

Proposta de um Modelo de Referência de Gerência de Configuração para um Processo de Reengenharia baseado em *Framework*

Marliane Aldivina Oliveira Ferreira, Maria Istela Cagnin

UNIVEM - Fundação Eurípides Soares da Rocha
Marília, São Paulo, Brasil, Caixa Postal 2041, CEP 17525-901

marlianeferreira@gmail.com, istela@univem.edu.br

Resumo. A Gerência de Configuração (GC) é um dos elementos essenciais para a garantia da qualidade, pois apóia o controle das mudanças nos artefatos produzidos durante um processo de desenvolvimento de software, a fim de manter a integridade e a consistência dos mesmos. Isso não é diferente no contexto de reengenharia. No entanto, atualmente, existe carência do uso de GC em processos de reengenharia de software. Este artigo tem como objetivo apresentar uma proposta de um modelo de referência para a atividade de GC de um processo ágil de reengenharia baseado em framework, denominado PARFAIT¹.

1 Introdução

Com o progresso da Engenharia de *Software*, a qualidade no desenvolvimento vem proporcionar uma melhoria no produto final entregue aos usuários (SOMMERVILLE, 2003; PRESSMAN, 2002). A qualidade não é tratada diferente no contexto de Reengenharia Ágil, pois visa entregar uma versão do sistema alvo o mais rápido possível, funcionando adequadamente (CAGNIN, 2005). Uma das maneiras de garantir a qualidade é por meio da Gerência de Configuração (GC).

Constata-se que as mudanças no *software* são inevitáveis durante todo o seu ciclo de vida para se adequar às necessidades dos usuários e/ou aos avanços tecnológicos. Assim, a GC é empregada como apoio, cuja responsabilidade está relacionada na identificação, na documentação, no armazenamento, no controle e no relato das mudanças ocorridas nos artefatos (CUNHA *et al.*, 2004; PRESSMAN, 2002). Além disso, GC facilita o controle das versões dos artefatos, pois várias versões podem estar em operação com diferentes configurações de *software* (PRESSMAN, 2002). De acordo com Sommerville (2003), diversas atividades são empregadas durante a GC: identificar os artefatos, criar um repositório de configuração, gerenciar as mudanças, gerenciar as versões, gerenciar as *releases*, realizar revisões e definir uma ferramenta de apoio.

¹ Processo Ágil de Reengenharia baseado em FrAmework no domínio de Sistemas de Informação com técnicas VV&T.

No entanto, apesar da importância de GC pode-se observar a carência na literatura do emprego de seu uso em Processos de Reengenharia (LEMOS *et al.*, 2003), pois, como no desenvolvimento de *software*, é necessário se preocupar com o controle sistemático das modificações que devem ser realizadas a fim de produzir um sistema alvo com qualidade. Para apoiar este controle é imprescindível a utilização de ferramentas específicas, como é o caso da *Concurrent Versions System* (CVS) (XIMBIOT, 2005) e da *Subversion* (COLLINS-SUSSMAN, *et al.*, 2006).

O processo de reengenharia de interesse deste trabalho, denominado PARFAIT (CAGNIN *et al.*, 2003), é baseado nas práticas de métodos ágeis (BECK, 2000) e utiliza como apoio computacional *frameworks* de aplicação (FAYAD *et al.*, 1999) a fim de gerar uma versão do sistema alvo o mais rápido possível. Cagnin (2005) ressalta que o emprego de GC no contexto de *frameworks* é similar a outro *software*, mas pode se tornar mais complexo devido a inúmeras versões do *framework* e ferramentas de apoio a ele associadas. Além disso, GC deve ser aplicada tanto no *framework* quanto nas versões dos sistemas criados a partir dele.

Assim, neste artigo apresenta-se uma proposta de um modelo de referência para a atividade de GC do PARFAIT. Para possibilitar isso, algumas normas e modelos de maturidade existentes serão tomados como base. Além disso, utilizou-se GQM (*Goal Question Metric*)² (BASILI, 1994) para elicitar os elementos (atividades e artefatos) desse modelo. O artigo está organizado da seguinte maneira: na Seção 2 apresentam-se as normas e modelos de maturidades presentes na literatura que apoiam a GC; na Seção 3 discutem-se alguns processos de reengenharia de *software*, explicitando a carência de GC em cada um deles; na Seção 4 discute-se a proposta do modelo de referência e, finalmente, na Seção 5 são apresentadas as conclusões e são discutidos os trabalhos futuros.

2 Normas e Modelos de Maturidade

Entre as normas e modelos de maturidade encontrados na literatura que tratam de GC têm-se: *CMMI* (SEI, 2001), *ISO/IEC 12207* (1998), *ISO/IEC 15504* (*ISO/IEC TR 15504*, 1999), *ISO/IEC 10007*(2005), MPS.BR (SOFTEX, 2005) e *PMBOK* (PMI, 2004). O *CMMI* apresenta a GC no nível 2 de maturidade como prática genérica. Os controles gerenciais e técnicos da GC são efetuados na seleção dos itens de configuração, no controle das mudanças dos artefatos, no estabelecimento das diretrizes para o desenvolvimento do produto com gerenciamento, na confiabilidade e nas atividades de configuração para o desenvolvedor e para o usuário. No *CMMI* as práticas específicas aplicadas na GC são: identificar os itens de configuração e transformá-los em *baselines*³, controlar as modificações dos itens de configuração e estabelecer a integridade das *baselines* (SEI 2001).

A norma NBR *ISO/IEC 12207* (1998) é a pioneira em tratar a GC e apresenta com detalhes a descrição das atividades que apoiam a empresa no desenvolvimento

² é um método que estabelece métricas quantitativas, tendo como base os objetivos ou as metas que deverão ser alcançados, com o apoio de um questionamento ou avaliação de como alcançá-los. Assim, estabelecem as bases que servirão de comparação para trabalhos futuros (BASILI *et al.*, 1994).

³ um conjunto de artefatos aceitos e controlados que poderá ser utilizado posteriormente, sendo que sua evolução é alcançada com as alterações, mediante uma solicitação e uma aprovação prévia (PRESSMAN, 2002).

desta prática. As atividades exclusivas de GC estão presentes na documentação do processo de apoio do ciclo de vida e constam de: identificar, controlar e avaliar a configuração, relatar a situação da configuração e gerenciar a liberação e a distribuição.

Já a norma *ISO/IEC 15504 (ISO/IEC TR 15504, 1999)* é composta por 9 partes cujo objetivo é elaborar um guia de orientação que examine e avalie o *software*, promovendo uma melhoria contínua dos processos. A GC encontra-se na quinta parte da norma, com a denominação de suporte. Esta norma descreve os processos básicos que devem ser seguidos em sua aplicação. Dentre os processos de interesse têm-se: criar uma estratégia de GC para executar todas as atividades; estabelecer o emprego de padrões, de procedimentos e de ferramentas de GC; identificar os artefatos que foram armazenados em repositórios, controlar as modificações com o formulário de solicitação de mudanças e de autorização, gerenciar o histórico do artefato e administrar a liberação de uma determinada versão somente após a revisão e autorização.

O Modelo de Referência para melhoria de processo de software (MR mps), criado pelo projeto de Melhoria de Processo de *Software* Brasileiro (MPS.BR) trata a GC no nível de maturidade F. Este modelo estabelece e mantém a integridade dos artefatos por meio das seguintes atividades: selecionar os artefatos com o emprego de critérios documentados; delegar responsabilidades para o desenvolvimento de cada artefato e *baseline*; identificar, armazenar, testar, revisar, alterar, definir, manter os artefatos e submetê-los a *baseline*; instituir uma política de GC; autorizar a criação ou a liberação das *baselines*; controlar as modificações e as liberações dos artefatos; registrar, relatar e analisar o impacto das solicitações de mudanças nos artefatos; assegurar a consistência e a completeza dos artefatos; controlar o armazenamento, o manuseio e a liberação dos artefatos; estabelecer e manter a integridade das *baselines* com os registros e a auditoria de GC (SOFTEX, 2005).

O *Project Management Body of Knowledge (PMBOK)* trata a GC na área de conhecimento de Gerenciamento de Integração de Projeto, com o gerenciamento das mudanças, as revisões e as aprovações das mudanças, os níveis de aprovação para a autorização e a validação das mudanças (PMI, 2004).

Dentre as normas citadas destaca-se a norma *ISO/IEC 10007 (2005)* que trata exclusivamente da GC, fornecendo um guia com diretrizes, cujo enfoque está voltado principalmente ao ciclo de vida do produto, atendendo aos requisitos de identificação e de rastreabilidade dos artefatos. Assim, sugere o planejamento, a identificação, o controle das alterações, a situação de configuração e as auditorias. A GC é dividida por essa norma em: *generalidade*, que enfoca os requisitos do cliente do produto final e todo contexto de execução; *planejamento*, que coordena todas as atividades de gestão de configuração; *identificação de configuração*, que seleciona os artefatos necessários com o emprego dos critérios de seleção relacionados aos requisitos, à segurança e ao risco, à tecnologia do projeto, à interface com os artefatos e ao suporte; *controle de alterações*, que possibilita a solicitação das alterações com a verificação do seu impacto, a análise de sua aprovação e, em seguida, o seu gerenciamento; *contabilização da situação da configuração*, que impulsiona o registro das informações importantes do artefato, por exemplo, o número de identificação da versão, a data efetiva da liberação da versão, a situação da revisão, o histórico do artefato; e *auditoria de configuração*, que deve ser aplicada nas revisões para analisar se os requisitos foram atingidos.

3 Processos de Reengenharia de *Software*

O processo de desenvolvimento de *software* pode ser definido como sendo o ciclo de vida do *software* que descreve a produção de um *software* desde o surgimento da proposta de desenvolvimento até a sua inutilização (PFLEEGER, 2004). Já, um processo de reengenharia descreve como migrar sistemas obsoletos para novas tecnologias. Entre os processos de reengenharia, destacam-se o PRE/OO, o Processo de Reengenharia Baseado em Componente Distribuído e o PARFAIT, sendo que este último, como mencionado, é de interesse deste trabalho.

O Processo de Reengenharia Orientada a Objeto (PRE/OO), composto por um grupo de padrões de reengenharia, se destaca por tratar das seguintes atividades de GC: identificação dos itens de configuração, documentação de todas as mudanças com o auxílio de um cabeçalho com informações sobre o responsável pela mudança, a data, o nome do documento, a descrição da mudança, a versão e a origem. Além disso, foi estabelecida a criação de uma *baseline* ao final de cada passo apresentado por este processo (LEMOS *et al.*, 2003). Como pode ser observado, poucos aspectos de GC foram considerados neste processo.

Já o Processo de Reengenharia Baseado em Componente Distribuído, desenvolvido por Fontanette *et al.* (2002), realiza a reengenharia com o apoio do método *Catalysis* (2006), que é baseado em Componentes Distribuídos, de diagramas da UML (*Unified Modeling Language*), da ferramenta MVCASE (PRADO e LUCRÉCIO, 2001) e do Sistema Transformacional Draco (GARCIA *et al.*, 2002). Os princípios adotados foram **Abstração** (auxilia na busca das características indispensáveis ao sistema), **Precisão** (tem como finalidade encontrar erros e problemas na modelagem) e **uso de Componentes “Plug-In”** (possibilitam a reutilização de componentes para a criação do sistema alvo) (FONTANETTE *et al.*, 2002a, 2002b). Deve-se destacar que este processo não emprega atividades de GC.

O PARFAIT é um processo ágil, incremental e iterativo, que visa migrar sistemas legados procedimentais para o paradigma orientado a objetos. Ele é um dos recursos do Arcabouço de Reengenharia Ágil (ARA) (CAGNIN, 2005), que apóia a reengenharia ágil baseada em linguagens de padrões (APPLETON, 1997) e *frameworks* (FAYAD *et al.*, 1999), cuja construção tenha sido baseada nessas linguagens. A linguagem de padrões é utilizada para facilitar o entendimento do domínio do sistema legado, a elaboração da documentação orientada a objetos do sistema e o uso do *framework* (ou seja, na instanciação do *framework* para a geração do sistema alvo).

PARFAIT trata de alguns itens de GC de forma manual, assim observou-se neste trabalho a necessidade de aperfeiçoar tal aspecto, criando um modelo de referência específico para isso. Neste processo, existe a necessidade de ter apoio computacional para controlar as mudanças ocorridas nas versões do sistema alvo. Isso foi parcialmente resolvido com o emprego da ferramenta *GREN-WizardVersionControl* (CAGNIN *et al.*, 2004), que controla mais especificamente as modificações feitas no sistema alvo em cada iteração do PARFAIT, uma vez que tais modificações podem ser perdidas em uma próxima instanciação do *framework*. Apesar do PARFAIT preocupar-se com a identificação dos itens de configuração logo no início da sua aplicação e ressaltar a importância de controlar as versões de todos os artefatos produzidos durante a reengenharia, não fornece nenhuma maneira sistemática de aplicar a GC.

Como *frameworks* também evoluem, o Processo de Evolução de *Framework* (PREF) pode ser executado em paralelo ao PARFAIT e utiliza um histórico de requisitos funcionais e não funcionais de sistemas que não foram cobertos pelo domínio do *framework* para decidir se deve ou não conduzir a evolução do *framework* (CAGNIN *et al.*, 2004a). Dentre as atividades do PREF, destaca-se a oitava, denominada Tratar do Gerenciamento de Controle de Configuração, que deve ser executada com certo cuidado, pois as evoluções em *frameworks* podem mudar o seu projeto e, conseqüentemente, dos sistemas gerados fazendo com que estes deixem de fornecer o comportamento desejado.

A partir do exposto, nota-se a carência de GC em processos de reengenharia de software, o que motivou esta pesquisa.

4 Proposta de um Modelo de Referência de GC para o PARFAIT

Para a criação do modelo de referência, foram seguidas duas etapas. A primeira etapa consistiu na seleção das atividades de GC essenciais para controlar as versões dos *frameworks* e das versões criadas a partir da instanciação desses sem afetar a agilidade do PARFAIT. Para isso, foram analisadas as normas e os modelos de maturidade CMMI, ISO/IEC 12207, ISO/IEC 15504, ISO/IEC 10007, PMBOK e MPS.BR. A segunda etapa foi realizada com base nas atividades selecionadas na etapa anterior, e tratou da seleção dos artefatos essenciais que devem ser considerados no modelo proposto.

Ressalta-se que além das práticas dos métodos ágeis, PARFAIT também emprega algumas práticas de modelagem ágil (AMBLER, 2004) com o objetivo de documentar o sistema legado com agilidade, promover o entendimento e facilitar a comunicação entre os engenheiros de *software* (CAGNIN, 2005). Assim, para selecionar as atividades e os artefatos de GC do modelo de referência proposto é necessário observar se esses não afetam cada uma das práticas ágeis consideradas pelo PARFAIT, a fim de garantir que o modelo de referência não afete a agilidade do processo.

Para apoiar ambas etapas, utilizou-se o método GQM em que foram estabelecidos os objetivos, as questões e as métricas para atender os objetivos estabelecidos para compor os elementos do modelo de referência. Dentre as métricas utilizadas, foi criada a métrica “grau de importância”, variando de 1 a 5: Grau 5 (extremamente importante): contempla o resultado das métricas de 100% a 80%; Grau 4 (importante): de 79% a 60%; Grau 3 (importância moderada): de 59% a 40%; Grau 2 (pouco importante): de 39% a 20%; Grau 1 (sem importância): de 19% a 0%.

No Quadro 1 apresenta-se o plano de mensuração GQM para selecionar as atividades de GC do modelo de referência do PARFAIT, criado na Etapa 1. Ressalta-se que as atividades de GC consideradas neste plano são aquelas definidas por Sommerville (2003). Em seguida, foi estabelecido o grau de importância para cada atividade considerada no plano de mensuração (Tabela 1). Para selecionar as atividades essenciais, observou-se aquelas da Tabela 1 com grau de importância 5 (identificar os artefatos, criar um repositório de configuração, gerenciar as mudanças, gerenciar as versões, definir a ferramenta de apoio a GC). Posteriormente, foi criado o plano de mensuração GQM para selecionar os artefatos do modelo de referência proposto (Etapa 2), apresentado no Quadro 2. Para isso, foram considerados somente os artefatos das atividades selecionadas na Etapa 1. Em seguida, foi determinado o grau de importância

de tais artefatos (Tabela 2) e selecionados os artefatos com grau de importância 5. No caso analisado, todos os artefatos considerados foram selecionados.

Quadro 1 Plano de mensuração GQM para a seleção das atividades de GC do modelo de referência

<p>Objetivo: Analisar as atividades de GC dos modelos de maturidade CMMI, ISO/IEC 12207, ISO/IEC 15504, ISO/IEC 10007, PMBOK e MPS.BR com o propósito de selecionar as atividades essenciais de GC para o PARFAIT sem afetar sua agilidade e apoiar o controle das versões tanto do <i>framework</i> quanto das aplicações geradas a parte dele, do ponto de vista de GC de <i>Software</i>, no contexto de processos de reengenharia.</p>
<p>Questão 1: A atividade X⁴ está presente na maioria das normas e modelos de maturidade? Métrica 1: Percentual de ocorrência da atividade X nas normas e modelos de maturidade Métrica 1.1: Valor do grau de importância Questão 2: A agilidade do PARFAIT não é prejudicada pela atividade X? Métrica 2: Percentual das práticas ágeis, atendidas pelo PARFAIT, que não são prejudicadas pela atividade X. Métrica 2.1: Valor do grau de importância Questão 3: A atividade X apóia o controle das versões do <i>framework</i> e das aplicações geradas a partir dele? Métrica 3: Percentual de apoio do controle de versões presente na atividade X. Métrica 3.1: Valor do grau de importância Métrica 3.2: Média ponderada do grau de importância obtidos nas métricas 1.1, 2.1 e 3.1 ...</p>

Tabela 1 Grau de importância das atividades de GC

Atividades de GC	Grau de importância	Justificativa
Identificar os Artefatos	5	Esta atividade está presente em 100% das normas e dos modelos de maturidade, não prejudica a agilidade do PARFAIT e apóia o controle dos artefatos em 100%.
Criar um Repositório de Configuração	5	Esta atividade está presente em 100% das normas e dos modelos de maturidade, não prejudica a agilidade do PARFAIT em 100% e apóia o controle dos artefatos em 100%.
Gerenciar as Mudanças	5	Esta atividade está presente em 100% das normas e dos modelos de maturidade, não prejudica a agilidade do PARFAIT em 100% e apóia o controle dos artefatos em 100%.
Gerenciar as Versões	5	Esta atividade está presente em 100% das normas e dos modelos de maturidade, não prejudica a agilidade do PARFAIT em 100% e apóia o controle dos artefatos em 100%.
Gerenciar as Releases	3	Esta atividade está presente em 50% das normas e dos modelos de maturidade, prejudica a agilidade do PARFAIT em 33 % e apóia o controle dos artefatos em 50%
Revisões de GC	3	Esta atividade está presente em 50% das normas e dos modelos de maturidade, prejudica a agilidade do PARFAIT em 33 % e apóia o controle dos artefatos 50%
Definir a Ferramenta de Apoio a GC	5	Esta atividade está presente em 100% das normas e dos modelos de maturidade, não prejudica a agilidade do PARFAIT em 100% e apóia o controle dos artefatos 100%

Sugere-se que seja feita uma validação, com um especialista em GC, a respeito dos graus de importância estabelecidos a cada um dos artefatos e atividades de GC.

⁴ Cada atividade de GC considerada por Sommerville (2003).

Quadro 2 Plano de mensuração GQM dos artefatos elaborados pelas atividades de GC

<p>Objetivo: Analisar os artefatos elaborados em GC presentes nos modelos de maturidade CMMI, ISO/IEC 12207, ISO/IEC 15504, ISO/IEC 10007, PMBOK e MPS.BR com o propósito de selecionar os artefatos essenciais de GC para o PARFAIT das atividades essenciais selecionadas sem afetar sua agilidade, e apoiar o controle das versões tanto do <i>framework</i> quanto das aplicações geradas a parte dele do ponto de vista de GC de Software, no contexto de processos de reengenharia.</p>
<p>Questão 22: O artefato Y⁵ está presente na maioria das normas e modelos de maturidade? Métrica 1: Percentual de ocorrência do artefato Y nas normas e modelos de maturidade Métrica 1.1: Valor do grau de importância</p>
<p>Questão 23: A agilidade do PARFAIT não é prejudicada pelo artefato Y? Métrica 4: Percentual das práticas de Modelagem Ágil atendidas pelo PARFAIT, que não são prejudicadas pelo artefato Y. Métrica 4.1: Valor de grau de importância Métrica 4.2: Média ponderada do grau de importância obtidos nas métricas 1.1 e 4.1 ...</p>

Tabela 2 Grau de importância dos artefatos elaborados pelas atividades de GC

Artefato	Grau de importância	Justificativa
Documento da baseline	5	Este artefato está presente em 100% das normas e dos modelos de maturidade, não prejudica a agilidade do PARFAIT.
Formulário de controle de mudança	5	Este artefato está presente em 100% das normas e dos modelos de maturidade, não prejudica a agilidade do PARFAIT.
Histórico dos Artefatos	5	Este artefato está presente em 100% das normas e dos modelos de maturidade, não prejudica a agilidade do PARFAIT.

A partir da seleção das atividades e artefatos essenciais de GC elabora-se a documentação do modelo de referência para guiar o engenheiro de software na sua utilização durante a aplicação do processo PARFAIT.

5 Conclusões

Como foi apresentado neste trabalho, é importante também considerar a GC no contexto de reengenharia de software. Uma vez que há carência disso na literatura, este trabalho teve como objetivo propor um modelo de referência para a atividade de GC de um processo de reengenharia. Para isso foram selecionadas as atividades e artefatos de GC, baseando-se em diversas normas e modelos de maturidade existentes, sem afetar a agilidade do processo. Como este trabalho está em andamento, a próxima fase será a elaboração da documentação completa do modelo de referência com base nas atividades e artefatos selecionados. Para analisar a aplicabilidade de tal modelo, será feito um estudo de caso de reengenharia tomando como base o processo PARFAIT e o modelo de referência criado. Como trabalhos futuros podem-se destacar: a) adaptação do Modelo de Referência de GC do PARFAIT para Processos de Reengenharia. Para isso, é necessário generalizar tal modelo observando as características básicas de processos de reengenharia ou de um conjunto de tais processos; b) adaptação do Modelo de Referência de GC do PARFAIT para processos de desenvolvimento de *software* baseados em *framework*; c) criar um meta-modelo para armazenar as informações relevantes do controle de GC do modelo de referência proposto para apoiar o desenvolvimento de uma ferramenta específica para automatizar o uso de tal modelo.

⁵ Todos os artefatos de GC presentes nas atividades selecionadas a partir da Tabela 1 com grau de importância 5.

Referências

- AMBER, S. W. (2004) Modelagem Ágil: Práticas eficazes para a Programação extrema e o Processo Unificado, Bookman, Porto Alegre.
- APPLETON, B. (1997) “Patterns and *software*: Essential concepts and terminology”, <http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html>, Janeiro/2006.
- BASILI, V. E.; CALDIERA, G.; ROMBACH, H. D. (1994) Goal Question Metric Paradigm, Encyclopedia of *Software Engineering*, John Wiley & Sons, New York.
- BECK, K. (2000). Extreme programming explained: Embrace change. second ed. Addison-Wesley
- CAGNIN, M. I. (2005) “PARFAIT: uma contribuição para a reengenharia de *software* baseada em linguagem de padrões e *frameworks*”, tese de doutorado do Instituto de Ciências Matemáticas e de Computação – ICMC/USP, São Carlos.
- CAGNIN, M. I.; MALDONADO, J. C.; GERMANO, F. S.; CHAN, A.; PENTEADO, R. D. (2003) “Um estudo de caso de reengenharia usando o processo PARFAIT”, In: SDMS'2003, Simpósio de Desenvolvimento e Manutenção de *Software* da Marinha, Rio de Janeiro, RJ, CD-ROM, 10 p.
- CAGNIN, M. I.; MALDONADO, J. C.; GERMANO, F. S.; PENTEADO, R. D.; BRAGA, R. T. (2004) “GREN-WizardVersionControl: Uma Ferramenta de Apoio ao Controle de Versão das Aplicações Criadas pelo *Framework* GREN”, In: Sessão de Ferramentas 2004, Simpósio Brasileiro de Eng. de *Software*, Brasília, DF, p. 73-78.
- CAGNIN, M. I.; MALDONADO, J. C.; MASIERO, P. C.; PENTEADO, R. D.; BRAGA, R. T. (2004a) “An Evolution Process for Application *Frameworks*”, In: I Workshop de Manutenção Moderna de *Software*, em conjunto com o XVIII Simpósio Brasileiro de Eng. de *Software*, Brasília, DF, CD-ROM, 8 p.
- COLLINS-SUSSMAN, B.; FITZPATRICK, B.W.; PILATO, C.M. (2006) “Version Control with Subversion, para Subversion 1.2”, http://svnbook.red-bean.com/nightly/pt_BR/svn-book.html.
- CUNHA, J. R. D.; PRADO, A. F.; SANTOS, A. C.; SOUZA NETO, R. M.; (2004) “Uma abordagem para o Processo de Gerenciamento de Configuração de *Software*”, In: 4ª ed., RESI – Revista eletrônica de S.I., Universidade F. de São Carlos, <http://www.presidentekennedy.br/resi/edicao04/artigo05.pdf>.
- FONTANETTE, V; GARCIA, V.C.; BOSSONARO, A. A.; PEREZ, A.B.; PRADO, A. F. (2002) “Reengenharia de *Software* usando Transformações (RST)”, In: Second Ibero-American Symposium on *Software Engineering* and Knowledge Engineering, Salvador.
- GARCIA, V. C., FONTANETTE, V., BOSSONARO, A., PEREZ, A. B., PRADO, A. F. (2002), “DDE – Draco Domain Editor”, In: XVI Simpósio Brasileiro de Eng. de *Software*, Gramado.
- ISO/IEC 10007 (2005) “Sistemas de Gestão da Qualidade – Diretrizes para a Gestão de Configuração”, ABNT - Associação brasileira de normas técnicas, Rio de Janeiro: ABNT.
- ISO/IEC 12207 (1998) “Tecnologia de Informação - Processos de ciclo de vida de *software*”, ABNT - Associação brasileira de normas técnicas Rio de Janeiro: ABNT.
- ISO/IEC TR 15504 – 5 (1999) “Information Technology – *Software* process assessment – Part 5: An assessment model and indicator guidance”, ISO/IEC - International Standard Organization and International Electrotechnical Commission.
- LEMOS, G. S.; RECCHIA, E. L.; PENTEADO, R. D.; BRAGA, R. T. V. (2003) “Padrões de Reengenharia Auxiliados por Diretrizes de Qualidade de *Software*”, In: The Second Latin American Conference on Pattern Languages of Programming. (Sugarloaf Plop), pp. 63-88, Pernambuco.
- PFLEEGER, S. L. (2004) Engenharia de *Software*, 2ª ed., Pearson Education
- PMI - Project Management Institute, “Sobre PMI – História”, <http://www.pmisc.org.br/portugues/introducao.htm>, Março/2006.
- PMI - Project Management Institute (2004) “Um Guia do Conjunto de Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)”, 3ª Edição, Four Campus Boulevard, Newtown Square, EUA.
- PRESSMAN, R. S. (2002) Engenharia de *Software*, 5ª ed., Rio de Janeiro: McGraw-Hill.
- SEI - *Software Engineering Institute* (2001) “Capability Maturity Model® Integration (CMMISM)”, Version 1.1, <http://www.sei.cmu.edu/cmmi/models/model-components-word.html>, Fevereiro/2006.
- SOFTEX (2006) “MPS.BR – Melhoria de Processo do *Software* Brasileiro (Guia Geral – Versão 1.0)”, <http://www.inf.ufes.br/~falbo/download/aulas/es-m/2005-2/Texto14.pdf>, Fevereiro/2006.

SOMMERVILLE, I. (2003) Engenharia de *Software*, 6ª ed., São Paulo: Addison Wesley.

XIMBIOT (2006) "Version Management with CVS", <http://ximbiot.com/cvs/manual/>, Fevereiro/2006.