Instituto de Ciências Matemáticas de São Carlos

Instituto de Ciências Matemáticas - São Carlos
Biblioteca

ISSN - 0103-2569

GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE

RENATO FERRARI PACHECO
ROSELY SANCHES

Nº 57

DEDALUS - Acervo - ICMSC
30300031377

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
Jun./1997

Class. -
Cutt. -
21.562

Instituto de Ciências Matemáticas de São Carlos

ISSN

GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE

RENATO FERRARI PACHECO
ROSELY SANCHES

Nº xx

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
Jul./1997

SUMÁRIO

1. INTRODUÇÃO 1

1.1 CONTEXTO 1

2. GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE 5

2.1 CONCEITOS FUNDAMENTAIS DE GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE 5

2.2 TAREFAS DE GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE 8

2.2.1 - Tarefa 1- Identificação da Configuração de Software 8

2.2.2 - Tarefa 2 - Controles de Mudança e de Versões 10

2.2.3 - Tarefa 3 - Auditoria da Configuração 14

2.2.4 - Tarefa 4 - Relato da Situação 14

2.2.5 - Tarefa 5 - Controle de interface 15

2.2.6 - Tarefa 6 - Controle de subcontratados e fornecedores 15

2.3 CONSIDERAÇÕES FINAIS 16

3. PLANO DE GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE 18

3.1 ELEMENTOS ENVOLVIDOS NO PLANO DE GERENCIAMENTO DE CONFIGURAÇÃO 18

3.2 ALGUNS PADRÕES DE PLANOS DE GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE 21

3.3 O PADRÃO IEEE STD 828-1990 24

3.3.1 - Introdução do Plano 25

3.3.2 - Gerenciamento 26

3.3.3 - Tarefas 29

3.3.4 - Cronograma 29

3.3.5 - Recursos 30

3.3.6 - Manutenção do Plano 31

3.4 RECOMENDAÇÕES PARA A ELABORAÇÃO DO PLANO DE GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE 31

REFERÊNCIAS BIBLIOGRÁFICAS 35

LISTA DE FIGURAS

Figura 2.1 Linha de referência / controle de concorrência 7

Figura 2.2 Processo de controle de mudanças 11

Figura 2.3 Árvore de revisões em um item de configuração, usando delta negativo ... 13

Figura 3.1 Elementos relacionados ao plano de gerenciamento de configuração 20

Figura 3.2 Exemplo de organograma para gerenciamento de configuração 27

LISTA DE QUADROS

Quadro 1.1 Tarefas de gerenciamento de configuração de software 2

Quadro 1.2 Padrões de Qualidade 3

Quadro 2.1 Fases de Desenvolvimento de Software 6

Quadro 2.2 Hierarquia dos itens de configuração 8

Quadro 2.3 Identificação dos itens de configuração 9

Quadro 3.1 Comparação entre padrões de planos de gerenciamento de configuração .23

Quadro 3.2 Estrutura do plano de gerenciamento de configuração 25

1. INTRODUÇÃO

1.1 Contexto

Durante o desenvolvimento de um software, é produzida uma grande quantidade de informações, tais como: especificações, planos de projeto, programas fonte, casos e planos de testes, manuais, arquivos de dados, etc. Cada um desses documentos é chamado de *item de configuração de software*. A *configuração de software* compreende todos os itens de configuração produzidos durante o processo de *engenharia de software*, ou seja, no processo de desenvolvimento disciplinado de sistemas [Pressman, 1995].

No período de desenvolvimento, há uma grande mutabilidade nos itens de configuração, pois itens novos podem ser criados e itens já existentes podem ser modificados, gerando novas *versões*. Para que haja maior controle no processo de desenvolvimento de sistemas, é necessário o estabelecimento de normas para a criação e alteração dos itens de configuração. A essas normas dá-se o nome de *gerenciamento de configuração de software*.

Gerenciamento de configuração de software (SCM) já foi definido de várias formas. Bersoff, Henderson e Siegel [Bersoff, 1980] o definem como "uma disciplina de identificação de um sistema em pontos discretos no tempo com a finalidade de controlar sistematicamente as mudanças em sua configuração e manter a integridade e a rastreabilidade dessa configuração, através de todo o ciclo de vida do software". Para Pressman [Pressman, 1995], gerenciamento de configuração de software é "uma atividade de

garantia de qualidade de software que é aplicada durante todas as fases do processo de engenharia de software". Feiler [Feiler, 1991] descreve gerenciamento de configuração de software como "uma disciplina que proporciona estabilidade para a produção de um sistema de software, por controlar a evolução do produto". Segundo Babich [Babich, 1986], "grerenciamento de configuração de software é a arte de identificar, organizar e controlar modificações no software que está sendo construído por uma equipe de programação".

O quadro 1.1 apresenta, resumidamente, as tarefas de gerenciamento de configuração de software e as questões centrais a serem abordadas por elas [Bersoff, 1979; Berlack, 1992; Capretz, 1994; Narayanaswamy, 1987; Pressman, 1995].

Quadro 1.1 - Tarefas de gerenciamento de configuração de software

Tarefa	Questão
Identificação	Como uma organização identifica e administra as muitas versões existentes dos itens de configuração de forma que possibilite que mudanças sejam feitas eficientemente?
Controle de Versão	Como uma organização controla as mudanças antes e depois que o software é liberado?
Controle de Mudanças	Quem tem a responsabilidade pela aprovação e pela determinação de prioridades para as mudanças?
Auditoria de Configuração	Como se pode garantir que as mudanças foram feitas adequadamente?
Relato de Situação	Qual o mecanismo usado para avisar outras pessoas sobre mudanças que são feitas?
Controle de Interface	Como gerenciar o efeito causado por alterações externas ao sistema?
Controle de Subcontratados e Fornecedores	Como garantir que módulos do sistema construídos por terceiros estejam corretos e coerentes com o restante dos sistema?

Alguns órgãos certificadores de qualidade de software (quadro 1.2) exigem o gerenciamento de configuração de software como requisito básico para que uma empresa inicie a melhoria de qualidade do processo de desenvolvimento de software.

Quadro 1.2 - Padrões de Qualidade

Padrões de Qualidade	Recomendação
ANSI/IEEE 983-1986	Exige gerenciamento de configuração de software como providência para garantia de qualidade.
CMM[1] /SEI	Exige gerenciamento de configuração de software para uma empresa sair do nível de maturidade 1, denominado caótico.

Quando se fala em gerenciamento de configuração de software em pequenas empresas, geralmente o pessoal da área de informática reage de duas maneiras: desconhecem o assunto ou o associam com uma tarefa burocrática que apenas "atrasa o serviço".

A primeira reação não se justifica, pois gerenciamento de configuração vem sendo estudado desde os anos sessenta [Berlack, 1992]. Inicialmente, era aplicado da mesma forma para software e hardware, sendo que no final dos anos setenta já havia padrões de gerenciamento de configuração específicos para software.

A segunda reação é resultado de uma visão distorcida, pois os ganhos em tempo e qualidade pela aplicação do gerenciamento de configuração são comprovados e mensuráveis. Esses ganhos podem ser reconhecidos constatando-se que o uso de gerenciamento de configuração de software

---

[1] CMM=Capability Maturity Model desenvolvido pelo Software Engeneering Institute (SEI) para a melhoria da qualidade do processo de desenvolvimento de software. Este modelo classifica as organizações em cinco estágios seqüenciais de maturidade (caótico, repetitível, definido, gerenciado e otimizado) e descreve os passos necessários para evolução [Paulk,1993].

facilita a comunicação entre as equipes de desenvolvedores relatando a situação do software a cada momento, assim como as alterações que foram efetuadas. Isso traz como conseqüência a redução no esforço necessário para efetuar alterações e a redução no número de erros. O resultado final é melhor cumprimento dos cronogramas, redução nos custos e melhora na qualidade do software [Berlack, 1992].

Essas reações podem não ocorrer se as tarefas de gerenciamento de configuração de software forem esclarecidas e orientadas através de um *Plano de Gerenciamento de Configuração* e apoiadas por ferramentas que automatizem algumas dessas tarefas.

No entanto, os padrões de *planos de gerenciamento de configuração de software* existentes (IEEE Std 828-1990, NASA-Sfw-DID-04, DoD-DID) se apresentam apenas como uma lista de tópicos que devem ser abordados na implantação das tarefas de gerenciamento de configuração de software. Essa lista é pouco esclarecedora e não é destinada especificamente para as necessidades de uma empresa de pequeno porte.

O mesmo ocorre com as ferramentas de gerenciamento de configuração de software existentes no meio acadêmico e no mercado (SCCS, RCS, Aide-de-Camp, AllChange, Continuus, CCC, ClearCase, CMS-Digital, CMCV-IBM, CMZ, CVS, MKS Source Integrity, Perforce, PMCS, PVCS, SCLM, SMS, Source Code Manager, VCS-UX). Essas ferramentas geralmente são dispendiosas e necessitam computadores de grande porte ou estações de trabalho para serem executadas. Isso se deve ao fato de serem construídas para serem usadas em grandes projetos. Outro obstáculo para que essas ferramentas sejam usadas é a dificuldade para instalá-las e utilizá-las, pois seus manuais são vagos e a operação complexa. Esses obstáculos colocam as ferramentas além do alcance das pequenas empresas de desenvolvimento de software, que geralmente trabalham com orçamentos reduzidos, projetos pequenos e pessoal inexperiente em gerenciamento de configuração de software.

2. GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE

Este capítulo inicia com a apresentação do processo de desenvolvimento de software e de um conceito fundamental para o Gerenciamento de configuração de software: a *linha de referência* (*baseline*). Posteriormente, serão vistas as tarefas de gerenciamento de configuração de software: Identificação da Configuração, Controle de Versão, Controle de Mudanças, Auditoria de Configuração, Relato de Situação, Controle de Interface e Controle de Vendedores e Subcontratadas.

2.1 Conceitos Fundamentais de Gerenciamento de Configuração de Software

O desenvolvimento de um software pode ser organizado de várias formas. A cada uma dessas formas de organização dá-se o nome de um *paradigma de ciclo de vida*. Os paradigmas mais estudados são o clássico, a prototipação, o modelo espiral e as técnicas de quarta geração [Pressman, 1995].

Um processo de desenvolvimento de software, independente do paradigma de ciclo de vida adotado, inclui as fases de engenharia de sistemas, análise de requisitos, projeto de software, codificação, testes e manutenção [Pressman, 1995]. O quadro 2.1 apresenta uma descrição resumida de cada uma dessas fases.

Quadro 2.1- Fases do Desenvolvimento de Software

Fases	Descrição
Engenharia de Sistemas	Coleta dos requisitos em nível do sistema, com uma pequena quantidade de projeto e análise de alto nível.
Análise de Requisitos	Compreensão do domínio da informação através dos requisitos coletados na fase anterior.
Projeto de Software	Desenvolvimento de quatro atributos distintos do software: estrutura de dados, arquitetura de software, detalhes procedimentais e caracterização de interfaces.
Codificação	Tradução do projeto de software numa forma legível para a máquina.
Testes	Realização de testes dos programas. Esses testes concentram-se nos aspectos lógicos internos do software e nos aspectos funcionais externos.
Manutenção	Modificações após o software ser liberado. Essas mudanças ocorrem por erros, adaptações, novos ambientes e novas funcionalidades.

Durante o processo de desenvolvimento de um software, é produzida uma grande quantidade de itens de informação. Esses itens são chamados *itens de configuração de software* e o conjunto dos mesmos compõem uma *configuração de software*.

A criação e as alterações de cada item de configuração de software devem ser acompanhadas e controladas pelo gerente do projeto. Para isso, é necessário o estabelecimento de pontos bem definidos dentro do processo de desenvolvimento do software: as *Linhas de Referência* (*baselines*) . Esses pontos podem ocorrer ao final de cada uma das fases do processo de desenvolvimento de software, ou de algum outro modo definido pela gerência.

Nos pontos estabelecidos pelas linhas de referência, os itens de configuração devem ser identificados, analisados, corrigidos, aprovados e armazenados ("congelados") como uma configuração de software. Os itens aprovados são armazenados em um local sob controle de acesso,

denominado *repositório dos itens de configuração*. Esses itens só poderão ser alterados após uma solicitação de alteração formalmente aprovada pelo gerente de configuração. Essa é uma forma de prover controle sobre a situação de cada um dos itens de configuração, evitando inconsistências.

O método utilizado para trabalhar com itens de configuração que já estão no repositório é chamado de *Check In/Check Out* [Bersoff, 1979; Harter, 1989], ou seja, conferência na entrada e conferência na saída. Quando for desejada uma alteração em algum item de configuração do repositório, uma cópia do item é colocada numa área de trabalho do desenvolvedor ("*check out*"). A partir desse momento, nenhum outro desenvolvedor poderá alterar o mesmo item, a isso dá-se o nome de *controle de concorrência*. Dentro de sua área, o desenvolvedor tem total liberdade de trabalho. Após o final das alterações no item de configuração, ele será revisado e recolocado no repositório ("*check in*"). Nesse momento, uma nova linha de referência deverá ser traçada, de modo que uma nova configuração contendo o item alterado seja formada e congelada no repositório (Figura 2.1).

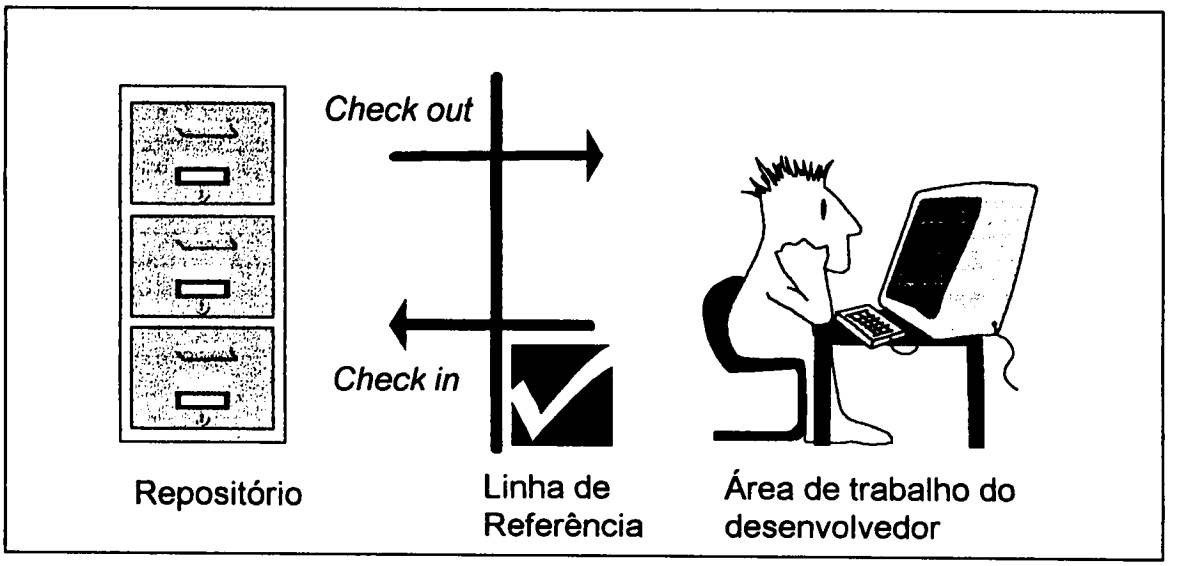

Figura 2.1 - Linha de Referência / Controle de concorrência

Depois do congelamento, o acesso é liberado, permitindo que outros desenvolvedores também executem alterações sobre esse item de configuração.

2.2 Tarefas de Gerenciamento de Configuração de Software

As tarefas de gerenciamento de configuração de software, que respondem às questões apresentadas no quadro 1.1 são descritas a seguir.

2.2.1 - Tarefa 1- Identificação da Configuração de Software

O primeiro passo para a identificação é selecionar os itens a serem gerenciados. Como exemplo, apresentamos no quadro 2.2 uma série de itens sugeridos por Pressman [Pressman, 1995]:

Quadro 2.2- Hierarquia dos itens de configuração

1.	Especificação do Sistema
2.	Plano de Projeto de Software
3.	Especificação de Requisitos do Software
4.	Manual Preliminar do Usuário
5.	Especificação do Projeto
	a) Descrição do Projeto de Dados
	b) Descrição do Projeto Arquitetural
	c) Descrições do Projeto Modular
	d) Descrições do Projeto de Interface
	e) Descrições de Objetos (se forem usadas técnicas orientadas a objetos)
6.	Listagem do código-fonte
7.	Planos, Procedimentos, Casos de Testes e Resultados Registrados
8.	Manuais Operacionais e de Instalação
9.	Programa Executável e Módulos Interligados
10.	Descrição do Banco de Dados
	a) Esquema e estrutura de arquivo
	b) Conteúdo inicial
11.	Manual do Usuário
12.	Documentos de Manutenção
	a) Relatórios de problemas de software
	b) Solicitações de manutenção
	c) Pedidos de mudança
13.	Padrões e procedimentos para engenharia de software
14.	Ferramentas de produção de software (editores, compiladores, CASE, etc.)

É importante que seja efetuada uma seleção dos itens relevantes, porque uma superdocumentação torna o gerenciamento de configuração muito oneroso. Geralmente, devem sofrer gerenciamento de configuração os

Instituto de Ciências Matemáticas Biblioteca

itens mais usados no ciclo de vida, os mais genéricos, os mais importantes para a segurança, os itens projetados para reuso e os que podem ser modificados por vários desenvolvedores ao mesmo tempo [Bersoff, 1992]. Somente os itens selecionados serão controlados, sendo que os outros itens poderão ser alterados livremente.

Após a seleção, deve-se descrever como esses itens se relacionam. Consideram-se cinco classes de relacionamento: *equivalência* (ex: BD em disco e em fita magnética), *dependência* (ex: a descrição do projeto modular é dependente da especificação do projeto), *derivação* (ex: código objeto é derivado do código fonte), *sucessão* (ex: a versão 1.2 é sucessora da versão 1.1) e *variante* (ex: versão para DOS ou para UNIX). A identificação desses relacionamentos é muito importante para a manutenção, pois permite que se localize rapidamente os itens afetados por cada alteração.

Depois de escolhidos os itens e estabelecidos os relacionamentos, deve-se criar um *esquema de identificação* dos itens com a atribuição de nomes únicos a cada um dos componentes, de forma que seja possível reconhecer a evolução de cada uma das versões dos componentes e a hierarquia existente entre componentes, a partir de seus nomes [Bersoff, 1979; Karakas, 1992]. Um exemplo simples para um pequeno programa cuja sigla é "AA" é apresentado no quadro 2.3. O esquema de identificação utiliza a combinação de nome do projeto, tipo de item, nome do item e versão.

Quadro 2.3 -Identificação dos itens de configuração

Item	Projeto	Tipo	Nome	Versão	Nome completo
Especificação do Sistema	AA	ES		1.1	AAES v1.1
Plano de Projeto	AA	PP		1.1	AAPP v1.1
Especificação de Requisitos do Software	AA	ER		1.1	AAER v1.1
Especificação de Projeto	AA	EP		1.1	AAEP v1.1
Programa Fonte	AA	PF	Prin	1.1	AAPFPrin v1.1
Programa Fonte (sub-rotinas)	AA	PF	Rot	1.1	AAPFRot v1.1
Plano e Casos de Testes	AA	TT		1.1	AATT v1.1
Nova versão das sub-rotinas	AA	PF	Rot	1.2	AAPFRot v1.2

Após o estabelecimento do esquema de identificação, devem ser planejadas as linhas de referência dentro do ciclo de vida do projeto. Geralmente, cria-se uma linha de referência ao final de cada fase do ciclo de vida do projeto e, periodicamente, depois de cada manutenção. Deve-se especificar quais itens fazem parte de cada uma das linhas de referência planejadas.

2.2.2 - Tarefa 2 - Controles de Mudança e de Versões

Dois controles básicos são instituídos no processo de gerenciamento de configuração de software: *Controle de Mudanças* e *Controle de Versões*.

a) Controle de Mudanças

Durante o processo de desenvolvimento de software mudanças descontroladas podem levar rapidamente ao caos [Presmann, 1995]. Assim, deve ser instituído na organização um processo que combine procedimentos humanos e ferramentas automatizadas para proporcionar um mecanismo de controle das mudanças. Esse processo deve ser implementado depois que uma linha de referência for fixada, antes disso somente um controle de mudanças informal precisa ser aplicado [Bersoff, 1979; Bersoff, 1984; Bersoff, 1992; Presmann, 1995]. A figura 2.2 ilustra um processo de controle de mudanças que pode ser implementado para os itens que já passaram por uma linha de referência.

De acordo com esse processo de controle de mudanças, quando um pedido de mudança é solicitado, primeiramente ele deve ser analisado, gerando um relatório de mudanças. Esse relatório é encaminhado para avaliação, se aprovado o relatório de mudança segue para o Gerente de Configuração. O Gerente de configuração controla o acesso aos itens no repositório, liberando-os para a equipe de desenvolvimento para que a mudança seja efetuada, e recebendo os itens, quando atualiza o repositório. Caso o pedido de mudança não seja aprovado, o relatório e o pedido são arquivados e é dado um retorno ao solicitante.

Esse controle possibilita que as mudanças sejam efetuadas, comunicadas e incorporadas de um modo disciplinado. Entretanto, para que esse controle ocorra de forma satisfatória, qualquer mudança que ocorra nos itens de configuração de software, após o estabelecimento de uma linha de referência, deve seguir efetivamente sempre o mesmo caminho [Bersoff, 1992].

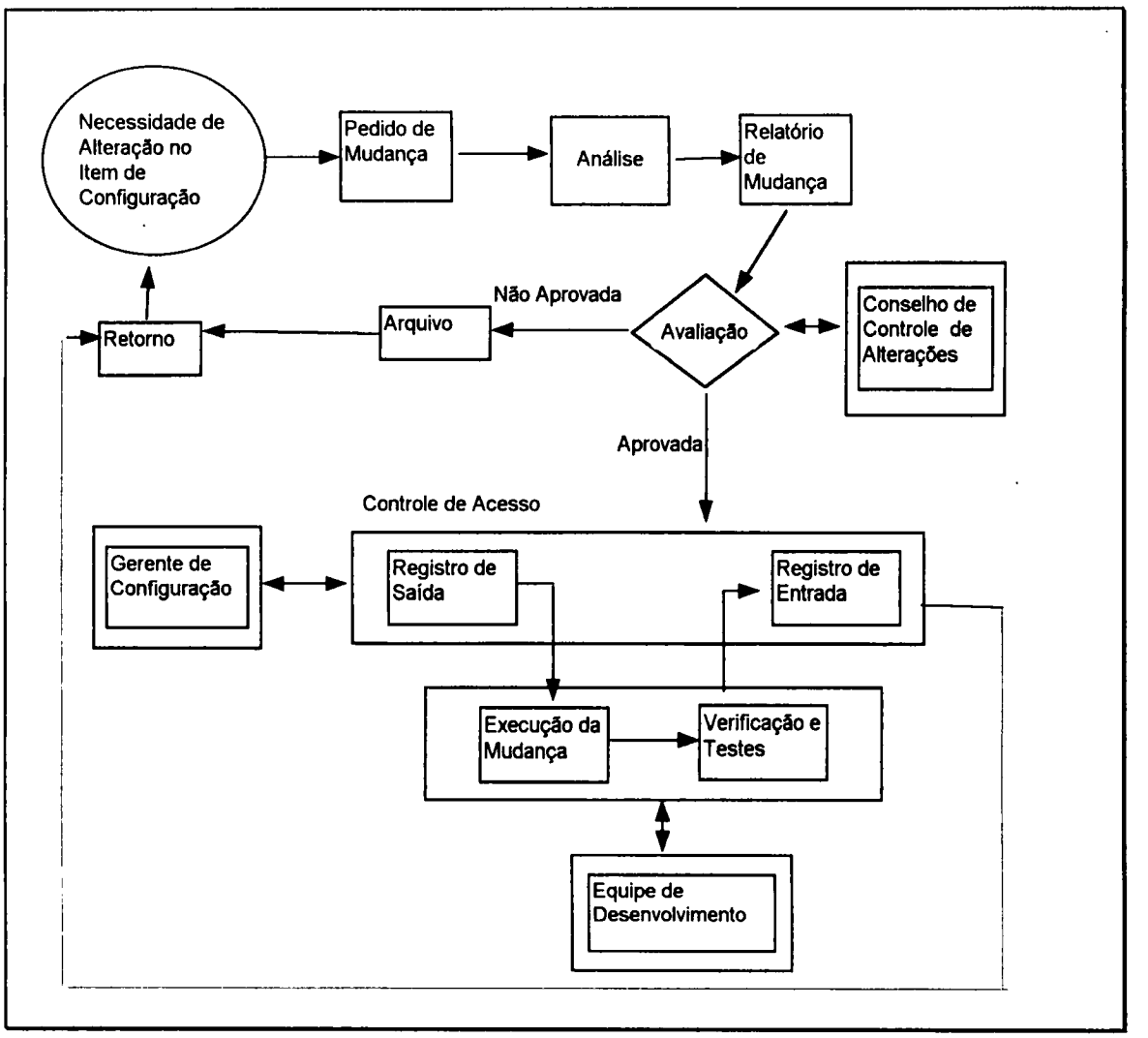

Figura 2.2 - Processo de Controle de Mudanças

No processo de controle de mudanças, as alterações aprovadas são efetuadas de maneira sincronizada. O objetivo dessa sincronização é evitar que duas pessoas efetuem, ao mesmo tempo, mudanças incompatíveis em um mesmo item, criando inconsistências [Bersoff, 1992]. O método mais utilizado para evitar inconsistências é controlar o acesso ao repositório, de

forma que, quando um desenvolvedor retira um item para alterações, ele bloqueia o acesso de escrita no item para os outros desenvolvedores (vide seção 2.1).

Os procedimentos de controle das mudanças asseguram que as mudanças em um software sejam feitas de modo controlado, permitindo-se prever o efeito das mesmas em todo o sistema. Procedimentos formais de organização e de controle das mudanças no sistema permitem que os pedidos de alteração possam ser considerados em conjunto com outros pedidos. Desse modo, os pedidos similares podem ser agrupados, e os pedidos incompatíveis entre si ou com os objetivos do sistema identificados. Também podem ser atribuídas prioridades aos pedidos e, de acordo com essas prioridades, pode-se gerar um cronograma.

b) Controle de Versões

Um item, ao ser desenvolvido, evolui até que atinja um estado em que atenda aos propósitos para o qual foi criado. Isso implica em diversas alterações, gerando uma *versão* (*revisão*) do item. Para estabelecer o controle sobre as diversas versões, todas as versões devem ser armazenadas e identificadas. Isso, geralmente, é feito com o auxílio de uma ferramenta.

A versão do item pode ser incluída no esquema de identificação (vide seção 2.2.1). É conveniente que o esquema de identificação das versões dos itens seja feito em forma de árvore, pois ao mesmo tempo em que mantém um histórico das versões dos itens, permite identificação única e ramificações a partir de qualquer versão (Figura 2.3).

Um item pode existir simultaneamente em duas ou mais formas diferentes que atendam a requisitos similares, neste caso temos *variantes* do item. Um exemplo seria o de duas subrotinas para retornar a data do sistema operacional: uma para Unix e outra para OS/2 (versões 2.1.1 e 2.2.1 na Figura 2.3).

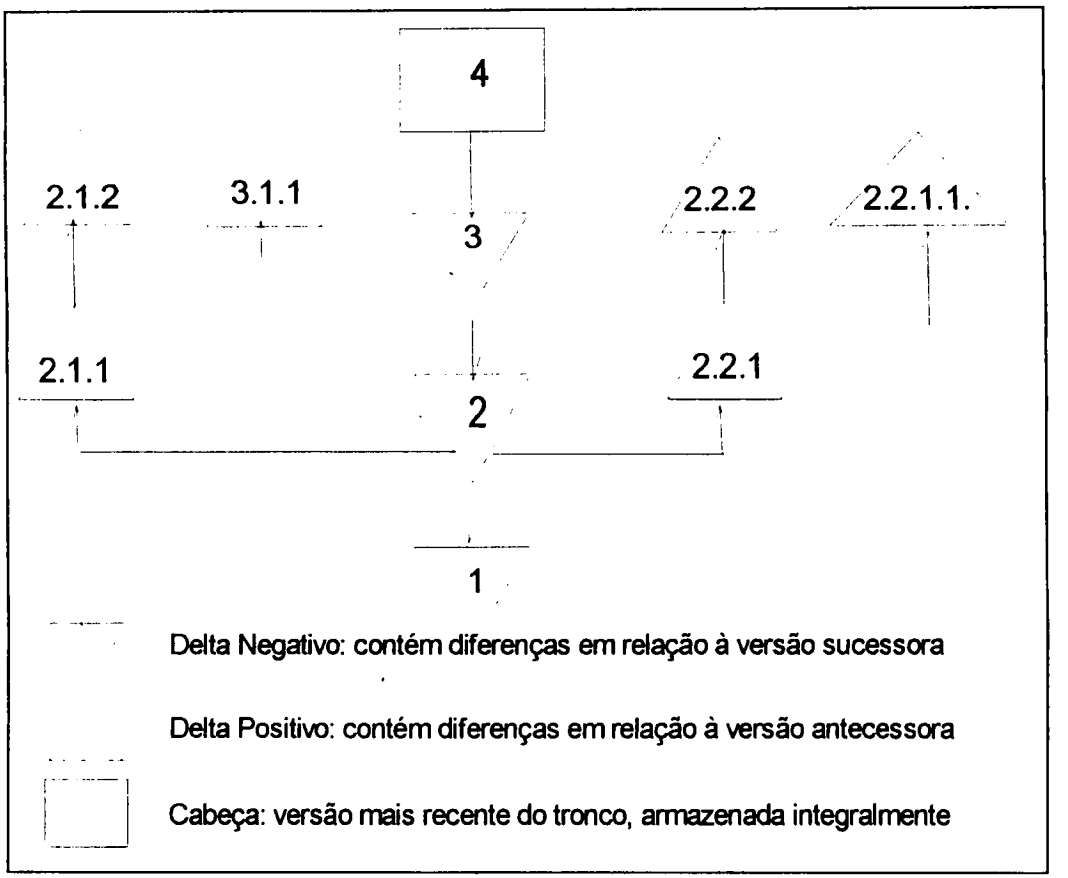

Figura 2.3- Árvore de revisões em um item de configuração, usando delta negativo.

Para minimizar o espaço de armazenamento das versões utiliza-se o conceito de *delta*, ou seja, são armazenadas uma versão completa e as diferenças entre as versões [Ambriola, 1990; Boas, 1994; Brown, 1991; Dart, 1991; Humphrey, 1989]. Há duas variações desse conceito: *delta negativo* e *delta positivo*. Com o delta negativo, armazena-se integralmente a versão mais recente e as diferenças (deltas) existentes até então. Com o delta positivo, armazena-se a versão mais antiga e para montar as versões mais recentes, processam-se as diferenças (deltas) armazenadas.

Os sistemas atuais de gerenciamento de versões utilizam o conceito de delta negativo no tronco, por ser mais comum a utilização de versões mais recentes do item de configuração [Boas, 1994]. A figura 2.3 representa um caso em que se utiliza delta negativo. A única versão armazenada integralmente é a 4. As outras versões são construídas, quando solicitadas, a partir da 4 e das diferenças armazenadas. Utiliza-se deltas negativos no

tronco da árvore, que representa o caminho principal de evolução do item. Os ramos representam as variantes dos itens e são obtidas pela utilização de delta positivo.

2.2.3 - Tarefa 3 - Auditoria da Configuração

A identificação e controle das alterações ajudam a manter ordem mas, para assegurar que a alteração foi implementada apropriadamente, há necessidade de auditorias na configuração do software [Sommerville, 1989].

Existem dois tipos de auditoria de configuração de software que são pré-requisitos para o estabelecimento das linhas de referência no ciclo desenvolvimento de software: a *Auditoria Funcional* e a *Auditoria Física*.

A auditoria funcional preocupa-se com aspectos internos dos arquivos, compreendendo uma verificação técnica formal na configuração de software, que deve ser realizada ao ser fixada uma linha de referência. Esta verificação é uma atividade de controle de qualidade que tenta descobrir omissões ou erros na configuração, que degradam os padrões de construção do software [Capretz, 1992; Capretz, 1994; Pressman, 1995].

A auditoria física é um processo administrativo que ocorre no final de cada fase do ciclo de vida do software e consiste em verificar se a configuração, a ser congelada pela linha de referência, está composta da versão mais recente dos itens de configuração, determinados para a fase do ciclo de vida específica [Bersoff, 1979; Bersoff, 1984; Capretz, 1992] e se os procedimentos e padrões foram devidamente aplicados.

2.2.4 - Tarefa 4 - Relato da Situação

O objetivo dessa tarefa de gerenciamento de configuração é relatar a todas as pessoas envolvidas no desenvolvimento e na manutenção do software as seguintes informações sobre as alterações na configuração de software:

a) O que aconteceu?

b) Quem o fez?

c) Quando aconteceu?

d) O que mais será afetado?

Para isso, deve ser criado um banco de dados sobre as ocorrências no gerenciamento de configuração. Esse banco de dados deve estar disponível aos desenvolvedores com acesso através de palavras chave. Além disso, deve ser gerado regularmente um relatório de situação para informar as alterações mais importantes. O acesso rápido às informações sobre a configuração agiliza o processo de desenvolvimento e melhora a comunicação entre as pessoas, o que é uma maneira de eliminar muitos problemas relativos à modificação do mesmo item de informação, com intenções diferentes e conflitantes.

2.2.5 - Tarefa 5 - Controle de interface

As atividades de controle de interface coordenam as mudanças nos itens de configuração que são afetados por itens que não estejam sendo controlados. Equipamentos, sistemas de software, software de suporte, assim como outros projetos devem ser examinados na busca por possíveis interfaces com o projeto sob controle.

Para cada interface o plano deve descrever:

a) O tipo de interface.

b) As unidades organizacionais afetadas.

c) Como será feito o controle sobre a interface.

d) Como os documentos de controle da interface serão aprovados.

2.2.6 - Tarefa 6 - Controle de subcontratados e fornecedores

As atividades de controle de subcontratados e fornecedores coordenam a forma como os itens que foram desenvolvidos por solicitação a

outras empresas ou foram adquiridos já prontos são testados e incorporados ao repositório do projeto.

Para itens subcontratados o plano deve descrever:

a) Os requisitos de gerenciamento de configuração de software a serem satisfeitos pelo subcontratado.

b) Como será feito o monitoramento sobre o subcontratado.

c) Como o código, documentação e dados externos serão testados, aceitos e adicionados ao projeto.

d) Como serão tratadas as questões de propriedade do código produzido, como direitos autorais e *royalties*.

Para itens adquiridos prontos o plano deve descrever:

a) Como serão recebidos, testados e colocados sob controle de gerenciamento de configuração.

b) Como as mudanças no software do fornecedor serão tratadas.

c) Se e como o fornecedor participará no processo de gerenciamento de mudança do projeto.

Itens de configuração poderão ser adquiridos de fornecedores, subcontratados, clientes, outros projetos ou outras fontes.

2.3 Considerações Finais

Neste capítulo, foram apresentados conceitos gerais e as principais tarefas de gerenciamento de configuração de software. Um estudo detalhado das necessidades específicas de cada ambiente de desenvolvimento de software, no que diz respeito ao gerenciamento de configuração de software, é necessário para que seja possível a execução das tarefas de forma mais adequada a cada situação. O documento gerado após esse estudo é chamado *Plano de Gerenciamento de Configuração de Software*. No próximo

capítulo, planos de gerenciamento de configuração de software são estudados em maiores detalhes.

3. PLANO DE GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE

Um plano representa um método através do qual tarefas podem ser definidas e executadas. Este capítulo apresenta elementos organizacionais que influenciam a preparação do plano de gerenciamento de configuração de software e alguns padrões de planos de gerenciamento de configuração de software. O padrão IEEE Std 828-1990 é estudado em detalhes. No final do capítulo são apresentadas recomendações para a elaboração de um plano de gerenciamento de configuração de software.

3.1 Elementos envolvidos no Plano de Gerenciamento de Configuração

O plano de gerenciamento de configuração de software é um documento que descreve quais atividades devem ser efetuadas na implantação e na administração do processo de gerenciamento de configuração de software. Neste plano também é descrito como e quando as atividades serão efetuadas, quem serão os responsáveis por elas e que recursos serão necessários [Whitgifh, 1991, IEEE Std 828-1990]. O plano de gerenciamento de configuração de software é desenvolvido de acordo com as peculiaridades de cada empresa, ou seja, ele depende das metas e soluções alternativas específicas a cada ambiente.

Dez elementos estão envolvidos no plano de gerenciamento de configuração de uma organização [Bounds, 1993]. Sete deles referem-se à etapa de *preparação* do plano e três à etapa de *implantação*.

Preparação do Plano:

1. **Plojeto do plano:** determinar a estrutura do plano de gerenciamento de configuração. Neste passo é recomendado que sejam estudados alguns padrões internacionais de gerenciamento de configuração de software.
2. **Processo:** descrever as tarefas de gerenciamento de configuração que serão implementadas e os controles que serão efetuados nas mesmas.
3. **Pessoas:** relacionar pessoas às funções, tarefas e responsabilidades envolvidas no gerenciamento de configuração.
4. **Ambiente:** analisar o tipo de cultura que existe na organização a fim de buscar uma solução de gerenciamento de configuração que seja apropriada a esta cultura.
5. **Sistema:** determinar os sistemas ou partes de sistemas que serão colocados sob o controle do gerenciamento de configuração e determinar os itens de informação que são relevantes para compor a configuração de software desses sistemas.
6. **Automatização:** decidir os requisitos mínimos que uma ferramenta de gerenciamento de configuração de software deverá apresentar em termos de funcionalidade, custo, plataforma necessária, facilidade de uso, etc.
7. **Decisões Gerenciais:** identificar como a gerência encara a utilização de ferramentas de gerenciamento de configuração de software que já existem no mercado e o desenvolvimento parcial ou integral de ferramentas de gerenciamento de configuração de software na própria organização.

Implantação do Plano:

1. **Elaboração do plano:** escrever o plano de forma a atender todos os requisitos identificados nas sete fases da etapa de preparação.
2. **Implantação de ferramentas de gerenciamento de configuração de software:** implantar as ferramentas adquiridas ou desenvolvidas para

auxiliar o gerenciamento de configuração. Este passo pode incluir treinamento.

3. **Elaboração da estratégia de implantação:** desenvolvimento de um roteiro para auxiliar a implantação do processo de gerenciamento de configuração na organização e para institucionalizar este processo. A institucionalização ocorre após intensiva realização das tarefas e utilização das ferramentas e a identificação de necessidades de alterações e complementações neste processo.

Os elementos relacionados ao plano de gerenciamento de configuração de software são representados na figura 3.1.

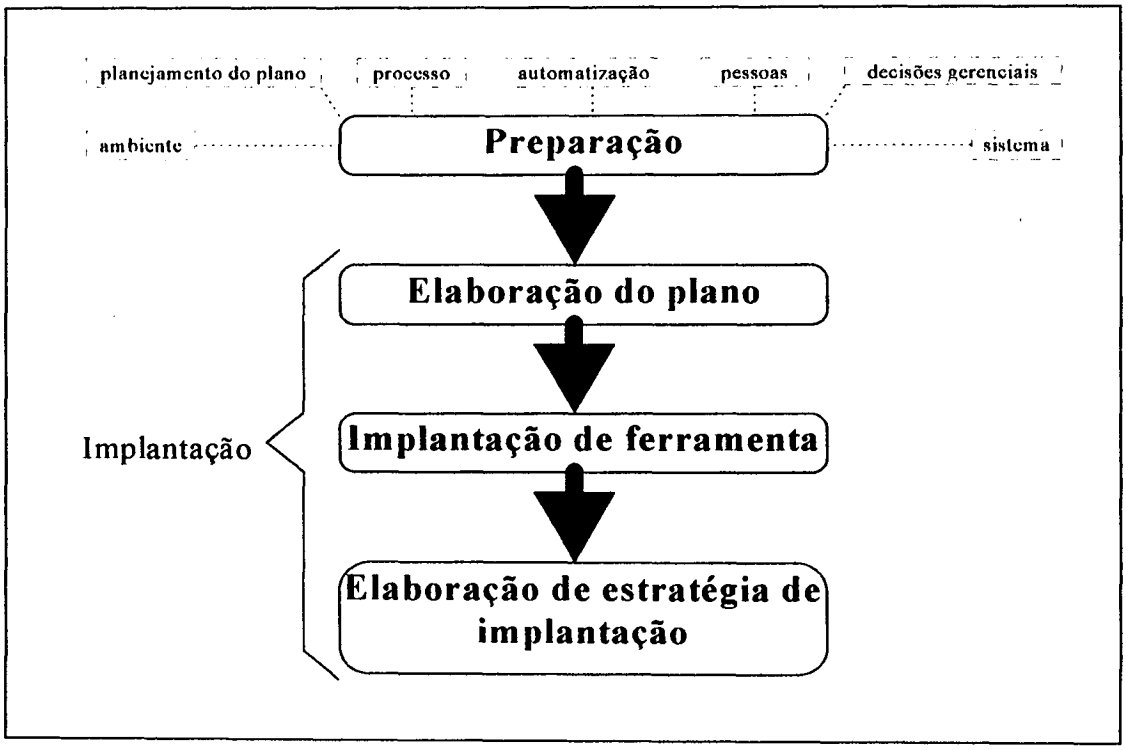

Figura 3.1 - Elementos relacionados ao plano de gerenciamento de configuração

A elaboração do plano é um dos três fatores chave para o sucesso em gerenciamento de configuração. Em muitos casos, o gerenciamento de configuração faz parte de um projeto de melhoria de toda a empresa, e neste caso, o plano deve estar em concordância com os outros planos relacionados com o esforço geral de melhoria.[Bounds, 1993]

3.2 Alguns Padrões de Planos de Gerenciamento de Configuração de Software

Ao iniciar o desenvolvimento de um plano de gerenciamento de configuração de software, é natural a dúvida sobre como começar e qual a forma mais adequada de apresentação das informações. É recomendado que sejam estudados alguns dos padrões internacionais de planos de gerenciamento de configuração e que seja escolhido como base aquele que tenha maior compatibilidade com o projeto para o qual o plano esteja sendo escrito [Buckley, 1993].

Alguns dos padrões mais amplamente utilizados são:

- *IEEE Standard for Configuration Management Plans* (IEEE Std 828-1990): Deve ser analisado juntamente com o IEEE Std 1042-1986, um guia de gerenciamento de configuração, e o IEEE Std 610.12-1990, um glossário de termos de engenharia de software. O grande valor deste padrão é que ele registra o consenso, entre os profissionais da área, sobre o que deve ser feito.
- *Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs* (MIL-STD-483A-1979): Especifica o formato e o conteúdo necessário para um plano de gerenciamento de configuração. O nível de detalhes é superficial e requer ampliação se *software* ou *firmware* forem incluídos.
- *NASA Software Configuration Management Plan Item Description* (NASA-Sfw-DID-04): Foi escrito para um segmento específico de indústria e é muito bem organizado, o que o torna fácil de usar. Os capítulos sobre controle de mudanças e documentação merecem destaque, porém o seu maior problema é que pede ao usuário que crie um plano para cada item de configuração, o que é trabalhoso e cria redundâncias. Outro problema é a sua superficialidade.

- *DoD Software Development Plan Data Item Description* juntamente com DoD-STD-2167A (DI-MCCR-80030A): A abordagem utilizada é introduzir as informações de planejamento de gerenciamento de configuração de software no próprio plano de desenvolvimento de software. O problema é que desta maneira as informações de gerenciamento de configuração ficam dispersas no plano de desenvolvimento, não sendo possível identificá-las separadamente. Se o contrato forçar o uso desta abordagem, o problema pode ser contornado colocando-se o plano de gerenciamento de configuração de software como um apêndice do plano de desenvolvimento de software.

Bounds e Dart [Bounds, 1993] analisaram os padrões três desses padrões (quadro 3.1) utilizando os critérios facilidade de uso, completitude, adaptabilidade, correção, consistência e adequação ao ciclo de vida do software.

•Facilidade de uso é medida pelo tempo e pelo esforço realizado para aprender como usar um padrão, e pelo esforço realizado para executar o plano. Atenção especial é dada àqueles que estão escrevendo um plano pela primeira vez.

•Completitude é conseguida quando o usuário do padrão não tem que procurar informações em outros lugares além do padrão para escrever um plano, ou seja, apenas com o padrão em mãos é possível escrever o plano.

•Adaptabilidade é a medida do quanto o padrão pode ser modificado de forma a atender às necessidades de cada projeto e com que facilidade isso pode ser feito.

•Correção é a medida em que o padrão contém apenas informações corretas, sem contradições ou informações inúteis ou impraticáveis.

•Consistência é obtida se o padrão é bem estruturado e se há uniformidade através de todo o documento, especialmente na nomenclatura.

•Adequação ao ciclo de vida do software só pode ser conseguida por um bom padrão e representa o quanto o padrão de gerenciamento de configuração e as tarefas de gerenciamento de configuração se encaixam no ciclo de vida do projeto.

O padrão IEEE obteve melhores resultados nos quatro primeiros critérios, empatando nos dois últimos. Essas características fazem com que o padrão IEEE seja o mais adequado para desenvolvimento de planos de gerenciamento de configuração de software de uso geral.

Quadro 3.1- Comparação entre padrões de planos de gerenciamento de configuração

Critério	IEEE	NASA	DoD
Facilidade de Uso	3	1	1
Completitude	2	1	0
Adaptabilidade	3	1	1
Correção	3	1	1
Consistência	3	3	3
Adequação ao ciclo de vida	1	1	1

0=não satisfaz requisitos mínimos
1=satisfaz os requisitos para um padrão regular
2=satisfaz os requisitos para um bom padrão
3=satisfaz os requisitos para um ótimo padrão

Outras características de destaque do padrão IEEE são:

- Não é um padrão específico para determinado ramo da indústria (o que ocorre com os da NASA e DoD), e ainda permite adaptações;
- Não exige grande experiência anterior para que um plano seja escrito;
- É aplicável a todo o ciclo de vida do *software*;
- Abrange o controle sobre empresas subcontratadas ou vendedoras de itens e as informações necessárias a este controle;
- Provê uma lista de interfaces possíveis, e as informações mínimas necessárias em cada uma.

Na próxima seção será apresentada a estrutura do plano de gerenciamento de configuração proposto pelo padrão IEEE Std 828-1990.

3.3 O Padrão IEEE Std 828-1990

Este padrão foi escrito pelo subcomitê de engenharia de software do Instituto dos Engenheiros Eletro-Eletrônicos dos Estados Unidos, para estabelecer o conteúdo mínimo necessário em um plano de gerenciamento de configuração de software e para definir as atividades e requisitos de gerenciamento de configuração de software necessários durante o desenvolvimento do software.

Este padrão é uma atualização do padrão do ano de 1983, ocorrida devido às novas práticas de engenharia de software e ao novo padrão de gerenciamento de configuração: IEEE Std 1042-1987. Os usuários do padrão poderão ser todos os envolvidos no planejamento das atividades de gerenciamento de configuração ou na execução de auditorias de gerenciamento de configuração de software.

O plano resultante da adoção deste padrão tem a estrutura apresentada no quadro 3.2.

Quadro 3.2 - Estrutura do Plano de Gerenciamento de Configuração

1	Introdução
2	Gerenciamento
	2.1 Organização
	2.2 Responsabilidades
	2.3 Políticas, diretivas e procedimentos aplicáveis
3	Tarefas
	3.1 Identificação da configuração
	3.1.1 Seleção dos itens de configuração
	3.1.2 Identificação dos itens de configuração
	3.1.3 Relacionamentos entre itens de configuração
	3.2 Controle de configuração
	3.2.1 Solicitação de mudanças
	3.2.2 Avaliação de mudanças
	3.2.3 Aprovação ou recusa de mudanças
	3.2.4 Implementação de mudanças
	3.3 Relato de situação da configuração
	3.4 Auditorias e revisões da configuração
	3.5 Controle de interface
	3.6 Controle de vendedores/subcontratados
4	Cronograma
5	Recursos
6	Manutenção do plano

3.3.1 - Introdução do Plano

Esta seção descreve o propósito do plano, as possíveis aplicações, termos chave e referências. Deve ser apresentado um resumo bem simplificado dos conceitos das tarefas de gerenciamento de configuração que serão efetuadas, para que as pessoas que tenham que interagir com gerenciamento de configuração o compreendam melhor. Quatro tópicos são importantes:

- Propósito: as primeiras necessidade de quem lê o plano são saber por que ele existe e quem deve lê-lo. Também é interessante que se faça uma análise de riscos da implantação do gerenciamento de configuração.
- Escopo: deve ser descrito onde o gerenciamento de configuração será aplicado, e as limitações e suposições sobre as quais o plano está baseado. Podem ser incluídas também previsões de custo e prazos, descrição do projeto onde será aplicado, participações dos clientes, ferramentas disponíveis, etc.

- Termos chave: devem ser definidos para que se tenha uma terminologia única entre todos os usuários do plano.
- Referências: todas as referências que o plano fizer a políticas, padrões, procedimentos, terminologias ou outros documentos devem ser identificadas, para que os usuários possam ter acesso.

3.3.2 - Gerenciamento

Esta seção identifica os responsáveis e as autoridades que devem acompanhar as atividades planejadas. Devem ser descritas as unidades organizacionais envolvidas com gerenciamento da configuração e as responsabilidades de cada unidade. Cada uma das tarefas de gerenciamento de configuração deve ter uma unidade responsável. É recomendável que se registre os atributos necessários e desejáveis para o ocupante de cada cargo.

Para grandes organizações, é geralmente necessário criar departamentos, cada um responsável por determinada tarefa para um único projeto, algumas vezes com hierarquias entre esses departamentos. Para projetos e/ou organizações menores, um departamento pode executar determinada tarefa para vários projetos ou um departamento pode ser composto de uma única pessoa ou mesmo podem ser atribuídas várias tarefas a uma única pessoa, desde que seja feita uma análise cuidadosa sobre os possíveis resultados dessa acumulação de tarefas. Não seria aconselhável, por exemplo, permitir que alguém fosse programador e gerente de configuração do mesmo projeto.

Whitgifh [Whitgifh, 1991] apresenta, em seu livro sobre métodos e ferramentas para gerenciamento de configuração de software, um organograma para um projeto de médio porte com aproximadamente quinze membros na equipe de projeto [figura 3.2].

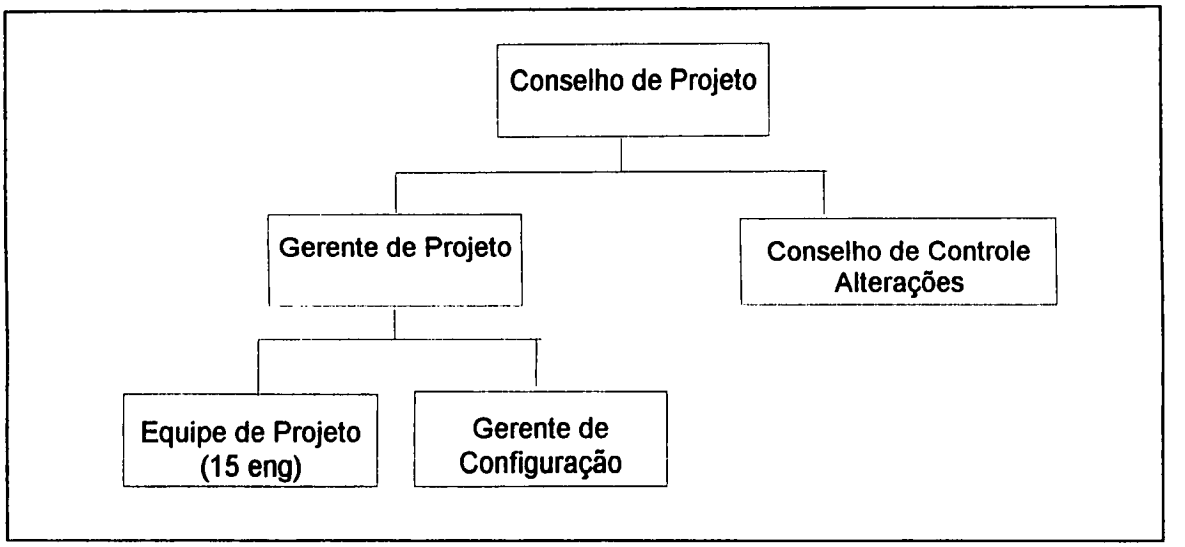

Figura 3.2- Exemplo de organograma para gerenciamento de configuração

Cada unidade neste organograma possui diversas tarefas. Apenas as tarefas associadas ao gerenciamento de configuração serão apresentadas.

O *conselho de projeto* é responsável pelos objetivos gerais do projeto e pela alocação dos recursos necessários. É composto por três membros: um executivo chefe, o gerente do projeto e um representante experiente do cliente. O gerente do projeto e o representante do cliente são indicados especialmente para cada projeto cuja configuração de software é colocada sobre controle. Assim, esse conselho difere de projeto a projeto. O conselho deve revisar e aprovar todos os planos do projeto, inclusive o plano de gerenciamento de configuração.

O *gerente de projeto* é responsável por garantir que o software seja executado no prazo, dentro do orçamento, e com o devido padrão de qualidade. Deve estar atento a desvios que estejam ocorrendo em relação aos planos, prevendo os problemas, e comunicando-os ao conselho de projeto. O gerente de projeto envia o plano de gerenciamento de configuração ao conselho de projeto para aprovação e garante que ele seja executado.

A *equipe de projeto* projeta e implementa o sistema de software, sob a coordenação do gerente de projeto. A equipe deve executar as tarefas de

gerenciamento de configuração, de acordo com o plano. Solicitações de alteração na configuração de software podem ser feitas por qualquer membro da equipe, ou pelo gerente de projeto, e são submetidas à aprovação do conselho de controle de alterações.

O *gerente de configuração* deve, primeiramente, escrever o plano de gerenciamento de configuração, junto com o gerente do projeto e, a partir de então, torna-se responsável pela implementação das tarefas rotineiras do plano. São tarefas do gerente de configuração:

- Criar a estrutura do repositório de itens de configuração de modo que ele atenda o esquema de identificação previsto no plano.
- Garantir a segurança e a integridade do repositório através de controle de acesso e cópias de segurança.
- Implementar as rotinas para criar novos itens de configuração para retirar (*check out*) e armazenar (*check in*) itens no repositório.
- Gerar os relatórios de situação da configuração, quando necessários.
- Gerenciar a liberação de versões de sistemas para os clientes.

O *conselho de controle de alterações* é responsável pela integridade do software produzido pelo projeto. É composto por três membros: o gerente do projeto, um representante do cliente e um coordenador de garantia de qualidade. Trabalhando com os relatórios de situação da configuração, o conselho é auxiliado por engenheiros da equipe de projeto sempre que necessário. O conselho de controle de alterações aprova ou rejeita as solicitações de alteração para itens congelados e monitora a execução das alterações que foram aprovadas. Se o impacto da alteração solicitada excede certos limites definidos, a solicitação de alteração deve ser aprovada também pelo conselho de projeto. Assim, o conselho de projeto atua como um conselho de controle de alterações de alto nível para o projeto.

3.3.3 - Tarefas

Tanto as tarefas técnicas quanto as tarefas administrativas devem ser identificadas e descritas detalhadamente.

As tarefas de gerenciamento de configuração de software propostas pelo plano IEEE estão de acordo com as descritas no capítulo 2 desta mini-dissertação e, desse modo, não serão aqui reapresentadas. As tarefas de gerenciamento de configuração são tradicionalmente classificadas em: identificação da configuração, controle da configuração, relato de situação e auditorias e revisões, controle de interfaces e controle de vendedores e subcontratados.

3.3.4 - Cronograma

Esta seção descreve como será a coordenação temporal entre as tarefas de gerenciamento de configuração e as outras atividades do projeto.

As tarefas de gerenciamento de configuração ocorrem em diferentes momentos do ciclo de vida do software. Algumas tarefas são constantes no decorrer de todo o projeto, algumas ocorrem em determinados momentos particulares e outras surgem durante o projeto.

O ponto mais importante do cronograma é definir quando o controle de mudanças começa a existir, ou seja, quando as linhas de referências serão estabelecidas. É muito importante que o início do controle de mudanças não seja nem muito precoce nem muito tardio.

A forma para a especificação da seqüência e coordenação das tarefas de gerenciamento de configuração de software pode ser a mesma utilizada para determinar cronograma de projetos, ou seja, através de um gráfico de Gantt, Pert ou Timeline. A determinação das datas pode ser feita utilizando-se datas absolutas, ou datas relativas a eventos a ocorrer.

3.3.5 - Recursos

Esta seção descreve os recursos que serão necessários para executar as tarefas de gerenciamento de configuração descritas no capítulo 2. São identificados ferramentas de software, pessoal, treinamento e equipamentos necessários.

O gerenciamento de configuração pode ser feito com uma combinação entre ferramentas de software e manual de procedimentos. As ferramentas tanto podem ser específicas para o gerenciamento de configuração como podem ser ferramentas genéricas. Também podem ser padronizadas para todos os projetos ou pode haver ferramentas específicas para cada projeto.

As ferramentas podem apoiar as seguintes funções:

- Manutenção da estrutura do repositório;
- Controle de acesso;
- Desenvolvimento de documentação;
- Controle de versões do código fonte;
- Geração de linhas de referências;
- Processamento, comunicação e autorização de mudanças;
- Relato de situação;
- Relacionamento entre problemas e alterações;
- Arquivamento, bloqueio e recuperação de itens controlados;
- Criação do plano de gerenciamento de configuração.

Para cada tipo de tarefa de gerenciamento de configuração, o plano deve identificar quais ferramentas, técnicas, equipamentos, pessoas, e treinamento são necessários e como cada recurso será obtido.

Para cada ferramenta de software, seja ela adquirida externamente ou desenvolvida juntamente com o projeto, o plano deve descrever ou

referenciar suas funções e deve identificar o controle de configuração que deve ser feito sobre a ferramenta.

3.3.6 - Manutenção do Plano

Esta seção descreve como manter o plano sempre atualizado. Isso envolve a identificação das atividades e responsabilidades necessárias para manter um controle contínuo sobre a execução do plano.

As seguintes informações devem estar presentes:

- Quem é responsável por monitorar o plano?
- Com que freqüências serão feitas as atualizações?
- Como as atualizações serão avaliadas e aprovadas?
- Como as alterações serão efetivadas e comunicadas?

O plano deve ser revisto no início de cada fase do projeto. Se as alterações propostas forem aprovadas, então todos os participantes da equipe devem ser informados.

O plano pode ser construído de forma que os detalhes de cada tarefa fiquem em documentos à parte ou em apêndices. Se esta forma for adotada, deve ser descrito como será feita a manutenção nesses outros documentos.

Como parte do projeto, o próprio plano pode ser colocado sob controle do gerenciamento de configuração.

3.4 Recomendações para a elaboração do Plano de Gerenciamento de Configuração de Software

Algumas pessoas acham que devem escrever um plano de gerenciamento de configuração apenas se isso for requisitado pelo cliente. Com isso perdem a oportunidade de estabelecer métodos uniformes para desenvolver sistemas de *software*. O momento da criação do plano de gerenciamento de software é excelente para que o gerente de configuração

vislumbre os problemas, antes que eles afetem os custos ou prazos do projeto [Berlack, 1992].

Escrever um plano de gerenciamento de configuração de software não é uma tarefa difícil, o mais difícil é determinar como executar o gerenciamento de configuração e quais tarefas serão implementadas [Bounds, 1993]. Para que o plano seja bem sucedido, é necessário que haja a colaboração da equipe de desenvolvimento, tanto na elaboração quanto na implementação.

A estrutura do plano deve seguir o padrão adotado pela empresa. Novamente vale salientar que o padrão IEEE é bastante adequado para empresas de desenvolvimento de software de pequeno porte. Se a empresa não faz gerenciamento de configuração e está elaborando um plano pela primeira vez, é recomendado que sejam lidos livros, padrões e, se possível, planos escritos para projetos semelhantes. Convém salientar que, como o plano apresenta características muito peculiares e valiosas da forma de trabalho da organização, é bastante difícil obter um plano de outra empresa[2]. Geralmente, a aquisição dos conhecimentos para elaboração da estrutura do plano deve ser feita em poucos dias. Quando o material mais interessante já estiver separado, um esboço do plano poderá ser escrito. Este esboço conterá apenas uma descrição sobre o que escrever em cada seção do plano.

O passo seguinte, no qual as tarefas de gerenciamento de configuração que serão implementadas são descritas, deve ser feito em conjunto com todos os grupos envolvidos com o gerenciamento de configuração. Serão definidos como fazer o controle de versões, qual será o esquema de identificação, como serão feitas as auditorias e revisões, como será efetuado o relato de situação, como será feito o controle de interface e o

---

[2] Boas, A.V. (CPqD - Telebrás) - Comunicação pessoal. (1996)

controle de subcontratadas e vendedores. Este é, na verdade, o passo mais difícil e mais importante do plano.

Os procedimentos detalhados para cada uma das tarefas são então escritos no plano, e é feita uma análise para verificar se nada foi esquecido. Caso falte algo, deve-se retornar ao passo anterior até que o processo esteja completo. Geralmente, os procedimentos não estão perfeitos em sua primeira descrição, mas, com o uso, pode-se aperfeiçoá-los. Um bom plano não precisa ser alterado freqüentemente e é fácil de entender. Além disso, um bom plano faz com que os participantes do projeto se interessem em segui-lo, e com que a gerência o apoie e custeie.

O plano de gerenciamento de configuração de software geralmente é bastante utilizado pelos desenvolvedores até que se acostumem com as tarefas de gerenciamento de configuração, momento em que o plano deixa de ser consultado com freqüência. Outros usuários do plano podem ser, por exemplo, as equipes de controle de qualidade, para fazer auditorias, e os líderes de projeto, na seleção de pessoal.

Em pequenos projetos, o plano de gerenciamento de configuração de software pode suprimir algumas das tarefas ou passos do gerenciamento de configuração presentes para grandes projetos. Por exemplo, o controle de subcontratados e fornecedores suprimido caso o software seja completamente desenvolvido na própria empresa e outros tópicos podem ser simplificados, como o controle de interface em ambientes bem estáveis. Nos pequenos projetos, o essencial é estabelecer os processos de controle de mudanças de versões, o esquema de identificação dos itens e políticas de liberação de versões. [CMFAQ, 1996]

É interessante notar que uma boa ferramenta de controle de versões já representa um grande benefício para o gerenciamento de configuração de software, por dispensar o controle manual sobre arquivos em diferentes diretórios. Se for possível automatizar também a composição da configuração dos sistemas, a produtividade aumentará bastante [Bounds, 1993; CMFAQ,

1996]. Quanto mais complexo o software em desenvolvimento e mais sofisticada a dinâmica de desenvolvimento, mais elaboradas devem ser as ferramentas de gerenciamento de configuração de software.

REFERÊNCIAS BIBLIOGRÁFICAS

[Ambriola, 1990] AMBRIOLA, V.; BENDIX, L.; CIANCARINI, P. *The Evolution of Configuration Management and Version Control.* Software Engineering Journal, v.5, n.6, p.303-10, 1990.

[Babich, 1986] BABICH, W. *Software Configuration Management.* 1.ed. Reading Mass. Addison-Wesley, 1986.

[Bersoff, 1979] BERSOFF, E.H.; HENDERSON, V.D.; SIEGEL, S.G. *Software Configuration Management: A Tutorial.* IEEE Computer, v.12, n.1, p.6-14, 1979.

[Bersoff, 1980] BERSOFF,E.H.; HENDERSON, V.D.; SIEGEL, S.G. *Software Configuration Management - An Investiment in Product Integrity*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[Bersoff, 1984] BERSOFF,E.H. *Elements of Software Configuration Management.* IEEE Transactions on Software Engineering, v.se-1.0, n.1, p.79-87, 1984.

[Berlack, 1992] BERLACK, H.R. *Software Configuration Management.* 1.Ed. New York, John Wiley, 1992.

[Boas, 1994] BOAS, A.V. *RCS-Revision Control System.* Departamento de Sistemas de Operação-Projeto ASSISTE CPqD-TELEBRÁS, 1994.

[Bounds,1993] BOUNDS, NADINE M.; DART, S. A. *Configuration Management (CM) Plans: The Begining to Your CM Solution. Pittsburg,* Software Engeneering Institute, Carnegie Mellon University, 1992 (Tech. Report CMU/SEI-92-TR ?).

[Brown, 1991] BROWN, H. *Like a Version.* Computer Languages, v.8, n.8, p.59-81, 1991.

[Buckley, 1993] BUCKLEY, F. J. *Implementing a Software Configuration Management Environment* IEEE Computer, v.27, n.2, p.56-61, 1994.

[Capretz, 1992] CAPRETZ M.A.M.. *CONFORM - A Software Maintenance Method Based on Software Configuration Management Discipline.* In: Conference on Software Maintenance, Orlando. Procedings. p.183-92, 1990.

[Capretz, 1994] CAPRETZ M.A.M.; MUNRO M. *Software Configuration Management Issues in the Manitenance of Existing Systems.* Software Maintenance: Research and Practice, v.6,p.1-14, 1994.

[CMFAQ, 1996] [EATON, D.W. - *Configuration Management Frequently Askey Quastions* - http//www.iac.honeywell.com/Pub/Tech/CM/CMFAQ.html, 1996 [Acesso em 11/09/96]

[Dart, 1991] DART, S. A. *Concepts in Configuration Management Systems.* In: International Workshop on Software Configuration Management, 3, Trondheim, 1991. Proceedings. p.1-18

[DoD-STD-2167A] DoD-STD-2167A, Military Standard for Defense System Software Development, Department of Defense. Jun, 1985.

[DI-MCCR-80030A] DoD DI-MCCR-80030A, DataItem Description for the Software Development Plan, Department of Defense, Jun, 1986

[Feiler, 1991] FEILER, P. H. *Configuration Management Models in Commercial Environments*. Pittsburgh, Software Engineering Institute, Carnegie Mellon University, 1991. (Tech.report CMU/SEI-91-TR-7)

[Harter, 1989] HARTER,R. *Configuration Management*. HP Professional, vol.3, n.6, jun. 1989.

[Humphey, 1988] HUMPHREY, W. S. *Characterizing the Software Process - A Maturity Framework*. IEEE Software, v.5, n.2, p.73-79, 1988.

[Humphrey, 1989] HUMPHREY, W. S. *Managing the Software Process*. 1.ed. Massachesetts. Addison-Wesley, 1989.

[ANSI/IEEE 983-1986] *Software Engineering Standards*, 3ª edição, IEEE, 1989.

[IEEE Std 610.12-1990] *IEEE Standard Glossary of Terminology in Software Engineering*. In:IEEE Software Engineering Standar Collection, p.07-83, 1991.

[IEEE Std 828-1990] *IEEE/ANSI Standard 828-1990*. Software Configuration Management Plans. In: IEEE Software Engineering Standard Collection, 1990.

IEEE Std 1042-1986] *IEEE/ANSI Standard 1042-1987: IEEE Guide to Software Configuration Management*. In: IEEE Software Engineering Standard Collection, 1987.

[Karakas,1992] KARAKAS, G. ; KAZAI, G.; KUNTH, E. *Project Management by Configuration Inheritance*. Annual Review of Automatic Programming, v.16, p.109-16, 1992.

[MIL-STD-483A-1979] DoD MIL-STD-483-A, Configuration Management Practices for Systems, Equipment, Muintions, and Computer Programs, 1979.

[Narayanaswamy, 1987] NARAYANASWAMY,K.; SCACCHI,W. *Maintening Configurations of Evolving Software*. IEEE Transactions on Software Engineering, v.se-13, n.3, p.324-34, 1987.

[NASA-Sfw-DID-04] NASA Sfs DID 04, *Software Configuration Management Plan Data Item Description*, National Aeronautics and Space Administration, Version 3.0. Out, 1986.

[Newman, 1995] NEWMAN, W.E.; *Software Configuration Management Process Definition*. Rear Adimiral, U.S. Navy, 1995.

[Ould, 1990] OULD, M.A. *Managing Software Engineering*. The Management of Risk and Quality, John Wiley, Chinchester, 1990.

[Pressman, 1995] PRESSMAN, R.S. *Engenharia de Software*. 3.Ed Rio de Janeiro. Makron Books, 1995.

[Paulk, 1993a] PAULK M. C. *et al. Capability Maturity Model for Software*. versão 1.1, Pittsburgh, Software Engineering Institute, Carnegie Melon University, 1993. (CMU/SEI-93-TR-24)

[Paulk, 1993b] PAULK M. C. *et al. Key Practices of the Capability Maturity Model*. versão 1.1. Pittsburgh, Software Engineering Institute, Carnegie Mellon University, 1993. (CMU/SEI-93-TR-25)

[Saiedian, 1995] SAIEDIAN, H. and KUZARA, R. *SEI Capability Maturity Model's Impact on Contractors*. IEEE Computer, v.28, n.1, p.16-26, 1995.

[Sanches, 1993] SANCHES, R. *A Influência do Software e de seu Processo de Manutenção no Esforço de Manutenção*. Tese (Doutorado), Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, São Paulo, 1993.

[Schneidwind, 1987] SCHNEIDWIND N.F. *The State of Software Maintenance*. IEEE Transaction on Software Engineering, v.se-13, n.3, p.303-310, 1987.

[Whitgifh, 1991] WHITGIFH, DAVID *Methods and Tools for Software Configuration Management*. John Wiley & Sons, 1991.