

# A INFLUÊNCIA DA CONFIGURAÇÃO DE SOFTWARE NA MANUTENÇÃO

Rosely Sanches

ICMSC-USP

Av. Dr. Carlos Botelho, 1465

13560-250, São Carlos, SP

email: rsanches@icmsc.sc.usp.br

Nicolau Reinhard

FEA-USP

Av. Prof. Luciano Gualberto, 908

05508 - São Paulo, SP

email: reinhard@sox.cce.usp.br

## RESUMO

*As dificuldades enfrentadas durante a fase de manutenção podem ser reduzidas com o melhoramento da configuração de software. Esse melhoramento depende de atividades de gerenciamento da configuração que devem ser efetuadas durante toda a construção do software e posteriormente em suas manutenções. Neste trabalho relatam-se os resultados de uma pesquisa sobre o gerenciamento da configuração de software, numa tentativa de demonstrar que certas características do mesmo podem contribuir para facilitar as futuras manutenções que o software venha a sofrer. Para obtenção dos dados utilizou-se a metodologia denominada Goal / Question / Metric (GQM) e foram realizadas entrevistas em 8 empresas, analisando-se 28 sistemas de software para computadores de grande porte e 86 manutenções que haviam sido efetuadas nos mesmos. Através de análises estatísticas, a configuração de software mostrou estar correlacionada com o esforço de manutenção. Além disso, a pesquisa também mostrou que, na maioria dos sistemas analisados, o código é o único item de informação da configuração efetivamente administrado. Os outros itens, quando existem, não podem ser considerados integros pois são mínimas as atividades de controle efetuadas sobre as alterações dos mesmos.*

**PALAVRAS CHAVE:** Manutenção de Software, Gerenciamento da Configuração de Software

## 1. INTRODUÇÃO

No processo de construção do software, independentemente da área de aplicação, do tamanho ou da complexidade do mesmo, ocorrem duas fases genéricas: definição e desenvolvimento.

Na *fase de definição* identificam-se as informações a serem processadas e as funções e desempenho desejados; estabelecem-se as interfaces; verifica-se se existem restrições de projeto e definem-se os critérios de validação do software. Na *fase de desenvolvimento* projetam-se a estrutura dos dados, a arquitetura do software e os detalhes procedimentais de implementação; traduzem-se os projetos para uma linguagem de programação e efetuam-se os testes (PRESSMAN, 1992).

Quando termina o processo de construção, o produto é liberado, o software é instalado para uso operacional e começa então a *fase de manutenção*. A definição de *manutenção* dada no

glossário de terminologias de engenharia de software do IEEE (ANSI/IEEE, 1983) é a seguinte:

*"Modificação de um software produto, depois de liberado, para corrigir falhas, melhorar o desempenho ou outros atributos, ou para adaptar o produto a um ambiente alterado."*

Por mais sofisticado que seja um sistema de software ou a instalação onde ele é processado, certos tipos de alteração sempre necessitam ser efetuados, o que torna a manutenção um fato da vida real, que não pode ser evitado. Entretanto, pode-se diminuir o esforço necessário às alterações do software, através do controle adequado dos fatores que afetam a facilidade de manutenção do mesmo ("maintainability"). Um desses fatores é a *Configuração de Software*.

Uma das primeiras pesquisas que forneceu um panorama geral da função manutenção foi efetuada por Lientz e Swanson em 1980 (LIE et alii, 1978; LIE et alii, 1980; LIE et alii, 1981) e envolveu 487 organizações de processamento de dados dos Estados Unidos. Embora o objetivo dessa pesquisa tenha sido diferente, ela identificou o fator *Qualidade do Produto* como um dos seis fatores mais influenciadores do esforço de manutenção. Esse fator, embora não completamente, corresponde ao que chamariamos hoje de qualidade da Configuração de Software.

Em um outro estudo, efetuado por DEKLEVA (1992), a técnica Delphi foi usada para a compilação dos principais fatores problemas na manutenção do software. Sessenta e dois profissionais de manutenção participaram de três rodadas realizadas em junho, julho e setembro de 1991, tentando chegar num consenso dos principais problemas com manutenção. Nessa pesquisa, um dos fatores identificados como "causador" de problemas foi a *Documentação do Sistema Incompleta ou Inexistente*. Tal fator, de certo modo, também está relacionado à Configuração de Software.

Outras pesquisas empíricas que envolvem, de algum modo a influência da configuração de software no esforço de manutenção são apresentadas resumidamente, no Quadro 1.

O propósito deste artigo é apresentar os resultados de uma pesquisa realizada em São Paulo, em 1993, onde se verificou a influência da configuração de software no esforço de manutenção (SANCHES, 1994). Essa pesquisa tratou conjuntamente de outros fatores relacionados ao software e ao processo de manutenção, porém apenas os resultados relacionados à configuração são apresentados neste artigo.

Na Seção 2 descreve-se a configuração de software e as atividades de gerenciamento que devem ser efetuadas nos itens de informação da configuração de modo que a mesma seja um "retrato" do software em qualquer momento de seu ciclo de vida. A pesquisa empírica é apresentada na Seção 3, através da metodologia denominada Goal/Question/Metric. O último passo da metodologia, o qual corresponde à análise dos dados, é apresentado na Seção 4. Finalmente, na Seção 5 são apresentadas as conclusões e contribuições deste trabalho.

**QUADRO 1**  
**Pesquisas Empíricas Relacionadas à Configuração de Software**

VARIAVEIS DEPENDENTES	METODOLOGIA	RESULTADOS RELATADOS	REFERENCIA
Pontos no teste de compreensão	Experiência de laboratório	Nos dois experimentos efetuados, os grupos que receberam os diagramas de estrutura de dados obtiveram mais pontos nos testes de compreensão que os grupos sem ajuda, ou com documentação do fluxo de controle	SHNEIDERMAN 1982
Gastos com manutenção	Estudo de campo	Os gastos com manutenção tendem a crescer com a idade do software. Programas re-desenvolvidos (substituição funcional) tendem a gastar menos com manutenção que os originais. Documentação inadequada aumenta significativamente os gastos com manutenção. Programas escritos em linguagem ASSEMBLER gastam mais com manutenção que os escritos em linguagem de nível mais alto	GUIMARAES, 1983
Areas problema no processo de manutenção	Estudo de campo	A maior parte dos problemas com manutenção estão na área de compreensão - falta entendimento do ambiente de programação e da documentação	DEAN e McCUNE, 1983
Preferência subjetiva, compreensão, tempo, qualidade e erros na implementação	Experiência de laboratório	As linguagens de projeto de programas (PDLs) mostraram-se superiores aos fluxogramas, em termos de qualidade dos projetos e na preferência subjetiva. Os fluxogramas não foram superiores em nenhuma situação testada	RAMSEY et alii, 1983
Compreensão do programador	Experiência de laboratório	O estudo comprovou a necessidade de software bem documentado	WEISSMAN, 1974
Tempo de composição, compreensão, rastreamento e modificação	Experiência de laboratório	Não foi notada nenhuma diferença entre grupos que usaram fluxogramas detalhados e os que não usaram	SHNEIDERMAN, 1977

## 2. A CONFIGURAÇÃO DE SOFTWARE E OS PROCEDIMENTOS DE GERENCIAMENTO DE CONFIGURAÇÃO

Durante o ciclo de vida do software, muitos itens de informação são produzidos. Esses itens englobam programas de computador (fonte e executável), documentos que descrevem o software (direcionados aos técnicos e aos usuários), estruturas de dados (contidas nos programas ou externas a ele), itens de teste, documentos de alteração, enfim toda a informação relacionada

ao software durante sua construção e posteriormente nas eventuais manutenções efetuadas no mesmo. A todos esses itens de informação dá-se o nome de *Configuração de Software* (PRESSMAN, 1992).

A configuração de software é uma arma poderosa para o profissional de manutenção entender um software escrito e modificado por diferentes pessoas. No entanto, é muito importante a existência de itens de informação completos e atualizados (SCHNEIDEWIND, 1987; CHARETTE, 1986; CURTIS et alii, 1989). A desatualização da configuração acontece porque, cada vez que o software é submetido a uma alteração formal, os programas são alterados e essas alterações podem implicar em mudanças também nos itens de informação relacionados ao software. Para que seja possível ter uma configuração de software atualizada, é necessário que os itens de informação sejam "controlados" durante todo o ciclo de vida do software. A existência de uma configuração adequada e a garantia que os itens de informação dessa configuração reflitam a realidade, depende de efetivos *Procedimentos de Gerenciamento da Configuração*.

Esses procedimentos garantem tanto os objetivos de confiabilidade e de facilidade de manutenção quanto os de cronograma, custo e qualidade (BOZNAK, 1990). As principais atividades de gerenciamento, sugeridas por Pressman (1992), são:

## 2.1 Planejamento da Composição da Configuração

Muitos dos itens de informação produzidos durante o ciclo de vida do software são documentos técnicos de trabalho que mostram idéias usadas no desenvolvimento e que estão sujeitos a freqüentes e regulares alterações. Outros são memorandos, minutas de reuniões, planos e propostas, etc. Apesar desses itens serem interessantes para a história do projeto, eles não são críticos para futuras manutenções do sistema. Um *planejamento* adequado *dos itens* que devem compor a configuração é importante primeiro porque a existência de informações completas sobre o software é um dos aspectos críticos da manutenção (SCHNEIDEWIND, 1987; CHARETTE, 1986; BERNS, 1984; LIENTZ e SWANSON, 1980; CURTIS et alii, 1989) e segundo porque uma "super documentação", além de ser desnecessária, pode consumir uma grande quantidade de recursos (CURTIS et alii, 1979).

Pressman (1992) fornece uma série de itens de configuração, os quais são importantes para a manutenção.

- *Especificação do Sistema*: Como o software sempre faz parte de um sistema maior, a primeira atividade, anterior à fase de definição, é a determinação do escopo do software. Dessa atividade resulta o documento de especificação do sistema que deve conter uma descrição em linhas gerais do software a ser desenvolvido, definindo seus objetivos, requisitos, restrições, as necessidades e expectativas do usuário e o porquê de ser necessário e oportuno desenvolver o sistema proposto.
- *Plano de Desenvolvimento do Software*: Antes de se iniciar a construção do software, é necessário alguns dados sobre o sistema a ser construído. Por exemplo: uma descrição em linhas gerais do sistema atual e suas deficiências; uma descrição geral do sistema a ser desenvolvido; dados que permitam avaliar a viabilidade do desenvolvimento e da operação do sistema proposto e uma descrição da relação do sistema proposto com o ambiente da empresa.

- *Especificação dos Requisitos*: Nesse documento os elementos alocados para o software são definidos em detalhes. Esse documento deve conter: os Requisitos Funcionais, que expressam todas as funções a serem processadas pelo software; os Requisitos de Informação, que expressam a necessidade de informação do software; os Requisitos de Interface, que expressam as interações do sistema com o meio ambiente e com outros sistemas; os Requisitos de Desempenho, que expressam os requisitos estáticos e os requisitos dinâmicos do sistema e os Requisitos de Qualidade, que expressam as características de qualidade necessárias ao software.
- *Projeto dos Dados*: O projeto dos dados é o primeiro das três atividades de projeto durante a construção do software. Deve conter a descrição lógica dos dados bem como suas características físicas de armazenamento e acesso.
- *Projeto Arquitetural*: O principal objetivo do projeto arquitetural é desenvolver uma estrutura modular do software e representar os relacionamentos de controle entre os módulos. Além disso o projeto arquitetural funda a estrutura dos programas e as estruturas de dados e define interfaces que possibilitam que os dados fluam através dos programas.
- *Projeto Procedimental*: O projeto procedimental ocorre depois que a estrutura dos dados e dos programas foram estabelecidas. A especificação procedimental define detalhadamente os algoritmos que implementam as funções da organização modular. Deve fornecer uma especificação precisa do processamento, incluindo seqüência de eventos e pontos exatos de decisão.
- *Código Fonte*: Depois de terminado o projeto, cada componente modular do software deve ser codificado para uma linguagem de programação apropriada produzindo as listagens fonte.
- *Programas Executáveis*: Correspondem o código executável de cada um dos programas do software.
- *Documentos de Certificação*: Certificação é uma avaliação realizada sem submeter o programa à execução experimental. Pode ser realizado por métodos informais tais como inspeção e "walkthrough" ou por verificação matemática formal.
- *Informações sobre os Testes*: O passo final da fase de desenvolvimento está associado com o teste do software. Testes de unidade tentam validar o desempenho funcional de um componente modular individual do software. Testes de integração fornece um meio para construção da arquitetura do software e ao mesmo tempo testa funções e interfaces. Testes de validação verificam se todos requisitos foram atendidos. As informações sobre os testes devem relacionar-se não apenas aos planos e procedimentos de teste, mas também à massa de dados utilizada e os resultados obtidos nos testes.
- *Manual do Usuário*: Esse manual deve estar concluído no final da fase de construção do software. É uma parte da documentação manipulada pelo usuário e deve conter as informações de seu interesse: os requisitos do software, a terminologia utilizada, os modelos de documentação utilizados e a indicação de como executar o software.
- *Documentos de Manutenção*: Para todas as alterações efetuadas no software deve existir os seguintes documentos: (1) Relatório dos Problemas do Software, (2) Pedido de Manutenção, (3) Ordem de Execução da Alteração (4) Todas as informações relacionadas à alteração efetuada. Essas informações dependem do tipo de alteração efetuada e podem ser: o novo Documento de Especificação dos Requisitos, a nova Descrição do Projeto Preliminar, a nova

Descrição do Projeto Detalhado, as novas Listagens Fonte, os novos Documentos de Certificação, as Informações sobre os novos Testes, os novos Programas Executáveis ou o novo Manual do Usuário.

## **2.2 Definição de um Esquema de Denominação dos Itens de Informação**

O termo "denominação" refere-se a um esquema de identificação, para todos os itens da Configuração. Essa identificação deve ser única para cada um dos componentes da configuração. Além disso, o esquema de denominação escolhido deve ser tal que itens relacionados, tais como especificação e projeto, tenham nomes relacionados. Isso pode ser conseguido usando-se um esquema de denominação hierárquico.

## **2.3 Definição de Procedimentos de Controle das Alterações nos Itens (Controle de Versões)**

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

## **2.4 Definição de Procedimentos de Verificação nos Itens**

A identificação e controle das alterações ajudam a manter ordem, mas para assegurar que a alteração foi implementada apropriadamente, há necessidade de verificações na configuração do software. Essa verificação é uma atividade de controle de qualidade que tenta descobrir omissões ou erros na configuração que degradem os padrões de construção do software (SOMMERVILLE, 1989).

## **2.5 Definição de Procedimentos de Comunicação das Alterações**

O objetivo dessa atividade do gerenciamento da configuração é relatar a todas as pessoas envolvidas no desenvolvimento e na manutenção do software, as seguintes informações sobre a configuração de software: item de informação alterado, a alteração efetuada, o responsável pela execução da alteração, o solicitante da alteração e os outros itens afetados pela alteração. Melhorar a comunicação entre as pessoas é uma maneira de eliminar muitos problemas relativos à modificação do mesmo item de informação, com intenções diferentes e conflitantes; desperdício de esforço/tempo na tentativa de construir um software para especificações obsoletas, etc (PRESSMAN, 1992).

## **3. A PESQUISA EMPÍRICA**

A metodologia utilizada neste trabalho denomina-se "Goal/Question/Metric" (GQM) (BASILI E WEISS, 1984) e já foi utilizada em muitos projetos de pesquisa em engenharia de software (ROMBACH, 1990; SHEPPERD, 1990; ROMBACH, 1987; BASILI, 1990; BASILI et alii, 1986; BASILI e ROMBACH, 1988; ROMBACH e ULERY, 1989; GALLAGHER e LYLE, 1991).



Essa metodologia fornece um mecanismo que orienta a determinação dos objetivos de interesse da análise e o refinamento de cada objetivo em um conjunto de questões, com o propósito de quantificar os mesmos. Tais questões definem o conjunto específico dos dados a serem coletados.

A metodologia GQM consiste seis passos básicos:

PASSO 1: Estabelecer os Objetivos da Coleta de Dados

PASSO 2: Desenvolver uma Lista de Questões de Interesse

PASSO 3: Estabelecer as Métricas

PASSO 4: Projetar e Testar o Formulário de Coleta de Dados

PASSO 5: Coletar e Validar os Dados

PASSO 6: Analisar os Dados

### 3.1 Objetivos da Coleta de Dados (Passo 1)

O objetivo da pesquisa é definido em termos de:

- *Objeto da Pesquisa*: o software produto e algumas manutenções efetuadas no mesmo
- *Propósito*: relacionar características da configuração de software com o esforço de manutenção
- *Perspectiva*: os resultados desta pesquisa interessam tanto aos profissionais de desenvolvimento e manutenção de software quanto aos administradores envolvidos com a área de manutenção.
- *Ambiente*: Ambiente de desenvolvimento e manutenção de software.

### 3.2 Lista de Questões de Interesse (Passo 2)

O objetivo da pesquisa pode ser refinado em questões a serem respondidas pelo estudo. Desse modo, a questão colocada foi:

*Existe relação entre a configuração de software e o esforço de manutenção?*

### 3.3 Métricas (Passo 3)

As métricas são definidas para descrever os dados que devem ser coletados para responder às questões.

#### • O índice do Esforço de Manutenção

O índice HH mede o esforço necessário a efetivação da manutenção. O esforço de manutenção está relacionado à facilidade em se efetuar os diversos passos de cada uma das etapas do processo de manutenção: entendimento, modificação e revalidação do software. No entanto, como não é comum que as empresas guardem informações sobre o esforço individual de cada uma dessas etapas, aqui é considerado o esforço global. A métrica utilizada para avaliar o índice do Esforço de Manutenção (Tabela 1) é uma métrica *objetiva e direta* (o índice de esforço de manutenção é obtido através do tempo necessário para a efetivação da manutenção, do tempo efetivamente dedicado a ela e do número de pessoas envolvidas).

**TABELA 1**  
**Determinação do Esforço de Manutenção (HH)**

Questões	Var.	Opções de Resposta	Valores Atribuídos
Tempo aproximado desde o pedido até a efetivação da manutenção	V74	a) até 2 dias b) de 3 a 6 dias c) de 7 a 14 dias d) de 15 a 29 dias e) mais de 30 dias	1.5 4.5 10.5 22.0 40.0
Tempo de trabalho diário efetivamente dedicado à manutenção	V75	a) muito pouco tempo b) metade do tempo c) todo o tempo	0.25 0.5 1.0
Tamanho da equipe que participou da manutenção	V91	a) 1 pessoa b) 2 ou 3 pessoas c) 4 ou 5 pessoas d) 6 ou 7 pessoas e) 8 ou mais pessoas	1.0 2.5 4.5 6.5 8.5
$HH = V74 \times V75 \times V91$			

### • O Índice da Configuração de Software

A métrica utilizada para avaliar esse índice é uma métrica *indireta* (a configuração de software é avaliada através dos itens de informação que a *compõem* e das *atividades de gerenciamento* efetuadas nesses itens) e *subjéctiva* (baseada nas informações da equipe de manutenção).

A nota de cada um dos itens de informação pode ser 0, 1, 2, 4, 8 ou 16. Por exemplo, a nota zero equivale a não existência do item na configuração e a nota dezesseis corresponde a um item que existe e no qual são efetuadas todas as atividades de gerenciamento (Tabela 2).

**TABELA 2**  
**Determinação da "Nota" do Item de Informação da Configuração de Software**

Item da Configuração	Existência do Item	Atividades de Gerenciamento			
		Identificação dos Itens	Controle das Alterações	Verificação nos Itens	Comunicação Alterações
	V5	V61	V62	V63	V64
Item	Não = 0 Sim = 1	Não = 0 Sim = 2	Não = 0 Sim = 2	Não = 0 Sim = 2	Não = 0 Sim = 2
$"Nota" \text{ do Item} = V5 \times V61 \times V62 \times V63 \times V64$					

Em princípio, pensou-se em utilizar um único índice para a configuração de software, o qual seria equivalente à soma das "notas" de cada um dos itens de informação da configuração. Porém, para melhor se visualizar os itens produzidos em cada uma das etapas do ciclo de vida do software, optou-se por utilizar cinco índices:

DEFSOF: mede a Configuração com relação aos itens relacionados à Definição do Software. É equivalente à soma das "notas" dos seguintes itens de informação: *Especificação do Sistema*, *Plano de Desenvolvimento do Software* e *Especificação dos Requisitos*.



**PROJET:** mede a Configuração com relação aos itens relacionados ao Projeto do Software. É equivalente à soma das "notas" dos seguintes itens de informação: *Projeto dos Dados, Projeto Arquitetural e Projeto Procedimental*.

**CODIFIC:** mede a Configuração com relação aos itens relacionados à Codificação do Software. É equivalente à soma das "notas" dos seguintes itens de informação: *Código Fonte e Programas Executáveis*.

**TESTE:** mede a Configuração com relação aos itens relacionados ao Teste do Software. É equivalente à soma das "notas" dos seguintes itens de informação: *Documentos de Certificação e Informações sobre os Testes*.

**MANOPE:** mede a Configuração com relação aos itens relacionados à Manutenção e Operação do Software. É equivalente à soma das "notas" dos seguintes itens de informação: *Manual do Usuário e Documentos de Manutenção*.

### **3.4 Projeto e Teste do Formulário de Coleta de Dados (Passo 4)**

Na pesquisa, o questionário representou apenas um instrumento de orientação para as entrevistas realizadas. Como ele seria preenchido pelo próprio elaborador e aplicador, não houve uma preocupação maior com o esclarecimento das questões no questionário. O interesse principal foi com sua validação, ou seja, saber se realmente estava se medindo aquilo que pensava-se estar medindo. Segundo KINEEAR e TAYLOR (1979), uma maneira de se validar o questionário é através da avaliação de seu conteúdo por especialistas (*validação por conteúdo*). Assim, submeteu-se o questionário a três especialistas em manutenção de software, que o analisaram cuidadosamente e que, após sugerirem algumas alterações, que foram implementadas, consideraram o conjunto como um instrumento válido para medir os fatores considerados em nosso estudo. Além disso, o questionário utilizado na pesquisa foi submetido a um *teste piloto* com a participação de 6 profissionais de 3 empresas escolhidas unicamente para esse objetivo. Pediu-se aos participantes do teste piloto que preenchessem o questionário o qual foi discutido com os respondentes, para verificar a existência de termos mal definidos e perguntas pouco claras.

### **3.5 Coleta e Validação dos Dados (Passo 5)**

A coleta de dados foi efetuada no período de janeiro a julho de 1993, através de visitas à 8 empresas do estado de São Paulo (dois bancos, quatro empresas de processamento de dados públicas e duas fábricas multinacionais). Foram coletados dados sobre o processo de desenvolvimento de 28 sistemas de software de grande porte e dados sobre 86 manutenções efetuadas nos mesmos. Diante da dificuldade em se encontrar empresas cujos sistemas de software possuíssem documentação adequada sobre o desenvolvimento e as manutenções efetuadas, optou-se por escolher sistemas para os quais ainda existisse na empresa pelo menos um profissional que tivesse participado do desenvolvimento e da manutenção.

A validação dos dados consiste em verificar os questionários quanto à integridade e consistência. Como o questionário foi a base para a entrevista, todos os problemas detectados foram resolvidos no momento em que ocorreram.

#### 4. ANÁLISE DOS DADOS (PASSO 6)

No último passo da metodologia, os dados coletados foram analisados e interpretados. A análise dos dados compreende Resultados Descritivos Básicos, que apresenta a distribuição de frequência associada aos fatores pesquisados, e Análise dos Relacionamentos, onde as associações entre a configuração de software (separada por itens de informação relacionados ao ciclo de vida) e o esforço de manutenção são estatisticamente analisadas.

##### 4.1 Resultados Descritivos Básicos

Como foi visto na Tabela 1, a determinação do esforço de manutenção é avaliada em função do tempo para a efetivação da manutenção, o tempo efetivamente dedicado a ela e o tamanho da equipe que participou da manutenção. Em função dessas variáveis geraram-se cinco níveis de esforço através dos quais as manutenções foram classificadas. A Tabela 3 resume o resultado dessa classificação.

Nota-se nessa tabela, que nenhuma manutenção foi classificada com esforço muito baixo e apesar da maioria ser classificada com esforço médio, uma porcentagem significativa classificou-se com esforço muito alto.

**TABELA 3**  
**Esforço de Manutenção - Classificação das Manutenções**

Classe (Homens/Hora)	%
Muito Baixo [0.38, 4.5]	0.0
Baixo [4.88, 11.0]	18.60
Médio [11.25, 25.0]	45.35
Alto [26.25, 65.0]	15.12
Muito Alto [68.25, 340.0]	20.93

A pesquisa também apresenta os itens de informação e as atividades de gerenciamento da configuração de software que foram sugeridas por PRESSMAN (1992) e solicita que seja informado, para os sistemas analisados, quais itens estão presentes na configuração e as atividades de gerenciamento efetuadas nos mesmos. A Tabela 4 resume a frequência das respostas para essa questão.

Pode-se notar a partir dessa tabela que apesar dos itens relacionados à definição do software estarem presentes na maioria dos sistemas pesquisados, também para a maioria, as únicas atividades de gerenciamento são o controle e a comunicação das alterações na Especificação dos Requisitos. Para os outros itens não existe nenhuma atividade de gerenciamento. Nota-se, também, que para a maioria dos sistemas, existe controle e comunicação das alterações no Projeto Arquitetural e todas as atividades de gerenciamento para o Projeto dos Dados. O Projeto Procedimental existe em cerca de metade dos sistemas de software analisados, sendo que, para a maioria, nenhuma atividade de gerenciamento é efetuada. Os itens relacionados à codificação estão presentes em todos os sistemas analisados e, na quase totalidade, são bem gerenciados. Os itens relacionados ao teste, à manutenção e à operação do software estão presentes em apenas metade dos sistemas analisados e, na maioria deles, nenhuma atividade de gerenciamento é efetuada.

**TABELA 4**  
**Composição e Gerenciamento da Configuração - Frequências de Respostas**

Ítems da Configuração	Composição		Atividades de Gerenciamento							
	Existência		Identificação dos Ítems		Controle das Alterações		Verificação nos Ítems		Comunicação Alterações	
	Sim %	Não %	Sim %	Não %	Sim %	Não %	Sim %	Não %	Sim %	Não %
<b>Ítems de Definição</b>										
Especificação do Sistema	57.1	42.9	25.0	75.0	21.4	78.6	7.1	92.9	21.4	78.6
Plano Desenvolvim. Softw.	60.7	39.3	25.0	75.0	17.9	82.1	7.1	92.9	14.3	85.7
Especif. Requisitos Softw.	85.7	14.3	46.4	53.3	53.6	46.4	39.3	60.7	57.1	42.9
<b>Ítems de Projeto</b>										
Descr. Projeto dos Dados	78.6	21.4	57.1	42.9	57.1	42.9	50.0	50.0	57.1	42.9
Descr. Proj. Arquitetural	75.0	25.0	46.4	53.6	53.6	46.4	50.0	50.0	53.6	46.4
Descr. Proj. Procedimental	57.1	42.9	21.4	78.6	25.0	75.0	10.7	89.3	14.3	85.7
<b>Ítems de Codificação</b>										
Código Fonte	100.0	0.0	96.4	3.6	100.0	0.0	42.9	57.1	100.0	0.0
Programas Executáveis	100.0	0.0	96.4	3.6	100.0	0.0	21.4	78.6	96.4	3.6
<b>Ítems de Teste</b>										
Doc. de Certificação	21.4	78.6	7.1	92.9	0.0	100.0	0.0	100.0	0.0	100.0
Informações sobre Testes	50.0	50.0	17.9	82.1	0.0	100.0	0.0	100.0	3.6	96.4
<b>Ítems Operação/Manuten</b>										
Manual do Usuário	75.0	25.0	53.6	46.4	42.9	57.1	28.6	71.4	35.7	64.3
Doc. de Manutenção	46.4	53.6	32.1	67.9	17.9	82.1	7.1	92.9	7.1	92.9

## 4.2 Análise dos Relacionamentos

Em resposta à questão da pesquisa, procurou-se efetuar análises que mostrassem o relacionamento entre o esforço de manutenção e as características da configuração de software consideradas.

### 4.2.1 Análise de Correlação

Numa primeira análise, procurou-se verificar qual a correlação entre cada um dos índices considerados e o esforço de manutenção. Para isso, utilizou-se a Análise de Correlação de Kendall. Os coeficientes da correlação de Kendall ( $\tau$  de Kendall) medem o grau relativo de concordância ou discordância entre duas variáveis, usando a "classificação" dos valores dos dados, ao invés dos valores em si.

Para o processo de análise, geraram-se 5 tabelas (uma para cada um dos índices: DEFSOF, PROJ, CODIFIC, TESTE, MANOPE). Cada uma das tabelas contendo, para cada valor possível do índice, o esforço médio de manutenção calculado sobre todas as ocorrências que apresentaram esse valor de índice. As 5 tabelas geradas, foram submetidas à análise de correlação de Kendall. O resultado é mostrado da Tabela 5.

TABELA 5

Relacionamento entre os Índices da Configuração de Software e o Esforço de Manutenção

Fatores	$\tau$ de Kendall e Nível de Significância
Configuração do Software: Itens de Manutenção e Operação	-0.61 (0.0061)
Configuração do Software: Itens de Definição	-0.49 (0.0116)
Configuração do Software: Itens de Teste Software	-1.00 (0.0143)
Configuração do Software: Itens de Projeto	-0.47 (0.0153)
Configuração do Software: Itens de Codificação	-1.00 (0.0415)

A partir dessa tabela, e considerando-se o nível de significância de 0.05, constata-se que existe correlação negativa entre todos os índices relativos à configuração de software e o esforço de manutenção.

#### 4.2.2 Identificação de Grupos de Manutenções Similares

Nessa análise, procurou-se verificar se as manutenções estudadas poderiam ser agrupadas em um conjunto de manutenções similares que pudessem ser identificadas por algumas características relevantes. Para tanto utilizou-se o procedimento estatístico denominado Análise de Cluster. Esse procedimento permite agrupar observações de um conjunto de dados multivariados em clusters de pontos "similares" (similar, nesse caso, significa que as observações ficariam próximas umas das outras se elas pudessem ser colocadas num espaço multi dimensional). Um método usado para formar clusters é através do fornecimento de um conjunto inicial de pontos "sementes". Nesse procedimento, essas "sementes" são usadas para inicializar a formação dos clusters.

Foram fornecidos como "sementes" três pontos com valores mínimos, médios e máximos em cada um dos índices considerados. As 86 manutenções pesquisadas, caracterizadas em termos dos índices DEFOSF, PROJET, CODIFIC, TESTE e MANOPE descritos no item 3.3, foram submetidas a esse procedimento de análise.

Separando-se as observações nos 3 conjuntos formados, calculou-se, para cada um deles, a média e o desvio padrão dos índices considerados na formação do cluster. A Tabela 6 apresenta o resumo dessa análise. Nota-se, com base nos desvios padrão, que as observações incluídas num mesmo cluster, quando comparadas com o conjunto total de observações, são mais homogêneas nos valores dos índices.

Observando-se essa tabela, pode-se notar que o primeiro cluster está constituído por observações que apresentam, em média, os melhores índices dos fatores relacionados à configuração de software. O terceiro cluster contém observações que apresentam, em média, os piores índices desses mesmos fatores. O segundo cluster apresenta situação intermediária.

A fim de se verificar qual o comportamento do esforço de manutenção em relação aos índices pesquisados calculou-se, para cada grupo, o esforço médio de manutenção (em homens/hora) dos sistemas de software. O resultado dessa análise é apresentado na Tabela 7.

**Tabela 6**  
**Características dos Clusters**

CARACTERÍSTICAS (Valores Mínimo e Máximo)	CLUSTERS			TOTAL
	1	2	3	
	Média Desvio	Média Desvio	Média Desvio	Média Desvio
Configuração de Software: Itens de Definição (0, 48)	40 10.47	11.23 7.89	1.39 1.58	10.36 12.49
Configuração de Software: Itens de Projeto (0, 48)	45 5.55	21.32 10.91	0.84 1.19	16.14 15.69
Configuração de Software: Itens de Codificação (0, 32)	30 3.70	21.62 5.76	14.71 3.41	19.91 6.65
Configuração de Software: Itens de Teste (0, 32)	3.5 0.93	1.04 0.83	0.13 0.34	0.94 1.16
Configuração de Software: Itens de Operação e Manutenção (0, 32)	27 5.55	8.11 6.34	1.26 2.67	7.40 8.77
Nº de Observações	8	47	31	86

**Tabela 7**  
**Outra Característica dos Clusters**

CARACTERÍSTICAS	CLUSTERS			TOTAL
	1	2	3	
	Média	Média	Média	Média
Esforço de Manutenção	9.33	15.79	73.62	36.03

Analisando-se conjuntamente as duas tabelas (Tabelas 6 e 7), verifica-se facilmente que existe uma correspondência entre os valores dos índices utilizados para a formação dos clusters e o esforço de manutenção. O cluster 1, por exemplo, que engloba as manutenções que exigiram, em média, menor esforço, apresenta também os melhores valores para os índices de configuração de software. Por outro lado, o cluster 3, que inclui as manutenções com piores índices é o que exige, em média, o maior esforço de manutenção.

## 5. CONCLUSÕES

Adotando-se a hipótese de que algumas características da configuração de software podem fornecer informações que expliquem o esforço de manutenção, obteve-se um conjunto de dados de 28 sistemas de software e 86 manutenções de 8 empresas paulistas. Esses dados foram analisados, procurando-se verificar a influência dessas características sobre o esforço de manutenção. Em duas análises estatísticas realizadas, a configuração de software, avaliada através dos itens de informação que a compõem e das atividades de gerenciamento efetuadas nos mesmos, mostrou ter correlação direta com o esforço de manutenção. Analisando-se separadamente a configuração através dos itens de informação produzidos em cada fase do ciclo de vida do software, verificou-se que os itens referentes à manutenção e operação do software, os itens referentes à definição, os itens referentes ao teste e os itens referentes ao projeto do software foram os que mais se correlacionaram com o esforço de manutenção. Os itens relacionados à codificação apresentaram uma correlação baixa, donde se pode inferir que não é

suficiente, apenas, manter um controle de versões do código, para se ter facilidade na manutenção. Existem outros itens da configuração mais influentes.

Embora a amostra não seja representativa, pode-se notar algumas características que de certo modo confirmam a imagem que se tem, informalmente, da maioria dos sistemas de software de grande porte atuais. Primeiramente, verifica-se, a partir da Tabela 3, que para esses sistemas o esforço de manutenção ainda continua alto. Nota-se também (Tabela 4), que quase todos os itens de configuração pesquisados existem, pelo menos, para a metade dos sistemas pesquisados. Porém, as atividades de gerenciamento efetuadas só são completas para os itens de codificação. Esse fato parece indicar que existe no início do desenvolvimento do software uma certa preocupação com a configuração. Porém com o decorrer do tempo, com as alterações que se processam e com as dificuldades encontradas, essa configuração não é atualizada completamente, mantendo-se apenas um controle de versões do código.

Os resultados desta pesquisa não são surpreendentes para a área de engenharia de software. A teoria relacionada reconhece amplamente o valor do gerenciamento da configuração como um dos atributos para a facilidade de manutenção do software. Porém essas teorias são normativas, colocam o que deveria se feito e raramente são submetidas a testes empíricos. Com isso, seus valores não são reconhecidos pela maioria dos profissionais de informática que poderiam fazer uso de suas conclusões.

A principal contribuição deste trabalho é justamente mostrar, através da pesquisa empírica, que a existência de uma configuração de software completa e íntegra é um fator determinante para a diminuir o esforço de manutenção. Além disso, esta pesquisa mostra que a principal dificuldade dos profissionais é gerenciar a configuração de software durante todo o ciclo de vida. Isso indica que a atividade de gerenciamento de configuração deve ser "automatizada" para que possa ser realizada adequadamente.

Finalmente, ressalta-se que a facilidade de manutenção ("maintainability") é uma das características através da qual um software é avaliado quanto à qualidade. Assim, a existência de uma configuração de software completa e íntegra contribui também para promover um software quanto sua qualidade.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

- ANSI/IEEE (1983) - An American National Standard IEEE Standard Glossary of Software Engineering Terminology - *IEEE Transaction on Software Engineering*
- BASILI, V.R. e WEISS, D.M. (1984) - A Methodology for Collecting Valid Software Engineering Data - *IEEE Transaction on Software Engineering*, Vol.SE-10, No. 6, p.728-738
- BASILI, V.R.; SELBY, R.W. e HUTCHENS, D.H. (1986) - Experimentation in Software Engineering - *IEEE Transaction on Software Engineering*, Vol.SE-12, No.7, p.733-743
- BASILI, V. e ROMBACH, H.D. (1988) - The Tame Project - *IEEE Transaction on Software Engineering*, Vol.14, No. 6, p.761-762
- BASILI, V.R. (1990) - Viewing Maintenance as Reuse-Oriented Software Development - *IEEE Software*, p.19-25
- BERNS, G.M. (1984) - Assessing Software Maintainability - *Communications of the ACM*, Vol.27, No. 1, p.14-23
- BOZNAK, R.G. (1990) - Improving Reliability and Maintainability Programs with Configuration Management - The Next Revolution - *Proc. Reliability and Maintainability Symposium*, p.109- 112



- CHARETTE; R.N. (1986) - *Software Engineering Environments* - McGraw- Hill, Inc. New York, NY
- CURTIS, B.; SHEPPARD, S.; MILLIMAN, P.; BORST, M. e LOVE T. (1979) - Measuring the Psychological Complexity of Software Maintenance Tasks With the Halsted and McCabe Metrics - *IEEE Transaction on Software Engineering*, Vol. SE-5, No 2, p.96-104
- CURTIS, B.; SHEPPARD, S.B.; BAILEY, E.K.; BAILEY, J. e BOEHM, D. (1989) - Experimental Evaluation of Software Documentation - *Journal of Systems and Software*, Vol. 9, p.167-207
- DEAN, J.S. e McCUNE, B.P. (1983) - An Informal Study of Software Maintenance Problems - *Proc. IEEE Conference on Software Maintenance*, p.137-139
- DEKLEVA S. (1992) - Delphi Study of Software Maintenance Problems - *Proc. IEEE Conference on Software Maintenance*, p. 10- 17
- GALLAGHER e LYLE (1991) - Using Program Slicing in Software Maintenance - *IEEE Transaction on Software Engineering*, Vol. 17, No. 8, p.759-761
- GUIMARAES, T. (1983) - Managing Application Program Maintenance Expenditures - *Communications of the ACM*, Vol. 26, No. 10, p.739-746
- KINNEAR, T.C. e TAYLOR, J. R. (1979) - *Marketing Research: An Applied Approach* - McGraw-Hill, Inc. New York, NY
- LIENTZ, B.P.; SWANSON, E.B. e TOMPKINS, G.E. (1978) - Characteristics of Application Software Maintenance - *Communications of the ACM*, Vol.21, No.6, p.466-471
- LIENTZ, B.P. e SWANSON, E.B. (1980) - *Software Maintenance Management* - Addison-Wesley Publishing Company - Philippines
- LIENTZ, B.P. e SWANSON, E.B. (1981) - Problems in Application Software Maintenance - *Communications of the ACM*, Vol. 24, No. 11, p.31-37
- PRESSMAN, R.S. (1992) - *Software Engineering: A Practitioner's Approach* - McGraw-Hill Book Co. 3<sup>o</sup> edition.
- RAMSEY, H.R.; ATWOOD, M.E. e VAN DOREN, J.R. (1983) - Flowcharts Versus Program Design Languages: An Experimental Comparison - *Communications of the ACM*, Vol. 26, No. 6, p.445-449
- ROMBACH, H.D. (1987) - A Controlled Experiment on the Impact of Software Structure on Maintainability - *IEEE Transaction on Software Engineering*, Vol. SE-13, No. 3), p.344-354
- ROMBACH, H.D. e ULERY, B.T. (1989) - Improving Software Maintenance through Measurement - *Proceedings of the IEEE*, Vol.77, No.4, p.581-595
- ROMBACH, H.D. (1990) - Design Measurement: Some Lessons Learned - *IEEE Software*, 17-24
- SANCHES, R. (1994) - A Influência do Software e de seu Processo de Manutenção no Esforço de Manutenção - *Tese de Doutorado - FEA-USP, março 1994.*
- SCHNEIDEWIND, N.F. (1987) - The State of Software Maintenance - *IEEE Transaction on Software Engineering*, Vol. SE-13, No.3, p.303-310
- SHEPPERD, M. (1990) - Design Metrics: An Empirical Analysis - *Software Engineering Journal*, Vol. 5, No. 1, p.3-10
- SHNEIDERMAN, B.; MAYER, R.; MCKAY, D. e HELLER, P. (1977) - Experimental Investigations of the Utility of Detailed Flowcharts in Programming - *Communications of the ACM*, Vol.20, No.6, p.373-381
- SHNEIDERMAN, B. (1982) - Control Flow and Data Structure Documentation: Two Experiments - *Communications of the ACM*, Vol 25, No. 1, p.55-63
- SOMMERVILLE, I. (1989) - *Software Engineering* - Addison-Wesley Publishing Company 3.ed.
- WEISSMAN, L. (1974) - Psychological Complexity of Computer - *SIGPLAN Notices*, Vol. 9, No. 6, p.25-36