representação até baixo nível dos conceitos de GCS (versão, configuração, etc) enquanto que a proposta deste trabalho aborda apenas conceitos de alto nível de GCS (linha base, rastreabilidade, etc.) haja vista a integração com uma ferramenta de GCS que fornece os conceitos de baixo nível. Além disso, o ADELE define *triggers* para executar processos de gerência de mudanças como parte do

processo de software, enquanto que o modelo proposto trata a **execução de processos de mudança em paralelo ao processo de software** por meio de regras de execução definidas em Gramática de Grafos.

O Odyssey-SCM [Murta 2006] foi desenvolvido no contexto do projeto Odyssey [Werner *et al.* 2003] que é um ambiente para reutilização de software que aplica técnicas de Engenharia de Domínio, Linha de Produtos e DBC (Desenvolvimento Baseado em Componentes). Nesse sentido, Murta (2006) definiu processos e ferramental de apoio para a GCS específicos para o contexto de DBC enquanto que a proposta deste trabalho é tratar a Gerência de Configuração de artefatos de software.

Todavia, ambas as propostas definem um modelo de integração com um ambiente de desenvolvimento de software, além de fornecer apoio ao processo de GCS, controle de versões, controle de modificações e gerenciamento da construção.

6. Considerações Finais

Este trabalho apresentou uma infra-estrutura de apoio à gerência de configuração de artefatos de software integrada a um ambiente de desenvolvimento de software, o ambiente WebAPSEE. Tal infra-estrutura definiu processos e ferramental de apoio à GCS baseados em modelos, padrões e normas atuais, visando facilitar a adoção dos processos de GCS pelas organizações de desenvolvimento de software.

Do ponto de vista de automação, foram integradas ao ambiente WebAPSEE macro-funcionalidades que auxiliam as tarefas da GCS durante o processo de desenvolvimento de software: Controle de Versões, Controle de Modificações, Apoio a Workspace, Gerenciamento da Construção, e Auditoria e Contabilidade. Tais funcionalidades são aplicadas especificamente aos artefatos de software, ou seja, fornecendo apoio a um nível de abstração mais alto se comparada com a gama de ferramentas existentes da área de GCS.

Ainda no contexto da infra-estrutura proposta, é importante ressaltar o apoio ao controle de modificações. Este controle é **executado paralelamente ao processo de desenvolvimento de software** pela máquina de execução do WebAPSEE, podendo ou não influenciar no fluxo de execução (flexível). Desta forma, a maneira como um processo de apoio é representado, neste caso o processo de mudança, torna-se mais adequada à realidade.

Com a integração não-intrusiva dos processos e técnicas de GCS ao ambiente de desenvolvimento de software, facilita-se a implementação dos resultados esperados relativos à GCS pelos modelos de maturidade de processos, tais como CMMI e MPS.Br. Isto ocorre devido a aspectos específicos e complexos da GCS serem disponibilizados de forma transparente, flexível, e com alto grau de automação ao usuário.

O presente trabalho encontra-se em evolução, com trabalho inicial publicado em [Sales 2007], e trabalhos futuros rumam em direção ao apoio de rastreabilidade dos artefatos em granularidade mais fina, melhoria e inclusão de processos de apoio fornecidos para o controle de modificações, definição de relatórios gerenciais mais robustos, aumento do poder de gerenciamento da construção, dentre outros.

Referências

- Abdala, M. A. D. (2006) “Uma abordagem para a gerência das modificações e da configuração em um ambiente integrado para o desenvolvimento e gestão de projetos de software”. Dissertação de Mestrado. São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 292 f.
- Asklund, U., and Bendix, L. (2003) “A Software Configuration Management Course” In: Lecture Notes in Computer Science (2649), p. 245-258.
- Aversano, L.; Cimitile, A.; Gallucci, P.; Villani, M.L. (2002) “FlowManager: a workflow management system based on Petri nets”. In: 26th Annual International Computer Software and Applications Conference. COMPSAC 2002.
- Ben-Shaul, Israel Z.; Kaiser, G. E (1997) “Federating Process-Centered Environments: the OZ Experience”. In: Automated Software Engineering, 5.
- Bruegge, Bernd *et al.* (2006) “Supporting Distributed Software Development with fine-grained Artefact Management”. In: IEEE International Conference on Global Software Engineering. IEEE Computer Society, Washington, DC.
- Chrissis, M.B., Konrad, M., Shrum, S. (2003) “CMMI: Guidelines for Process Integration and Product Improvement”. Addison-Wesley: Boston, MA.
- CVS. CVS. Disponível em: <http://ximbiot.com/cvs/cvshome/>. Acesso em: jan. 2008.
- Dart, S., (1991) “Concepts in Configuration Management Systems”. In: International Workshop on Software Configuration Management (SCM), Trondheim, Norway.
- Estublier, Jack and Casallas, R. (1994) “The Adele Software Configuration Manager”. In: Trends in Software Configuration Management, Edited by W. Tichy; J. Wiley.
- Estublier, Jack. (2000) “Software Configuration Management: a Roadmap”. In: Conference On The Future Of Software Engineering (Limerick, Ireland, June 04 - 11, 2000) – ICSE '00, ACM Press, New York, NY, p. 279-289.
- Estublier, J., Leblang, D., Van Der Hoek, A., et al. (2005) “Impact of Software Engineering Research on the Practice of Software Configuration Management”. In: ACM Transactions on Software Engineering and Methodology (TOSEM), v. 14, n. 4.
- Harrison, W., Ossher, H., Tarr, P. (2000) “Software Engineering Tools and Environments: A Roadmap”. In: Conference On The Future Of Software Engineering (Limerick, Ireland, June 04 - 11, 2000) – ICSE '00, ACM Press, New York, NY.
- IEEE (1987) “Std 1042 - IEEE Guide to Software Configuration Management”, Institute of Electrical and Electronics Engineers.
- IEEE (1990) “Std 610.12 - IEEE Standard Glossary of Software Engineering Terminology”, Institute of Electrical and Electronics Engineers.
- IEEE (2005) “Std 828 - IEEE Standard for Software Configuration Management Plans”, Institute of Electrical and Electronics Engineers.
- ISO 10007 (1995) “Quality Management - Guidelines for Configuration Management”, International Organization for Standardization.
- ISO 12207 (1998) “Tecnologia de Informação - Processos de ciclo de vida de Software”, Associação Brasileira de Normas Técnicas, Rio de Janeiro: ABNT.
- Jalote, P. (2000) “CMM in Practice: Processes for Executing Software Projects at Infosys”. Addison-Wesley Publishing Company, 2000.

- Lima, Adailton *et al.* (2006) “Gerência Flexível de Processos de Software com o Ambiente WebAPSEE”. In: Sessão de Ferramentas do Simpósio Brasileiro de Engenharia de Software (SBES), 20. Florianópolis: Informática-UFSC.
- Lima Reis, Carla A. (2003) “Uma Abordagem Flexível para Execução de Processos de Software Evolutivos”. Tese de Doutorado. Porto Alegre: PPGC da UFRGS, 2003.
- Lima Reis, Carla A.; Reis, Rodrigo Q. (2007) “Laboratório de Engenharia de Software e Inteligência Artificial: Construção do ambiente WebAPSEE”. In: ProQuality (UFLA), v. 3, p. 43-48, 2007.
- MCT (2002) “Qualidade e Produtividade no Setor de Software Brasileiro”, Ministério de Ciência e Tecnologia, Secretaria de Política de Informática, Brasília, DF, Brasil.
- Montoni, M. A.; Santos, G.; Rocha, A. R. C.; Figueiredo, S.; C. Silva Filho, R.; Barreto, A.; Cerdeiral, C.; Lupo, P. (2006) “Taba Workstation: Supporting Software Process Deployment Based on CMMI and MR-MPS.BR.” In: Lecture Notes in Computer Science (LNCS), presented in International Conference on Product Focused Software Process Improvement, Amsterdam - Holanda, p. 249-262.
- Murta, L.G.P. (2006) “Gerência de Configuração do Desenvolvimento Baseado em Componentes”. 213 f. Tese de Doutorado em Ciências em Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, UFRJ.
- Nutter, David. *et al.* (2002) “An Artefact Repository to Supported Distributed Software Engineering”. In: 26th Annual International Computer Software and Applications Conference. COMPSAC 2002, p. 1081- 1086
- Paxiúba, Carla. *et al.* (2005) “Towards an Event Recording Mechanism for a Process-based Environment”. In: Seminário Integrado de Software e Hardware, 32. São Leopoldo: SBC, 2005.
- Paulk, M. C.; Curtis, B.; Chrissis, M. B.; Weber, C. V. (1993) “Capability maturity model” (Version 1.1). In: IEEE Softw. 10, 4 (July), p. 18–27.
- Pressman, Roger S. (2005) “Software Engineering: a practioner’s approach”. McGrawHill, 6th edition.
- Sales, Ernani de O.; Costa, Anderson J. S. (2007) “Uma Proposta para Reutilização de Processos de Software para o Ambiente WebAPSEE”, TCC. Belém: CBCC-UFPA. Fevereiro, 85 pp.
- Sales, Ernani de O. (2007) “Gestão de Configuração integrada a Gerência de Processos de Software no Ambiente WebAPSEE”. In: XXXIII Conferencia Latinoamericana en Informática - CLEI'2007. San José, Costa Rica, Outubro.
- SEI. CMMI Web Site. Disponível em: www.sei.cmu.edu/cmmi/. Acesso em: jan. 2008.
- Softex. MPS.BR – Melhoria de Processos de Software Brasileiro. Disponível em: www.softex.br/mpsbr/. Acesso em: jan. 2008.
- SVN. Subversion. Disponível em: <http://subversion.tigris.org/>. Acesso em: jan. 2008.
- Werner, C.M.L., Mangan, M.A.S., Murta, L.G.P., *et al.* (2003) “OdysseyShare: an Environment for Collaborative Component-Based Development”. In: IEEE Conference On Information Reuse And Integration (IRI). Las Vegas, USA.
- Yan, Jun; Yang, Yun; Raikundalia, Gitesh K. (2003) “Decentralised Coordination for Software Process Enactment”. In: Lecture Notes in Computer Science, Heidelberg: Springer Berlin.