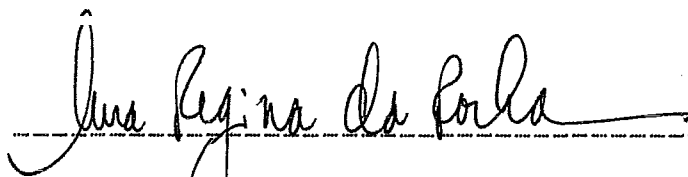


UMA FERRAMENTA AUTOMATIZADA PARA  
GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE

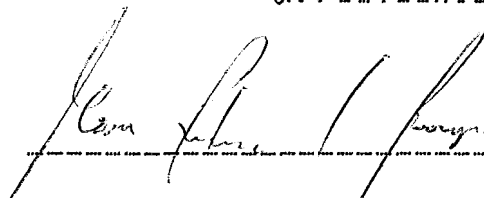
Leonardo Falcão Koblitz

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS  
PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DA UNIVERSIDADE  
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS  
EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

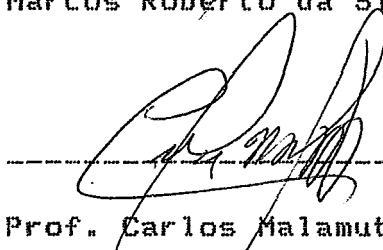
Aprovada por:



Prof. Ana Regina C. da Rocha, D.Sc.  
(Presidente)



Prof. Marcos Roberto da Silva Borges, Ph.D.



Prof. Carlos Malamut, D.Sc.

RIO DE JANEIRO, RJ - BRASIL.

ABRIL DE 1991

**KOBLITZ, LEONARDO FALCÃO**

**UMA FERRAMENTA AUTOMATIZADA PARA GERENCIAMENTO DA  
CONFIGURAÇÃO DE SOFTWARE CRIO DE JANEIRO] 1991**

**X, 160 p. 29,7 cm (COPPE/UFRJ, M. Sc., Engenharia de  
Sistemas e Computação, 1991)**

**Tese - Universidade Federal da Rio de Janeiro, COPPE.**

**i. Ferramentas Automatizadas ■ - COPPE/UFRJ I ■ - TÍTULO  
(Série).**

Ao meu pai por ter conquistado  
a grande mulher que é minha  
mãe, Maria Laura.

## AGRADECIMENTOS

A Ana Regina, minha orientadora, pela dedicação e paciência,

A minha família, que esteve sempre ao meu lado.

Aos meus tios Armando e José.

Aos amigos fraternos :

Jailton da Gosta Ferreira, pela oportunidade e exemplo que me deu.

Eustério Benitz pelos conselhos e amizade que sempre demonstrou,

Eduardo Valdetaro, pela ajuda na implementação e pelo seu senso profissional.

Cláudio Franco de Abreu, Tereza Aguiar e Guilherme Travassos, pelas boas sugestões a este trabalho,

Cacilda Jorge minha amiga das horas de aperta,

Aos colegas de trabalho Becker, Cristina, Dilmar, Joana, Miriam e Sergio, que foram responsáveis indiretos pela conclusão deste trabalho, mesmo sem se darem conta disto.

Aos colegas da CNEN, em especial ao Luis Arrieta.

Aos membros da banca Carlos Malamut e Marcos Borges.



**Resumo** da Tese apresentada à COPPE/UFRJ como parte dos requisitos **necessários** para a **obtenção** do grau de **Mestre em Ciências (M. Sc.)**

UMA FERRAMENTA AUTOMATIZADA PARA **GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE**

**Leonardo Falcão Koblitz**

Abril de 1991

Orientador : **Profa.** Ana Regina Cavalcanti da Rocha

Programa : Engenharia de **Sistemas e Computação**

Este trabalho apresenta o **SAGCS**, uma ferramenta **automatizada** para gerenciamento da **configuração de software** para **ambientes DOS**. **Além de controlar o acessos aos arquivos sob seu controle, fornece vários relatórios, que permitem o usuário acompanhar precisamente a evolução do sistema.**

Em conjunto com esta ferramenta, são sugeridos um modelo e procedimentos para elaboração e implantação de um plano de gerenciamento da configuração de software que melhor atenda as características específicas de cada projeto de software.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.)

UMA FERRAMENTA AUTOMATIZADA PARA GERENCIAMENTO DA  
CONFIGURAÇÃO DE SOFTWARE

Leonardo Falcão Koblitz

April of 1991

Thesis Supervisor: Ana Regina Cavalcanti da Rocha

Department : System Engineering and Computer

This work presents the system SAGCS which was designed as an automated tool for software configuration management for DOS environment. Besides controlling the access to the files under its management, SAGCS provides several reports which allow the user to follow very closely the evolution of the system.

This work additionally suggests a model and its procedures for designing and implementing a software configuration management plan which fits best the specific characteristics and requirements of each software project.

**Í N D I C E**

**SUMÁRIO**

**página**

**ABSTRACT**

**■ - INTRODUÇÃO**

**11. GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE**

**2.1 - INTRODUÇÃO**

**2.2 - COMO ESTABELECE O GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE** 12

**2.3 - FUNÇÕES DO GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE** 13

**2.3.1 - IDENTIFICAÇÃO** 15

**2.3.2 - CONTROLE** 21

**2.3.3 - AUDITORIA** 28

**2.3.4 - CONTABILIDADE DA SITUAÇÃO** 31

**2.4 - CUSTOS DO GERENCIAMENTO DA CONFIGURAÇÃO** 35

**2.5 - FERRAMENTAS AUTOMATIZADAS** 40

**2.5.ã - EXEMPLOS DE FERRAMENTAS** 41

**2.6 - NOTAS FINAIS** 45

**III. PLANO DE GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE**

**3.1 - INTRODUÇÃO** 47

**3.2 - DESCRIÇÃO DE UM PLANO DE GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE** 48

**3.3 - NOTAS FINAIS** 182

<b>IV. SISTEMA AUTOMATIZADO PARA GERENCIAHENTO DA CONFIGURAÇÃO DE SOFTWARE</b>	
4.1 - INTRODUÇÃO	103
4.2 - LIMITAÇÕES DO SISTEMA ATUAL	184
4.3 - DESCRIÇÃO GERAL	105
4.3.1 - INTERFACE COM OUTROS PRODUTOS E PROJETOS	112
4.3.2 - REBULSITOS DE INFORMAÇÃO	113
4.3.3 - REQUISITOS FUNCIONAIS	116
4.3.3.1 - DESCRIÇÃO DOS PROCESSOS	116
4.4 - ESPECIFICAÇÃO E PROJETO DO PROTÓTIPO DO SAGCS	124
4.5 - IMPLEMENTAÇÃO	128
<b>V. CONCLUSÃO</b>	<b>141</b>
<b>VI. HANUAL DO USUÁRIO</b>	<b>145</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>157</b>

## CAPÍTULO I

### INTRODUÇÃO

Manter a correspondência entre as formas não executável e executável do software e a consistência entre as várias representações da forma não executável constitui um grande desafio no desenvolvimento e manutenção de um sistema.

Tal correspondência não é nada mais do que a garantia que as duas formas do software e suas representações expressam exatamente o software pretendido, desde a sua especificação até a sua forma executável (BERSOFF, 1980).

A ausência de um controle eficaz sobre o software ocasiona inúmeros problemas, tais como: a realização de alterações conflitantes ou indevidas, dificuldade de se rastrear a evolução de um software ou mesmo de se levantar precisamente a configuração corrente.

Este problema cresce ainda mais quando um software é distribuído por várias instalações em uma empresa. Muitas vezes, cada instalação está trabalhando com uma versão (configuração) diferente do software, sem se dar conta disso. O resultado é o caos, os gerentes de cada

uma das instalações não se entendem, os relatórios divergem entre si e muito esforço é desperdiçado na solução do problema.

Apesar de já existirem no mercado algumas ferramentas para o gerenciamento da configuração de software (MOTTA, 1990), estas são pouco utilizadas em virtude de dois fatores:

O primeiro envolve o fato destas ferramentas em sua maioria estarem disponíveis em ambientes de grande porte e serem muito caras. Já o segundo fator, está relacionado ao pouco desconhecimento ou falta de conscientização por parte do responsável pelo software, da importância de se gerenciar a configuração de um software para manter a sua consistência e rastreabilidade.

O Sistema Automatizado para Gerenciamento da Configuração de Software (SAGCS) proposto neste trabalho de tese, tem como objetivo principal, suprir o usuário de microcomputador com um ambiente barato e de fácil utilização para o acompanhamento da evolução de um sistema em desenvolvimento.

As várias versões (configurações) que compõem o sistema são identificadas através de um rótulo de nome simbólico e qualquer informação sobre **uma** configuração ou os itens que a compõem estão disponíveis aos usuários.



Vários tipos de relatórios são fornecidos pelo SAGCS, respondendo perguntas tais como : quando um arquivo foi registrado, quem modificou um arquivo, porque o arquivo ou que configuração foram modificados, entre outras informações.

Esta tese está dividida em cinco capítulos:

**Capítulo I** - Apresenta uma introdução ao trabalho e levanta alguns pontos intimamente ligados a aplicação do gerenciamento da configuração de software.

**Capítulo II** - Contém uma revisão bibliográfica sobre o gerenciamento da configuração de software.

**Capítulo III** - Propõe um modelo de Plano de Gerenciamento da Configuração de Software (PGCS) e apresenta os pontos críticos a serem considerados na sua confecção.

**Capítulo IV** - Descreve o Sistema Automatizado para Gerenciamento da Configuração de Software (SAGCS).

**Capítulo V** - Apresenta as conclusões deste trabalho apontando para alguns caminhos a serem seguidos.

**Apêndice A** - Apresenta o Manual de utilização do SAGCS.

## CAPÍTULO II

### GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE

#### 2.1 INTRODUÇÃO

Grande parte dos problemas no desenvolvimento de um sistema são devidos às deficiências na definição dos requisitos e ao controle precário da documentação e suas alterações.

É muito comum que a configuração exata de um sistema só seja conhecida por uma única pessoa. Como a configuração do software é uma história sem fim de alterações (DUNN, 1982), o problema aumenta ainda mais, quando este é distribuído por várias instalações. A alteração em uma das instalações do software sem a sua comunicação e consentimento das partes interessadas, acarreta inúmeras dificuldades.

Assim, o desconhecimento da configuração corrente, das alterações conflitantes e falhas de comunicação são comuns e produzem gastos excessivos de recursos (DUNN, 1990).

O Gerenciamento da Configuração (GC) tem sido utilizado com bastante sucesso por muito tempo na indústria. No entanto, pela visão do GC tradicional o software era apenas um único item entre tantos outros que

compunham o sistema (figura II.1). Assim sendo as partes que compunham o software não eram identificadas e controladas individualmente.

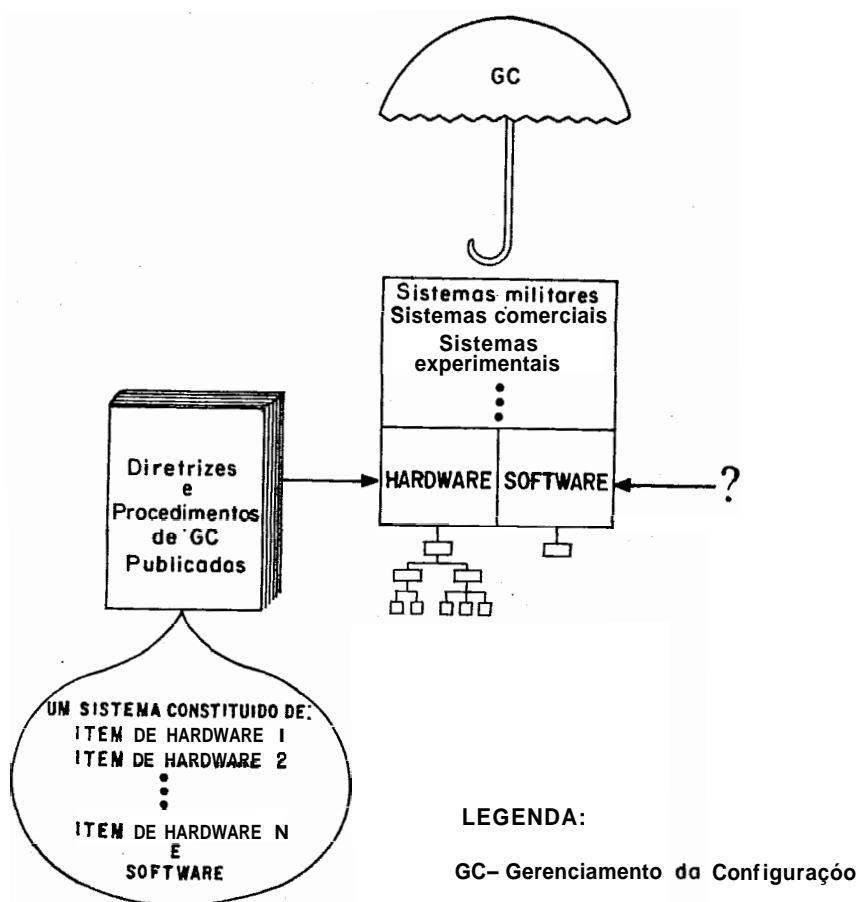


Figura II.1 - O software visto pelo gerenciamento da configuração tradicional

Fonte - BERSOFF (1984)

Esta visão é antiquada, visto que, atualmente, ao contrário de anos atrás, o software desempenha cada vez mais as funções executadas por um sistema.

Segundo BERSOFF et al (1980):

"O gerenciamento da configuração é a disciplina de identificar a configuração de um sistema em pontos discretos no tempo, com o propósito de sistematicamente controlar as alterações na configuração e manter a integridade e a rastreabilidade da configuração por todo o ciclo de vida do sistema".

O Gerenciamento da Configuração de Software (GCS) é então, apenas o gerenciamento da configuração para sistemas que possuam funções que são desempenhadas predominantemente pelo software.

O Gerenciamento da Configuração de Software (GCS) é uma das disciplinas da Engenharia de Software que é melhor entendida (TILBURG, 1985). Apesar disto, os problemas de comunicação entre a antiga comunidade de gerenciamento da configuração e a comunidade técnica de software são decorrentes das visões diferentes do processo de desenvolvimento do software e da utilização de mesmos termos com significados diferentes. Alguns termos como Gerenciamento da Configuração (GC), "baseline", Contabilidade da Situação apresentam consonância de opiniões, ao contrário dos termos versão e revisão por exemplo (ALDERSEN, 1985).

TILBURG (1985) chama de macro o nível de controle do Gerenciamento da Configuração de Software o qual corresponde às atividades do Gerenciamento da Configuração

tradicional. Este nível compreende a identificação das especificações e documentação, a administração das alterações feitas nas representações do projeto e no protótipo.

Já micro Gerenciamento da Configuração corresponde ao nível de controle do Gerenciamento da Configuração de Software que coordena as alterações detalhadas que ocorrem no processo de desenvolvimento do projeto (alterações evolucionárias). Estas alterações são essencialmente internas às atividades técnicas.

Como os níveis macro e micro possuem características diferentes, geralmente as pessoas responsáveis pelas decisões de controle não são qualificadas para atuarem nos dois níveis.

O nível micro é responsável por um volume de documentação muito superior ao volume do nível macro. A razão disto é que o nível micro fornece uma visão muito mais profunda do processo de engenharia.

É importante notar que as alterações interativas feitas durante o processo de desenvolvimento são confundidas com erros. Sendo assim as comparações de estatísticas de erro entre o software (macro e micro) e o hardware são pelo menos perigosas.

Concluindo, POPE (1983) observa que o Gerenciamento da Configuração de Software é o veículo para documentar e controlar o ciclo de desenvolvimento de um sistema e

apresenta as seguintes vantagens:

- i) identificação precisa da configuração corrente da definição de um item, com a rastreabilidade de configurações anteriores;
- ii) a habilidade para reproduzir ambientes definidos ("baselines") para permitir diagnósticos e correção de problemas;
- iii) uma estrutura formal para avaliar os impactos oriundos de alterações propostas;
- iv) os procedimentos para notificar oficialmente os documentos formalmente estabelecidos, com suas configurações e alterações aprovadas e liberadas.

Um conceito fundamental em Gerenciamento da Configuração é o de "Baseline", que no jargão de Gerenciamento da Configuração, é um ponto de referência, ou um plateau no desenvolvimento de um sistema. Uma "baseline" é definida no final de cada estágio do ciclo de vida do sistema (BERSOFF, 1980).

A "baseline", isto é, uma base comum, serve como uma base para a negociação entre as partes interessadas no produto do software.

Durante o plano de desenvolvimento de um sistema, é

definida uma sucessão de "baselines" que coincidirão com os marcos estabelecidos no ciclo de vida. Estas "baselines" são refinadas até o marco do ciclo de vida, e são chamadas de "baseline"/atualização para-ser-estabelecida. Quando o marco é alcançado, é dito que a "baseline" foi estabelecida.

"Baseline"/atualização para-ser-estabelecida é o conjunto de documentos, materiais ou equipamentos em um ponto do ciclo de vida que ainda não foi auditado ou está sendo auditado, ou foi reprovado ou está esperando a decisão de uma autoridade para ser formalmente estabelecida.

BOEHM (1981), observa que o processo marco-"baseline" geralmente trabalha da seguinte maneira:

- i) uma versão de um produto intermediário ou final é desenvolvida;
- ii) esta versão inicial de um produto é verificada e validada;
- iii) uma revisão formal do produto determina se o produto está ou não em condições satisfatórias para se proceder a próxima fase (isto é, se o marco foi alcançado ou não). Caso não tenha sido, o processo retorna ao primeiro passo;
- iv) se o produto é satisfatório, ele é dito "baselined", ou seja, posto sob processo formal de controle de alteração.

Para assegurar que a integridade do produto foi observada a cada ponto, uma sêrie de revisões são feitas para o estabelecimento de cada "baseline".

As "baselines" e suas atualizações são um mecanismo fundamental no gerenciamento da configuração.

Algumas das vantagens de se estabelecer "baselines" em um produto são:

- i) alterações só serão feitas quando houver um acordo entre as partes interessadas;
- ii) qualquer versão definitiva do produto está disponível a qualquer hora.

O número de "baselines" pode variar dependendo do tamanho do projeto. Geralmente existem as "baselines" funcional, alocada, de projeto, de produto e operacional.

Estes nomes foram mantidos devido ao seu longo uso no gerenciamento da configuração tradicional. Tipicamente cada uma destas "baselines" apresentam (WILBURN, 1983) (BERSOFF, 1980):

- i) "baseline funcional" é um produto do estágio de definição dos requisitos do sistema. Nela, tanto o usuário quanto o desenvolvedor formalizam o seu entendimento e concordância sobre o que exatamente o sistema irá fazer;



- ii) "baseline alocada" define precisamente quais as funções serão manuais e desempenhadas pelo hardware ou software. Dependendo do projeto pode resultar dois documentos: uma especificação para hardware e outra para software;
- iii) "baseline de projeto" é um produto do estágio de projeto. Esta "baseline" inclui entre outras coisas, diagramas estruturados, pseudocódigos, especificações, etc. Só após a sua aprovação é que o sistema será finalmente construído;
- iv) "baseline de produto" é o conjunto de toda a documentação e listagens de computador descrevendo o primeiro produto de software;
- v) "baseline operacional" é o produto que é finalmente implementado e inclui toda a documentação associada ao projeto. Nesta "baseline" o produto de software já foi melhorado e otimizado.

Cada "baseline" e suas atualizações, coletivamente representam a evolução do software durante cada um dos seus estágios do ciclo de vida (BOEHM, 1981).

Todavia, para um projeto pequeno ou que apresente uma baixa complexidade, o mecanismo de se migrar de uma "baseline" para outra (por exemplo, da funcional para a alocada) pode ser muito formal, bastando apenas o estabelecimento de uma única "baseline" e suas

atualizações.

Concluindo, a formalização do mecanismo de rotulação e atualização é a manifestação da disciplina chamada Gerenciamento da Configuração de Software.

## 2.2 COMO ESTABELECEER O GERENCIAMENTO DA CONFIGURAÇÃO

Para se estabelecer um sistema de Gerenciamento da Configuração é necessário uma análise detalhada dos requisitos de informação provenientes dos usuários. O objetivo desta análise é eliminar ou diminuir o grande volume de dados inúteis ao Gerenciamento da Configuração.

Informação e dados diferenciam-se pela razão dos dados serem um conjunto de fatos não estruturados. Estes dados só têm valor quando puderem ser organizados em pacotes de informação. É necessário então, minimizar o volume de dados não usados e não estruturados (EVANS, 1986).

Devido ao grande volume de dados utilizados no Gerenciamento da Configuração, um sistema de Gerenciamento da Configuração deve ter em seu núcleo um banco de dados.

Após a análise dos dados, é necessário a implementação de uma equipe de Gerenciamento da Configuração e de um Plano de Gerenciamento da Configuração de Software (PGCS) o qual estabelecerá as diretrizes, procedimentos e ferramentas a serem utilizadas.

Este plano é a ponte entre a teoria e a prática do Gerenciamento da Configuração. Geralmente é produzido um plano para cada projeto.

No caso de vários projetos sendo desenvolvidos ao mesmo tempo, pode ser útil emitir um único plano padrão de Gerenciamento da Configuração para todos. Neste caso, só estarão esboçados no plano os procedimentos básicos e enfatizados os aspectos mais importantes.

O Plano de Gerenciamento da Configuração de Software (PGCS) não deve ser visto como estático e rígido e dependendo do tempo de vida do sistema, o próprio plano estará sujeito a uma evolução.

O PGCS é o elemento chave do controle da configuração e contém um conjunto de procedimentos, decisões e diretrizes que irão reger todo o desenvolvimento do projeto (BRYON, 1980) (KHAN, 1986).

### **2.3 - FUNÇÕES DO GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE**

O gerenciamento da configuração de software apresenta quatro funções separadas (não independentes):

- i) identificação da configuração do software, cujo papel é fornecer rótulos para que as partes que compõem o sistema possam ser visualizadas;
- ii) controle da configuração do software, cujo papel está em gerenciar as alterações nas partes

identificadas que compõem o sistema;

iii) auditoria da configuração do software, que fornece o mecanismo para se verificar o estado corrente do sistema com o estipulado formalmente;

iv) contabilidade da situação da configuração do software, que fornece o mecanismo para acompanhar e registrar a evolução do sistema.

Outra dúvida frequente é quanto à utilização do Gerenciamento da Configuração. O Gerenciamento da Configuração deve ser aplicado em qualquer tipo de sistema, não importando se é uma simples tarefa a ser executada em uma máquina de calcular ou um sistema de tráfego aéreo.

O Gerenciamento da Configuração deve ser sempre aplicado, variando apenas o seu grau de utilização. A sua utilização não é restrita a algum estágio do ciclo de vida do sistema. Ao contrário, pode e deve ser utilizado em qualquer estágio do ciclo de vida do sistema.

Entretanto, o seu uso será mais eficaz será quando for implementado no início do projeto.

O Gerenciamento da Configuração de Software é um veículo importante de comunicação e concordância entre as várias partes envolvidas no desenvolvimento de um sistema. O controle e a documentação de todo o ciclo de desenvolvimento de um sistema é alcançado através de

conceitos, decisões, procedimentos e utilização de ferramentas que são especificadas no Plano de Gerenciamento da Configuração de Software.

É importante ter sempre em mente que o software é parte de uma entidade chamada sistema. A interação entre o software e o hardware sempre deve acompanhar-nos quando do desenvolvimento de um sistema.

BERSOFF et al (1980) observam:

"O conceito de Gerenciamento da Configuração de Software não pode ser separado das considerações de sistema e hardware. Não se deve perder a visão desta perspectiva quando proceder nos aspectos do GCS."

### 2.3.1. IDENTIFICAÇÃO

É preciso impor disciplina no desenvolvimento do software para que se possa controlar os resultados do ciclo de vida do sistema. A identificação das diversas partes que compõem o software é crucial para se impor esta disciplina.

Por identificação, entende-se a determinação das partes constituintes do software, a determinação do relacionamento destas partes, a designação de um rótulo e um nome para cada parte e a apresentação tabular ou gráfica do software identificado (WILBURN, 1983).

A identificação da configuração refere-se à descrição completa das características físicas e funcionais de um produto, por exemplo, o número de linhas de código, os processos desempenhados, requisitos de armazenamento e tudo o que for necessário para construir, testar, operar e reparar o software (EVANS, 1986).

O propósito básico da identificação da configuração é facilitar o particionamento das "baselines" em um número progressivamente maior de partes, durante os estágios do ciclo de vida (BERSOFF, 1980).

A motivação para o particionamento do software, é que frequentemente algumas partes do software se estabilizam durante o desenvolvimento do sistema, possibilitando o direcionamento da atenção do gerente do projeto para as partes mais instáveis.

Estas partes são chamadas de Itens de Configuração de Software (ICS), mais comumente conhecidos como Itens de Configuração de Programa de Computador (ICPC), podem ainda ser divididos em Componentes de Programas de Computador (CPC) (por exemplo, submódulos) sempre que houver necessidade de se dividir por conveniência uma parte complexa de um programa em partes mais fáceis de se trabalhar (BRYON, 1981) (VINCENT, 1988).

BERSOFF (1980) contudo não usa a nomenclatura dos CPC, apenas utiliza ICS/ICPC de níveis mais baixos para o controle da complexidade de um problema.

Estes itens podem ser:

- i) documentos de requisitos;
- ii) documentos de projeto;
- iii) programas fonte;
- iv) código compilado, linkeditado ou executável;
- v) documentos do usuário e instruções para início e recuperação depois de uma falha;
- vi) comandos usados na compilação, linkedição e execução para cada versão de programas executáveis, incluindo versões do compilador, etc.;
- vii) pedidos de alterações (se aprovada ou não), com a história das autorizações e ações tomadas;
- viii) ambiente de execução (hardware e software) requerido para cada programa executável;
- ix) conjuntos de dados de teste e resultados esperados (para todas as versões);
- x) registro dos resultados dos testes (para todas as versões).

Uma dúvida comum se refere à identificação de um Item de Configuração (IC). Na realidade não existe uma regra. Geralmente os ICs não devem ultrapassar a fronteira entre computadores ou programas independentes. Cada IC deve englobar **todas** as operações funcionalmente relacionadas dentro de um **único** computador (EVANS, 1986).

Outro elemento importante é o Objeto de Projeto (OP). Objetos de Projeto são construções que ainda não estão sob a esfera do GC/GCS. São peças do software que ainda estão sendo desenvolvidas, modeladas ou formadas. O conjunto de ICSs com os objetos de Projeto formam a arquitetura do sistema. Os Objetos de Projeto ocupam os níveis mais baixos em uma arquitetura "Top-down".

Com a evolução do sistema, o número de ICSs aumenta e de Objetos de Projeto diminui. É dito que houve uma transição dos Objetos de Projeto para Itens de Configuração de Software, os quais estão sob a responsabilidade do GC.

A configuração do software é então o arranjo relativo dessas partes, que estão relacionadas através de um caminho bem definido. Para um gerente da configuração do software o sistema se parece como um conjunto de ICS e um ou mais componentes do hardware.

A figura (II.2) mostra o processo de conversão (congelamento) de um Objeto de Projeto em um Item de Configuração de Software.

O arranjo desses itens relacionados, descritos a respeito de um estágio particular do ciclo de vida de um sistema, é o que o gerente da configuração etiqueta de "baseline" ou atualização.



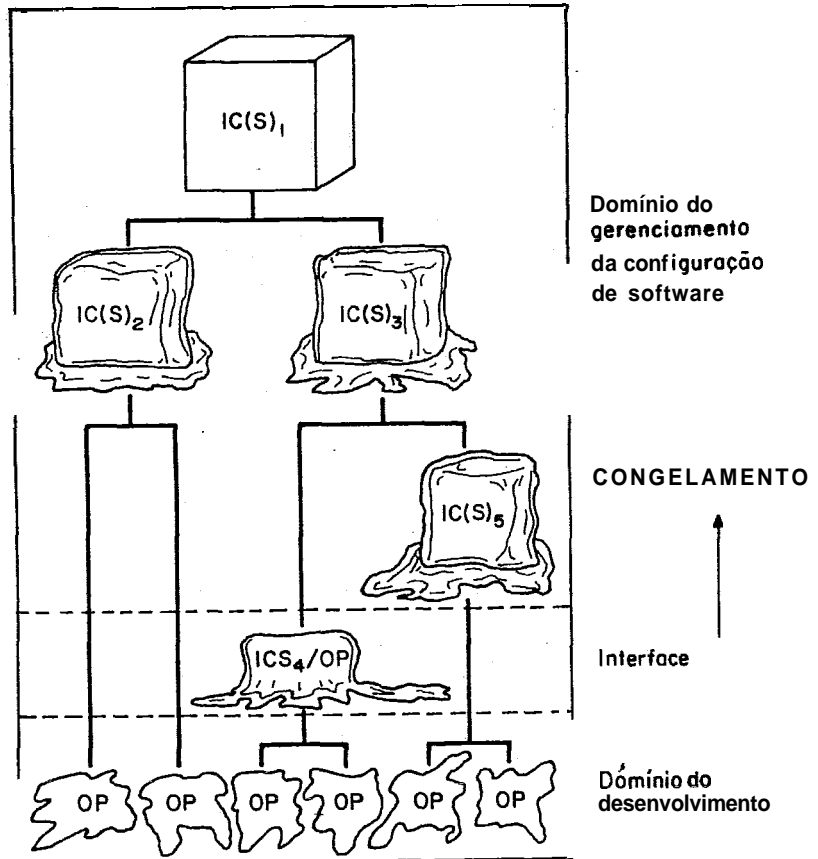
Todos os Itens de Configuração de Software exceto a especificação dos requisitos do sistema geralmente dependem de algum item ou itens já estabelecidos na configuração do software. Estes itens estão conectados em rede a qual indica as dependências entre diferentes itens. Geralmente esta rede é uma árvore (POPE, 1983).

Com a travessia pelos vários estágios do ciclo de vida, várias representações do software vão surgindo.

A identificação destas representações é crucial para se manter a consistência entre elas. Algumas representações manifestam-se explicitamente (especificações, por exemplo) enquanto outras representações manifestam-se implicitamente (códigos fonte armazenados em discos magnéticos) (BERSOFF, 1980).

Neste caso, a replicação de um mesmo rótulo para todas as representações é utilizada e é adicionado mais um rótulo especificando o tipo de representação.

Para se evitar um grande número de "baselines", o gerenciamento da configuração de software usa o conceito de versão. Versão é uma pequena perturbação na configuração do sistema. É mais conveniente se pensar que duas configurações são versões de uma mesma "baseline" em vez de que estas duas versões representam dois sistemas diferentes, cada um com suas "baselines". As versões diferenciam-se em pelo menos um ICS nos níveis mais baixos das estruturas de ICS (geralmente uma árvore).



**LEGENDA:**

ICS - ítem de configuração de software,  
( Sob o gerenciamento da configuração ).

OP - Objeto de projeto,  
( Sob o desenvolvimento ).

ICS<sub>i</sub>/OP - Sendo liberado para o gerenciamento da configuração .

( ) - Ainda não definido se o item é hardware ou software .

Figura II.2 - Interface entre o desenvolvimento e o gerenciamento da configuração de software..

Fonte - BERSOFF (1984)

Outra observação importante, é que um elemento que foi identificado não desaparece, contudo nos níveis mais baixos da estrutura de ICS pode ocorrer uma fundição entre dois

ICS do mesmo nível. Neste caso o ICS é rotulado como retirado.

Concluindo, a meta da identificação do gerenciamento da configuração é fornecer uma identificação para cada versão de cada ICS e ainda manter informação a respeito dos interrelacionamentos dos ICS (WILBURN, 1983).

### 2.3.2 CONTROLE

Controle é o veículo de comunicação ordenada das alterações realizadas sobre pontos já estabelecidos. A palavra comunicação deve ser enfatizada, pois ela é fundamental no processo de controle. Todos os afetados e envolvidos por uma alteração devem ser comunicados para que se obtenha uma concordância ou não da proposta de alteração.

Este controle não deve ser visto como uma ação polícialasca, ao contrário, garante que o desenvolvimento não se desviará de suas metas e que o usuário receberá um produto que satisfaça ou mesmo supere as suas expectativas (BERLACK, 1981).

Como se controlar o que não se vê? Esta pergunta é respondida pelo controle da configuração que fornece um quadro de referência comum e independente, o qual é utilizado por todos, fornecendo, assim, a obtenção e a manutenção da visibilidade.

Os produtos liberados durante os estágios do ciclo de vida representam as diversas configurações do sistema, cada uma sendo criada para transmitir um ou mais atributos do sistema. O processo de controle da configuração tem que assegurar o relacionamento harmonioso entre as múltiplas configurações em face das alterações contínuas.

Pode-se concluir que o controle das configurações pode ser obtido pelo controle dos produtos liberados durante os estágios do ciclo de vida do sistema (BRYAN, 1981).

Formalmente o controle de configuração de software pode ser definido como (BERSOFF, 1980):

"O controle da configuração de software é a harmonização dos processos pela qual a porção de software de um sistema pode obter e manter a visibilidade por toda a sua jornada através do ciclo de vida. Ela fornece as ferramentas (procedimentos, documentação e um corpo organizacional) para controlar tanto a implementação do sistema quanto qualquer alteração sobre ele."

O processo para controlar a implementação e a alteração de um sistema inclui os seguintes elementos (BERLACK, 1981):

- i) um método para identificar o que está para ser implementado ou modificado;

- ii) um método para avaliação dos recursos necessários para gerar um novo produto ou os impactos provenientes de uma alteração;
- iii) um recurso capaz de criar o produto ou implementar a alteração necessária;
- iv) um recurso para garantir a identificação da configuração do sistema contida em um produto novo ou modificado, verificando se o desenvolvimento de produtos e as alterações seguiram as práticas, decisões e procedimentos estabelecidos, podendo aprová-las ou não;
- v) um fórum para rever e aprovar os novos produtos e alterações propostas, estabelecer prioridades, cronogramas e situações correntes.

O processo de controle da configuração deve começar pelo menos no início do desenvolvimento do sistema. Tanto as organizações de usuários, compradores (contratante) ou contratado (desenvolvedores) devem interagir para o estabelecimento de diretrizes a serem seguidas. O consenso entre as três partes é um elemento integral do controle da configuração de software (BERSOFF, 1980).

As ferramentas utilizadas para o controle das alterações são (BERSOFF, 1980):

- i) ferramenta de documentação: Esta ferramenta se compõe principalmente das documentações das "baselines" e dos formulários de alteração. Estabelece um quadro de referência comum, para possibilitar uma revisão concorrente de toda documentação relativa ao projeto por todos os participantes;
- ii) procedimentos: É o conjunto de passos que têm que ser seguidos na execução de uma modificação;
- iii) conselho de controle da configuração: Constitui o fórum para controlar o desenvolvimento do sistema.

O Conselho de Controle (de alteração) da Configuração (CCC) ou Conselho de alteração é o ponto focal para o controle da configuração. Pode ser visto como uma Suprema Corte e é constituído para resolver as questões que não puderam ser resolvidas em um nível mais baixo.

É um corpo votante, apesar do presidente do conselho ter a última palavra. O Conselho de Controle da Configuração deve ser constituído e interagir tanto nas organizações de desenvolvimento (contratado) quanto nas de compra (contratante). Deve ser constituído com representantes da gerência, das organizações de usuário e desenvolvimento e representantes do GCS. Sempre que houver necessidade o Conselho de Controle da Configuração pode solicitar representantes de outra organização funcional,

que atuarão como consultores (TILBURG, 1985). O Conselho de Controle da Configuração deve ter participação por todo o sistema, não ficando restrito ao software.

BERSOFF et al (1980), afirmam:

"O Conselho de Controle da Configuração (CCC), como ponto focal do controle da configuração, deve ser organizado para ser responsável pelos requisitos do projeto. Na ausência da consciência deste princípio por parte do comprador e/ou usuário, o vendedor deve preencher esta lacuna com um conjunto de CCC internos. Na ausência da consciência por parte do vendedor, o comprador deve impor o estabelecimento de um CCC integrado."

O CCC tem duas responsabilidades fundamentais (BERSOFF, 1980):

- i) aprovar, monitorar e controlar a conversão de objetos de projeto em itens de configuração de sistema (software);
- ii) aprovar, monitorar e controlar alterações para o sistema.

A aprovação das "baselines" e alterações nas mesmas só são possíveis com o consentimento do Conselho de Controle de Configuração. No caso de uma modificação de emergência, não há necessidade de se reunir o CCC, contudo na próxima

data de reunião estas modificações devem ser notificadas para a sua aprovação ou não. É importante notar que as alterações rejeitadas são arquivadas para futuras referências.

O controle da configuração do software utiliza vários tipos de formulários. O grau de aplicação destes formulários dependerá do projeto e o gerente do Gerenciamento da Configuração tem que visualizar a fronteira entre a extensão necessária para assegurar o nível de controle requerido para o projeto e a burocracia.

Nunca é demasiado lembrar que a documentação básica do controle da configuração é a "baseline" e suas atualizações. O ciclo de desenvolvimento, identificação, controle, auditoria e contabilidade da situação é fundamental para a obtenção da integridade de um produto de software.

Quando as alterações são decorrentes do processo de revisão (alterações evolucionárias) o processo é direto. Os problemas surgem com as alterações externas (revolucionárias).

Alterações revolucionárias são referenciadas através de um formulário chamado Proposta de Alteração de Engenharia (PAE). Este formulário é utilizado para documentar todas as alterações revolucionárias e para ser usado pelo Conselho de Controle da Configuração.



Alterações revolucionárias ocorrem quando da alteração dos requisitos da implementação, do modelo ou de algum problema no desenvolvimento do sistema.

Relatórios do tipo Relatório de Incidentes (BERSOFF, 1980) ou Relatório de Problema e/ou Correção de Problema de Sistema (VINCENT, 1983) são outros tipos de formulários que podem originar uma Proposta de Alteração de Engenharia. São utilizados entre outras coisas para documentar uma má função do sistema.

Estes formulários encontram seus primeiros usos durante a Auditoria da configuração das "baselines" de produto e operacional. São o meio pelo qual a auditoria independente comunica qualquer deficiência para o desenvolvedor e o CCC.

Após a liberação do sistema são utilizados pelos usuários para relatar qualquer deficiência do sistema.

O formulário mais perigoso do controle da configuração é o Formulário de Alteração de Emergência (Remendo). O remendo é uma alteração para uma representação executável ou executando do software. Os remendos são alterações temporárias para resolver problemas críticos e imediatos.

Segundo BERSOFF et al (1980):

"Os formulários de controle da configuração fornecem algumas das ferramentas necessárias para se implementar um efetivo programa de GCS. Todos os pedidos de alteração e problemas

apresentados pelo sistema devem ser documentados e registrados para uma perfeita coordenação e acompanhamento da execução de uma alteração aprovada".

### 2.3.3 AUDITORIA

Auditoria da Configuração de Software (ACS) é o meio pelo qual o gerenciamento pode aumentar a visibilidade, estabelecer a rastreabilidade e assegurar que as integridades técnicas e administrativas do produto estão sendo satisfeitas por todo o ciclo de vida do projeto (WILBURN, 1983).

A Auditoria da Configuração de Software fornece o mecanismo para se comparar o estado corrente do software com o descrito nas "baselines" ou documentos de requisitos. O processo de auditoria é pré-requisito para formalização da "baseline" pelo Conselho de Controle da Configuração. Entretanto, segundo CHARETT (1988), a função de auditoria é entre as funções de GCS, a que menos é realizada com rigor.

Sendo assim, o estabelecimento formal de uma "baseline" fica a mercê apenas da vontade ou sentimento da gerência do projeto.

A função de auditar a configuração de um sistema requer que os responsáveis pela auditoria tenham uma grande experiência em desenvolvimento de software, para assim

poder analisar e avaliar o produto com precisão.

É indispensável que a pessoa ou equipe responsável pela auditoria do produto sejam totalmente independentes da equipe de desenvolvimento. O auditor deve ter sempre em mente o que está faltando no produto analisado, ao invés do que está presente.

A extensão e o grau em que a função de auditoria será aplicada, deve ser descrita no Plano de Gerenciamento da Configuração do Software e dependerá do tipo de projeto a ser desenvolvido.

A auditoria da configuração do software serve para dois propósitos:

- i) **verificação da configuração:** assegurar que cada Item da Configuração de Software (ICS) na "baseline" para-ser-estabelecida é logicamente relacionado com o ICS correspondente nas "baselines" precedentes, isto é, se é rastreável;
- ii) **validação da configuração:** assegurar que a configuração identificada satisfaz as funções sistema/software para serem desempenhadas por cada ponto de referência escolhido.

Como se pode notar a função de auditoria intercepta outras disciplinas como Garantia da Qualidade, Verificação

e Validação e Teste & Avaliação. O grau de interseção destas disciplinas com auditoria dependerá de como a gerência do projeto define cada uma delas.

É a aplicação da Auditoria da Configuração de Software nas "baselines" para-ser-estabelecida que irá fornecer ao Conselho de Controle da Configuração as informação para o estabelecimento formal das "baselines". A determinação da estrutura de ICSs e o relacionamento dos seus conteúdos com as informações disponíveis é o processo comum de auditoria em todas as "baselines". A auditoria enfoca os seguintes pontos nas "baselines" (BERSOFF, 1984):

- i) funcional: é o estabelecimento de um traço comum entre os requisitos globais do sistema e a concepção do sistema;
- ii) alocada: verifica se as funções do software foram definidas, se são rastreáveis com as funções do sistema identificadas na "baseline" funcional, se foram elaboradas baseadas em modelos e algoritmos implementáveis e se foram identificadas como ICSs;
- iii) projeto: verifica se os modelos e algoritmos definidos na "baseline alocada foram detalhados o suficiente para permitir uma codificação não ambígua;
- iv) produto: verifica se as representações

executáveis de ICSs desempenham adequadamente em ambientes de teste;

- v) **operacional:** verifica se as representações executáveis de ICSs desempenham adequadamente em ambiente operacional.

Concluindo, a auditoria da configuração torna visível para o gerente do projeto, ou da configuração a situação corrente do software auditado. Das quatro funções do Gerenciamento da Configuração a auditoria da configuração é a que requer mais pessoas (peritos) e recursos (dinheiro) sendo responsável por um custo superior ao das três outras funções combinadas. O resultado da auditoria pode ser o estabelecimento de uma "baseline", o redirecionamento de uma "baseline" e o ajustamento dos recursos aplicados no projeto.

#### **2.3.4 CONTABILIDADE DA SITUAÇÃO**

Durante o desenvolvimento de um sistema (software) um atraso segue a decisão de se incorporar uma alteração em relação a implementação real desta alteração. Um mecanismo é necessário para se manter o registro da evolução do sistema e para se saber onde que ele está a qualquer hora em relação ao que consta em uma "baseline" estabelecida (BERSOFF, 1984)..

A contabilidade da situação da configuração de software (CSCS) fornece este mecanismo e ferramentas para se

determinar quais eventos aconteceram durante o desenvolvimento do sistema (software) e como e quando eles ocorreram.

Segundo BRYON (1980), a Contabilidade da Situação da Configuração de Software (CSCS) é provavelmente a parte menos entendida do gerenciamento da configuração. Geralmente é vista como um conjunto volumoso e caro de relatórios. A CSCS é um processo administrativo e os relatórios são só um meio de realizar este processo.

Sua intenção é facilitar o processamento das alterações. Desta maneira, as alterações não são perdidas nem atrasadas antes do estabelecimento formal da "baseline".

A Contabilidade da Situação da Configuração de Software requer que todas as alterações sejam formalmente registradas desde quando a primeira proposta de alteração for registrada oficialmente, até a sua desaprovação, aprovação ou sua incorporação ao projeto.

A Contabilidade da Situação da Configuração de Software fornece informação para se saber exatamente qual a configuração corrente do software e como e quando ela foi realizada. O rastreamento administrativo e o relatório de todos os itens de software formalmente identificados e controlados e a manutenção de registros para suportar a auditoria da configuração do software são a CSCS.

A Contabilidade da Situação da Configuração de Software deve registrar e relatar (WILBURN, 1983):

- i) a data e hora na qual cada representação da "baseline" e atualização foi estabelecida;
- ii) a data e hora na qual cada ICS foi estabelecido;
- iii) informação descritiva sobre cada ICS;
- iv) a situação de cada Proposta de Alteração de Engenharia (aprovado, desaprovado, esperando ação);
- v) informação descritiva de cada Proposta de Alteração de Engenharia;
- vi) situação da alteração;
- vii) informação descritiva sobre cada alteração;
- viii) a situação da documentação técnica e/ou administrativa relativa a uma "baseline";
- ix) deficiências em uma "baseline para-ser-estabelecida descobertas durante a auditoria.

A Contabilidade da Situação da Configuração de Software tem três funções: registrar dados, armazenar dados e apresentar dados. A função registrar dados deve armazenar todos os eventos que tenham importância para o projeto, tais como: o que, quando e como. As informações

especificadas no Plano de Gerenciamento da Configuração do Software devem ser **tornadas** como requisitos **mínimos**. Um bom conselho é, na dúvida, registre.

Para qualquer tipo de projeto devem ser registrados:

- i) detalhes de Itens de Configuração de Software;
- ii) sumários de revisão de "baseline";
- iii) deficiências identificadas da "baseline";
- iv) alterações na "baseline";
- v) minutas do Conselho de Controle da Configuração;
- vi) propostas de alteração;
- vii) relatórios de problema e/ou correção de problema de sistema;
- viii) sumários de auditoria.

A segunda função deve resultar em uma base de dados com todas as informações registradas. Tem como propósito suprir informações para os relatórios e suprir dados para a análise do projeto. Devido ao grande volume de dados é recomendável o uso de ferramentas automatizadas.

Conforme SIBLEY (1981), um Sistema gerenciador de



Configurações está intimamente relacionado com um Sistema gerenciador de Banco de Dados e um dicionário de dados.

Este sistema deve ser ativo, ou seja, deve forçar que as diretrizes de acessos aos dados sejam seguidas, ao contrário de um sistema passivo que apesar de suportar estas diretrizes, permite que elas sejam desrespeitadas.

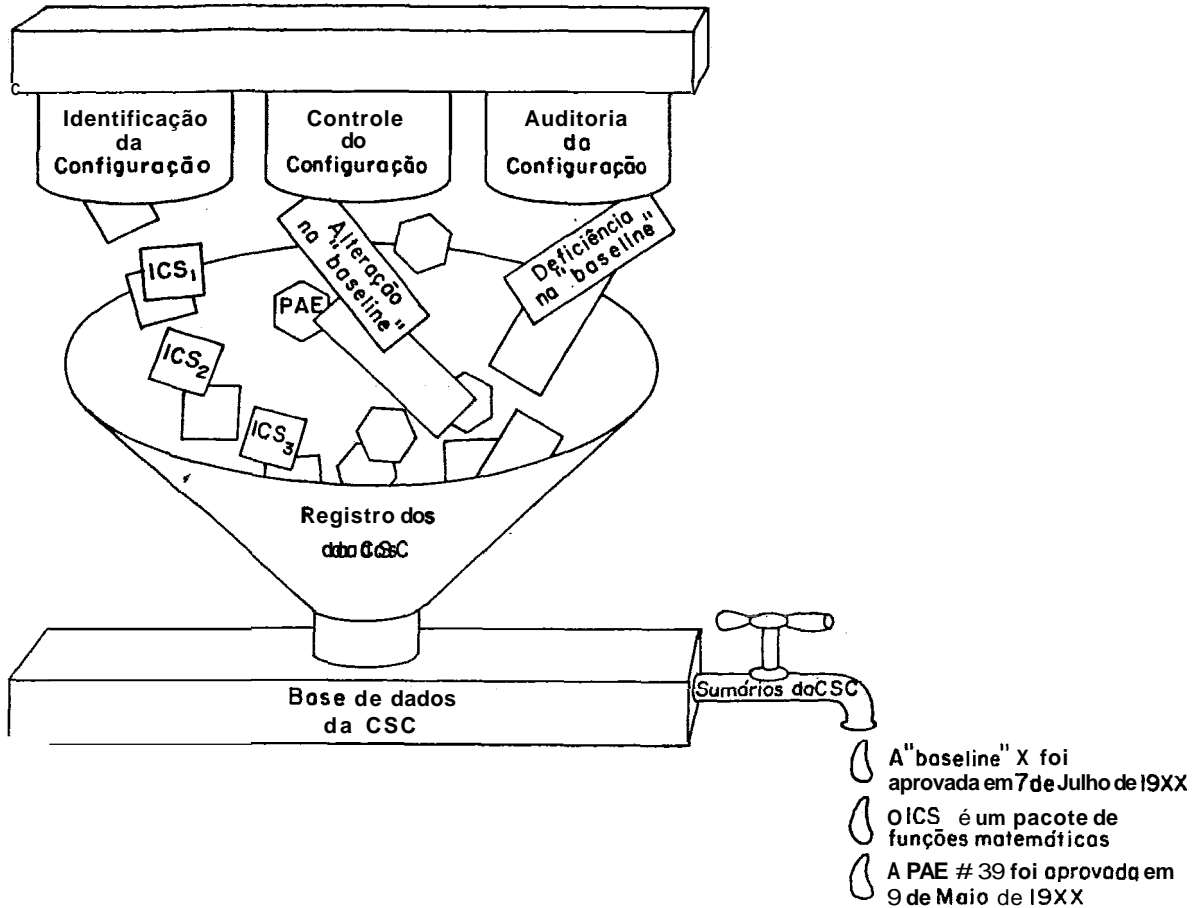
A terceira função, apresentar dados, fornece um meio documentado de comunicação entre todas as partes interessadas no projeto. Ela deve preparar relatórios programados e estar apta a fornecer outros relatórios não usuais.

Concluindo, o volume de informações detalhadas requeridas por um programa e os meios de rastreamento dependem das necessidades e decisões do responsável pelo projeto mas, de uma maneira ou de outra, o rastreamento tem que ser feito.

A figura II.3 fornece uma visão geral da Contabilidade da Situação da Configuração de Software.

## **2.4 CUSTOS DO GERENCIAMENTO DA CONFIGURAÇÃO**

O modelo de custo RCA PRICE S (FORMAN, 1981), de avaliação de preços de software, foi desenvolvido pela RCA



**LEGENDA:**

- PAE - Proposta de alteração de engenhario
- ICS - Ítem de configuração de software
- CSC- Contabilidade da situação da configuração

Figura II.3 - Contabilidade da situação da configuração - visão geral

para medir o custo do Gerenciamento da Configuração de Software. Este modelo apresenta os custos do Gerenciamento da Configuração em conjunto com os da Garantia de Qualidade.

O modelo apresenta inúmeras variáveis de entrada, sendo que os resultados que serão apresentados mais adiante só levam em consideração o número de instruções e as datas de início e final do projeto. Tais resultados foram obtidos somente em valores médios da RCA.

Os casos apresentados variam de mil a um milhão de instruções e de dois meses a vinte anos, sendo que todos os projetos começaram em 01/12/30. Para cada número de instruções existe três cronogramas: acelerado, normal e estendido.

Observando a tabela (figura II.4), verifica-se:

- i) O percentual do custo independe do tipo de cronograma;
- ii) O percentual do custo depende do número de instruções;
- iii) O aumento do número de instruções não é proporcional ao aumento do percentual do custo;
- iv) Para programas com um número pequeno de instruções, há pouca diferença em gerenciar um

projeto com um cronograma acelerado ou normal;

- v) Programas com cronograma normal tem o maior percentual do custo total.

A figura (II.5) apresenta outra característica interessante, a inclinação diminui drasticamente no nível de cinco mil instruções (programa pequeno). Isto ocorre devido que nesta hora quase todos os procedimentos de gerenciamento da configuração já foram introduzidos.

Nº instruções	programa acelerado	programa normal	programa estendido
1.000	12,0%	12,5%	10,5%
2.500	13,0%	13,5%	12,2%
5.000	13,9%	14,8%	13,5%
7.500	14,2%	14,9%	13,8%
10.000	14,4%	15,2%	14,0%
25.000	15,4%	16,5%	15,1%
50.000	16,0%	17,1%	15,6%
100.000	16,5%	17,4%	15,7%
500.000	17,8%	18,5%	17,5%
1.000.000	18,0%	19,5%	18,3%

Figura II.4 - Número de instruções versus o percentual do custo de da Garantia da Qualidade/ Gerenciamento da Configuração de Software

O modelo de custo não fornece uma estimativa precisa nos cronogramas acelerado e estendido visto que, ao contrário do cronograma normal que apresenta estágios bem definidos e previsíveis, os procedimentos de Gerenciamento da Configuração não podem ser aplicados sem o gasto extra de recursos.

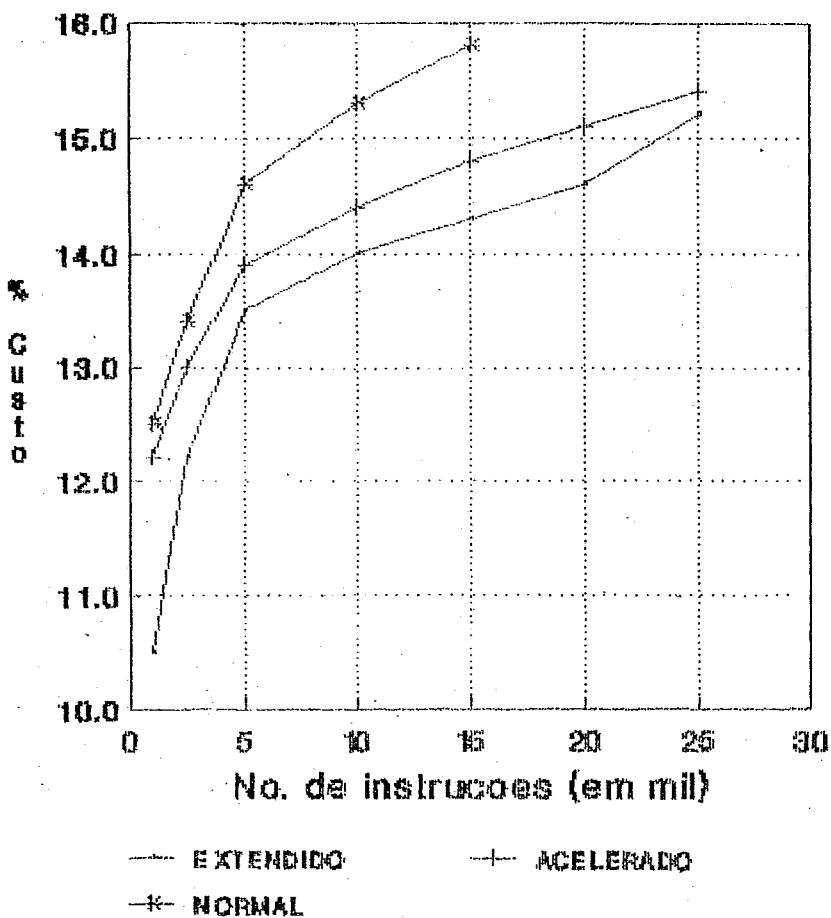


Figura II.5 - Porcentagem total do custo de gerenciamento da configuração de software versus o número de instruções.

Fonte - FORMAN (1981)

Como o modelo é baseado fundamentalmente no número de

Instruções, a estimativa do número de instruções que um determinado projeto irá ter é tida pela comparação com projetos com funções similares ao do projeto em questão.

## 2.5 FERRAMENTAS AUTOMATIZADAS

Segundo ALDERSON (1985), o Gerenciamento da Configuração foi identificado em 1983 como a maior área na qual a produtividade podia ser melhorada pelo fornecimento e uso de ferramentas automatizadas.

MOQUIN (1985), lista os tipos de ferramentas para software manuais e automatizadas para o controle e gerenciamento de alterações sobre o ciclo de vida de um projeto. Estes sistemas podem ser vistos como unidades funcionais, que no futuro poderiam ser integrados em um único sistema de controle da configuração.

As ferramentas por área de aplicação são:

### i) IDENTIFICAÇÃO

- Sistemas de controle de versão;
- Sistemas de gerenciamento de código;
- Sistemas de gerenciamento de módulo;
- Sistemas de gerenciamento de biblioteca;
- Sistemas de suporte de geração.

### ii) CONTROLE

- Sistemas de rastreabilidade de requisitos;
- Sistemas de suporte a especificação;
- Sistemas de cronograma PERT/CPM;
- Modelagem de custo e sistemas de Estimativa/determinação.

iii) AUDITORIA E REVISÃO

- Sistemas de gerenciamento de "baselines"

iv) OUTROS

- Sistemas de geração de relatórios;
- Sistemas de controle de bibliotecas;
- Sistemas de arquivamento, e
- Sistemas de recuperação e "Rollback".

MOQUIN (1985) conclui afirmando:

"Ferramentas automatizadas para a evolução do software, parecem ser as menos entendidas, as menos disponíveis e as menos aplicadas. Ao mesmo tempo, parecem ser as mais necessárias, as mais desejadas e as de maior custo efetivo."

2.5.1 EXEMPLOS DE FERRAMENTAS

"Softool's Change and Configuration Control" (LOBBA,

1985) é uma ferramenta automatizada para o gerenciamento da configuração, na qual a informação é organizada em uma hierarquia de estruturas de dados. Esta hierarquia apresenta um banco de dados no seu nível mais alto que é seguido hierarquicamente por sistema/ configuração/ módulo/ texto.

A estrutura Módulo pode representar diferentes programas e rotinas que completam uma configuração específica e a estrutura Texto mantém os códigos fonte, objeto e a documentação que acompanha o módulo. Em qualquer nível hierárquico há um texto associado a ele, o qual fornece um mecanismo conveniente para documentação.

Esta ferramenta permite ao usuário entrar com o nome e a descrição de uma alteração. Este nome torna-se uma chave na qual uma ou mais alterações relacionadas podem ser acessadas.

Permite também ao usuário manter uma completa auditoria de todas as modificações feitas, possibilitando o exame e a recriação de qualquer alteração realizada.

A "Softtool's" possibilita ao usuário vários tipos de relatórios e também limita a capacidade de acesso do usuário a uma estrutura específica. Isto é feito através de um controle de acesso baseado na estrutura, usuário e classe de usuário.



A inabilidade de sistemas de gerenciamento de configuração de gerenciar acesso concorrente a arquivos compartilhados por vários usuários é superada pelo o VMLIB da PANSOPHIC (16) que é um sistema de biblioteca de códigos fonte.

O VMLIB possui também interfaces diretas para o ambiente de produção DOS ou OS. Este sistema permite que o administrador da base de dados restrinja o acesso à alguns arquivos através de senhas individuais, classes de usuários com privilégios de acesso diferenciado ou pelo tipo de aplicação a ser compartilhada.

VALLINO (1987) analisa vários Sistemas Gerenciadores de Código Fonte (SGCF) para microcomputadores. Estes SGCFs tem como funções básicas controlar o armazenamento e a geração das múltiplas revisões de um módulo de código fonte e fornecer um mecanismo para gerar qualquer revisão solicitada.

Estes Sistemas Gerenciadores de Código Fonte armazenam as alterações em um arquivo fonte em termo de deltas, ou seja, a informação comum às duas revisões não é duplicada.

Cada delta representa então, somente as diferenças

entre duas revisões de um arquivo. Ao ser solicitada uma revisão, o sistema tendo por base um arquivo original aplica os deltas um de cada vez, até que a revisão solicitada seja gerada.

Expansão automática de palavras-chave inseridas no código fonte, é outra característica apresentada pelos SGCFs. Elas permitem a identificação da revisão bem como do código objeto e executável originados por esta revisão. Neste caso é necessário uma ferramenta para examinar o código objeto ou executável para procurar a palavra-chave inserida, que geralmente está embutida em uma sequência de caracteres.

Os vários Sistemas Gerenciadores de Código Fonte analisados variam quanto a estrutura de armazenamento dos arquivos mantidos por eles. Alguns usam um único arquivo ou vários arquivos e/ou diretórios, este sendo um pouco mais difíceis de serem usados.

Outro dispositivo encontrado em quase todos Sistemas Gerenciadores de Código Fonte é o Rótulo de versão simbólico. Este rótulo permite o estabelecimento de uma versão através da seleção dos módulos que a compõem.

Concluindo, não importa quão poderosa seja a ferramenta, se não houver consciência por parte da equipe responsável pelo desenvolvimento do sistema. Sem esta

consciência, sempre haverá maneiras de se burlar o sistema ou ignorá-lo. Cada ferramenta apresenta vários dispositivos, uns mais, outros menos poderosos, mas de qualquer maneira, melhores do que se manter e controlar as alterações manualmente.

## 2.6 NOTAS FINAIS

Apesar de ser uma das disciplinas mais bem entendidas da Engenharia de Software, o Gerenciamento da Configuração de Software apresenta problemas na sua implementação devido às dificuldades de comunicação com o gerenciamento da configuração tradicional. A falta de padronização dos termos utilizados por ambos e a pouca oferta de ferramentas automatizadas de controle são os principais obstáculos enfrentados pelo Gerenciamento da Configuração de Software. Além disso, o Gerenciamento da Configuração de Software ainda é visto como uma atividade burocrática e enfadonha pela comunidade de processamento de dados.

Através das suas funções: identificação, controle, auditoria e contabilidade da situação, o Gerenciamento da Configuração de Software fornece a todas as partes interessadas no sistema, um meio de comunicação claro e consistente sobre o estado corrente do sistema em desenvolvimento. Assim sendo, a utilização do Gerenciamento da Configuração de Software é fundamental para a obtenção da integridade de um sistema.

A economia de recursos na fase de manutenção devido ao uso do Gerenciamento da Configuração de Software justifica a utilização de parte destes recursos para a sua implantação no início do desenvolvimento. Conforme foi visto, Gerenciamento da Configuração de Software é a área de maior aumento de produtividade com o fornecimento de ferramentas automatizadas. Assim sendo, o Gerenciamento da Configuração de Software é uma ferramenta da Garantia da Qualidade para a comunicação e rastreamento de alterações por todo o ciclo de vida do sistema obtendo então, os custos e cronogramas previstos.

Os benefícios provenientes do Gerenciamento da Configuração de Software estão presentes no desenvolvimento e manutenção do software, no controle de documentos e alterações e no rastreamento e previsão de problemas.

Por fim, o treinamento de profissionais no Gerenciamento da Configuração de Software é vital devido ao fato de que com o avanço da tecnologia na próxima década mais dados serão manipulados e novas formas de representação surgirão para serem controladas.

## CAPÍTULO III

### PLANO DE GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE

#### 3.1 INTRODUÇÃO

Este capítulo apresenta uma proposta para a elaboração de um Plano de Gerenciamento da Configuração de Software (PGCS).

São descritas as políticas, os procedimentos, uma estrutura organizacional, ferramentas e documentação de referência para auxiliar o Gerente da Configuração na preparação de um Plano de Gerenciamento da Configuração de Software que atenda o nível de controle esperado, mas que não dificulte o fluxo normal do desenvolvimento do software.

O modelo de plano sugerido tem como **base** a padrão IEEE para Planos de Gerenciamento da Configuração de Software (IEEE 828-1983), que aplica-se a todo ciclo de vida de um software crítico (software cuja falha pode acarretar impacto na segurança ou causar grandes perdas sociais ou econômicas).

### 3.2 DESCRIÇÃO DE UM PLANO DE GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE

Ao se elaborar um PGCS segundo o padrão IEEE para PGCS, todas as seções têm que estar presentes. Caso uma seção não tenha aplicação no sistema a ser desenvolvido, esta tem que estar presente mesmo assim.

Neste caso, virá acompanhada da frase " não existe informação pertinente para esta seção " e uma justificativa para a sua ausência.

A proposta do IEEE para PGCS se compõe das seguintes seções descritas a seguir :

#### Seção 1 - INTRODUÇÃO

Esta seção deve fornecer uma visão geral do Plano.

##### Seção 1.1 - Propósito

Esta subseção delinea o propósito específico de um PGCS.

##### Seção 1.2 - Escopo

Esta subseção deve identificar os itens a serem produzidos e usados, as organizações, as atividades, e as fases do ciclo de vida do software para as quais o Plano se

aplica.

### Seção 1.3 - Definições e Siglas

Esta subseção deve definir ou fornecer uma referência de todas as definições de todos os termos e acronismos requeridos para a interpretação do PGCS.

### Seção 1.4 - Referências

Esta subseção deve fornecer uma lista de todos os documentos referenciados no PGCS, indicando aonde estes documentos podem ser localizados.

Os documentos abaixo relacionados foram utilizados na preparação deste plano :

- IEEE Std 828-1983, IEEE Standard for Software Configuration Management Plans
- IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology
- BERSOFF (1980), Software Configuration Management - An Investment in Product Integrity
- BRYON (1980), Tutorial Software Configuration Management
- VINCENT (1988), Software Quality Assurance - Practice

and Implementation

- WILBURN (1983), Guidelines - Software Configuration Management

## Seção 2 - GEAENCXAMENTQ

### Seção 2.1 - Organização

O diagrama funcional abaixo (figura III.1), ressalta as atividades de Garantia do Produto de Software e seus Tem como propósito evitar a minimizar a duplicação dos esforços decorrentes da aplicação das atividades de Garantia da Qualidade de Software, este & Avaliação, Verificação & validação e Gerenciamento da Configuração de Software, tendo em vista, que não existe uma total concordância quanto as fronteiras de atuação e aplicação de cada uma deseac atividades.

É sugerida a implantação do Círculo de Garantia da Qualidade Principal (CGQP), que possui ascendência sobre todos os outros círculos da qualidade envolvidos no processo de produção de software.

Abaixo são descritos cada componente do diagrama dando ênfase aos aspectos relacionados à Garantia do Produto de Software.



#### - **Círculo de Garantia da Qualidade Principal (CGQP)**

Este círculo apesar de estar administrativamente sob a Divisão de Gerência de Sistemas de Computação, não está sob o controle gerencial desta Divisão. Reporta-se direto ao Gerente do Projeto mantendo assim, sua autonomia e autoridade para a execução de sua atividade. O Círculo da Garantia da Qualidade Principal tem poder concorrente a Divisão de Projeto e Manutenção de Sistemas para aprovar os planos de Teste & Avaliação, Verificação & validação, e a Configuração de Software e Garantia da Qualidade de Software.

#### - **Conselho de Controle da Configuração (CCC)**

É um grupo permanente com o poder de revisão e aprovação sobre qualquer alteração proposta. Tem, também, como responsabilidades, o acompanhamento das implementações das alterações aprovadas segundo as políticas e procedimentos da organização.

Apesar do Conselho de Controle da Configuração ter controle direto sob o material controlado, tem que haver concordância por parte do Círculo da Garantia da Qualidade Principal para a efetivação das suas ações.

O Conselho de Controle da Configuração reporta-se diretamente ao Gerente do Projeto.

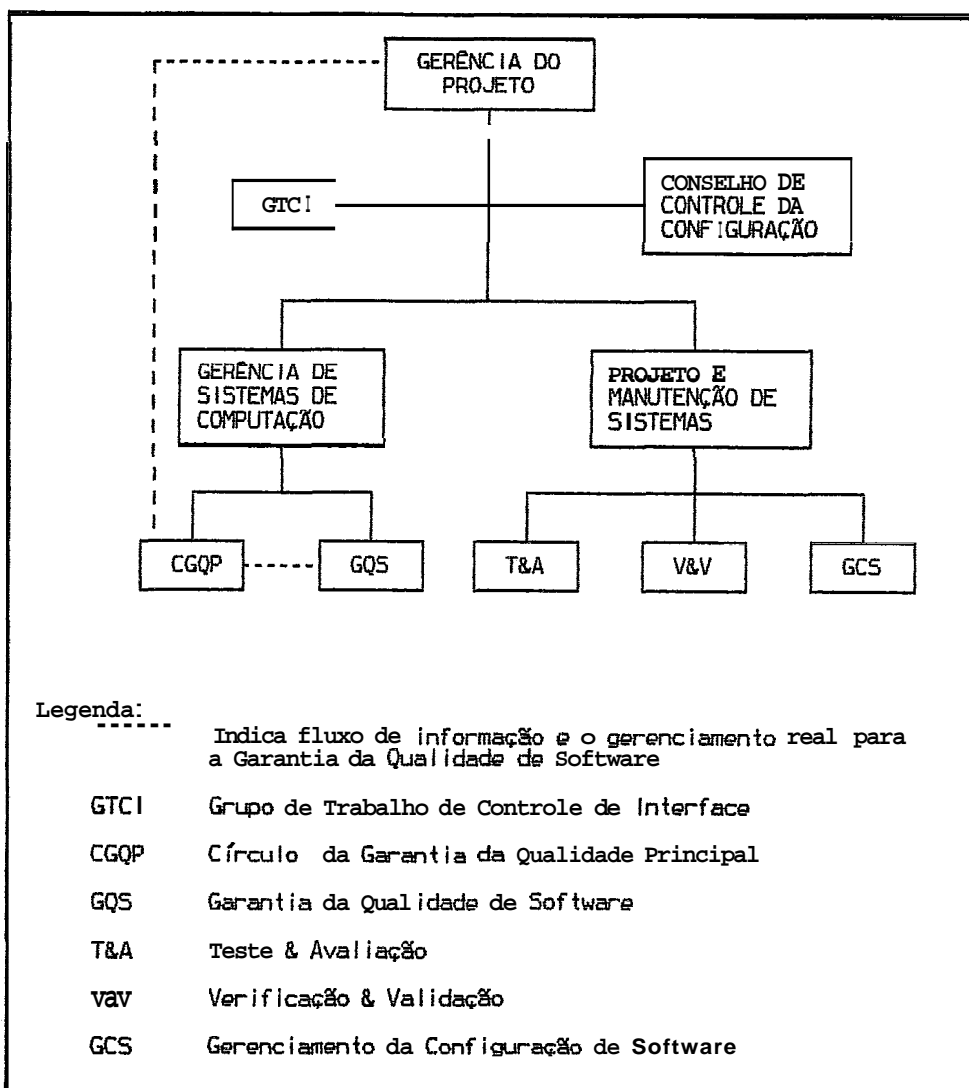


Figura III.1 - Modelo de organização para melhor utilização das funções da garantia do produto

#### **- Divisão de Projeto e Manutenção de Sistemas (DPMS)**

Tem como função o desenvolvimento e a manutenção de sistemas. Sob o ponto de vista da garantia do produto, deve gerenciar os dados, para assegurar que toda a documentação do projeto, antes da sua aprovação para controle, é preparada de acordo com os padrões da indústria e da organização. Estes procedimentos serão, então, monitorados pela a Divisão de Garantia da Qualidade de Software.

#### **- Divisão de Gerência de Sistemas de Computação (DGSC)**

Tem como função, além da administração do projeto, fornecer todo o apoio administrativo à Divisão da Garantia da Qualidade de Software (DGQS) e ao Círculo da Garantia da Qualidade Principal.

#### **- Grupo de Trabalho de Controle de Interface (GTCI)**

É apontado pelo Gerente do projeto com o propósito de controlar todas as iterações de interface dentro da organização contratada. Deve ser constituído de pelo menos um membro de cada um dos contratados e membros dos contratantes que participam no desenvolvimento do software.

- Garantia da Qualidade de Software, Gerenciamento da Configuração de Software, Verificação & validação e Teste & Avaliação

Estas funções são integradas em um único processo de controle, evitando-se a duplicação de esforços e melhorando a comunicação.

Destas quatro funções da Garantia do Produto, apenas o administrador de Garantia da Qualidade de Software se reporta diretamente ao Círculo da Garantia da Qualidade Principal. O Gerenciamento da Configuração de Software entre elas é a mais independente, tendo que ter estreito relacionamento com a gerência do projeto.

## Seção 2.2 - Responsabilidades

- Gerente do Projeta

É o responsável pela realização do projeto durante todas as fases do ciclo de vida .

- Gerente da Gerenciamento da Configuração de Software

Tem como responsabilidades :

i) gerenciar e dirigir o Gerenciamento da Configuração de Software de acordo com as políticas, procedimentos e diretrizes estabelecidas pelo Conselho de Controle da

Configuração;

- ii) coordenar as funções de Gerenciamento da Configuração de Software e de Desenvolvimento do sistema para integrar os esforços necessários à realização do projeto;
- iii) fornecer direção técnica sobre o Gerenciamento da Configuração de Software para todos os grupos envolvidos no desenvolvimento do produto de software, implementando procedimentos e padrões de Gerenciamento da Configuração de Software;
- iv) manter registro dos encontros do Conselho de Controle da Configuração;
- v) manter informações para o Conselho de Controle da Configuração através de arquivos de auditoria de Gerenciamento da Configuração de Software;
- vi) estabelecer e manter arquivos centrais com a situação de desenvolvimento do sistema. Mantendo os originais dos módulos e das documentações pertinentes;
- vii) indicar o bibliotecário.

**- Conselho de Controle da Configuração (CCC)**

Tem como responsabilidades :

- i) avaliar todos os pedidos de modificação de documentos de projeto e módulos já sob Gerenciamento da Configuração de Software;
- ii) avaliar o impacto das alterações propostas, designando prioridades e recursos para a realização das alterações aprovadas;
- iii) aprovar, monitorar e controlar a conversão de Objetos de Projeto em Itens de Configuração de Software (ICSS);
- iv) o Conselho de Controle da Configuração deve também participar como revisor do processo de desenvolvimento (alterações evolucionárias).

#### - Divisão de Projeto e Manutenção de Sistemas (DPMS)

Tem como responsabilidades :

- i) controlar o Projeto de Software nas fases de Desenvolvimento, Projeto, Codificação, Testes e Implementação. Deve também, se comunicar com o Gerente de Gerenciamento da Configuração de Software para assegurar o cumprimento do PGCQ;
- ii) fazer as avaliações de impacto de uma alteração

proposta quanto ao cronograma, orçamento, quando solicitadas.

**- Divisão de Gerência de Sistemas de Computação (DGSC)**

Tem como responsabilidades :

- i) fornecer todo o suporte administrativo para todas as equipes envolvidas no projeto;
- ii) fornecer relatórios gerenciais quando solicitados.

**- Círculo da Garantia da Qualidade Principal (CGQP)**

Tem como responsabilidades :

- i) assegurar que todos os padrões e planos estabelecidos pelas divisões de Garantia da Qualidade de Software, Gerenciamento da Configuração de Software, Teste & Avaliação e Verificação & validação estão sendo seguidos;
- ii) trabalhar em conjunto com o Conselho de Controle da Configuração para assegurar uma total concordância quanto a seleção e identificação dos ICS, natureza do impacto das alterações propostas, aceitação ou rejeição de

alterações e documentos internos;

iii) selecionar e coordenar comentários dos usuários e revisores, documentando-os por meio de formulários apropriados;

iv) monitorar as atividades dos subcontratados;

v) monitorar as atividades dos subcontratantes.

#### - Grupo de Trabalho de Controle de Interface (GTCl)

Tem como responsabilidades :

i) Assegurar que a identificação dos itens de configuração estão em conformidade com as interfaces funcionais e físicas estabelecidas;

ii) Servir como ligação de comunicação oficial entre os participantes do programa, para resolver problemas de interface, coordenar Propostas de Alteração de Engenharia (PAE) (no tocante as interfaces) e estabelecer uma concordância nas interfaces entre documentos.

#### - Bibliotecário

Tem como responsabilidades :

i) manter e controlar fisicamente as cópias



principais das "baselines" aprovadas em conjunto com o Contratante, para serem usadas como referência para qualquer alteração externa proposta;

- ii) manter e controlar fisicamente as "baselines" internas aprovadas e módulos de programas que ainda não foram aprovados pelo contratante;
- iii) manter (registrada) uma referência cruzada dos usuários de uma subrotina versus as modificações realizadas que interfiram direta ou indiretamente nestas subrotinas, avisando-os automaticamente quando alterações ocorrerem;
- iv) fornecer a versão corrente de programas ou segmentos de programas para testes;
- v) manter as versões correntes e anteriores dos programas em desenvolvimento.

### **Seção 2.3 - Controle da Interface**

Desenho de Controle de Interface (DCI) é um documento que mostra os requisitos das interfaces funcionais e físicas de um item de configuração (equipamento ou programa de computação) a qual afeta o projeto ou operação de outro itens de configuração.

Tem como responsabilidades :

- i) a gerência do Grupo de Trabalho de Controle de Interface tem que ser designada pela atividade de Controle de Interface do contratado, através da coordenação com o contratante;
- ii) manter os detalhes das interfaces durante a fase de validação devem ser tratadas pelo o critério estabelecido para o sistema e não deve ser estabelecida como parte da atividade de controle de interface. Com isto, os contatos entre as partes, têm a flexibilidade necessária para que as negociações fiquem dentro das faixas estabelecidas para o sistema;
- iii) desenvolver a identificação da configuração cobrindo os requisitos de interface do sistema que podem ser delegadas contratualmente em parte ou no todo para o Controle de Interface do Contratado;
- iv) cabe ao Contratante determinar os requisitos para o Controle de Interface e requisitos de instalação durante a fase de instalação;
- v) estabelecer as responsabilidades, os relacionamentos e requisitos que têm que ser

especificados nos contratos;

vi) no caso da aquisição de programas durante o ciclo de projeto deve formar um Grupo de Trabalho de Controle de Interface cuja funções serão determinadas pela a atividade de Controle de Interface;

vii) a aquisição de programas que não requerem todo o ciclo de desenvolvimento, ou seja, vão diretamente da validação para a produção, o Grupo de Trabalho e Controle de Interface não é aplicado e o Contratante especificará outros procedimentos para a atividade de Controle de interface;

viii) os Documentos de Controle de Interface (DCI) serão usados para registrar os acordos de projeto entre os contratados e o contratante. Devem refletir os efeitos de qualquer desistência ou desvio sobre as interfaces.

#### Seção 2.4 - Implementação da Plana de Gerenciamento da Configuração de Software

A partir do momento em que foi tomada a decisão da execução de um sistema, muitas das atenções estão voltadas

para a formação da equipe de desenvolvimento estudo de orçamento e cronograma, entre outras preocupações. É neste ponto, ou seja, logo a seguir ao início do projeto, que alguém na organização que contratou o serviço (recomendável) e na organização desenvolvedora (mandatório) deve direcionar sua atenção para como será gerenciada a configuração deste novo sistema.

Como primeiro passo para esta tarefa, é necessário que seja designado um gerente da configuração com a responsabilidade de produzir um PGCS. Este plano irá delinear então, como a organização de Gerenciamento da Configuração irá aplicar sua porção de recursos do projeto por todo o ciclo de vida do sistema.

Para o caso de um sistema que envolva o desenvolvimento de hardware e software, é uma boa prática de engenharia a aplicação do PGCS em conjunto com o Plano de Gerenciamento da Configuração de Hardware (PGCH) e o Plano de Gerenciamento da Configuração de Sistema (PGCSt). Neste caso, o PGCSt será preparado logo a seguir ao início do projeto e os outros dois planos, após a "baseline" alocada ser aprovada.

## Seção 2.5 - Políticas, Diretrizes e Procedimentos aplicáveis

- Inclusão de um item de Configuração de Software (ICS) na

## Biblioteca Mestre de Programas

- i) A equipe de Gerenciamento da Configuração de Software faz uma auditoria sobre o módulo de software. Verifica se os requisitos de formato, padronização, rastreabilidade e testes estão segundo os resultados esperados;
- ii) Preparação de um Documento de Descrição de Versão para cada módulo a ser incluído;
- iii) Liberação de uma cópia do módulo (no meio apropriado), juntamente com o Documento de Descrição de Versão e uma listagem do código fonte para o responsável pelo controle da configuração de software;
- iv) Os membros da equipe de desenvolvimento realizam o teste de integração (quando necessário) da nova "baseline";
- v) São preenchidos os Diários de Item de Configuração (DIC), o Índice de item de Configuração (IIC) e o Relatório de Gerenciamento da Configuração de Software.

- Liberação formal de material controlada

- i) O Desenvolvedor deve preencher os campos específicos do formulário do Relatório de Gerenciamento da Configuração de Software (RGCS), o qual, é usado para notificar o Gerente de Gerenciamento da Configuração de Software o pedido de liberação ou inclusão de um Item de Configuração de Software (ICS);
  
- ii) O Gerente de Gerenciamento da Configuração de Software autoriza a liberação pelo Bibliotecário de uma cópia para o desenvolvedor, registrando os dados referentes a liberação (data, justificativa, nome do solicitante, etc...);
  
- iii) O Desenvolvedor após os testes de aceitação e integração preenche um Relatório de Gerenciamento da Configuração de Software notificando o sucesso ou não (através de um Relatório de Problema e/ou Correção de Problema de Sistema) dos testes;
  
- i.) .. caso de sucesso, o Gerente do Gerenciamento da Configuração de Software guarda a cópia principal do Item de Configuração do Software (ICS), atualiza a versão e cria uma descrição

desta versão.

- Procedimento para o estabelecimento de uma "Baseline"

- i) O Desenvolvedor apresenta uma "Baseline";
- ii) O Conselho de Controle da Configuração pede a função de Identificação do Gerenciamento da Configuração de Software um esquema com a árvore de itens de configuração de software do sistema;
- iii) Esta árvore é enviada para a função de Auditoria. Caso seja encontradas deficiências estas são relatadas ao Conselho de Controle da Configuração, caso contrário o sancionamento desta "baseline" é proposto;
- iv) Se não houverem deficiências, o Conselho de Controle da Configuração analisa os relatórios de Auditoria e sanciona (estabelece) a "baseline". Se houver alguma deficiência estas são enviadas ao Desenvolvedor e o processo de estabelecimento da "baseline" retornará ao passo a).

- Procedimento para o controle e escolha dos Itens de Configuração de Software (ICS)

- i) todas as formas de cada ICS tem que ser

- consistentes;
- ii) nenhum ICS pode ser incluído ou modificado sem autorização;
  - iii) nenhum ICS pode conter desvios não autorizados a partir dos padrões estabelecidos;
  - iv) tanto a versão corrente como a anterior de um ICS tem que estar disponíveis para serem rapidamente armazenadas e recuperadas;
  - v) A identificação a nível de linhas de código não é necessária;
  - vi) os ICSs não devem ultrapassar as fronteiras entre computadores e/ou programas independentes;
  - vii) processos com grande interação devem ser um único ICS;
  - viii) programas com cronogramas diferentes de desenvolvimento, ou teste, ou liberação devem ser ICSs separados;
  - ix) cada ICS requer a sua própria árvore de especificação, revisões e auditorias, testes de aceitação e validação, e serem contabilizados individualmente;



x) o número de ICSs não deve ser muito grande, visto que há necessidade de se documentar cada ICS separadamente.

## - Guia para classificação de alterações Classe I e II

### Classe I

- i) qualquer alteração afetando custo, cronograma, desempenho, ou modificando a especificação;
- ii) alterações técnicas na identificação da "baseline" funcional ou alocada de um ICS;
- iii) alterações técnicas na identificação da "baseline" de produto;
- iv) alterações envolvendo desempenho fora das tolerâncias especificadas do produto ou afetando as características de interface do produto;
- v) alterações com impacto na garantia e liberação de material;
- vi) alterações afetando equipamentos fornecidos no tocante a compatibilidade com outros

equipamentos e com o suporte e treinamento.

- vii) todas as alterações Classe I tem que ser aprovadas pelo Conselho de Controle da Configuração do Contratado, do Contratante e ser referendada pelo Circulo da Garantia da Qualidade Principal.

#### Classe II

- i) Qualquer outra alteração que não seja da Classe I;
- ii) Alterações Classe II podem ser aprovadas pelo Conselho de Controle da Configuração do Contratado;
- iii) Cabe ao Contratante concordar na classificação de uma alteração como alteração Classe II.

#### - Procedimentos na utilização dos documentos de controle

Os documentos utilizados para controle são :

- i) Relatório de Problema e/ou Correção de Problema de Sistemas (RPCPS);

- ii) Proposta de Alteração de Engenharia (PAE);
- iii) Notificação de Alteração de Software (NAS);
- iv) Relatório de Revisão de Documento (RRD);
- v) Documento de Descrição de Versão (DDV);
- vi) Formulário de Rastreamento de Documentos (FRD);
- vii) Formulário de Alteração de Emergência (FAE)

A descrição de cada um deles e a sua utilização estão a seguir :

**i) RPCPS - Relatório de Problema e/ou Correção de Problema de Sistema**

**Descrição :** É um documento composto de cinco seções (figuras (III.2) e (III.3) e é utilizado para dois propósitos distintos. As duas primeiras seções (a-b) descrevem uma discrepância em um programa, na documentação, em um outro software de apoio, ou no hardware de interface, enquanto que as três últimas seções (c-e) são utilizadas para responder as questões levantadas.

**Procedimento :**

- 1 - A Divisão de Projeto e Manutenção de Sistemas produz um RPCPS para registrar uma discrepância conhecida ou suspeita;
- 2 - A função responsável pelo Controle da Configuração registra a situação corrente deste RPCPS e retorna a Divisão de Projeto e Manutenção de Sistemas para análise;
- 3 - A Divisão de Projeto e Manutenção de Sistemas analisa o possível defeito, identificando-o e verificando o impacto decorrente da correção ou não deste problema e preenche as seções (c-e) do RPCPS;
- 4 - O Conselho de Controle da Configuração revisa este RPCPS, comenta e envia para o Círculo da Garantia da Qualidade Principal;
- 5 - O Círculo da Garantia da Qualidade Principal revê as impressões do Conselho de Controle da Configuração e o RPCPS, comenta e retorna ao Conselho de Controle da Configuração;
- 6 - O Conselho de Controle da Configuração aprova ou não uma determinada ação;
- 7 - Caso o Conselho de Controle da Configuração negue a aprovação este RPCPS é contabilizado;
- 8 - Caso o Conselho de Controle da Configuração aprove, é verificada a necessidade ou não da preparação de uma

PAE;

9 - Caso seja aprovada, é solicitada a preparação de uma PAE;

10 - Caso contrário, a alteração é implementada;

## ii) PAE - Proposta de Alteração de Engenharia

**Descrição :** Documento utilizado para se submeter uma proposta de alteração a uma "baseline". Pode ser utilizado também para relatar um problema encontrado pelo usuário após o sistema ser liberado e um pedido direto de alteração por parte do usuário.

### **Procedimento :**

1 - Um pedido de preparação de uma PAE surge decorrente de um RPCPS ou de um usuário;

2 - A Divisão de Projeto e Manutenção de Sistemas prepara uma PAE e envia ao responsável pela função de Controle da Configuração;

3 - Esta então, registra esta PAE e analisa-a para determinar a classe desta alteração;

4 - Caso a alteração seja classificada como Classe I e aprovada, então esta alteração é classificada como tal e enviada a Divisão de Projeto e Manutenção de Sistemas para ser implementada;

4.1 - Caso não seja aprovada é registrada, e enviado um relatório mensal para o Círculo da Garantia da

RELATÓRIO DE PROBLEMA E/OU CORREÇÃO DE PROBLEMA DE SISTEMA (RPCPS)		
DATA : _____		Nº DE CONTROLE : _____
A. DEFINIÇÃO DO PROBLEMA		
1. ORIGINADOR :	2. DATA :	3. SUBSISTEMA AFETADO :
4. TÍTULO DO PROBLEMA :		
5. DESCRIÇÃO DO PROBLEMA :		
HARDWARE : <input type="checkbox"/> SOFTWARE : <input type="checkbox"/> IC : _____      ICPC : _____ COMP : _____      CPC : _____ SUBCOMP : _____      MOD. : _____		
<input type="checkbox"/> DESCONHECIDO INTEGRAÇÃO/CONSTRUÇÃO      INTEGRAÇÃO/CONSTRUÇÃO      INTEGRAÇÃO/CONSTRUÇÃO OU Nº DA VERSÃO      OU Nº DA VERSÃO      OU Nº DA VERSÃO Nº DA PARTE COMP/SUBCOMP : _____		
6. CONFIGURAÇÃO :		
Nº DO DISCO : _____      REV. : _____      DATA : _____ Nº DA FITR : _____      REV. : _____      DATA : _____ Nº DA PARTE COMP/SUBCOMP : _____		
7. DOCUMENTAÇÃO :      AFETADA : _____      NÃO AFETADA : _____		
NOHE DO DOCUMENTO : _____ Nº DO DOCUMENTO : _____      GATA : _____		
B. DESIGNAÇÃO DA AÇÃO CORRETIVA		
1. DESIGNADO PARA :	2. DATA DE RESPOSTA	3. PRIORIDADE
		EMERGENCIA : <input type="checkbox"/> MANTER : <input type="checkbox"/> ROTINA : <input type="checkbox"/> DEFERIDA : <input type="checkbox"/> URGENTE : <input type="checkbox"/> DUPLICADA : <input type="checkbox"/>
4. APROVADO POR :		5. DATA
C. AÇÃO CORRETIVA		
1. ANÁLISE/SOLUÇÃO DO PROBLEMA (ANEXE FOLHAS EXTRAS SE NECESSÁRIO)		
NOTA: LISTE TODAS AS NOVAS VERSÕES, REVISÕES, ID. NÍDIA, ETC.		
2. ASSINATURA ENGENHEIRO/PROGRAMADOR		3. DATA :
4. APROVAÇÃO DA ORGANIZAÇÃO		5. DATA :

Figura III.2 - Relatório de Problema e/ou Correção de Problema de Sistema - parte 1

Fonte : VINCENT (1988)

D. VERIFICAÇÃO		
1. REQUISITOS DE TESTE (ANEXE FOLHAS EXTRAS SE NECESSARIO)		
2. RESULTADOS DE TESTE (ANEXE FOLHAS EXTRAS SE NECESSARIO)		
3. APROVADO <input type="checkbox"/>	ASSINATURA DE APROVAÇÃO :	3. DATA :
4. REJEITADO <input type="checkbox"/>	ASSINATURA DE REJEIÇÃO :	5. DATA :
E. GARANTIA DA QUALIDADE		
1. NOTAS : (ANEXE FOLHAS EXTRAS SE NECESSARIO)		
3. APROVADO <input type="checkbox"/>	ASSINATURA DE APROVAÇÃO :	3. DATA :
4. REJEITADO <input type="checkbox"/>	ASSINATURA DE REJEIÇÃO :	5. DATA :

Figura III.3 - Relatório de Problema e/ou Correção de  
Problema de Sistema - parte 2

Fonte : VINCENT (1988)

Qualidade Principal para comentários;

5 - Caso seja classificada como alteração Classe II, então;

5.1 - O Conselho de Controle da Configuração reve e comenta esta PAE e envia suas impressões para o Círculo da Garantia da Qualidade Principal para comentários;

5.2 - O Círculo da Garantia da Qualidade Principal revê, comenta e retorna ao Conselho de Controle da Configuração para aprovação ou não;

5.3 - Caso seja negada, a situação e esta PAE são registradas;

5.4 - Caso seja aprovada a situação e esta PAE são registradas e a Divisão de Projeto e Manutenção de Sistemas executa esta alteração..

### **iii) NAS - Notificação de Alteração de Software**

**Descrição :** Documento utilizado para documentar as alterações a serem feitas na documentação do software..

#### **Procedimento :**

1 - Após a aprovação da implementação de uma Proposta de Alteração de engenharia (PAE) ter sido testada e aprovada pelo Conselho de Controle da Configuração e o



Círculo da Garantia da Qualidade Principal, uma NAS é preparada pela Divisão de Projeto e Manutenção de Sistemas;

2 - A NAS é, então, inserida na especificação do software;

3 - Após a revisão da documentação ser feita, o responsável pela função de Contabilidade da Situação da Configuração registra e mantém as documentações antigas e novas para o acompanhamento da rastreabilidade do software.

#### iv) DDV - Documento de Descrição de Versão

**Descrição :** Documento que acompanha a liberação de um ICS. Deve conter uma relação com todos os materiais liberados, bibliografia dos documentos referenciados, descrição operacional, instruções de instalação e possíveis erros;

#### **Procedimento :**

1 - Ao final do processo PAE-NAS o Bibliotecário prepara um Documento de Descrição de Versão para acompanhar a liberação de uma versão de um ICS.

#### v) RRD - Relatório de Revisão de Documento

**Descrição :** Documento utilizado para apresentar os

comentários produzidos durante uma seção de "walkthrough", revisões de ciclo de vida e auditorias conduzidas pela Garantia da Qualidade de Software.

**Procedimento :**

- 1 - É preenchido durante uma seção de "walkthrough" conduzido pelo Círculo da Garantia da Qualidade Principal.

**vi) Formulário de Rastreamento de Documento**

**Descrição :** Documento utilizado para identificar todos os documentos afetados por uma deficiência a ser corrigida do sistema.

**Procedimento :**

- 1 - Resulta frequentemente após a resolução de um RPCPS-NAS.

**vii) FAE - Formulário de Alteração de Emergência**

**Descrição :** O FAE (remendo) é utilizado para registrar uma alteração de uma representação executável de um software sem tempo para uma aprovação formal. É uma solução temporária para problemas críticos, geralmente decorrente de uma RI urgente. Este documento pode ser suprimido ao se usar uma Proposta de Alteração de Engenharia com opção do tipo preliminar.

**Procedimento :**

- 1 - Informar ao usuário e ao gerente do projeto o impacto proveniente da alteração corretiva;
- 2 - Caso seja dado o consentimento, preencher uma PAE (ou um FAE) com opção preliminar e implementar a alteração;
- 3 - Depois preencher um Relatório de Problema e/ou Correção de Problema de Sistemas especificando o problema anterior para análise e confirmação da alteração pelo Conselho de Controle da Configuração e o Círculo da Garantia da Qualidade Principal.

#### **viii) RGCS - Relatório de Gerenciamento da Configuração de Software**

**Descrição :** Usado para solicitar ou colocar um ICS sob Gerenciamento da Configuração.

#### **Procedimento :**

- 1 - O Desenvolvedor solicita um item de configuração controlado;
- 1.1 - O Bibliotecário preenche uma ficha de liberação e registra os dados do solicitante, justificativa de liberação, data e outras informações que forem necessárias.

2 - O desenvolvedor solicita através deste formulário que um ICS seja colocado sob o gerenciamento da configuração de software;

2.1 - O Bibliotecário então preenche o documento com os dados relativos ao registro do ICS.

#### - Convenções de nomes para módulos e programas

Os nomes dos programas deverão conter oito caracteres, sendo que os dois últimos identificarão a versão e revisão do programa.

Por versão entende-se o incremento das características do programa, enquanto que revisão será apenas devido a uma correção.

#### - Identificação de ambiente

Cada "baseline" terá sua representação em palavras sobre papel e em disquetes separados, usando as convenções de rótulo especificadas na seção 3.1 deste plano.

#### - Diretrizes de Controle

- i) As decisões do Conselho de Controle da Configuração serão através do voto, contudo o Presidente do Conselho de Controle da

Configuração terá a palavra final, não importando qualquer posição contrária;

ii) É de responsabilidade do Conselho de Controle da Configuração do contratante em conjunto com o desenvolvedor a aprovação ou rejeição de Alterações Classe I;

iii) É de responsabilidade da função de Gerenciamento da Configuração aprovar ou rejeitar Alterações Classe II. Entretanto, um relatório mensal com estas alterações deve ser apresentado ao Conselho de Controle da Configuração e Garantia da Qualidade.

### Seção 3 - ATIVIDADES DO GERENCIAMENTO DA CONFIGURAÇÃO DE SOFTWARE

#### Seção 3.1 - Identificação da configuração

O número de "baselines" dependerá principalmente do tipo de projeto que se quer controlar. É sugerido o estabelecimento de cinco "baselines" que coincidirão com o final das fases do ciclo de vida do sistema, são elas: Funcional, Alocada, de Projeto, de Produto e Operacional.

Abaixo estão relacionadas as "baselines" com os estágios de ciclo de vida correspondentes, as revisões e auditorias necessárias ao estabelecimento destas "baselines".

A Baseline Funcional é um documento com a concordância entre o desenvolvedor (Contratado) e o usuário (Contratante) sobre quais as funções o sistema irá apresentar. Geralmente é uma Especificação dos Requisitos da Sistema.

ESTÁGIOS	BASELINES	REVISÕES
Estudo	Funcional	Revisão Conceitual de Projeto inicial
Projeto	Alocada	Revisão de Requisitos de Software
Desenvolvimento	Projeto	Revisão de Projeto Preliminar
		Revisão de Projeto Crítica
	Produto	Revisão de Desenvolvimento Final e Auditoria Funcional
Operação	Operacional	Revisão de pós-operação e Auditoria Física

A Baseline Alocada consta da Especificação dos Requisitos de Software, o PGCS, o Manual do Usuário e o Plano de Testes..

A Baseline de Projeto tem como produto a Especificação do Projeto de Arquitetura do Sistema, com a estrutura de árvore de módulos, e a especificação detalhada de cada módulo..

A Baseline de Produto é formada pela implementação e teste integrado e aceitação do Sistema.. Consta de várias representações (códigos fonte, objeto e executável) do sistema e do relatório de testes..

A Baseline Operacional é formada pelo produto de software na sua fase aprimorada, já operacional e a implementação de tabu a documentação relacionada ao produto..

- Identificação das "Baselines" e dos Itens da Configuração de Software (ICS)(ICPC) que as compõem.

São apresentados dois esquemas de identificação :

i) Este esquema utiliza uma notação matemática e é mais formal. A notação hierárquica decimal é utilizada para indicar os pais e filhos de um ICS(ICPC). Por exemplo, o ICS1.1 é pai dos ICS1.1.1 e ICS1.1.2, mas é filho do ICS1.

As Baselines são identificadas segundo o esquema abaixo:

$$R \begin{matrix} [B & ] \\ m & ij \end{matrix}^v$$

Onde B = Baseline, i estabelece o tipo de "baseline", j esta o número da revisão da "baseline" e R como serão as suas representações.

i = 1(funcional), 2(alocada), 3(projeto),  
4(prodoto) e 5(operacional)

j = 0,1,2,... atualizações da "baseline"

R = Representação da "baseline", m = 1(Palavras

sobre páginas), 2(armazenamento magnético), 3(listagem fonte), 4(executável)

A seguir é apresentado um exemplo que retrata este esquema de identificação associando-o aos itens relacionados a cada "baseline":



BASELINE	REPRESENTAÇÃO	IDENTIFICAÇÃO
funcional	Especificação	$[R(B)]_1^{10}$
funcional revista	Especificação	$[R(B)]_1^{11}$
alocada	Especificação	$[R(B)]_1^{20}$
:	:	:
:	:	:
operacional	Especificação, fonte, objeto e executável	$[R(B)]_m^{40}$ $m = 1, 2, 3, 4$

ii) A figura (III.4) apresenta o segundo esquema de identificação. Neste caso, tem-se um software (SINGEV) sendo desenvolvido pela CNEN. As "baselines" funcional e alocada já estão sob o gerenciamento da configuração de software e são constituídas por um e dois ICSs (ICPCs) respectivamente.

O ICS da "baseline" funcional (ER1, especificação de requisitos 1) passou para o controle formal do Gerenciamento da Configuração de Software em 3 de outubro de 1983, tendo assim, o seu número de revisão igual a 031083.

Os dois últimos ICSs/ICPCs (ICPC (4) e ICPC(5)) já estão previstos para a "baseline" de projeto, embora aguardando o seu estabelecimento formal pelo Conselho de Controle da Configuração

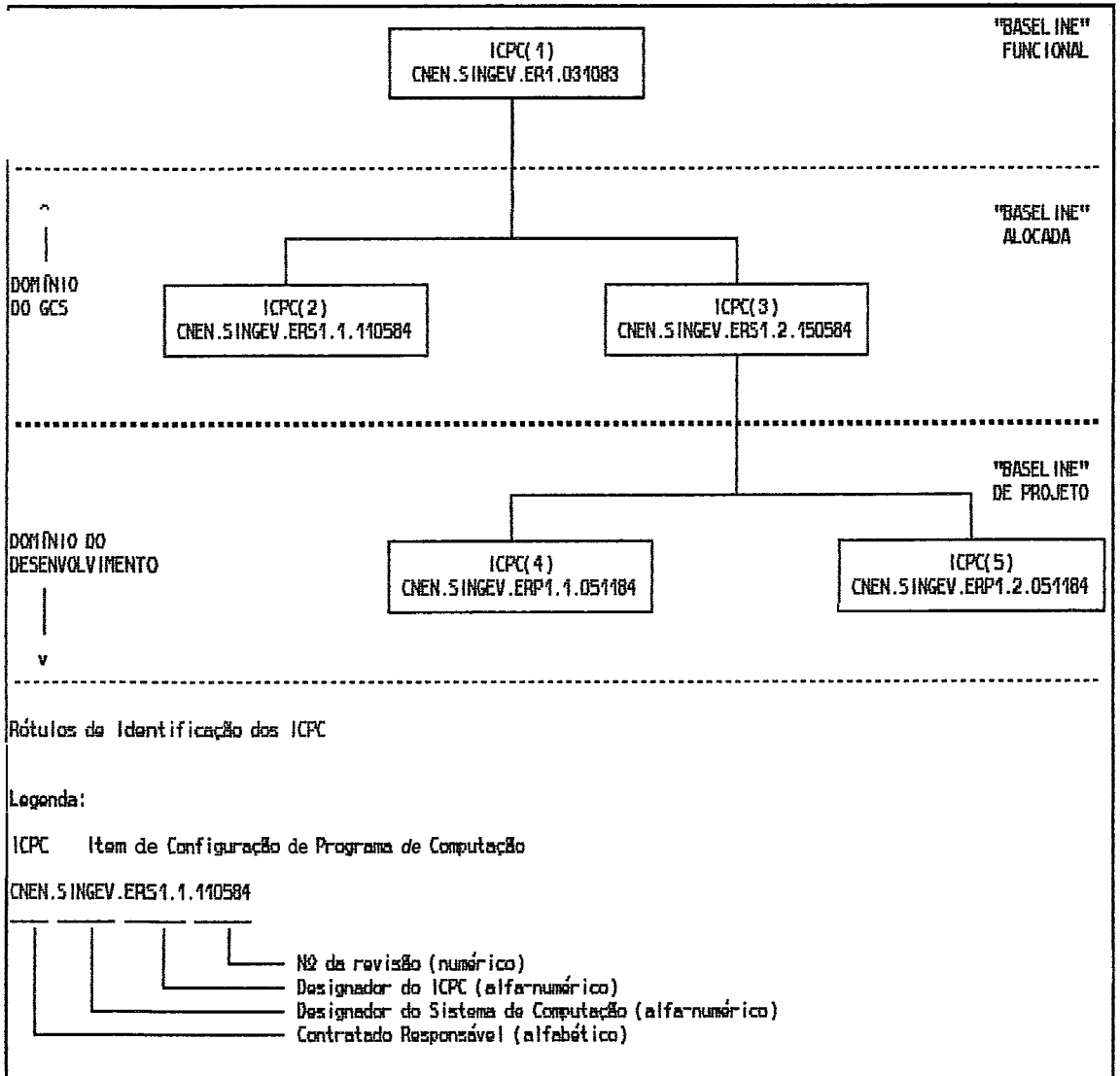


Figura III.4 - Exemplo de um esquema de identificação

### Seção 3.2 - Controle da Configuração

A seguir são sugeridos pontos a serem considerados quando da aplicação do Controle da Configuração em um projeto de software :

- i) as políticas e os procedimentos de Controle da Configuração, devem ser estabelecidos pelo o Contratante e o Desenvolvedor antes do início do projeto;
- ii) o Controle da Configuração, bem como o Conselho de Controle da Configuração devem começar no início do projeto;
- iii) o nível de controle sobre um projeto deve ser proporcional ao tamanho do projeto, à sua complexidade, ao orçamento e ao cronograma.. Deve-se evitar que o controle torne-se um entrave ao fluxo normal do desenvolvimento do projeto;
- vi) um Plano de Gerenciamento da Configuração de Software tem que ser estabelecido;
- v) cabe ao desenvolvedor insistir com o Contratante

da necessidade do envolvimento entre eles no processo de revisão, o qual será feito através do Conselho de Controle da Configuração;

vi) cabe ao Contratante impor o envolvimento com o Desenvolvedor no processo de revisão, o qual será feito através do Conselho de Controle da Configuração.

#### - Conselho de Controle da Configuração

Conselho de Controle da Configuração será o ponto focal do Controle da Configuração e é constituído para resolver questões não resolvidas nos níveis mais baixos.

O Conselho de Controle da Configuração deve ser visto como um fórum multi-disciplinar, através do qual o gerenciamento pode monitorar e controlar o projeto e a implementação do sistema. Deve ter representantes da gerência, dos usuários, do contratado e da Gerência da configuração.

A figura (III.5) mostra as várias variantes possíveis na formação de um Conselho de Controle da Configuração com a participação ou não de representantes dos usuários, da organização contratante e contratado.

i) o Bloco A é formado de representantes dos usuários com o poder de veto sobre todos os

outros Conselhos de Controle da Configuração;

ii) caso o usuários formem um Conselho de Controle da Configuração (ou seja, existe o Bloco A) a

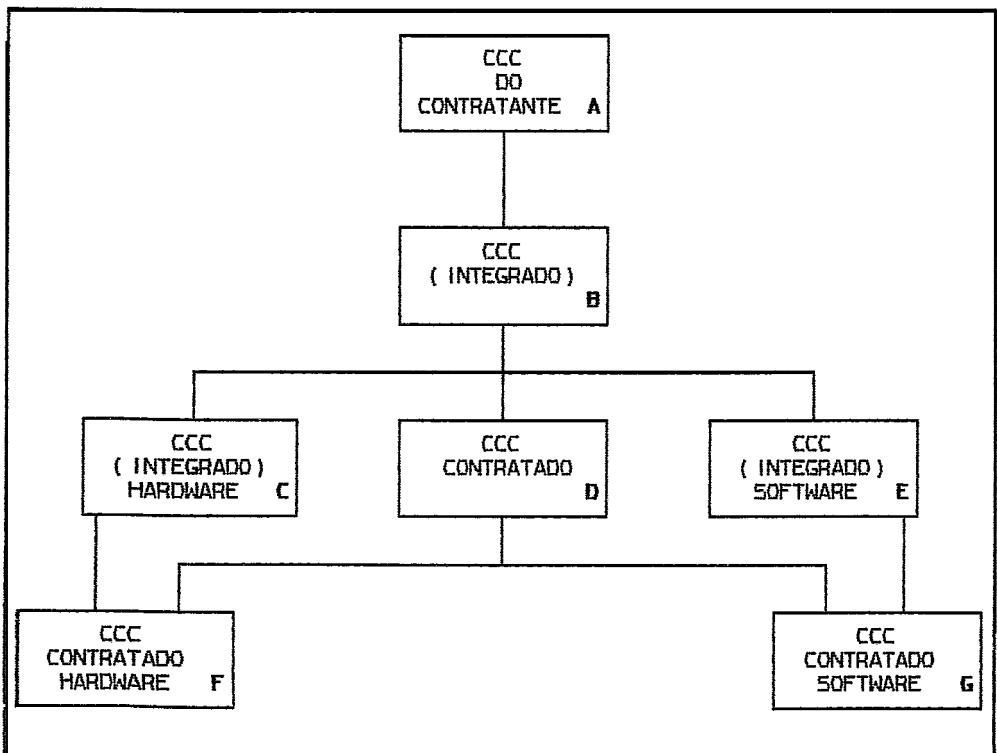


Figura III.5 - Variações possíveis na formação do Conselho de Controle da Configuração

Fonte : BERSOFF (1980)

organização do Contratado tem que estabelecer um Conselho de Controle da Configuração integrado, com representantes dos usuários, do contratante e do contratado (Bloco B);

iii) caso o Bloco B seja estabelecido o Conselho de Controle da Configuração integrado e desmembrado em dois Conselho de Controle da Configuração: os Conselhos de Controle da Configuração de Software e de Hardware (CCCS e CCCH) (Blocos C e D);

iv) caso o Bloco B não seja estabelecido o Conselho de Controle da Configuração do contratado estabelece os seus CCCS e CCCH e então, deve solicitar a presença de consultores externos para poder captar as perspectivas dos usuários (Blocos D, F, G);

v) É uma boa política para o contratado a criação dos seus próprios Conselho de Controle da Configurações para desempenharem o controle interno de todos os projetos de responsabilidade do Contratado.

Sugere-se a seguinte formação para os Conselhos de Controle da Configuração do contratante e do contratado :

## CONTRATANTE

## CONTRATADO

.. Gerente do Projeto	.. Gerente do Projeto
.. Engenheiro de Sistemas	.. Engenheiro de Sistemas e/ou Hardware
.. Engenheiro de Software	.. Engenheiro de Software
.. Gerente da Configuração de Hardware	.. Gerente da Configuração de Hardware
.. Representante da Garantia da Qualidade de Software	.. Representante da Garantia da Qualidade de Software
.. Representante dos Usuários	

### Seção 3.3 - Contabilidade da Situação da Configuração

Deverão ser contabilizados :

- i) todos os encontros com o Conselho de Controle da Configuração do contratato e do contratante;
- ii) o histórico de cada ICS, contendo sua identificação, data em que foi posto sob Gerenciamento da Configuração de Software, função e hierarquia (seus pai e filhos);

iii) a situação de todas as especificações;

vi) a situação de todos os documentos de controle;

v) todos os dados referentes ao estabelecimento de uma "baseline" (documentos que a compõe, data do seu estabelecimento, das revisões e auditorias).

#### Seção 3.4 - Auditorias e Revisões

Tem como responsabilidades :

- i) Verificar que cada ICS na "Baseline" é logicamente relacionado com a especificação do mesmo ICS nas "baselines" anteriores;.pm 8
- ii) Validar que os requisitos de software/sistema são satisfeitos através da configuração de ICSs especificados na "Baseline".

As deficiências encontradas terão que ser relatadas ao Conselho de Controle da Configuração através de um relatório de auditoria, o qual, identificará os ICSs que apresentaram problemas.

Serão objetos de auditoria os produtos relacionados pela gerência do projeto como essenciais, no mínimo as



especificações de requisitos do sistema, do software, da base de dados, de projeto, entre outras.

Estas auditorias devem ser feitas durante a evolução de cada uma das "baselines" para se evitar alguma surpresa na hora da aprovação formal de uma "baseline".

Quando o produto estiver para ser liberado serão necessárias duas auditorias: funcional e física.

A auditoria funcional terá como objetivo verificar se cada requisito presente na "baseline" alocada está sendo satisfeito. A auditoria física terá como objetivo assegurar que todos os elementos de software, hardware e operacionais estão completos.

#### Seção 4 - FERRAMENTAS, TÉCNICAS E METODOLOGIAS

##### - Ferramentas de Identificação

- . Gerador de árvore de ICSS

- . Rotulador de ICSS

##### - Controle da Código

Para controlar o relacionamento desde o código fonte ao executável, serão utilizadas as seguintes convenções:

###### i) Fonte ----- Objeto

Manter o mesmo nome, mais o número da versão/revisão.

## ii) Objeto----- Executável

Usar o Mapa de Carga para identificar os códigos objetos que foram linkeditados para formar o código executável.

## iii) Mapa de Carga ----- Executável

Usar um registro cabeçalho. Esta identificação será impressa na parte superior do Mapa de Carga e terá que ser feita em tempo de linkedição.

Caso um módulo "Stub" seja utilizado, o nome do código objeto será o mesmo do código fonte, mais um sufixo (ST)

## iv) Fonte ----- Cópia

Manter uma biblioteca com todos os códigos fontes e objetos anteriores e a última forma do programa executável.

## - Ferramentas de Controle

Um exemplo de documento de controle da configuração de software é apresentado nas figuras (III.5) a (III.6). Um sistema automatizado de geração de documentos para o gerenciamento da configuração é de grande utilidade ao Gerente da configuração.

## - Ferramentas de Auditoria

As ferramentas abaixo são utilizadas pelos auditores

no seu trabalho nas várias fases do desenvolvimento do software..

- i) Analizadores lógicos;
- ii) Fluxogramas automáticos;
- iii) Simuladores..

#### **- Ferramentas para Contabilidade da Situação**

A função da Contabilidade da Situação deve registrar para apresentar a gerência do projeto os seguintes documentos :

- i) sumários da gerência da configuração;
- ii) minutas dos encontros do Conselho de Controle da Configuração e do Círculo da Garantia da Qualidade Principal;
- iii) notas liberadas;
- iv) diário de Itens da Configuração;
- v) Índice de Itens da Configuração;
- vi) diário de Propostas de Alteração de Engenharia;

PROPOSTA DE ALTERAÇÃO DE ENGENHARIA PAE		<input type="checkbox"/> REGISTRO DO PROBLEMA	<input type="checkbox"/> PAE
1. PARA :		2. DE :	
3. Nº DO ORIGINADOR	4. CONTATO (NOME E TEL)	5. PRIORIDADE DA PAE <input type="checkbox"/> EMERGENCIA <input type="checkbox"/> URGENTE <input type="checkbox"/> ROTINA	
6. APLICAÇÃO : BASELINE/VERSÃO	7. EXECUTIVA BASELINE/VERSÃO	8. DATA DO PROBLEMA	
9. JOB/CICLO/ID.DO PROGRAMAR			
10. TÍTULO DO PROBLEMA/ALTERAÇÃO			
11. DESCRIÇÃO DO PROBLEMA/ALTERAÇÃO   LISTAR TODOS OS ANEXOS E DOCUMENTOS REFEREM - CIADOS) ( SE FOR PRECISO MAIS ESPAÇO, USE O ITEM 15 )			
12. EFEITOS SOBRE O USUÁRIO ( SE FOR PRECISO MAIS ESPAÇO, USE O ITEM 15 )			
13. SOLUÇÃO/JUSTIFICATIVA RECOMENDADAS ( SE FOR PRECISO MAIS ESPAÇO USE O ITEM 15 )			
14. DATA	NOME E TÍTULO DA AUTORIDADE	ASSINATURA :	

Figura III.5 - Proposta de Alteração de Engenharia -  
parte 1

Fonte - VINCENT (1988)

15. OBSERVAÇÕES (SE FOR PRECISO MAIS ESPRO USAR UMA FOLHA DE PAPEL SEPARADA)		
USUÁRIO		AÇÃO
16. USUÁRIO <input type="checkbox"/> APROVA <input type="checkbox"/> REJEITA (PODE INCLUIR QUALQUER COMENTARIO)		
17. DATA	18. NOME E TITULO	19. ASSINATURA
AGÊNCIA RESPONSÁVEL DESIGNADA (somente o registro do problema)		
18. AÇÃO TOMADA		
<input type="checkbox"/> RESOLVIDO PELO SUPORTE AO USUÁRIO <input type="checkbox"/> PAE DUPLICADA : Nº _____ <input type="checkbox"/> IDENTIFICADA COMO URGENTE OU ROTINEIRA <input type="checkbox"/> CANCELADA PELO ORIGINADOR <input type="checkbox"/> FORMALIZADA UMA PAE DE EMERGENCIA <input type="checkbox"/> CANCELADA POR DOCUMENTAÇÃO INSUFICIENTE <input type="checkbox"/> CANCELADA POR IDENTIFICAÇÃO INSUFICIENTE		
19. DATA	E TITULO	ASSINATURA
AGÊNCIA PROPONENTE		
20. CLASSE DA PAE <input type="checkbox"/> I <input type="checkbox"/> II	21. CÓDIGO DE JUSTIFICAÇÃO	22. Nº DA PAE
23. TIPO DA PAE <input type="checkbox"/> PRELIMINAR <input type="checkbox"/> FORMAL	24. CUSTOS/ECONOMIAS ESTIMADAS	
25. OUTROS SISTEMAS AFETADOS		
26. IDENTIFICAÇÃO DA ALTERAÇÃO (MARQUE UM EM CADA COLUMNA)		
<input type="checkbox"/> FUNCIONAL/ALOCADA <input type="checkbox"/> MAIOR <input type="checkbox"/> MANUTENÇÃO <input type="checkbox"/> TECNICA/PRODUTO <input type="checkbox"/> MENOR <input type="checkbox"/> MODIFICAÇÃO		
27. IMPLEMENTAÇÃO PROJETADA		
28. AUTORIZAÇÃO DE APROVAÇÃO (MARQUE A AGENCIA E R AÇÃO)		
<input type="checkbox"/> AGENCIA PROPONENTE <input type="checkbox"/> AGENCIA RESPONSÁVEL <input type="checkbox"/> APROVADO <input type="checkbox"/> REJEITADO		
29. DATA	NOHE E TITULO	ASSINATURA

Figura III.6 - Proposta de Alteração de Engenharia  
parte 2

vii) diário de Relatório de Problema e/ou Correção de Problema de Sistema.

Exemplos de documentos utilizados pela a Contabilidade da situação são apresentados a seguir, nas figuras (III.7) e (III.8):

#### Seção 5 - CONTROLE DE FORNECEDOR(CONTRATADO)

A contratação de serviços de terceiros apresenta um alto grau de insatisfação por parte dos contratantes.

Tudo isso, pode ser evitado se for observado o seguinte :

- i) avaliar as condições do fornecedor de satisfazer : requisitos do projeto e de Gerenciamento da Configuração;
- ii) verificar se foi descrito adequadamente um método para assegurar a concordância entre o fornecedor e o contratante;
- iii) impor ao fornecedor a formação de um Conselho de Controle da Configuração;

Data de Registro : _____
Componente : _____
Propósito : _____
Natureza ( ICPC/CPC ) : _____
Baseline : _____
Número Básico : _____
Número do Volume : _____
Número da Revisão : _____
Identificação da alteração : _____
Data da Compilação : _____
Data da Aprovação : _____
Comentários : _____
_____
_____
_____
_____

Figura III.8 - Fôlha do diário de Itens de  
Configuração de Software (ICS)

Fonte - VINCENT (1988)

Registro de desenvolvimento de ICPC - seção 1	
Número de ICPC e Nomenclatura	
N. da especificação em Desenvolvimento	
Data de Emissão	
Data de Aprovação	
Horário da Revisão de Projeto Conceptual	Horário da Revisão de Requisitos de Software
início	Início
Término	Término
Horário da Revisão de Projeto Preliminar	Horário da Revisão de Projeto Crítico
Início	Início
Término	Término
Horário da Revisão de Desenvolvimento Final	Horário de Auditoria Funcional de Configuração
Início	Início
Término	Término
Horário de Auditoria Física de Configuração	Horário de Revisão de Pós-Operação
Documentos de teste de Qualificação	

Figura III.9 - Fôlha do índice de item de Configuração de Software

Fonte - VINCENT (1988)



- iv) verificar se os requisitos de Gerenciamento da Configuração adotados pelo fornecedor satisfazem aos requisitos estabelecidos pelo contratante (Identificação, Controle e Contabilidade da Situação);
- v) ajustar os requisitos de Gerenciamento da Configuração à cada fornecedor, levando em consideração a complexidade e importância do item:
- vi) validar os produtos do fornecedor (satisfação dos requisitos, desempenho técnico) através de revisões de projeto e auditorias;
- vii) preparar os requisitos de Gerenciamento da Configuração para controlar o fornecedor;
- viii) aprender, sempre, através dos procedimentos anteriores aplicados no controle do fornecedor;
- ix) indicar como será realizada a integração com os dados técnicos do fornecedor.

As ações abaixo foram consideradas as mais importantes na interação entre o contratante e o fornecedor/contratado

- i) especificar a documentação exata requerida nos contratos;

- ii) designar quem será autorizado a decidir sobre questões que aparecerem durante o desenvolvimento do projeto;
- iii) trabalhar mais estreitamente até definir claramente a linguagem final do contrato;
- iv) manter o mesmo grupo do início ao final do projeto.

#### Seção 6 - RETENÇÃO E COLEÇÃO DE REGISTROS

Relação mínima dos documentos a serem retidos :

- i) detalhes dos Itens de Configuração de Software;
- ii) sumários de revisões de "baselines";
- iii) deficiências nas "baselines" identificadas;
- iv) alterações nas "baselines";
- v) notas do Conselho de Controle da Configuração;
- vi) propostas de alteração;
- vii) relatórios de incidentes de software;

viii) sumários de auditoria.

Em caso de dúvida sobre registrar ou não um item, registre-o.

Dependendo do tamanho e complexidade dos dados a serem armazenados, é necessário que seja estabelecida uma Base de Dados completa e organizada. Esta Sistema de Suporte de Base de Dados tem que ter as seguintes características :

i) Rastreabilidade/recuperabilidade

Qualquer elemento (ICS ou Baseline) sob controle pode ser rastreado ou recuperado para qualquer versão anterior;

ii) Segurança

O acesso ao material controlado só é possível através de autorização ou pelo nível de privilégio do usuário;

iii) Liberação

A biblioteca tem que ter um mecanismo de distribuição padrão para a entrega de material (fonte, executável) ao usuário;

iv) Automação

O Sistema de Suporte de Base de Dados deve ter um alto grau de automação, minimizando ao máximo a intervenção do

responsável pela operação nas tarefas de manutenção do sistema.

v) Consistência

O sistema tem que assegurar a consistência dos códigos e dos dados;

vi) Estatísticas

Emitir vários relatórios com estatísticas dos dados armazenados.

### 3.3 NOTAS FINAIS

Os procedimentos, recomendações, estruturas organizacionais propostos e descritos acima visam ajudar na preparação de um Plano de Gerenciamento da Configuração de Software (PGCS) que atenda especificamente a um sistema/software em desenvolvimento ou já desenvolvido.

No entanto, esta proposta, pode ser modificada de acordo com as necessidades do projeto em questão, seções podem ser incluídas, como por exemplo, para o controle de dados confidenciais e novos tópicos podem ser ressaltados ou adaptados segundo a experiência adquirida na aplicação de outros PGCS.

## CAPÍTULO IV

SISTEMA AUTOMATIZADO PARA GERENCIAMENTO DA CONFIGURAÇÃO  
DE SOFTWARE

## 4.1 INTRODUÇÃO

Apesar de ter o consenso geral da comunidade de Sistemas de Computação quanto a importância de sua aplicação na obtenção de um produto de software com qualidade, o Gerenciamento da Configuração de Software (GCS) tem sido relegado a um segundo plano, ou mesmo, não tem sido empregado nos projetos de sistemas.

A justificativa mais frequente de não se empregar o Gerenciamento da Configuração de Software (GCS), está no grande esforço manual requerido no levantamento e classificação dos dados dos projetos, e na falta de ferramentas automatizadas adequadas à realização desta tarefa.

As poucas ferramentas automatizadas de GCS no mercado, geralmente, são para ambientes de grande porte e requerem um grande esforço por parte do usuário na sua utilização.

Com o incremento cada vez maior de utilização de microcomputadores, constatamos a necessidade de difundir os conceitos de Gerenciamento da Configuração de Software e oferecer ao usuário de microcomputador uma ferramenta automatizada, que fosse ao mesmo tempo poderosa e de fácil utilização.

#### 4.2 LIMITAÇÕES DO SISTEMA ATUAL

Atualmente o processo de gerenciamento da configuração é um processo altamente manual, que envolve coletar cada item de configuração em uma biblioteca central, indexar a biblioteca e gerenciar os procedimentos para o desenvolvimento e alteração do sistema.

Devido à utilização dos computadores, o número de revisões e versões das informações manipuladas é cada vez maior. A taxa em que estas informações são produzidas é bem maior do que é possível documentá-las manualmente.

Este atraso entre as informações geradas e suas documentações correspondentes, impossibilita o acompanhamento real do desenvolvimento de um sistema.

Abaixo estão listadas as principais desvantagens da utilização de um sistema de gerenciamento da configuração de software manual :

- i) dificuldade de manipular o grande volume de dados produzidos durante o desenvolvimento do sistema;
- ii) dificuldade de gerar relatórios com referências cruzadas das informações retidas pelo sistema;
- iii) imprecisão no fornecimento da configuração corrente devido a defasagem da documentação e as alterações em andamento;
- iv) a rastreabilidade de um item de configuração é penosa e demorada;
- v) as informações geralmente estão dispersas, dificultando o seu gerenciamento.

#### 4.3 DESCRIÇÃO GERAL

O Sistema Automatizado para Gerenciamento da Configuração de Software (SAGCS) proposto neste trabalho, está baseado, principalmente, na ferramenta da Softool Corporation " Change and Configuration Control - CCC/DM Turnkey " (LOBBA , 1985) e nas características dos gerenciadores de código fonte apresentadas por VALLINO (1987).

Configuração de Software engloba as seguintes funções do gerenciamento da configuração de software :

- i) a identificação de todos os componentes (itens) de uma aplicação e seus interrelacionamentos. Estes componentes podem ser qualquer tipo de informação legível por máquina;
- ii) o controle da configuração, restringe o acesso às informações, especificando quais alterações podem ser feitas e por quem;
- iii) a contabilidade da situação que fornece através de uma grande variedade de relatórios, as informações necessárias ao acompanhamento do sistema.

O SAGCS gerencia quatro ambientes (configurações) distintos: configuração de desenvolvimento, configuração aprovada, configuração de produção e configuração formal. Uma configuração base de trabalho é definida no início, segundo as características de cada projeto.

Esta configuração base será comum para todas as outras configurações durante o ciclo de vida do projeto. Os ambientes (configurações) podem ser divididos, então, em níveis, os quais podem ser visualizados como subdiretórios.



- ii) a configuração formal é onde está registrada a evolução do sistema. Uma vez que uma revisão do sistema (sua configuração), foi colocada sob este ambiente, ou seja posta sob o gerenciamento da configuração de software, não pode mais ser retirada;
- ii) configuração de desenvolvimento, é nesta configuração que os itens que compõem a aplicação são retirados e inseridos para serem testados e alterados;
- iii) após passarem pelos testes e serem validadas as alterações presentes na configuração de desenvolvimento são transferidas para a configuração aprovada;
- iv) A configuração de produção é utilizada tanto pela equipe de produção quanto para manutenções de emergência quando o ciclo normal de desenvolvimento/ aprovação é por demais lento. Entretanto, caso as alterações tenham que ser incorporadas na próxima revisão do sistema, estas terão que seguir o caminho formal de controle.

O Sistema Automatizado para Gerenciamento da Configuração de Software (SAGCS) define os níveis fonte, objeto e executável. Em cada um desses níveis são colocados os itens (arquivos) que compõem a aplicação.

A evolução do sistema pode ser visualizada na figura (IV.1).

O sistema mantém apenas as revisões da configuração formal e as três configurações correntes. O estabelecimento de uma nova revisão da configuração formal ocorre no momento em que o gerente do projeto decide que as alterações presentes na configuração aprovada podem servir como um novo ponto de referência no ciclo de vida do sistema. Neste momento, o conteúdo da configuração formal passa a ser identificado como a nova revisão da configuração formal. As três configurações: desenvolvimento, aprovada e produto são destruídas e são criadas outras três com o mesmo conteúdo da configuração formal anteriormente estabelecida.

O SAGCS baseia todo o seu controle na associação dos pedidos de alteração aos itens armazenados em sua base de dados. Para que um item possa ser retirado do SAGCS é necessário que o item já tenha sido associado a um pedido de alteração. Esta associação é feita pelo bibliotecário ou qualquer outra pessoa designada para esta tarefa.

Os documentos de alteração têm um ciclo de vida bem definido, composto de cinco estágios: situação designada,

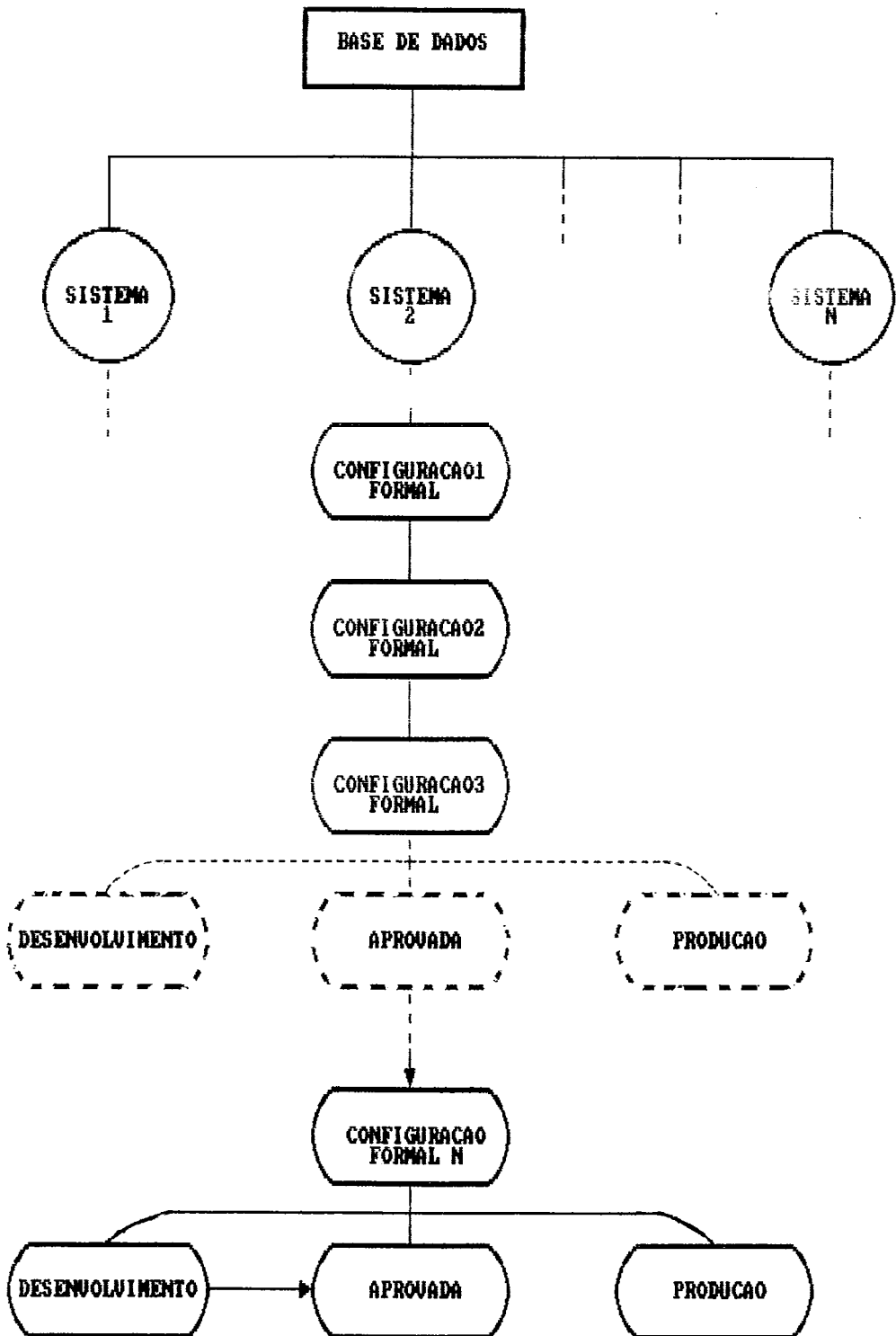


Figura IV.1 - Evolução das configurações no sistema SAGCS

situação de desenvolvimento, situação de teste, situação aprovada e situação rejeitada.

A primeira situação é a designada, quando um documento de alteração é associado a um ítem. A partir do momento que o ítem é retirado da configuração de desenvolvimento para que seja alterado, a situação do documento passa para desenvolvimento.

Quando o ítem reitorria ao controle do IRGCS, a situação do documento passa para teste, a partir daí para aprovada ou rejeitada após a aprovação ou não do gerente do projeto.

Caso seja aprovada, todas as alterações feitas sobre os ítems relacionados ao documento são copiadas para a configuração aprovada. Caso contrário, a alteração foi rejeitada, as modificações dos ítems na configuração de desenvolvimento relacionadas ao documento são destruídas e os ítems voltam a ser como eram anteriormente.

O diagrama de transição de estados, a seguir, mostra as várias situações que um documento de alteração pode assumir (figura IV.2).

Um ítem não pode ser retirado, novamente, da base de dados do SAGCS enquanto a situação do documento associado ao ítem não for igual a aprovada ou rejeitada, visto que o SAGCS não permite desenvolvimento paralelo (ramos) de um

item, possuindo apenas um caminho de desenvolvimento (tronco). Para tal, o item tem que voltar ao controle do

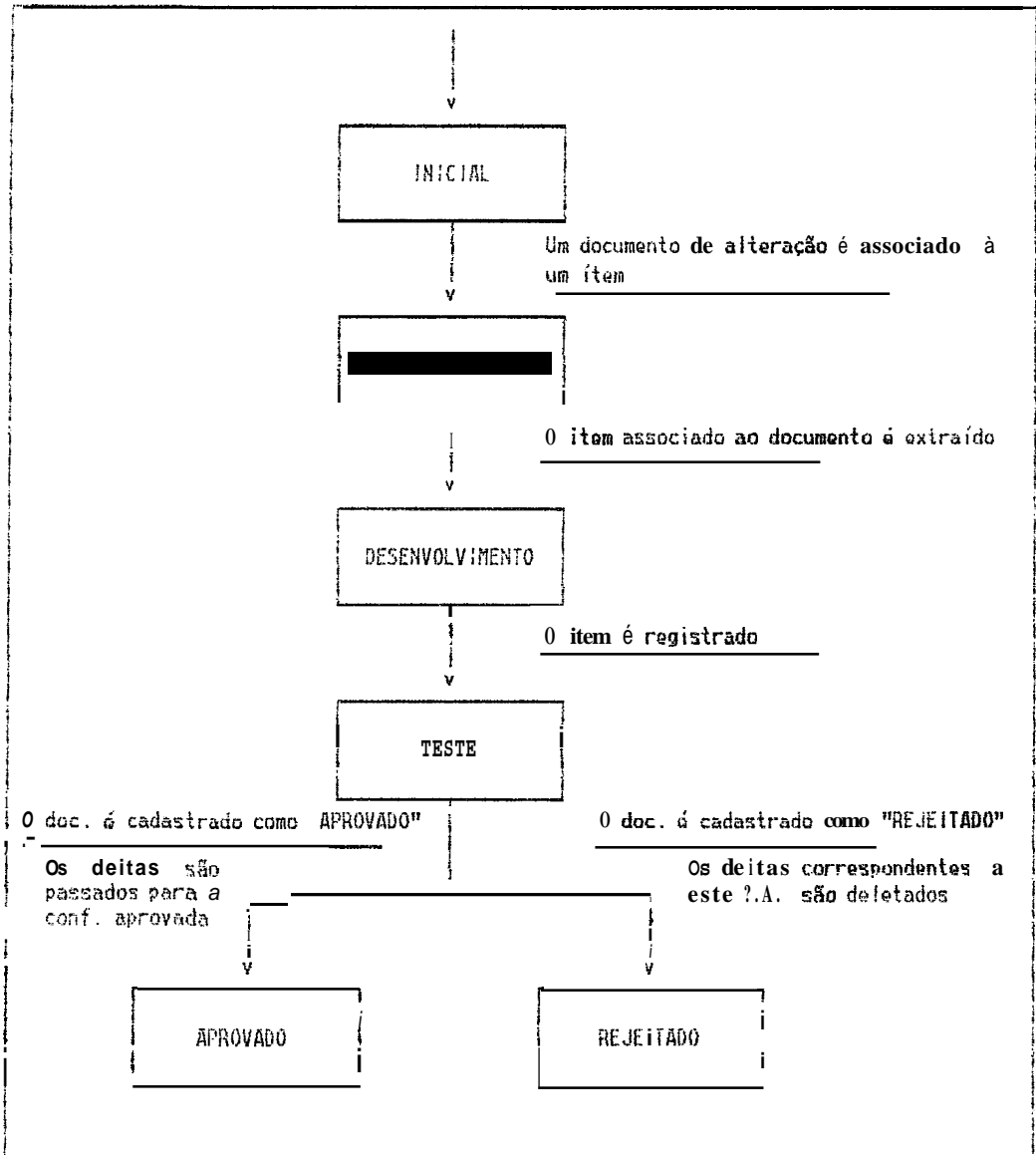


Figura IV.2 - Diagrama de transição de estados correspondendo a situação dos pedidos de alteração

SAGCS, por meio do mesmo usuário que o retirou e associado ao mesmo pedido de alteração, só assim, podendo ser retirado novamente.

#### **4.3.1 INTERFACE COM OUTROS PRODUTOS E PROJETOS**

Em conjunto com o Sistema Automatizado para Gerenciamento da Configuração de Software são sugeridos os seguintes produtos :

##### **i) Relator de problemas**

Este tipo de produto permite a confecção de formulários de controle para que os usuários possam entrar com as informações diretamente no sistema. Os tipos de formulários são especificados no plano de gerenciamento da configuração de software.

##### **ii) Gerador automático de programas**

Automatiza o processo de geração da forma executável. Esta ferramenta, já disponível em alguns ambientes (por exemplo, MAKE no UNIX), deve garantir que os códigos objeto utilizados na construção de um sistema foram gerados a partir das versões corretas de todos os arquivos fonte. No futuro estas ferramentas devem incorporar técnicas de inteligência artificial para gerenciar inconsistências ao invés de eliminá-las.

### iii) Documento de descrição de versão

Permite ao usuário especificar a exata combinação dos componentes do sistema que serão usados para se gerar um relatório detalhado a respeito de um sistema, versão ou item.

### iv) Referência cruzada

Através das informações fornecidas pelo usuário, identifica as partes do sistema e documentos afetados por uma modificação proposta.

## 4.3.2 REQUISITOS DE INFORMAÇÃO

Os requisitos de informação do SAGCS foram estipulados levando-se em conta o ambiente formal de gerenciamento da configuração de software, onde os documentos de controle e sua utilização são definidos no Plano de Gerenciamento da Configuração de Software para serem os veículos de registro de todos os fatos relevantes que ocorrerem durante o desenvolvimento do sistema.

As várias formas e representações dos itens que compõem cada "baseline" se constituem na principal ferramenta de acompanhamento e rastreabilidade da evolução do sistema:

- i) documentos de controle de alteração, tais como, Proposta de Alteração de Engenharia, Relatório de Problema e/ou Correção de Problema de Sistema e outros, com identificação, data, justificativa e autor;
- ii) definição da estrutura de identificação dos itens em conjunto com a data, o tipo de operação (Registro/Extração) , justificativa para a operação e usuário;
- iii) Relatório de Alteração Detalhado, cujo propósito é descrever detalhadamente as modificações feitas sobre a versão de um item;
- iv) Relatório de Interdependência de Item, cujo propósito é apresentar as interdependências de um item, mostrando os relacionamentos na forma FONTE para OBJETO e este para EXECUTÁVEL e, também, como uma alteração irá afetar outros itens;
- v) Relatório de História do Projeto, cujo propósito é guardar toda a história do projeto através de um relato completo de todos documentos de alteração associados aos itens e suas situações,



data do estabelecimento das configurações, data das operações sobre os itens, etc;

vi) Relatório de Alteração, cujo propósito é conter um sumário de todas as alterações feitas aos itens de um determinado nível;

vii) Relatório de Situação dos Documentos de Alteração, cujo propósito é conter a lista da situação corrente de todos os documentos de alteração;

viii) Relatório de História do Item, cujo é mostrar todas as situações (bloqueado, desbloqueado ou extraído) que um item teve durante o ciclo de vida do projeto.

A situação normal de um item de configuração é a situação bloqueada, ou seja, quando o item está sob gerenciamento da configuração de software. A partir do momento que um documento de alteração, por exemplo, uma Proposta de Alteração de Engenharia, é associado a um item, a situação desse item passa para desbloqueada. Quando o item é solicitado da configuração de desenvolvimento para alteração, sua situação passa para extraída, voltando a ter a situação bloqueada após o seu retorno (registro) ao controle da configuração.

### 4.3.3 REQUISITOS FUNCIONAIS

O modelo de especificação funcional utilizado foi o proposto por ROCHA (1987) e foi utilizado o método Análise Estruturada proposto por GANE (1984) . Diagramas de transição de estados foram utilizados para descrever os comportamentos dos itens e documentos de alteração utilizados no sistema. Como proposto por GANE (1987) após o último esboço do diagrama de fluxo de dados expandido foi realizada uma análise de entidade-relacionamento para se ter uma melhor visão das estruturas de dados representadas no sistema. O resultado desta análise permitiu a inclusão e exclusão de entidades segundo as necessidades do sistema. Com isso, o diagrama de fluxo de dados foi refeito para ressaltar uma melhor compreensão dos dados que compõem o sistema.

As funções do Sistema Automatizado de Gerenciamento da Configuração de Software (SAGCS) são apresentadas através do diagrama de fluxo de dados a seguir:

#### a) Diagrama de fluxos de dados

As figuras (IV.3) e (IV.4) apresentam o DFD de nível 0 e nível 1 respectivamente do sistema SAGCS.

#### 4.3.3.1 DESCRIÇÃO DOS PROCESSOS

**P1 - CADASTRAR USUÁRIO**

## Descrição :

O Bibliotecário cadastra os usuários do sistema, especificando os privilégios de acesso a cada item ou sistema.

entradas :

b - P1 - usuário

saída :

P1 - D1 - usuário

**P2 - CONTROLAR PEDIDOS DE ALTERAÇÃO**

## Descrição :

Através desse processo o bibliotecário associa a um documento de alteração um item presente na configuração de desenvolvimento. Então, sob ordens do gerente do projeto ou gerente da configuração, o bibliotecário pode alterar a situação de um documento.

Caso a situação passe para aprovada, todos os itens associados ao documento são copiados na configuração aprovada, caso contrário, se a situação for passada para rejeitada as modificações inseridas nos itens da configuração de desenvolvimento são destruídas.

entradas :

b - P2 - pedido\_alteração

D1 - P2 - usuário

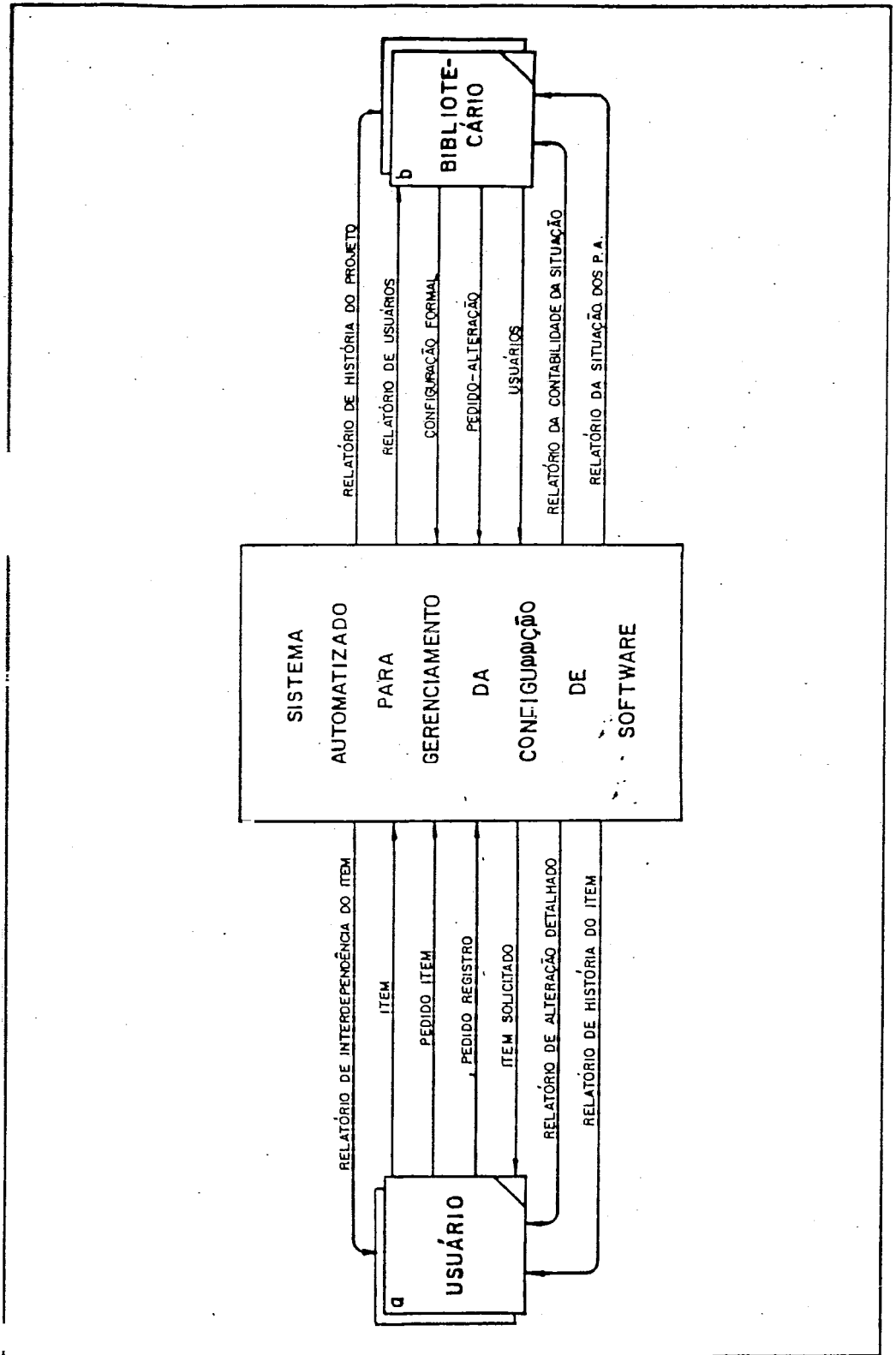
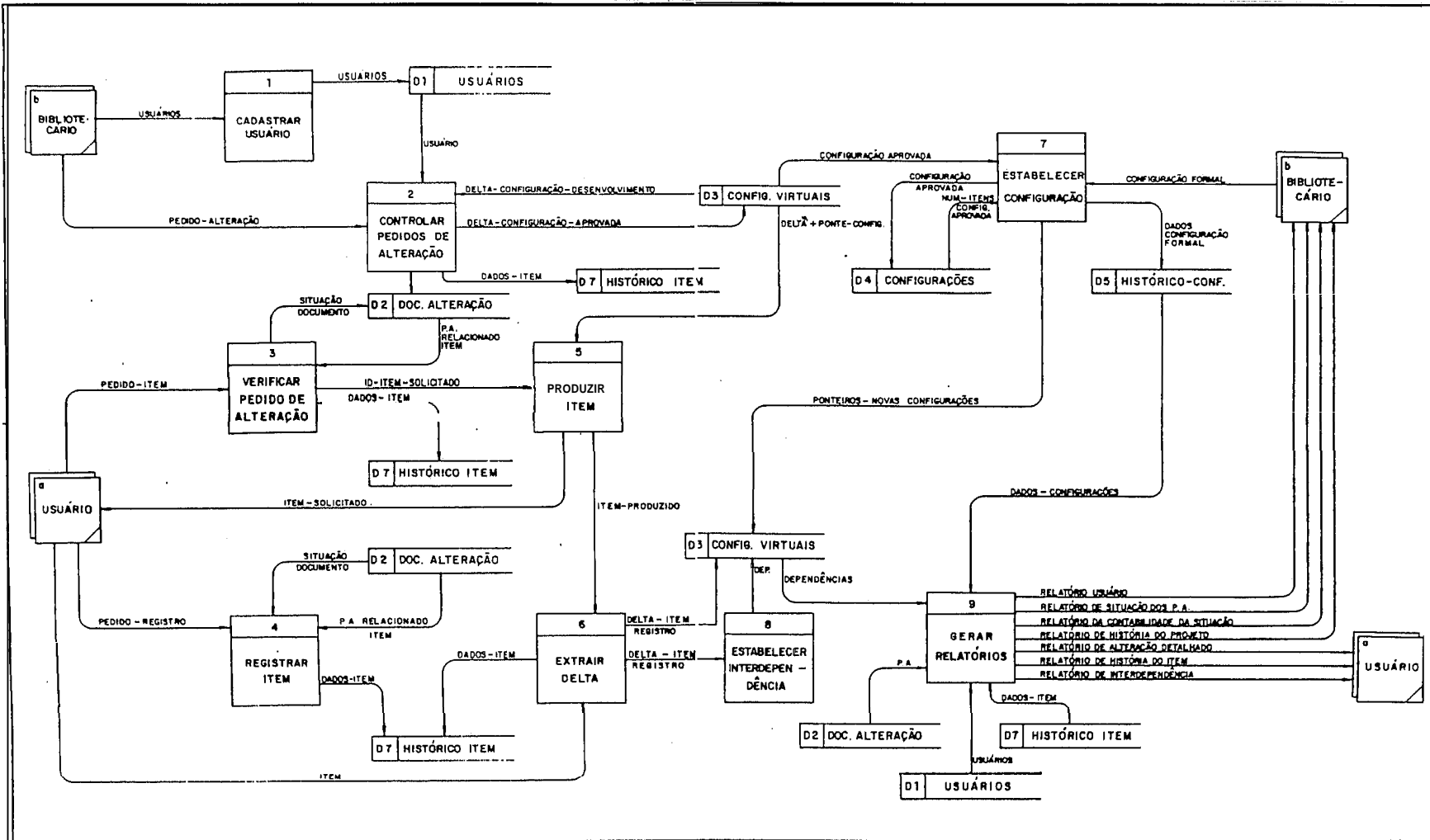


Figura IV.3 - DFD nível contexto do SAGCS

FIGURA IV.4 - DFD Nível 1 do SAGCS



D3 - P2 - delta\_config\_desenvolvimento

saídas :

P2 - D2 - pedido\_alteração

P2 - D7 - dados\_item

P2 - D3 - delta\_config\_desenvolvimento\_aprovado

### **P3 - VERIFICAR PEDIDO DE ALTERAÇÃO**

Descrição :

Verifica se o item solicitado pelo usuário está associado a um documento de alteração. Caso esteja, envia a identificação do item para que ele seja produzido. Neste caso, a situação do documento passa para desenvolvimento.

entradas :

a - P3 - pedido\_item

D2 - P3 - pedido\_alteração\_relacionado\_item

saídas :

P3 - D2 - situação\_documento

P3 - P5 - Id.item solicitado

P3 - D7 - dados\_item

### **P4 - REGISTRAR ÍTEM**

Descrição :

Este processo coloca um item sob gerenciamento da configuração de software. Se uma versão anterior do

item já existe e está associada a um documento de alteração (situação igual a desenvolvimento) , então, é produzida uma versão delta deste item e colocada na configuração de desenvolvimento..

entradas :

a -P4 - pedido\_registro

D2 - P4 - situação\_documento

D2 - P4 - pedido\_alteração\_relacionado\_item

saídas :

P4 - P5 - Id.item\_reg

P4 - D7 - dados\_item

#### P5 - PRODUZIR ÍTEM

Descrição :

Caso não seja a última versão do item, aplica sobre a versão base do item os deltas necessários à produção da versão solicitada.

entradas :

D3 - P5 - delta + ponteiros\_config

D4 - P5 - item

P3 - P5 - Id.item\_solicitado

P4 - P5 - Id.item\_reg

saídas :

P5 - a - item\_solicitado

P5 - P6 - item\_produzido

## P6 - EXTRAIR DELTA

Descrição :

Compara duas versões de um item. Através das diferenças encontradas é produzido um arquivo delta referente à versão anterior (caso o arquivo base seja o último) ou posterior (caso o arquivo base seja o primeiro).

entradas :

P5 - P6 - item\_produzido

a - P6 - item

saídas :

P6 - D7 - dados\_item

P6 - D3 - delta\_item\_registro

P6 - P8 - delta\_item\_registro

## P7 - ESTABELEECER CONFIGURAÇÃO

Descrição :

No momento em que uma configuração é estabelecida o conteúdo da configuração aprovada passa a ser a próxima configuração formal. As configurações virtuais anteriores (de desenvolvimento, aprovada e produção) são destruídas e são criadas outras com o mesmo conteúdo da nova configuração formal.



entradas :

b - P7 - configuração formal

D3 - P7 - configuração aprovada

D4 - P7 - números\_itens\_config\_aprovada

saídas :

P7 - D5 - dados\_config\_formal

P7 - D4 - configuração aprovada

P7 - D3 - ponteiros\_novas\_configurações

#### **P8 - ESTABELECEER INTERDEPENDÊNCIA**

Descrição :

Os itens (arquivos) presentes no nível FONTE e OBJETO com os mesmos nomes são associados automaticamente e os itens do nível OBJETO são associados ao único item do nível EXECUTÁVEL.

entradas :

P6 - P8 - delta\_item\_registro

saídas :

P8 - D3 - dependências

#### **P9 - GERAR RELATÓRIOS**

Descrição :

Gera os seguintes relatórios: relatório da contabilidade da situação, relatório de usuários, relatório de situação de pedidos de alteração, relatório de história do projeto, relatório de alteração detalhado, relatório de história do item e relatório de

interdependência do item.

entradas :

D5 - P9 - dados\_configurações

D3 - P9 - dependências

D2 - P9 - pedidos\_alteração

D7 - P9 - dados\_item

D1 - P9 - usuários

saídas :

P9 - b - relatório da contabilidade da situação

P9 - b - relatório de usuários

P9 - b - relatório de situação dos pedidos de alteração

P9 - b - relatório de história do projeto

P9 - a - relatório de alteração detalhado

P9 - a - relatório de história do item

P9 - a - relatório de interdependência do item

#### 4.4 ESPECIFICAÇÃO E PROJETO DO PROTÓTIPO DO SAGCS

O protótipo do SAGCS gerencia vários projetos e fornece ao usuário as seguintes capacidades :

- i) identificação dos itens e configurações sob o seu controle: rotula os itens extraídos (figura IV.5), informando o nome, tipo e versão do item, a data e hora do registro e os

comentários associados ao item;

- ii) controla o acesso aos itens por usuários não cadastrados ao projeto: impõe regras quanto a manipulação de um item, tal como: um item só pode ser registrado pelo usuário que o extraiu;
- iii) contabiliza todas as operações feitas em itens e configurações, fornecendo inúmeros relatórios detalhados aos usuários.

SSAGCS - Relatorio de Itens do Usuario : ADMIN    Data : 23/2/1991 12:51

\$Projeto: INI Usuario: ADMIN

\$Item: CALC.FOR:1        \$Registro: 28/1/1991 'as 20:35  
 \$Ambiente: COMPILADOR MICROSOFT FORTRAN  
 \$Extracao: 28/1/1991 'as 20:38  
 \$Usuario: ANA, Pedida de alteracao: P3  
 \$Comentario: ALTERACAO ARGUMENTOS DE ENTRADA

```
C
  INTEGER C, A(10), MASSA(12)
C
  X = X + 1
  IF (A(1) .LT. MASSA(1)) THEN
    DO 10 I = 1,12
      MASSA(I) = I + 1
10  CONTINUE
  ENDIF
  RETURN
  END
```

Figura IV.5 - Exemplo de rotulação automática de um item extraído

Este protótipo, está restrito ao ambiente de configuração aprovada, que mantém a evolução aprovada do projeto sob o seu controle, mantendo assim a rastreabilidade do projeto.

O protótipo gerencia apenas o nível fonte. Contudo, os arquivos objetos e executáveis podem ser gerados facilmente através de um utilitário similar ao MAKE, disponível em vários ambientes.

Uma configuração aprovada é constituída por um ou mais itens (arquivos fonte) já registrados, ou seja, já sob o controle do SAGCS. A associação dos itens a uma configuração é feita através de um nome dado pelo usuário a um conjunto de itens que comporão a configuração. Este nome é chamado rótulo de versão simbólico (figura IV.6).

Uma vez que, uma versão de um arquivo fonte foi associada a uma configuração, só as versões com números iguais ou superiores as versões associadas à configuração, poderão compor a próxima configuração.

Esta restrição tem como objetivo manter a rastreabilidade da evolução do sistema. Devido a isso, um rótulo de versão simbólico não pode ser associado às versões de arquivos fonte anteriores às versões que compõem a última configuração.

Para o projeto do protótipo do SAGCS foi utilizado o método Projeto Estruturado (PAGE-JONES, 1980). A passagem da especificação funcional para os gráficos de estrutura seguiu os passos descritos por NOGUEIRA (1988).

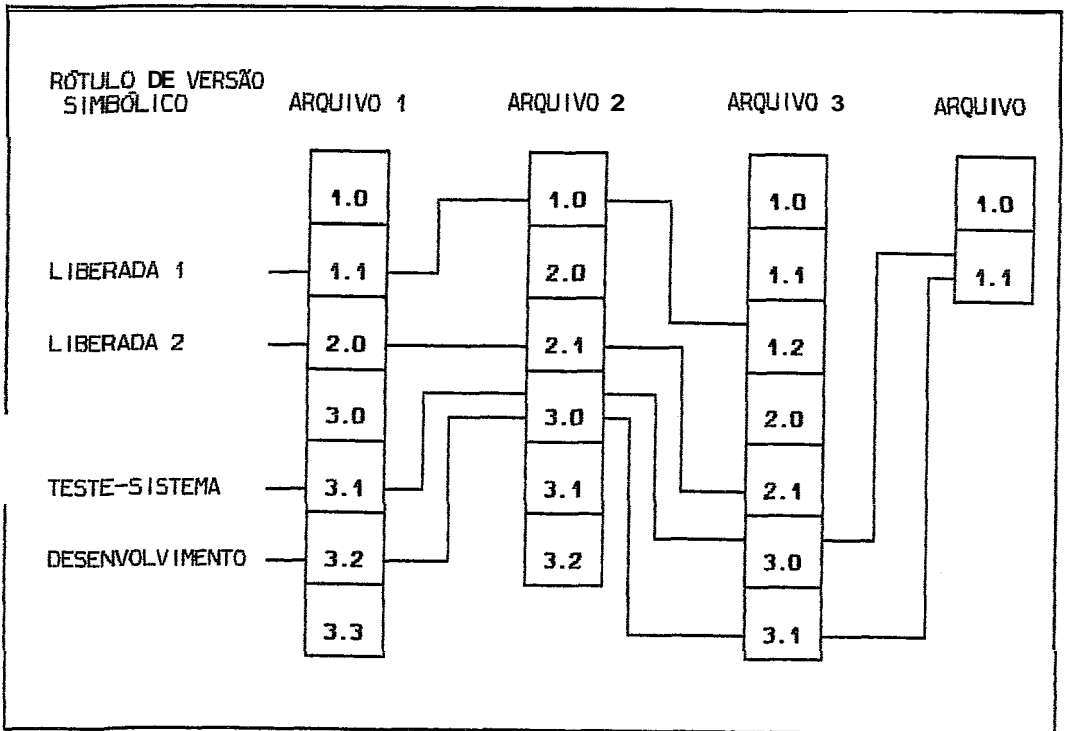


Figura IV.6 - Estabelecimento de uma configuração através do Rótulo de versão simbólico

Fonte : VALLINO (1987)

O SAGCS foi dividido em "jobs", cada um originando um gráfico de estrutura. A divisão dos gráficos de estrutura em programas levou em consideração o seguinte (NOGUEIRA, 1988) :

- i) módulos que podem ser executados por pacotes de software;
- ii) requisitos de segurança;
- iii) recursos disponíveis;

iv) módulos mais utilizados devem estar em um mesmo pacote.

A estrutura modular do protótipo do SAGCS é apresentada através das figuras (IV.7) a (IV.11).

#### 4.5 IMPLEMENTAÇÃO

O protótipo do Sistema Automatizado para Gerenciamento da Configuração de Software (SAGCS) foi implementado através da linguagem Turbo C da Borland. Requer um ambiente operacional mínimo de um microcomputador compatível com o IBM/XT com 640 Kb de memória, um disco rígido de 20Mb e uma unidade de disco flexível.

O SAGCS apresenta um ambiente de fácil utilização através de menus interativos. A tela com as funções do SAGCS (figura IV.12) permite ao usuário gerenciar todos os itens e configurações presentes no desenvolvimento de um projeto. Vários tipos de relatórios possibilitam o acompanhamento e a contabilidade da situação do projeto.

A figura (IV.13) , apresenta um exemplo da tela de extração de um item sob o controle do SAGCS.

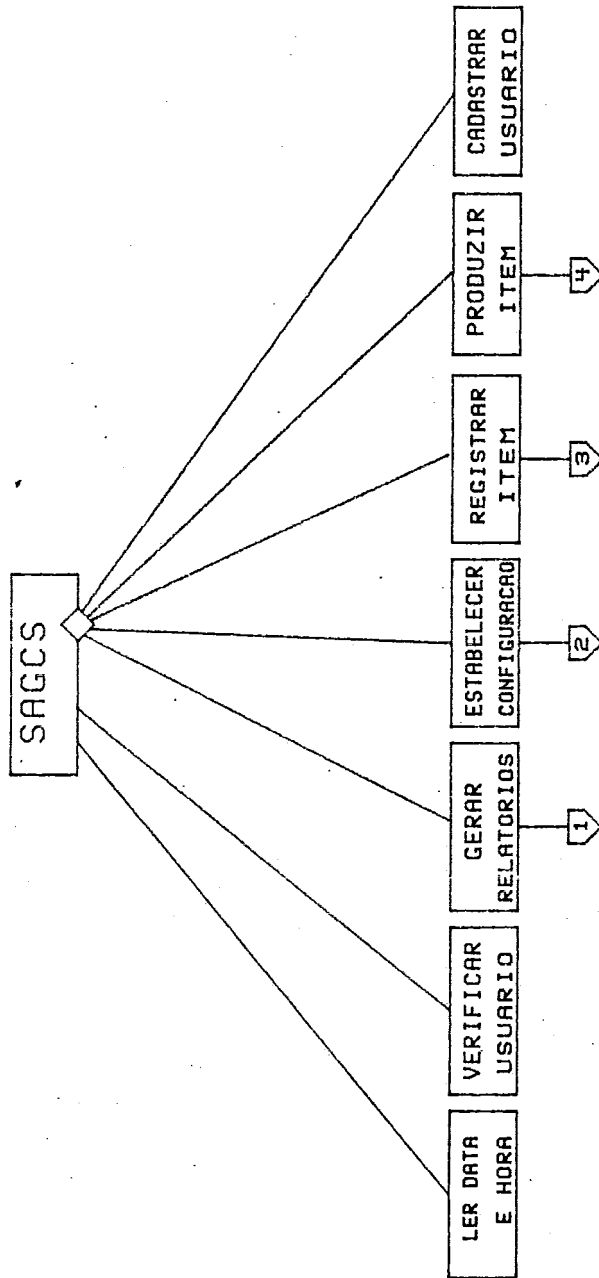
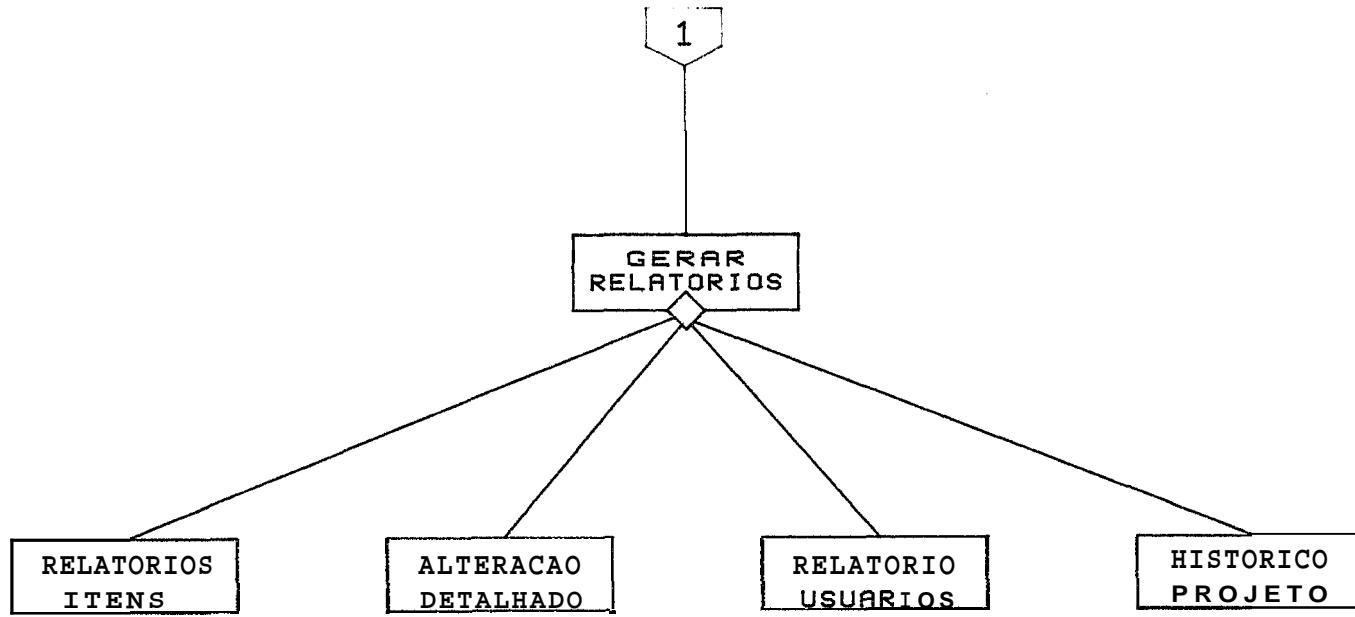


Figura IV.7 - Estrutura modular do protótipo do SAGCS

FIGURA IV.8 - Estrutura modular de Gerar Relatórios





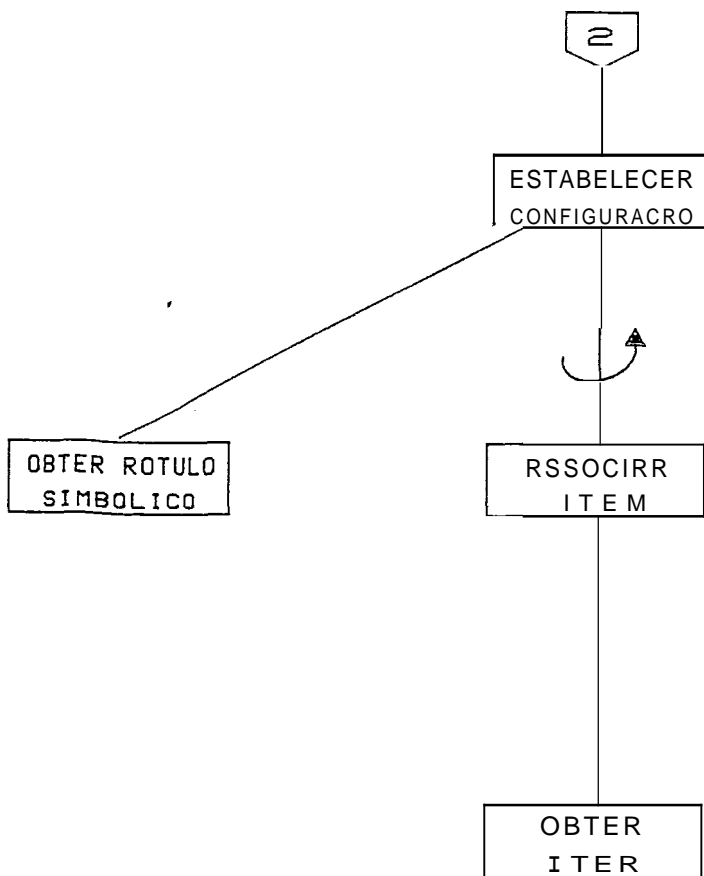


Figura IV.9 - Estrutura modular de Estabelecer Configuração

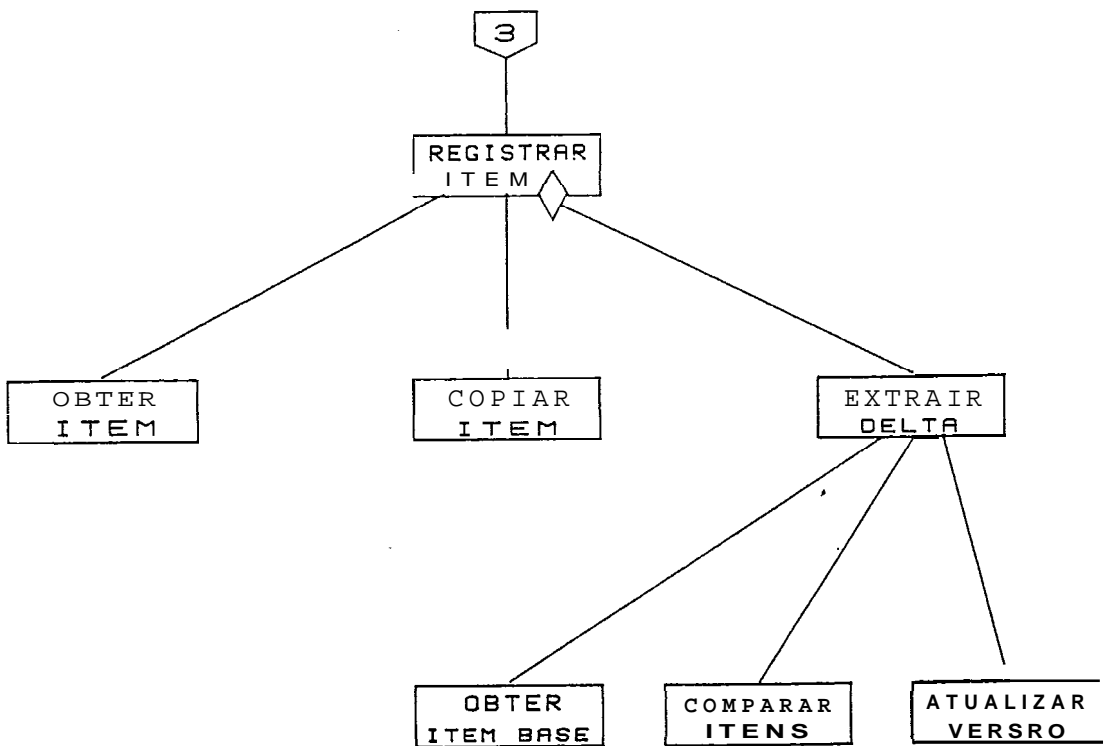


Figura IV.10 - Estrutura modular de Registrar Item

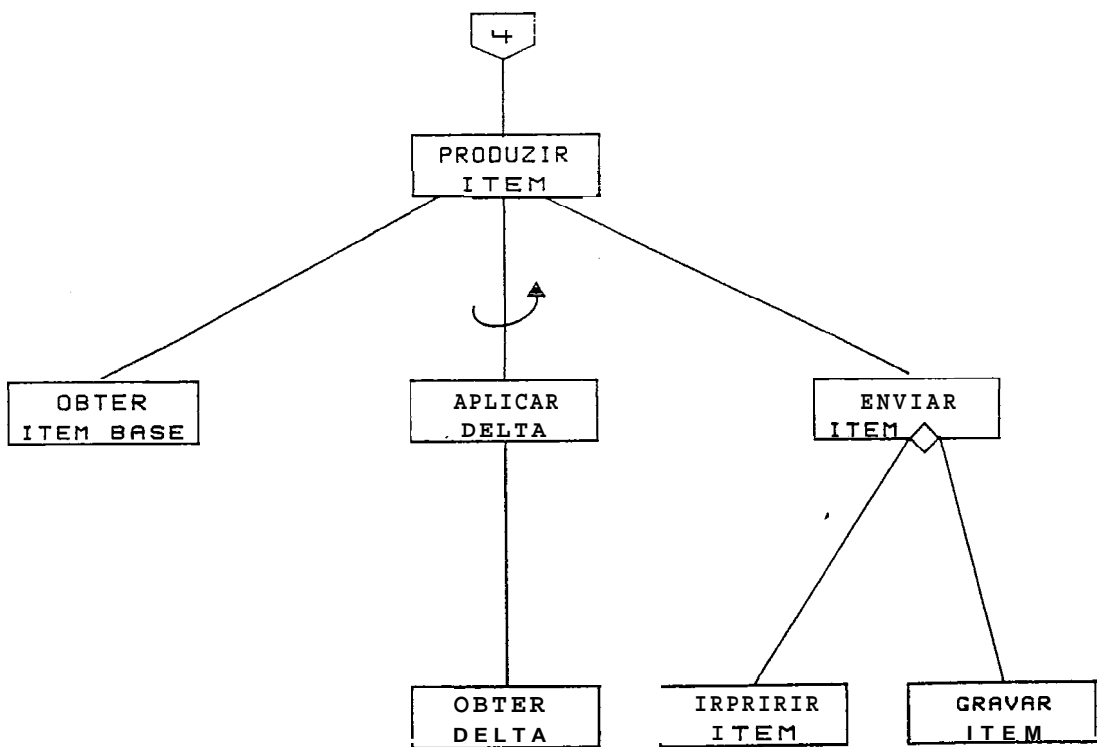


Figura IV.11 - Estrutura modular de Produzir Item

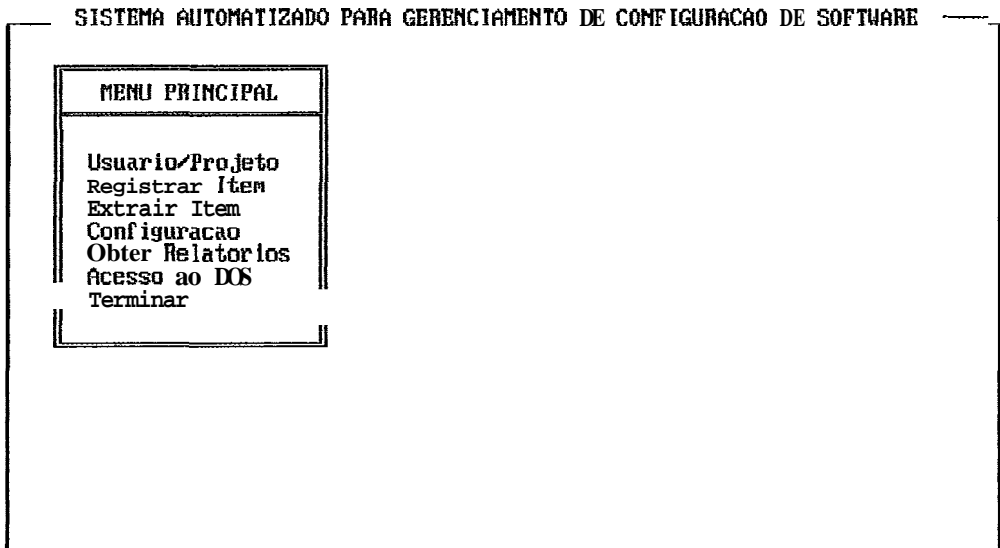


Figura IV.12 - Tela do menu principal do SAGCS

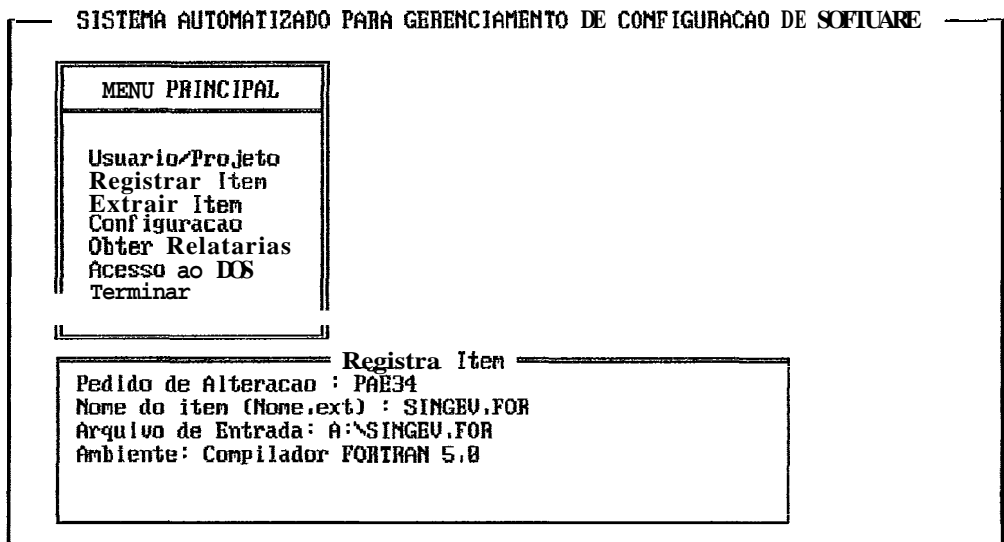


Figura IV.13 - Tela de extração de um item no SAGCS

Devido ao grande volume de dados que este sistema tem que manipular, optou-se por só armazenar as diferenças entre duas versões consecutivas de um código fonte. A

economia é substancial além de facilitar o controle das diversas versões produzidas (MILLER, 1987).

As diferenças entre as duas versões consecutivas são expressas através de deltas. Deltas são arquivos que contêm as operações e os dados necessários para se reproduzir uma determinada versão. A aplicação dos dados armazenados no delta sobre um arquivo base (completo) produz a revisão desejada. Um exemplo da aplicação de deltas é apresentado a seguir (figura IV.14):

A aplicação dos deltas sobre um arquivo base pode ser direta ou reversa. Na aplicação direta, a primeira versão do código fonte é mantida integralmente na base de dados e as versões deste código são reproduzidas através da aplicação dos seus deltas correspondentes. Já na aplicação reversa, a última versão é que é mantida integralmente. Na implementação do SAGCS optou-se pela aplicação reversa, em virtude da última versão de um código ser a mais solicitada durante o desenvolvimento de um sistema e com isso evitando-se o tempo gasto na aplicação dos deltas na produção do arquivo desejado.

Para a produção dos deltas é necessário que haja um comparador de arquivos. Vários algoritmos de comparação foram estudados. O algoritmo descrito por CORTESI (1984) apesar de não ter sido utilizado, por não produzir sempre um delta com o número mínimo de instruções, merece ser mencionado.

Este algoritmo ao contrário dos outros algoritmos de comparação, não procura as diferenças entre dois arquivos, mas sim as suas semelhanças e independe do tamanho dos arquivos sendo comparados ao contrário dos outros algoritmos de comparação. O algoritmo pode ser resumido pela aplicação das seguintes regras :

- i) Regra 1 - se uma linha aparece somente uma vez nos dois arquivos sendo comparados, então elas têm que ser as mesmas e são ditas casadas.
- ii) Regra 2 - se as linhas adjacentes a um par casado de linhas são idênticas, então este par é também denominado casado.

Aplicando-se estas regras nos dois arquivos, temos :

- i) as linhas que satisfazem as regras um e dois são consideradas casadas;
- ii) as linhas que aparecem apenas no arquivo novo são consideradas linhas inseridas;
- iii) as linhas que só aparecem no arquivo base são consideradas linhas apagadas;
- iv) as linhas que sobraem são consideradas as diferenças entre os dois arquivos.

ARQUIVO BASE	
LINHA	IDENTIFICAÇÃO
1	Linha 4 do Arquivo base
2	Esta linha será alterada na revisão
3	
4	A primeira revisão terá 2 linhas adicionar antes desta
5	Esta linha será apagada na próxima revisão
6	
7	Fim do arquivo base
DELTA	
M 2,1	Na linha 2, apague uma linha e substitua uma linha
I 4,2	Esta é a segunda linha revisada Antes da linha 4 insira 2 linhas Esta linha foi inserida antes da linha 4 Esta outra também
A 6,1	
PRIMEIRA REVISÃO	
1	Linha 4 do arquivo base
2	Esta é a segunda linha revisada
3	
4	Esta linha foi inserida antes da linha 4
5	Esta outra também
6	
7	
8	Fim do arquivo base

## Legenda:

- M - Modificar
- I - Inserir
- A - Apagar

Figura IV.14 - Revisão de um arquivo através da  
aplicação de um delta

Fonte - VALLINO (1987)

O algoritmo utilizado (MILLER, 1987) na construção do comparador do SAGCS é ótimo na produção de deltas com as instruções COPIAR e ADICIONAR, ou seja, se um arquivo B pode ser construído a partir de um arquivo A com n instruções do tipo copiar e adicionar, então este algoritmo produz no máximo n instruções. A instrução COPIAR, referencia uma linha do arquivo base e a instrução ADICIONAR mantém a informação a ser inserida no arquivo sendo gerado. Um exemplo de um delta produzido pela aplicação deste algoritmo é exposto a seguir :

Se o arquivo B = acab é construído a partir do arquivo A = ab, então com a aplicação deste algoritmo tem-se o seguinte delta:

DELTA	ARQUIVO SENDO FORMADO
COPIAR o símbolo 1 (a) começando na linha 1 .....	a
ADICIONAR c .....	ac
COPIAR os dois símbolos (ab) começando na linha 1 ..	acab



Qualquer versão, então, pode ser criada pela simples aplicação dos deltas referentes a ela sobre a versão base. A utilização de deltas só é eficaz, quando os dois arquivos sendo comparados não diferem muito.

Para dar uma melhor noção das características do protótipo do SAGCS, este foi incluído na tabela (tabela IV.1) apresentada por MOTTA (1990) que compara várias ferramentas de gerenciamento da configuração de software disponíveis no mercado : Source Code Control System (SCCS) disponível nos ambientes UNIX, Revision Control System (RCS), Project Automatic Librarian (PAL), Code Management System (CMS) e Module Management System (MMS) disponíveis em ambiente VAX e o Sistema de Controle de Versões proposto por MOTTA (1990).

Em virtude da própria filosofia de especificação deste protótipo, este só será utilizado eficazmente se associado em conjunto com um Plano de Gerenciamento da Configuração de Software, o qual estabelecerá os documentos de controle necessários à condução do projeto e suas ações correlatas.

Muitas das restrições apresentadas, tal como, a impossibilidade de se excluir um item já registrado, têm como objetivo forçar ou convencer o usuário da necessidade de todo um ciclo de garantia do produto, até que o item seja posto sob o gerenciamento da configuração de software.

FACILIDADES	SCCS	R65	PAL	CMS/ MMS	SCV	SAGCS
<u>Manipulação de revisões</u>						
Armazena (Delta/Inteira)	D	D	I	D	I	D
Rotula (S/N)	S	S	N	S	S	S
Recupera (Copia/Reserva)	C	C/R	C/R	C/R	C/R	C/R
Remove (S/N)	S	N	S	S	S	N
Altera-estado (S/N)	N	S	-	S	S	N
Documenta as mudanças (S/N)	S	S	S	S	S	S
Identifica (Código/Executável)	C	C/E	C	C	C	C
Facilita a geração Executável	N	S	S	S	S	N
<u>Manipulação de configurações</u>						
Gera (S/N)	N	S	S	S	S	S
Recupera (Copia/Reserva)	N	C/R	C/R	C/R	C/R	C/R
Rotula (S/N)	N	S	N	S	S	S
Atualiza (Substitui/Gera nova)	N	S	G	N	G	G
Remove (S/N)	N	-	S	S	S	N*
Documenta as mudanças	N	N	N	N	S	S
Facilita a geração do Executável	N	S	S	S	S	N
Modelo da organização e do processo de desenvolvimento	N	N	N	N	S	N
* - Só a última configuração						

Tabela IV.1 - Comparação entre vários sistemas de gerenciamento da configuração e o SAGCS

Fonte : MOTTA (1990)

A utilização do SAGCS então, fornece um ambiente de fácil interação para o usuário de microcomputador. Através dela, toda evolução do sistema pode ser acompanhada, todas alterações podem ser documentadas e todas as configurações dos sistemas sob seu controle estarão sempre disponíveis.

## CAPÍTULO V

### CONCLUSÃO

Os conceitos de gerenciamento da configuração de software ainda são pouco difundidos e por consequência muito pouco aplicados.

Mesmo na área de computação, é raro encontrar quem tenha uma noção precisa dos termos e da filosofia desta disciplina da engenharia de software.

O que se vê na prática, são tentativas de se acompanhar a evolução de um sistema através de métodos próprios para cada instalação. Estes métodos, geralmente são insuficientes e muito esforço é gasto para se saber quais são as configurações correntes e anteriores (rastreadibilidade) de um sistema.

A se perdurar a filosofia: "pague-me um pouco agora e muito mais adiante" em detrimento de "pague-me um pouco mais agora e um pouco menos adiante", por parte tanto do desenvolvedor quanto do solicitante do produto de software,

a aplicação do gerenciamento da configuração de software sempre será descartada, devido ao custo inicial para a sua implantação e aplicação.

Contudo, todo recurso empregado em gerenciamento da configuração de software, será ressarcido facilmente pela redução dos custos de manutenção e pela satisfação do solicitante do produto de software.

A aplicação do Plano de Gerenciamento da Configuração de Software (PGCS) é fundamental em qualquer projeto. Entretanto, o gerente do projeto tem que estar atento à não deixar que aplicação desse plano altere o fluxo normal de desenvolvimento do projeto, tornando-o um entrave burocrático.

Ao final de cada projeto, o PGCS deve ser avaliado para se diagnosticar os possíveis melhoramentos e ajustes a serem incorporados nos próximos planos.

A ferramenta automatizada, SAGCS, proposta neste trabalho de tese, tenta suprir em parte a carência de ferramentas automatizadas para gerenciamento da configuração de software para ambientes DOS que fosse ao mesmo tempo barata e poderosa.

Para torná-la **mais** poderosa, deveria ser implementada a versão *rede* da SAGCS, *ria qual* fosse permitido a

desenvolvimento paralelo. Assim, toda equipe de desenvolvimento teria sempre a disposição a configuração corrente do sistema, além de permitir que mais de uma pessoa trabalhe sobre um mesmo item ao mesmo tempo.

Outro aspecto importante, é a necessidade de se incorporar ao SAGCS um dispositivo de proteção aos dados sob seu controle.

Como sugestão para pesquisa, visando diminuir os custos do gerenciamento da configuração de software e assim a sua aplicação mais efetiva, são apresentadas alguns tipos de ferramentas importantes para esta área:

Como a função de auditoria é a que mais consome recursos empregados na aplicação de gerenciamento da configuração de software, devido a necessidade de ser realizada por peritos, seria de grande utilidade um sistema especialista para auxiliar os auditores do sistema através de uma base de conhecimento com as experiências adquiridas em projetos de software anteriores.

Um gerador de documentos de controle que facilitaria enormemente a confecção e a gerência desses documentos por parte do gerente da configuração.

Um gerador de árvore de itens, mostrando de forma gráfica os itens que compõem a configuração de um sistema.

Tal ferramenta, possibilitaria também a inclusão de um item, na base de dados do SAGCS por exemplo, via esta interface gráfica. Neste caso, seria gerado um esqueleto do módulo (específico para cada linguagem), o qual alertaria que um módulo que fará parte do sistema estaria em desenvolvimento por uma das equipes.

Por fim, um ponto a ser considerado na construção de ferramentas automatizadas para gerenciamento da configuração de software, é que apesar da solicitação dos usuários por ambientes mais flexíveis, a construção e utilização destas ferramentas geralmente não podem estar dissociadas do aspecto formal imposto por esta disciplina. Pois, o gerenciamento da configuração de software é uma atividade de controle e acompanhamento e requer todo um ciclo de garantia do produto.

## APÊNDICE I

## MANUAL DO USUÁRIO

O protótipo do Sistema Automatizado para Gerenciamento da Configuração de Software (SAGCS) gerencia vários sistemas através do controle de todas as configurações estabelecidas durante o ciclo de vida de um sistema. Nesta versão, o SAGCS não gerencia caminhos paralelos de desenvolvimento (ramos), só controlando um único caminho (principal ou tronco).

A utilização eficaz do SAGCS só será conseguida em conjunto com um Plano de Gerenciamento da Configuração de Software. Neste plano, deverão constar os tipos de documentos e procedimentos de controle para o projeto.

No início de sua instalação o sistema SAGCS cria um subdiretório chamado SAGCS, onde estarão suas bases de dados. Neste momento, só um usuário especial, o ADMIN associado ao projeto INI terá acesso as bases de dados do SAGCS.

A proteção aos dados é feita através da associação

usuário-projeto. Para que, um usuário possa utilizar o sistema o ADMIN deve cadastrá-lo a um projeto específico.

Este sistema é facilmente utilizado através de menus interativos, onde o usuário pode escolher a opção realçada ou digitar a primeira letra da opção desejada. Cada opção pode ser cancelada por meio da tecla ESC, fazendo que o sistema retorne ao nível anterior.

Para melhor compreensão, este manual será apresentado ao usuário através das telas produzidas pelo sistema.

Para ativar o SAGCS basta digitar SAGCS no diretório corrente do sistema. Como resposta é mostrada a tela de apresentação do sistema (figura AI.1) na qual é solicitado o nome do usuário.

<b>SISTEMA AUTOMATIZADO PARA GERENCIAMENTO DE CONFIGURACAO DE SOFTWARE</b>	
<b>COPPE/UFAJ</b>	<b>PROGRAMA DE ENGENHARIA DE SISTEMAS</b>
<div style="border: 1px solid black; padding: 5px;"><p><b>Data: 16-2-1991</b> <b>Hora: 15:50:36</b> <b>Digite &lt;ESC&gt; para terminar</b> <b>Usuario : ADMIN Projeto : INI</b></p></div>	

figura AI.1 - Tela de apresentação do sistema SAGCS

No início como nenhum usuário foi cadastrado, só o usuário designado para gerenciar o sistema terá acesso



através do nome ADMIN e projeto INI.

Tendo o sistema aceito o usuário, o menu principal é apresentado com as opções do sistema (figura AI.2).

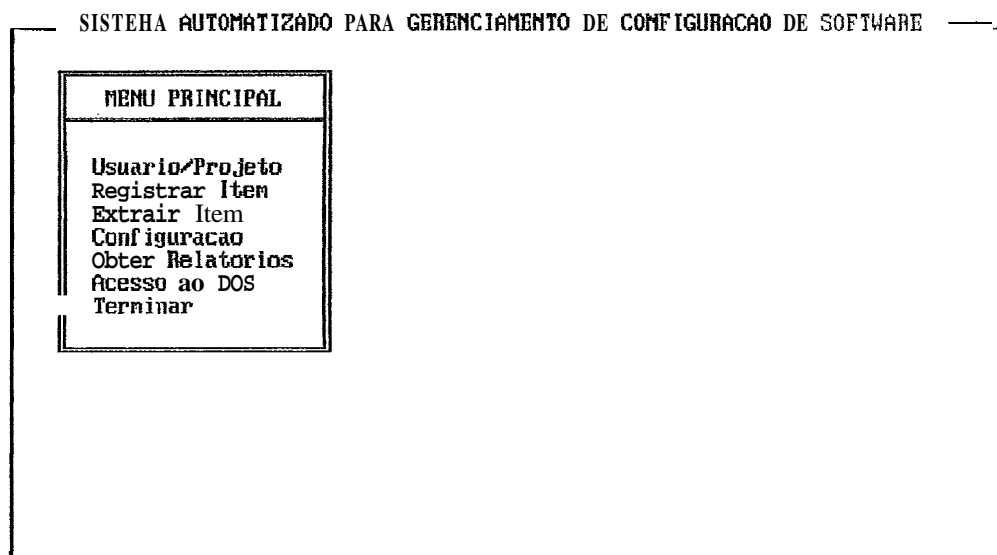


figura AI.2 - Tela do menu principal do SAGCS

Cada opção do menu é descrita a seguir :

### 1 - USUÁRIO/PROJETO

Esta tela (figura AI.3) permite inserir ou apagar as associações usuário-projeto e a finalização de um projeto. Esta última opção exclui todos os dados sobre um projeto na base de dados e só deve ser utilizado no término do projeto após terem sido feitas cópias de segurança.

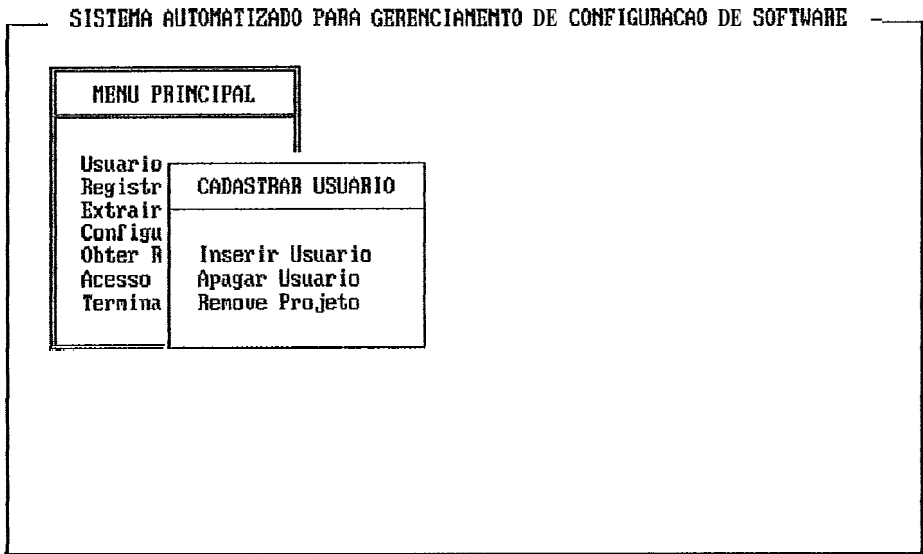


figura AI.3 - Tela com as opções de Usuário-Projeto  
mensagens de erro :

```

----> usuário já cadastrado
----> usuário não existe
  
```

## 2 - REGISTRAR ÍTEM

A tela "registrar item" (figura AI.4) solicita o nome do pedido de alteração. Se o item está sendo registrado pela primeira vez, basta passar para o próximo campo, entrar com o nome do arquivo que contém o item e o nome do item.

A abreviação do nome do pedido de alteração deve ser usada acompanhada de um número sequencial para facilitar a associação entre o item alterado e o documento que autorizou a sua alteração. Por exemplo, para um documento estabelecido no Plano de Gerenciamento da Configuração de Software, como, Proposta de Alteração de Engenharia e sendo a terceira vez que é utilizado no projeto, deveria ser

abreviado como PAE03.

Após o pedido de alteração, o usuário deve entrar respectivamente, com o nome que o item será registrado na base de dados do SAACS e o local e identificação (caminho, nome e tipo) do arquivo a ser registrado. Finalmente, é dada ao usuário a opção de indicar o ambiente operacional do item.

SISTEMA AUTOMATIZADO PARA GERENCIAMENTO DE CONFIGURACAO DE SOFTWARE	
<p><b>MENU PRINCIPAL</b></p> <p>           Usuario/Projeto            Registrar item            Extrair Item            Configuracao            Obter Relatorios            Acesso ao DOS            Terminar         </p>	
	<p><b>Registra Item</b></p> <p>           Pedido de Alteracao : PAE34            Nome do item (Nome.ext) : SINGEU.FOR            Arquivo de Entrada: A:\SINGEU.FOR            Ambiente: <del>Compilador</del> FORTRAN 5.0         </p>

figura AI.4 - Tela de registro de um item

mensagens de erro :

---> este item já está associado a outro pedido de alteração

### 3 - EXTRAIR ÍTEM

Esta opção, segue o padrão inverso da opção anterior. Neste caso o usuário escolhe o item a ser extraído e o local e identificação onde o item será gravado.

Caso não seja a última versão do item, não é necessário o nome do pedido de alteração. Neste caso, a identificação do item terá que vir acompanhada do número da versão a ser extraída (figura AI.5).

SISTEMA AUTOMATIZADO PARA GERENCIAMENTO DE CONFIGURACAO DE SOFTWARE	
<b>MENU PRINCIPAL</b> Usuario/Projeto Registrar Item Extrair Item Configuracao Obter Relatorios Acesso ao DOS Terminar	
===== Extrair Item =====	
Pedido de Alteracao : PAE20 Nome do item (Nome.ext) : MINU.FOR Digite o numero da versao : 5 Arquivo de Saida: A:\DESENU\MINU.FOR Comentario : Alteracao no algortino de inversao	

Figura AI.5 - Tela de extração de um item sob controle do SAGCS

mensagens de erro :

---> o item solicitado não existe

### 4 - CONFIGURAÇÃO

Esta opção permite o usuário manipular a configuração do sistema.

Estabelecer configuração, lista todas as versões com os números iguais ou superiores a última configuração estabelecida (figura AI.6). O usuário então, marca as versões que farão parte desta nova configuração, associando um nome simbólico para ela.

Através da opção "Apagar configuração", o usuário pode excluir uma configuração estabelecida. Entretanto, só é possível apagar a última configuração estabelecida, isto em virtude de se manter a rastreabilidade do projeto.

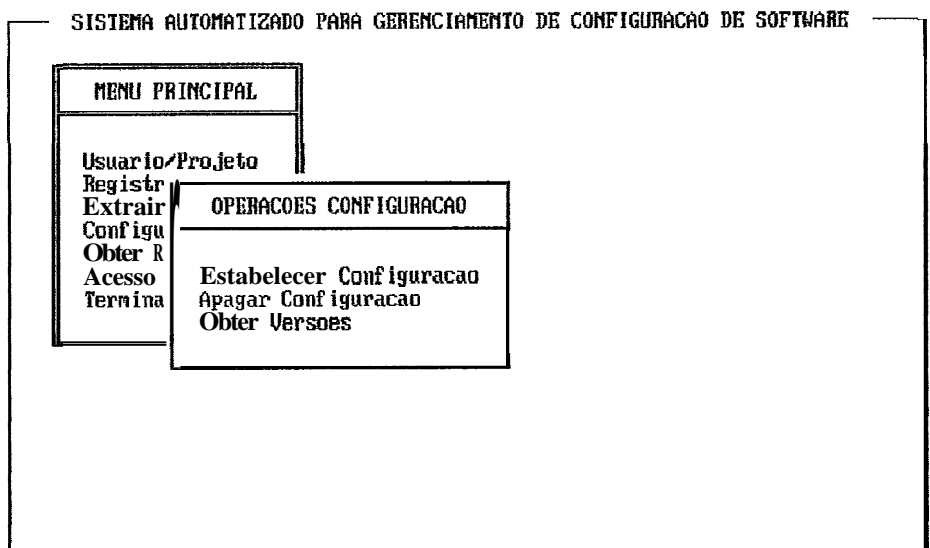


figura AI.6 - Tela da opção Configuração

No exemplo abaixo, o sistema apresenta todos os itens que podem fazer parte da próxima configuração. Versões diferentes de um mesmo item, entretanto não podem fazer parte de uma única configuração.

SAGCS - Estabelecer Configuracao - Relacao de Itens			
Nome	Tipo	Versao	
MINU	C	2	
DEGRAF	C	2	
CALC	FOR	2	
CONUERTE	FOR	1	
SINGEV	C	3	
RAIZQUAD	C	1	
RAIZQUAD	C	2	
GRAF	C	1	

<Enter> : Marca/Desmarca Item - <Esc> Aceita configuracao

Figura AI.7 - Escolha dos itens que farão parte de uma configuração

A opção "Obter versões", permite que se extraia todos os itens relacionados a uma configuração.

## 5 - OBTER RELATÓRIOS

Esta tela permite a escolha de seis tipos de relatórios. O conteúdo e as telas correspondentes de cada um deles são listadas a seguir:

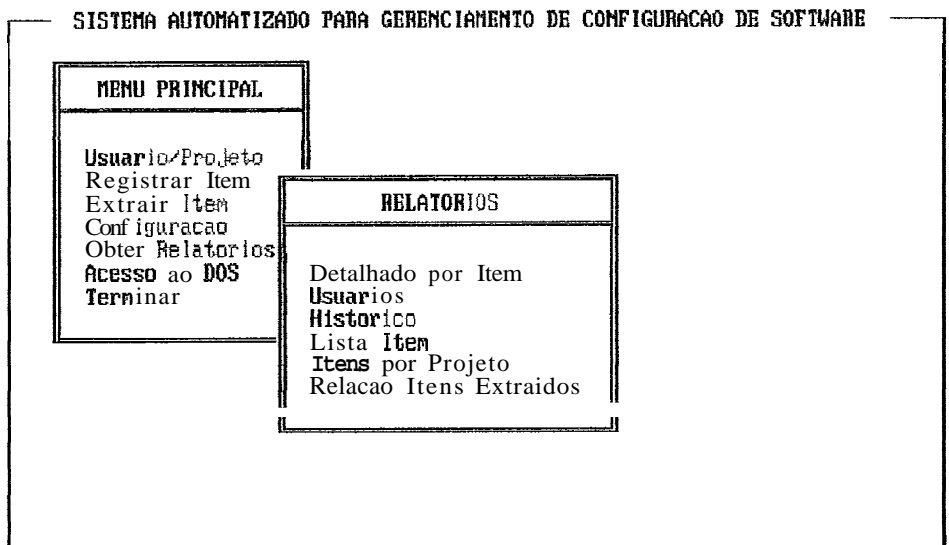


Figura AI.8 - Tela com os relatórios do SAGCS

### 5.1 - RELATÓRIO DETALHADO POR ÍTEM

A tela deste relatório solicita a identificação do item, caso a versão seja omitida, a última versão do item será listada.

Um exemplo deste relatório é mostrado a seguir :

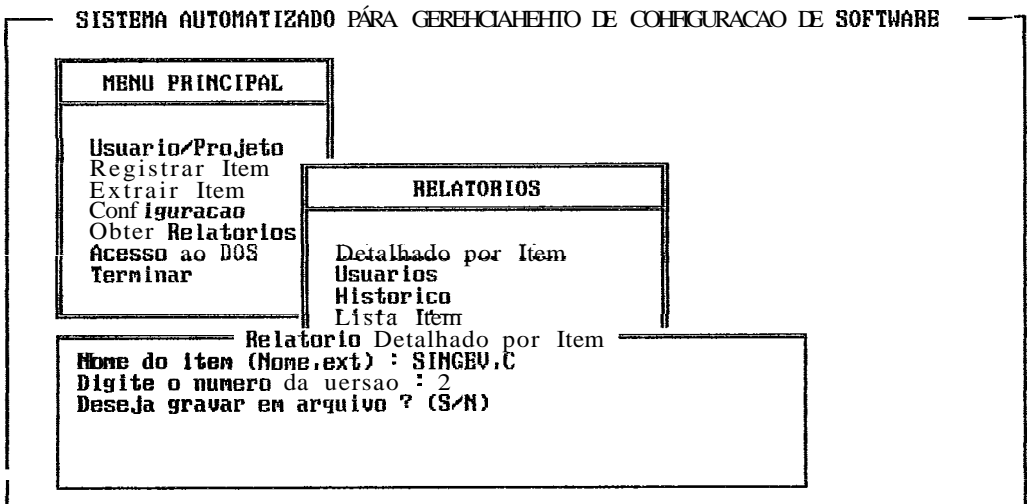


Figura AI.9 - Tela com a opção relatório detalhado

O relatório abaixo fornece ao usuário informações sobre quem fez a alteração, através de que documento, quando foi realizada e o que foi modificado.

SAGCS - Relatório Detalhado de Itens Data : 23/2/1991 11:34

Projeto: INI

Item: SINGEV.C:2 Registro: 28/1/1991 'as 20:44  
 Ambiente: HICRO XT COM PLACA ACELERADORA  
 Extração: 28/1/1991 'as 20:47  
 Usuário: TERESA, Pedido de alteração: P6  
 Comentário: CORRECAO ERRO DE LBITURR

SAGCS - Relatório Detalhado - Delta Correspondente Data : 23/2/1991 11:34

Projeto: INI, Usuario: ADMIN

Liçtagem do Delta:

```
A:
int
ARQLookUp(lista,index,arq,tipo,dim)
ARQauxLista *lista:
unsigned int index:
char *arq[]:
char *tipo[]:
unsigned int dim:
C,1,8
A:
unsigned int i:
C,1,4
A:
/* Se ja' foi marcado, entao e' um toggle para desmarcar. Tudo bem */
if (strcmp(lista[index],ARQmark,MARK)==0) return(0):
Aperte a tecla de espaco para continuar.
```

Figura A.10 - Relatório de Alteração detalhado

## 5.2 - RELATÓRIO USUÁRIOS

Este relatório lista todos os usuários correspondentes a um projeto.



### 5.3 - RELATÓRIO DE PROJETO

Este relatório lista informações sobre todas as configurações estabelecidas, especificando os itens que as compõem.

Abaixo é listada a evolução de um sistema através de suas configurações.

```

SAGCS - Relatório de Historico do Projeto          Data : 23/2/1991  11:36

Projeto: INI, Usuario: ADMIN

Configuracao 1 : CONFIGURACAO DE TESTE           Estabelecida em 28/1/1991 `as 20:50
Comentario : LIBERADA PARA TESTE E VERIFICACAO
Relacao de itens:
    MINU.C:1
    DEGRAF.C:2
    CALC.FOR:1
    CONVERTE.FOR:1

Configuracao 2 : PRODUCAO INTERNA                Estabelecida em 28/1/1991 `as 20:54
Comentario : LIBERADA PARA A GARANTIA DO PRODUTO
Relacao de itens:
    MINU.C:1
    CALC.FOR:1
    CONVERTE.FOR:1
    SINGEV.C:3
    RAIZQUAD.C:1
Aperte a tecla de espaco para continuar,

```

Figura A1.11 - Relatório com a evolução de um sistema

### 5.4 - RELATÓRIO LISTA ÍTEM

Lista todas as operações realizadas sobre um item.

SAGCS - Relatório do arquivo de itens

Data : 23/2/1991 11:37

Projeto: INI, Usuario: ADMIN

Item: DEGRAF.C:1 Registro: 28/1/1991 `as 28:28  
Ambiente: COMPILADOR TURBO C VERSAO 5.0  
Extracao: 28/1/1991 `as 28:31  
Usuario: ADMIN, Pedido de alteracao: PZ  
Comentario: CORRECAO TELAS GRAFICAS

Item: DEGRAF.C:2 Registro: 28/1/1991 `as 28:33  
Ambiente: COMPILADOR TURBO C COM OPCOES /D E /F

Figura AI.12 - Relatório com todas operações sobre um  
Ítem

## 5.5 - RELATÓRIO ITENS POR PROJETO

Lista todos os ítems com suas operações referente a um projeto específico.

## 5.6 - RELATÓRIO DE ITENS EXTRAÍDOS

Lista todos os ítems extraídos por um usuário.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALDERSEN, A. (1985), "Configuration Management in Eclipse", Conference Proceedings - MILCOMP '85 Military Computers, Graphics and Software pp. 226-30.
- BERLACK, H. (1981), "Implementing Software configuration control in the structured programming", 1981 ACM Workshop/Symposium on Measurement and Evaluation of Software Quality vol. 10 n.1 pp. 55-57.
- BERSOFF, E., et al (1980), "Software Configuration Management - An Investment in Product Integrity", Prentice-Hall, Englewood Cliffs, N. J..
- BERSOFF, E., et al (1984) , "Elements of Software Configuration Management", IEEE Trans. Software Engineering vol. SE-10 n. 1 pp. 79-87.
- BOEHM, B. (1981), "Software Engineering Economics", Prentice-Hall, Cliffs, N. J.
- BRYAN, W., et al (1981), "An approach to Software Configuration Control", 1981 ACM Workshop/Symposium on Measurement Configuration Control, pp. 33-47.
- BRYON, W., et al (1980), "Tutorial Software Configuration Management", IEEE computer society's Fourth International Computer Software & Applications Conference..

- CHARETTE, R. (1988), "Software Engineering Environments - Concepts and Technology", MacGraw Hill Inc.
- CORTESI, D. (1984 ), "What's the Diff? A file comparator for CP/M Plus", Dr. Dobb's Journal agosto.
- DUNN, R. e ULLMAN (1982), "Quality Assurance for computer software", MacGraw Hill, Inc.
- DUNN, R. (1990), "Software Quality: Concepts and plans", Prentice-Hall, Inc.
- EVANS, W. e FURR, G. (1986), "The effective Configuration Management Environment", The IEEE Computer Society 'Tenth Annual International Computer Software and Applications Conference, pp. 168-73.
- FORMAN, J. (1981), "How much does configuration cost?", IEEE Software Engineering Standards Applications Workshop, pp. 41-44.
- GANE, C. (1987), "Rapid System Development - using structured analysis and relational technology", Rapid System Development Inc.
- GANE, C. e SARSON, T. (1984), "Análise Estruturada de Sistemas", Livros Técnicos e Científicos.
- IEEE Std 729 (1983), "IEEE Standard Glossary of Software Engineerin Terminology".
- IEEE Std 730 (1984), "IEEE Standard for Software Quality Assurance Plans ".

IEEE Std 020 (1983), "IEEE Standard for Software Configuration Management Plans".

KHAN, M. (1985), "A practical Software Configuration Management Plan", J. INF. SYST. MANAGE vol. 3 n. 1 pp. 42-45.

LOBBA, A. (1985), "Automated Configuration Management" ■ Conference on Software tools, 1985, N.Y., N.Y., IEEE Computer Society, pp. 100-103.

MELLIER, P. (1980), "The Practical guide to Structured Systems Design", Yourdon Press.

MILLER, W. (1987), " A Software tools sampler", Prentice-Hall, Inc.

MOQUIN, B. (1985), "Software Configuration Management tools change management vs. change control", Fourth International Conference'PHOENIX Conference on Computers and Communications, pp. 97-100.

MOTTA, M. e VELASCO, F. (1990), "Um ambiente para controle de versões de software - Ambiente MS-DOS", X Congresso Sociedade Brasileira de Computação - anais, pp. 201-214.

, PANSOPHIC Systems, Inc. (1986), "VMLIB Functional Overview", PANSOPHIC Systems, Inc.

- POPE, A. (1983), "Software Configuration Management: A Quality Assurance tool", IEEE Engineering Management Conference. The new industrial revolution, pp. 59-66.
- ROCHA, A. (1987), "Análise e Projeto Estruturado de Sistemas", Editora Campus Ltda.
- SIBLEY, E. , et al (1981) , "The Software Configuration Management Database", AFIPS Conference Proceedings vol. 50 National Computer Conference pp. 249-55.
- STANLEY, M. (1983), "Software Configuration Management in an integrated programing support environment, Royal Signals and radar establshment Memorandum - 3578, Controller HMBO London).
- TILBURG, R. (1985), "Software Configuration Management an update", Fourth International Conference'PHOENIX Conference on Computers and Communications pp. 101-5.
- VALLINO, J. (1987), "Tracking Code Modules", Tech PC Journal vol. 5 n. 9. pp. 123-131
- VINCENT, J., et al (1988) , "Software Quality Assurance - v. 1 Practice and Implementation", Prentice-Hall, Inc.
- WILBURN, N. (1983), "Guidelines - Software Configuration Management", Hanford Engineering Development Laboratory - HEDL-TC-2263.
- YOURDON, E. (1989), "Modern Structured Analysis", Prentice-Hall International, Inc.