

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**  
(Bacharelado)

**SOFTWARE DE APOIO A MANUTENÇÃO DE SISTEMAS**  
**BASEADO EM NORMAS DE QUALIDADE**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE  
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA  
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA  
COMPUTAÇÃO — BACHARELADO

**CHARLES HOPPE**

BLUMENAU, DEZEMBRO/1999

1999/2-05

# **SOFTWARE DE APOIO A MANUTENÇÃO DE SISTEMAS BASEADO EM NORMAS DE QUALIDADE**

**CHARLES HOPPE**

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Everaldo Artur Grahl — Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

## **BANCA EXAMINADORA**

---

Prof. Everaldo Artur Grahl

---

Prof. Marcel Hugo

---

Prof. Oscar Dalfovo

# ÍNDICE

Lista de Figuras .....	VII
Lista de Tabelas .....	VIII
Lista de Quadros .....	IX
Agradecimentos .....	XII
Resumo .....	XII
Abstract.....	XIV
1 Introdução .....	15
1.1 Origem.....	15
1.2 Objetivos.....	16
1.3 Organização .....	16
2 Manutenção de Software .....	18
2.1 Conceito.....	18
2.2 Tipos de Manutenção.....	19
2.2.1 Manutenção Corretiva .....	20
2.2.2 Manutenção Adaptativa.....	20
2.2.3 Manutenção Aperfeiçoante.....	21
2.3 Formas de Manutenção.....	22
2.3.1 Manutenção estruturada de software .....	22
2.3.2 Manutenção não-estruturada de software .....	22
2.4 Objetivos da Manutenção de Software.....	23
2.5 Problemas na Manutenção de Software.....	24
2.6 Tarefas da Manutenção.....	26
2.6.1 Organização para a Manutenção.....	26

2.6.2 Relatórios.....	27
2.6.3 O Fluxo dos eventos .....	28
2.6.4 Conservação dos registros .....	30
2.6.5 Avaliação.....	30
2.6.6. Tarefas técnicas da Manutenção de Software.....	30
3 Normas e modelos de qualidade .....	33
3.1 Norma ISO/IEC 12207 – Processos de ciclo de vida de software.....	33
3.1.1 Processos fundamentais.....	34
3.1.2 Processos de Apoio.....	35
3.1.3 Processos organizacionais .....	36
3.2 Manutenção de software segundo a norma ISO/IEC 12207.....	37
3.2.1 Implementação do processo manutenção .....	38
3.2.2 Análise do problema e da modificação.....	38
3.2.3 Implementação da modificação .....	39
3.2.4 Revisão/Aceitação da Manutenção.....	40
3.2.5 Migração.....	40
3.2.6 Descontinuação do software.....	41
3.2.7 Roteiro de Manutenção.....	42
3.3 ISO 9000-3 .....	45
3.3.1 Estrutura do sistema de qualidade .....	47
3.3.2 Atividades do ciclo de vida do software.....	48
3.3.3 Atividades de suporte do sistema da qualidade .....	50
3.4 Manutenção de Software segundo a norma ISO 9000-2 .....	51
3.4.1 Plano de Manutenção.....	51

3.4.2 Procedimentos de liberação .....	52
3.5 Spice .....	53
3.5.1 Cliente-fornecedor .....	55
3.5.2 Engenharia .....	56
3.5.3 Suporte.....	57
3.5.4 Gerência.....	58
3.5.5 Organização .....	59
3.5.6 Níveis de Capacitação .....	60
3.6 Manutenção de Software Segundo SPICE .....	61
3.7 Comparativo entre normas e modelos .....	62
3.7.1 Explicação do comparativo .....	64
4 Proposta de Manutenção.....	68
4.1 Implantação do processo .....	71
4.1.1 Registrar os problemas/modificações do software .....	71
4.1.2 Receber os registros do problema/modificações .....	72
4.1.3 Analisar autenticidade do pedido .....	73
4.1.4 Registrar a pendência.....	74
4.2 Análise do problema e modificação .....	74
4.2.1 Priorizar as solicitações .....	75
4.2.2 Análise do problema e modificação .....	76
4.2.3 Registrar causas do problema .....	77
4.2.4 Registrar alterações da modificação .....	78
4.2.5 Aceitação e homologação da solução .....	78
4.3 Implantação da solução .....	79

4.3.1 Verificar coerência das informações .....	79
4.3.2 Registrar as alterações .....	80
4.3.3 Testes preliminares .....	80
4.3.4 Avaliar partes do software não modificados .....	81
4.3.5 Registros de envio para testes finais .....	82
4.4 Revisão/Testes e aceitação .....	82
4.4.1 Efetuar plano de testes .....	83
4.4.2 Executar plano de testes .....	84
4.4.3 Validar os resultados .....	84
4.4.4 Registrar liberação .....	85
4.5 Qualificação da equipe .....	85
4.6 Verificação Periódica .....	86
4.7 Formulários propostos .....	86
5 Software de apoio a Manutenção de Sistemas.....	90
5.1 Diagrama de contexto e DFD .....	90
5.2 Modelo Entidade Relacionamento.....	92
5.3 Dicionário de Dados .....	93
5.4 Implementação do software.....	98
5.5 Principais telas do software .....	99
5.5.1 Telas de Cadastro.....	99
5.5.2 Telas de pendência e análise.....	101
5.5.3 Telas de registro.....	104
5.5.4 Relatórios.....	104
6 Conclusões.....	106
Referências Bibliográficas.....	107

## LISTA DE FIGURAS

Figura 1 – Alocação de Tempo para Manutenção .....	19
Figura 2 – Organização para Manutenção .....	27
Figura 3 – Fluxo dos Eventos .....	29
Figura 4 – Roteiro de Manutenção .....	43
Figura 5 - Estrutura da norma ISO 9000-3 .....	47
Figura 6 – Avaliação de Processo de Software – SPICE .....	54
Figura 7 – Diagrama de Contexto .....	90
Figura 8 – DFD.....	91
Figura 9 – Modelo Entidade Relacionamento – Lógico.....	92
Figura 10 – Dicionário de Dados.....	93
Figura 11 – Modelo Entidade Relacionamento – Físico .....	97
Figura 12 - Tela de Manutenção de Sistemas.....	99
Figura 13 - Tela de Manutenção Componentes.....	100
Figura 14 - Tela de Manutenção de Funcionários .....	100
Figura 15 - Tela de Manutenção de Clientes.....	101
Figura 16 - Tela de Solicitação de Correção .....	102
Figura 17 - Tela de Solicitação de Modificação.....	102
Figura 18 - Tela de Análise da Pendência de Correção.....	103
Figura 19 - Tela de Análise da Pendência de Modificação .....	103
Figura 20 - Tela de Registro de Manutenção .....	104
Figura 21 - Relatório de Pendências de Correções.....	105
Figura 22 - Relatório de Registro de Manutenção.....	105

## LISTA DE TABELAS

Tabela 1 – A série ISO 9000 .....	45
Tabela 2 – Processos Cliente/Fornecedor.....	56
Tabela 3 – Processos de Engenharia de Software .....	57
Tabela 4 – Processos de Suporte de Software .....	58
Tabela 5 – Processos de Gerência de Software .....	59
Tabela 6 – Processos de Organização.....	59
Tabela 7 – Níveis de Capacitação .....	60
Tabela 8 – Comparativo entre as normas de Qualidade .....	62



## LISTA DE QUADROS

Quadro 01 – Ciclo de Vida do Processo Proposto de Manutenção.....	70
Quadro 02 – Plano de Implantação do Processo de Manutenção.....	71
Quadro 03 – Tarefas para registro de problema .....	71
Quadro 04 – Tarefas para registro de modificação.....	72
Quadro 05 – Tarefas do recebimento do registro de problema .....	72
Quadro 06 – Tarefas do recebimento do registro de modificação.....	73
Quadro 07 – Análise de Autenticidade do Pedido de correção .....	73
Quadro 08 – Análise de autenticidade de modificação do software.....	74
Quadro 09 – Tarefas de Registro de Pendência.....	74
Quadro 10 – Plano de Análise do Problema e Modificação.....	75
Quadro 11 – Tarefas de Priorização de Solicitações .....	75
Quadro 12 – Tarefas da Análise do Problema.....	76
Quadro 13 - Tarefas da Análise da Modificação do Software .....	77
Quadro 14 – Tarefas para Registrar Causas do Problema.....	77
Quadro 15 – Tarefas de Registro de Alterações da Modificação .....	78
Quadro 16 – Tarefas da Aceitação e Homologação da Solução.....	78
Quadro 17 – Plano de Implementação da Solução .....	79
Quadro 18 – Tarefas da Verificação de Coerência da Análise.....	80
Quadro 19 – Tarefas de Registro de Alterações .....	80
Quadro 20 – Tarefas dos Testes Preliminares .....	81
Quadro 21 – Tarefas da Avaliação das partes do software não modificadas .....	81
Quadro 22 – Tarefas do Registro de Envio para Testes Finais .....	82

Quadro 23 – Tarefas de Revisão, Testes e Aceitação.....	83
Quadro 24 – Tarefas do Plano de Testes .....	83
Quadro 25 – Tarefas da Execução dos testes .....	84
Quadro 26 – Tarefas da Validação dos testes.....	84
Quadro 27 – Tarefas da Liberação do Software .....	85
Quadro 28 – Plano de qualificação de equipe .....	85
Quadro 29 – Tarefas da Verificação Periódica.....	86
Quadro 30 – Formulário de Solicitação de Correção .....	87
Quadro 31 – Formulário de Solicitação de Modificação.....	87
Quadro 32 – Formulário de Análise da Solicitação de Problema .....	88
Quadro 33 – Formulário de Análise de Solicitação de Modificação.....	88
Quadro 34 – Formulário de Registro de Manutenção .....	89

## DEDICATÓRIA

Dedico este trabalho aos meus pais Heinz e Erika por toda ajuda, incentivo, compreensão e carinho que tiveram comigo durante mais esta etapa da minha vida.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus pela vida, saúde e por tudo que ele representa na minha vida.

A minha amiga e companheira Deise Lilian Stoeckke pelo carinho e compreensão que teve comigo nesta difícil fase da minha vida.

Aos professores do curso de Bacharelado em Ciências da Computação da Universidade Regional de Blumenau, que de alguma forma contribuíram para o meu aperfeiçoamento acadêmico.

Agradeço especialmente ao Professor Everaldo Artur Grahl por toda sua dedicação e orientação durante a confecção deste trabalho.

## **RESUMO**

Este trabalho descreve conceitos de manutenção de software e apresenta um estudo comparativo entre três normas de qualidade ISO/IEC 12207, ISO 9000-3 e SPICE. Propõe-se um processo de manutenção de software baseado nas normas estudadas, bem como se desenvolveu um software para o apoio a manutenção de software.

## **ABSTRACT**

*This work describes concepts of software maintenance. It presents a comparative study among three quality standards ISO/IEC 12207, ISO 9000-3 and SPICE. Proposes a process of software maintenance. It also presents a software for the support the systems maintenance.*



# 1 INTRODUÇÃO

## 1.1 ORIGEM

A engenharia de software é considerada uma das áreas da computação mais importantes desta década, devido à crescente disseminação do uso do software, como parte integrante das mais variadas áreas da sociedade. Entretanto o desenvolvimento, manutenção e demais tarefas relacionadas ao software, são relativamente novos se comparadas às áreas tradicionais da engenharia. Como resultado, não há na produção de software o mesmo rigor existente nos projetos tradicionais de engenharia. Assim, existe um nível elevado de insatisfação com o software, tanto por parte dos usuários, que não vêem as suas necessidades atendidas, quanto com relação à organização que o desenvolve, por não conseguir torná-lo econômica e comercialmente viável [KRA98].

Atualmente, uma das atividades do desenvolvimento de software, que apresenta dificuldades e falta de apoio aos profissionais é a manutenção de software. Então os profissionais da área de desenvolvimento de software, fundamentalmente relacionados a manutenção de software, têm por tendência executar manutenções no sistema de forma incorreta ou até mesmo apenas resolvendo problemas atuais e não se preocupando com possíveis alterações que o software necessitará no futuro.

Segundo [FOU94] e [PAR90], a manutenção de software, nos dias atuais, é uma tarefa relativamente importante, visto a velocidade com que o ambiente empresarial é modificado e devido seu alto custo, sendo que cerca de 60% do custo do software de uma empresa vem da fase de manutenção, seja ela, adaptativa(adaptar o software a realidade da empresa), corretiva (corrigir erros do software) ou aperfeiçoante (realizar melhoramentos no programa).

Na maioria das empresas as manutenções são mal estruturadas e feitas de maneira desorganizada e individualizada, quase que intuitivamente, sem cumprimento de métodos ou padrões específicos. Para uma manutenção adequada de um software, devem ser



considerados os testes, o tempo que levou para a conclusão da manutenção e a metodologia empregada.

Dentro deste contexto, a comunidade de informática vem criando normas para regular e orientar a atividade de produção de software. Como exemplos pode-se citar as normas ISO 9000-3, ISO/IEC 12207, SPICE que foram escolhidas pôr serem normas mais conhecidas e de mais fácil acesso, embora trabalhem os processos em níveis diferentes. Mesmo com o auxílio das regras definidas pelas normas, a sua aplicação fica prejudicada sem o auxílio de uma ferramenta que facilite sua adoção.

## **1.2 OBJETIVOS**

Este trabalho possui como objetivo geral especificar e implementar um software de apoio a manutenção de sistemas baseado em três normas internacionais referentes a processo de software (ISO/IEC 12207, ISO 9000-3 e SPICE).

Os objetivos específicos são:

- a) estudar as normas de qualidade citadas e suas recomendações para a manutenção de sistemas;
- b) propor um processo de manutenção baseado no comparativo das normas estudadas;
- c) desenvolver um software que dê suporte ao processo proposto de manutenção.

## **1.3 ORGANIZAÇÃO**

A seguir serão descritos brevemente cada capítulo do trabalho.

O primeiro capítulo apresenta uma introdução, incluindo a origem, objetivos e organização do trabalho.

O segundo capítulo apresenta os conceitos de definições de manutenção de software;

O terceiro capítulo apresenta as quatro normas de qualidade e também demonstra detalhadamente o processo de manutenção de cada norma. E ainda demonstra o comparativo entre as normas de qualidade e a teoria de manutenção de sistemas

O quarto capítulo relata o processo de manutenção proposto neste trabalho.

O quinto capítulo apresenta a especificação e implementação do software de apoio a manutenção de sistemas.

O sexto capítulo apresenta as conclusões e as sugestões do trabalho.

## 2 MANUTENÇÃO DE SOFTWARE

### 2.1 CONCEITO

Manutenção de software é certamente, bem mais do que consertar erros. Para muitos autores o termo “Manutenção de Software” é utilizado de forma errônea, pois o mesmo normalmente, já está associado a consertar algo que não funciona. Sabe-se porém, que a manutenção de software abrange muito mais atividades do que somente o reparo de algo defeituoso.

Segundo [BEL93], manutenção de software inclui procedimentos para assegurar que os programas funcionem adequadamente, corrigi-los quando necessário e incrementando-os com novas funções.

Para [FOU94], manutenção de software é o trabalho feito em um software depois que se torna operacional. Isto inclui: entendimento e documentação dos sistemas existentes, adição de novas funções, descoberta e correção de erros, respostas a perguntas de usuários e pessoal de operações, treinamento de pessoal novo nos sistemas, rescrita, reestruturação, conversão e eliminação de software, gerenciamento de software de um sistema operacional; e muitas outras atividades que são incluídas na execução bem sucedida de um sistema de software.

Segundo [PRE95], defini-se como manutenção de software o conjunto de atividades necessárias para manter um software operacional, aplicados sobre um programa existente. Estão relacionados nestas atividades entre outras a determinação de novos quesitos, geração de código, reprojeto e teste do sistema.

Em [PAR90], manutenção de software, em geral, é definida como qualquer trabalho em qualquer sistema de software feito depois que ele se torna operacional ou passa para a produção. Pode-se dizer com isto, que as mudança ocorridas durante o desenvolvimento não são considerados como “manutenção”. A manutenção inicia a partir do momento que o sistema se torna operacional.

Um termo também utilizado é o de “melhorias”. As melhorias são às vezes consideradas como parte do desenvolvimento, porém, se considerada a definição de [PAR90] ela deve ser tratada como parte da manutenção de software. Embora as melhorias tenham algumas características de desenvolvimento como análise, projeto, codificação e testes, há uma diferença entre atividades desenvolvidas num software existente, com estruturas pré definidas e atividades em algo não concretizado.

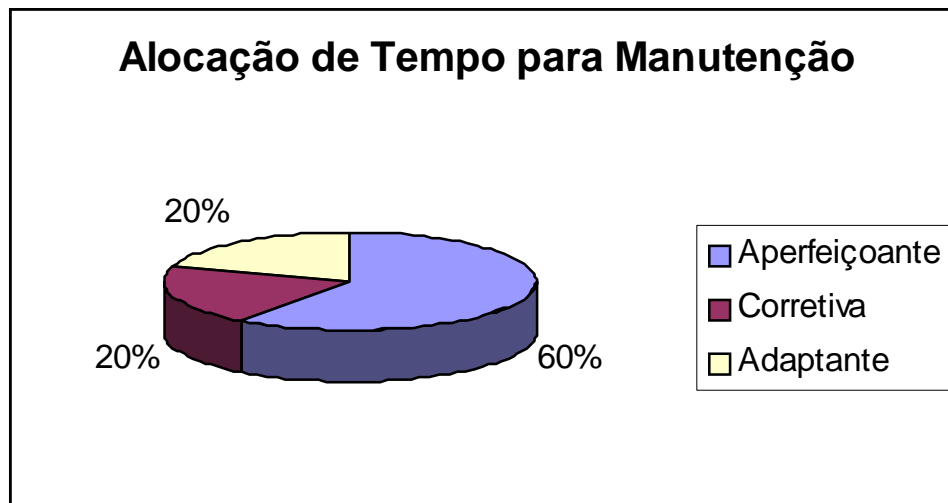
## 2.2 TIPOS DE MANUTENÇÃO

Conforme já descrito anteriormente, a manutenção de software é muito mais do que consertar erros. A maioria dos autores adota para classificar os tipos de manutenção, a classificação realizada por [SWA76], onde este divide-a em três categorias: aperfeiçoadora, adaptativa e corretiva.

Uma minoria de autores subdivide a classificação em outros tipos, porém, apenas há uma manutenção de posicionamento das atividades de manutenção. A maior parte do tempo recai em uma ou mais destas três categorias como mostra a figura 1. A seguir são expostos os tipos de manutenção e qual a quantidade de tempo alocada para cada tipo.

Fonte: [PAR90]

Figura 1 – Alocação de Tempo para Manutenção



### 2.2.1 MANUTENÇÃO CORRETIVA

Esta manutenção ocorre porque não é coerente presumir que a atividade de testes descobrirá todos os erros latentes num grande sistema de software.

Durante a utilização do software ocorrerão erros que serão relatados ao desenvolvedor, para que o mesmo diagnostique e os corrija [PRE95].

Então pode-se dizer que a manutenção corretiva é a correção de problemas que são abortados ou não atendem os requisitos ou especificações originais. Essa categoria consiste em atividades normalmente consideradas “apaga incêndio” ou “quebra galho” para que o sistema permaneça operacional. A manutenção corretiva é geralmente um processo onde um erro precisa ser corrigido imediatamente. As estimativas indicam que apenas 20% de todo o tempo utilizado em manutenção são alocados para esta categoria [PAR90].

Este tipo de manutenção tende a ser o mais desgastante para o desenvolvedor, devido a rapidez em que a alteração necessita acontecer, e devido ao processo de pesquisa dentro do software para localizar os problemas. A manutenção corretiva se executada de forma não organizada pode causar graves problemas no decorrer da vida útil do sistema.

São exemplos de manutenções corretivas o conserto de uma tela que devolve um valor inválido, a correção de um relatório que mostre valores incorretos, ou o conserto de um cancelamento anormal em produção causado por uma transação de entrada inválida.

## **2.2.2 MANUTENÇÃO ADAPTATIVA**

Esta manutenção ocorre pela rápida mudança que é encontrada em cada aspecto de computação. Novas gerações de hardware parecem ser anunciados num ciclo de 24 meses. Novos sistemas operacionais ou novos lançamentos de antigos aparecem regularmente, equipamentos periféricos e outros elementos de sistemas são freqüentemente atualizados ou modificados. A vida útil dos aplicativos por outro lado, pode facilmente ultrapassar 10 anos, vivendo mais que o ambiente de sistema para qual foram originalmente desenvolvidos. Por conseguinte, esta é uma atividade que modifica o software para que ele tenha como interface adequada com o ambiente mutante [PRE95].

Esta manutenção adaptativa é efetivada quando há alguma necessidade de mudança decorrente de um fator externo, havendo assim uma mudança da forma de operação do software. Estas mudanças normalmente não são controladas pelos responsáveis pela

manutenção do software e consistem principalmente em mudanças no hardware, no sistema operacional, nas ferramentas do sistema operacional e nos terminais. É importante salientar que todas as mudanças feitas pelos usuários nas especificações dos requisitos são consideradas manutenção aperfeiçoativa e não adaptativa. As estimativas indicam que aproximadamente 20% de todo o tempo utilizado são alocados para esta categoria.

Como exemplos da manutenção adaptativa, pode-se citar a adição de novos periféricos como leitores ópticas, sensores, etc., introdução de nova versão do sistema operacional.

### **2.2.3 Manutenção Aperfeiçoativa**

São todas as mudanças, inclusões e exclusões, modificações feitas em um sistema para atender as necessidades de evolução ou expansão dos usuários. São consideradas manutenção aperfeiçoante. Também são consideradas as atividades, projetos para facilitar e entendimento e a utilização de programa como a reestruturação ou as atualizações da documentação normalmente considerados manutenções preventivas, além da otimização para que o programa tenha uma melhor performance. As estimativas indicam que mais de 60% de todo tempo utilizado em manutenção são alocados para esta categoria.

Segundo [PAR90], objetiva o aperfeiçoamento do software. Ela inclui as melhorias de performance, mudança iniciadas por usuários que pedem novas características e assim por diante. Estas são essencialmente modificações por causa da mudança dos requisitos internos.

Esta manutenção ocorre quando o software é bem sucedido, pois é evidente que com o decorrer dos anos de utilização, novas características e novas necessidades serão solicitadas pelos usuários.

Este tipo de manutenção é o que menos resistência encontra por parte dos desenvolvedores, pois ela não está somente ligada a correção de problemas, mas também ao desenvolvimento de novos procedimentos dentro do software.

Como exemplo, poderia ser citado a inclusão de novas funções, inclusão de novos relatórios, modificações de relatórios existentes, reorganização do banco de dados para otimizar o desempenho e otimizar a utilização da memória.

## **2.3 FORMAS DE MANUTENÇÃO**

### **2.3.1 MANUTENÇÃO ESTRUTURADA DO SOFTWARE**

Em sua forma ideal, a manutenção estruturada de software é um processo de desenvolvimento continuado. Entretanto, são necessárias técnicas para a manutenção estruturada do software.

A atenção atual na engenharia do software está no novo desenvolvimento de sistemas. Os sistemas desenvolvidos usando as tecnologias estruturadas precisam de técnicas para a implementação e manutenção estruturada.

Segundo [PAR90], os objetivos dessas técnicas são:

- mínima ação da distinção entre o teste, integração e instalação.
- preservação da integridade do projeto inicial de sistemas, embora fazendo manutenção.

Para que seja possível executar a manutenção estruturada de software, é necessário que anteriormente tenha sido aplicado uma metodologia de software. Contudo, uma configuração ideal do software não garante uma manutenção sem problemas, o esforço utilizado para a tarefa é reduzido e a quantidade de mudanças executadas é aumentada.

### **2.3.2 MANUTENÇÃO NÃO-ESTRUTURADA DE SOFTWARE.**

Para [PAR90], até que o software não-estruturado tenha recebido uma “modernização” total ou parcial, ou até que seja parcial ou completamente reescrito usando as tecnologias estruturadas, ele pode continuar sendo um fardo. A necessidade de agrupar ( e até mesmo desenvolver) técnicas para lidar com essa manutenção de software é ainda maior do que para o software estruturado.

Se no momento que ocorrer um pedido de manutenção, o único elemento disponível de uma configuração de software for o código-fonte, a atividade de manutenção inicia-se com uma penosa avaliação do código, freqüentemente complicada pela documentação ruim. Características sutis, tais como estrutura de programa, estrutura de dados globais, interfaces do sistema, desempenho e/ou restrições do projeto, são difíceis de ser verificados e freqüentemente são mal interpretados. As ramificações das mudanças que finalmente são

feitas no código são difíceis de ser avaliadas. Estes de regressão são impossíveis de ser executados porque não existe nenhum registro de testes.

Este tipo de manutenção exige um trabalho árduo pelo responsável e tende a agravar-se com o decorrer dos tempos, este é o preço que se paga devido ao software não ter sido desenvolvido utilizando uma metodologia bem definida.

## 2.4 OBJETIVOS DA MANUTENÇÃO DE SOFTWARE

Segundo [PAR90] a manutenção de software é uma atividade extremamente importante, mas bastante negligenciada. Sua importância é evidente, estima-se que cerca de 60% dos gastos com o software vem da manutenção. Todavia, este número continuará a crescer por muito tempo, enquanto continuarem a aumentar os estoques de código por meio do desenvolvimento numa velocidade mais rápida do que o código torna-se obsoleto.

Dentro deste contexto, pode-se no futuro não muito distante, prever uma organização de software “baseada na manutenção” que não mais pode produzir novo software, porque está gastando todos os seus recursos disponíveis mantendo o software antigo.

Tanto para atualização quanto para o reparo, três funções principais são envolvidas na manutenção de software:

a) Entender o Software Existente: isso implica a necessidade de uma boa documentação, boa relação entre os requisitos e o código e um código bem estruturado e bem formatado.

Os auxílios aqui são bastante discutidos, programação estruturada, formatação automática e auditores de código para checagem de concordância, com padrões, a fim de melhorar a legibilidade do código.

b) Modificar o Software Existente: Isso implica a necessidade do software, hardware e estruturas de dados fáceis de se expandir e que minimizam os efeitos colaterais das mudanças, mais uma documentação de fácil atualização.



As técnicas de modularização e abstração de dados das linguagens facilitam diminuir os efeitos colaterais das mudanças. Entretanto, poderá haver um preço na manutenção. No passado, alguns sistemas com programas altamente acoplados e estruturas de dados associados tinham dificuldades com a atualização de banco de dados. Esse pode não ser um problema com as capacidades de dicionários de dados de hoje, mas as interações ainda não foram investigadas. Outros auxílios à modificação são o código estruturado, técnicas de gerenciamento da configuração, bibliotecas de suporte à programação e sistemas de construção de processos.

c) Reavaliar o software modificado: isso implica a necessidade de estruturas de software que facilitem o reteste selético, além de recursos para fazer testes mais completos e eficientes.

Esta tarefa inclui principalmente os sistemas de gerenciamento de dados de teste, programas comparadores e analisadores de estrutura do programa com alguma capacidade limitada para análise seletiva de teste.

## **2.5 PROBLEMAS NA MANUTENÇÃO DE SOFTWARE**

Há diversos aspectos no problema de manutenção, alguns deles podendo ser solucionados pelo projeto correto e alguns não. Os principais problemas encarados por um departamento de manutenção são os seguintes conforme pesquisa realizada em [PRE95], [FOU94], [BEL93] e [PAR90]:

a) Programas são colocados em manutenção ainda tendo um número significativo de erros; assim, o que é chamado manutenção na realidade torna-se uma continuação de um esforço de teste. Esse parece ser um ponto obvio, mas geralmente é ignorado. O que está acontecendo é que os programadores de manutenção estão sofrendo pelos pecados dos programadores originais.

- b) Há o problema contínuo de atualizar o programa para novos compiladores, novos sistemas operacionais e outros novos softwares do sistema. Não é provável que este trabalho termine no futuro imediato.
- c) Frequentemente é difícil ou impossível rastrear a evolução do software através de muitas versões ou lançamentos. As mudanças não estão adequadamente documentadas. Outrora, também é muito complicado rastrear o processo pelo qual o software foi criado.
- d) Muitas vezes é excepcionalmente difícil entender o programa “de outra pessoa”. A dificuldade aumenta à medida que o número de elementos, de uma configuração de software diminui. Contudo, em muitas ocasiões a “outra pessoa” frequentemente não está por perto para explicar. A mobilidade entre o pessoal da área de software é elevada. Não podemos contar com uma explicação pessoal do desenvolvimento de software quando a manutenção for necessária.
- e) A documentação, normalmente, é ruim ou não existe. O reconhecimento de que o software deve ser documentado é o primeiro passo, mas a documentação deve ser compreensível, e consistente com o código-fonte para ter algum valor. Algumas experiências indicam que os programadores de manutenção trabalhariam melhor se removessem todos os comentários que acompanham um programa. E depois tentassem encontrar um erro ou implementar algumas melhorias. Nitidamente, muitas organizações estão agora, pagando o preço por padrões ruins de documentação no passado.
- f) A maioria dos softwares não é projetada para sofrer mudanças. A menos que um método de projeto acomode mudanças mediante conceitos tais como independência funcional ou classes de objetos. As modificações no software são difíceis e propensas a erros.

Encara-se o problema básico de que as pessoas geralmente não gostam de fazer manutenção. Esse trabalho não é considerado muito atraente, normalmente não possui grandes salários e sofre das frustrações óbvias de tentar consertar o trabalho sujo de uma outra pessoa [SCU98].

Todos os problemas anteriores podem, em parte, ser atribuídos ao grande número de programas existentes que foram desenvolvidos sem levar em consideração a engenharia

de software. Uma metodologia disciplinada não deve ser vista como a solução para todos os males. Contudo, a engenharia de software oferece pelo menos soluções parciais para cada problema associado à manutenção [PRE95].

## **2.6 TAREFAS DA MANUTENÇÃO**

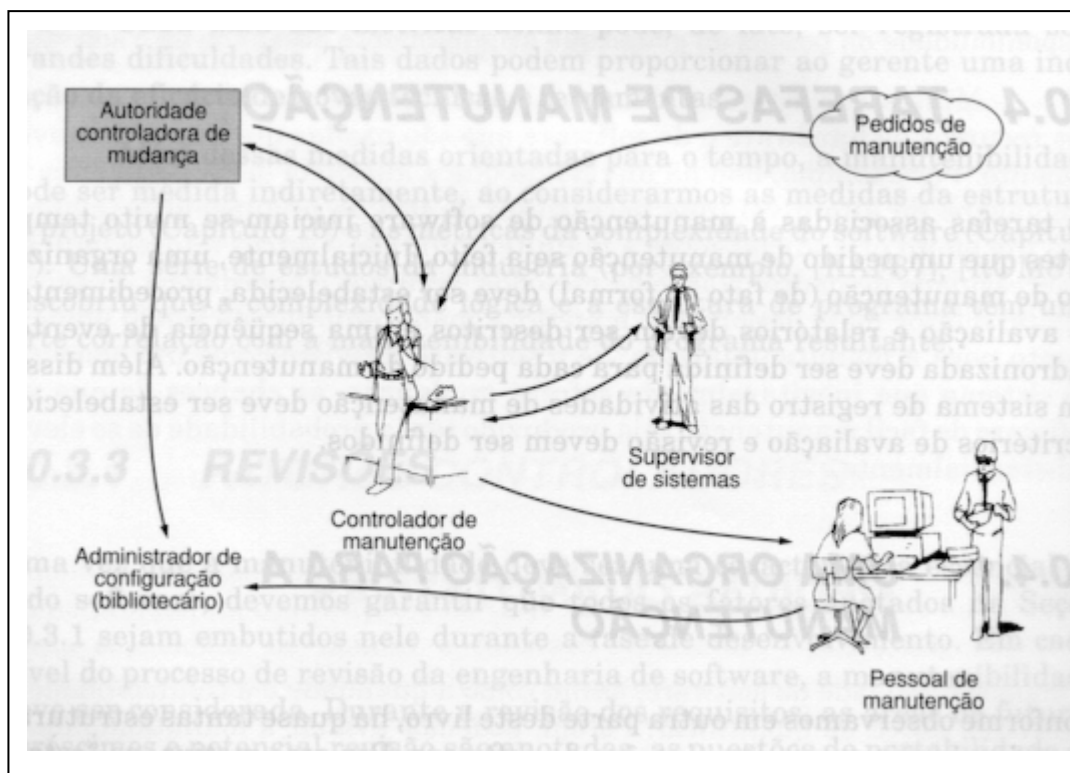
As tarefas associadas à manutenção de software iniciam-se muito tempo antes que um pedido de manutenção seja feito. Inicialmente, uma organização de manutenção (de fato ou formal) deve ser estabelecido procedimentos de avaliação e relatórios devem ser definidos para cada pedido de manutenção.

Em [PRE95], existe uma descrição de tarefas genéricas para a manutenção de software como é demonstrado a seguir.

### **2.6.1 ORGANIZAÇÃO PARA A MANUTENÇÃO**

Embora não seja necessário estabelecer um organização de manutenção formal, uma delegação de responsabilidade informal é absolutamente essencial. Conforme mostra a figura abaixo, os pedidos de manutenção devem ser canalizados a um controlador de manutenção que apresenta cada pedido para ser avaliado por um supervisor de sistemas. Este por sua vez é um membro do pessoal técnico a quem foi atribuída a responsabilidade de se familiarizar com um pequeno subconjunto de programas de produção. Logo que uma avaliação é feita, uma autoridade controladora das mudanças deve determinar a ação a ser tomada.

Figura 2 – Organização para Manutenção



Fonte: [PRE95]

Este tipo de organização reduz a confusão e melhora o fluxo de atividade de manutenção. Uma vez que todos os pedidos afunilam para um único indivíduo(ou grupo). Quando as responsabilidades são atribuídas antes do início da atividade de manutenção, a confusão é grandemente reduzida.

## 2.6.2 RELATÓRIOS

Nesta fase todos os pedidos de manutenção de software devem ser apresentados padronizadamente. O desenvolvedor de software normalmente elabora um formulário de pedido de manutenção que é preenchido pelo usuário solicitante. Se um erro for encontrado, uma descrição completa das circunstâncias que levaram ao erro (inclusive dados de entrada, listagens e outros materiais de apoio) devem ser incluída. Para pedidos de manutenção adaptativa os aperfeçoante, uma breve especificação de mudança é submetida.

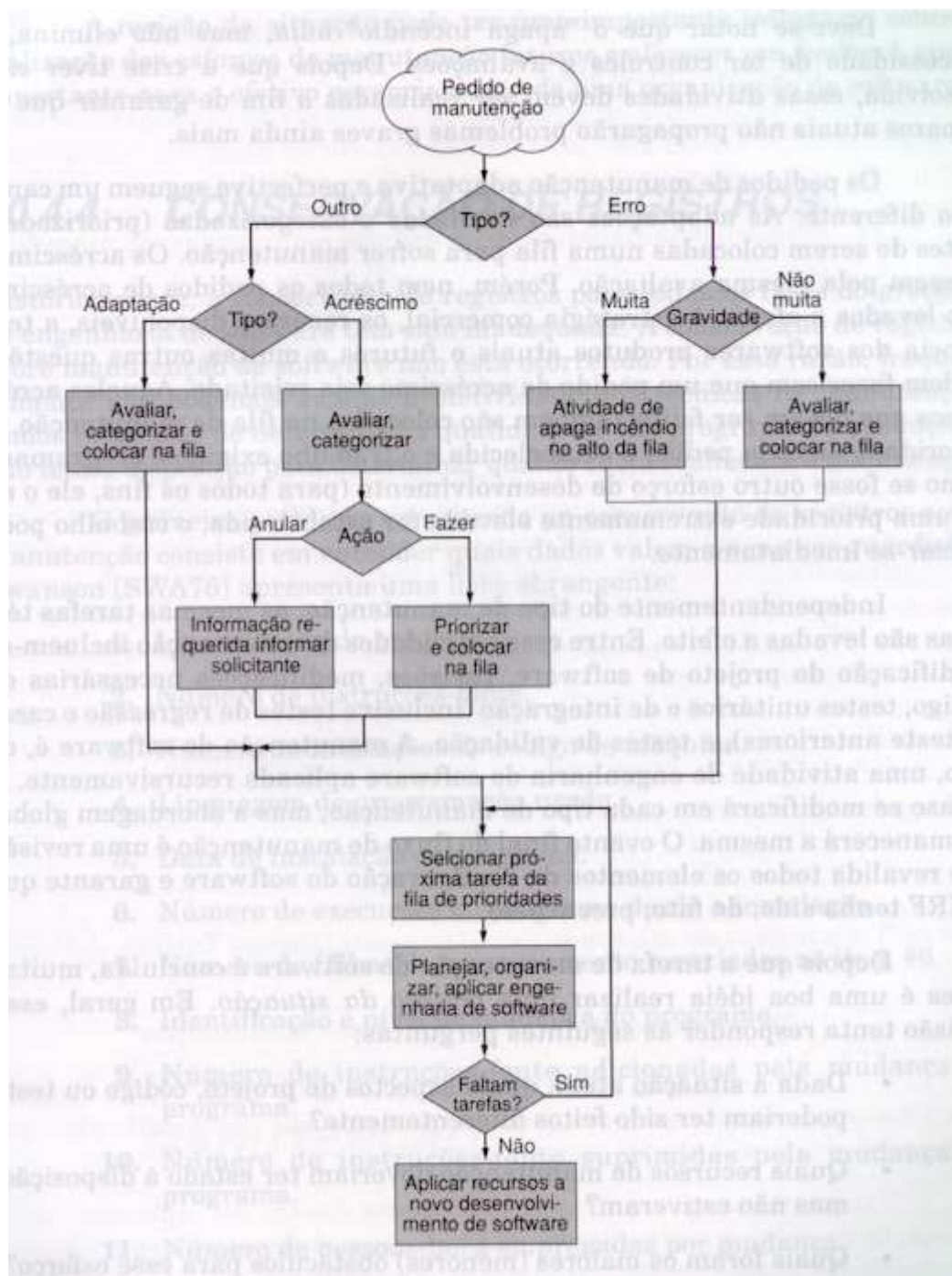
### 2.6.3 O FLUXO DOS EVENTOS

Como mostra a figura a 3, o primeiro requisito consiste em determinar o tipo de manutenção onde, é necessário um consenso entre desenvolvedor e usuário, se um erro grave é encontrado é rapidamente designada uma equipe para a análise do problema. Para outros erros o pedido de manutenção é analisado e programado em conjunto com outras tarefas que exijam recursos de desenvolvimento.

Em certos casos, um erro pode ser tão grave que os controles normais devem ser abandonados temporariamente. O código deve ser modificado imediatamente. Esse modo “apaga incêndio”, para manutenção corretiva, é reservado somente para situação de crise, e deve representar um porcentagem muito pequena de todas as atividades de manutenção.

Os pedidos de manutenção adaptativa e aperfeiçoante seguem um caminho diferente. As Adaptações são avaliadas e categorizadas antes de serem colocadas numa fila para sofrer manutenção. Os acréscimos passam pela mesma avaliação, porém nem todos são levadas a efeito, dependendo de vários aspectos relacionados a administração da empresa.

Figura 3 – Fluxo dos Eventos



Fonte : [PRE95]

## 2.6.4 CONSERVAÇÃO DOS REGISTROS

Historicamente, a conservação de registros para todas as fases do processo de engenharia de software tem sido inadequado. A conservação de registros sobre manutenção de software não está ocorrendo. Por essa razão, freqüentemente não é possível avaliar a efetividade dos técnicas de manutenção, é difícil de determinar a qualidade de um programa de produção e muitas vezes não há disposição para determinar quanto custa realmente a manutenção.

O primeiro problema encontrado na conservação de registros sobre manutenção consiste em entender quais dados valem a pena ser guardados. Em [SWA76] é apresentada uma lista abrangente.

## **2.6.5 AVALIAÇÃO**

Uma avaliação das atividades de manutenção do software freqüentemente é complicada pela falta de dados concretos. Se a conservação de registros for iniciada, uma série de medidas do desempenho de manutenção pode ser desenvolvida. Novamente em [SWA76], sete medidas são descritas para proporcionar uma estrutura quantitativa a partir da qual podem ser tomadas decisões sobre a técnica de desenvolvimento, escolha de linguagem, projeções sobre o esforço de manutenção, alocação de recursos e muitas outras questões. Obviamente, tais dados podem ser aplicados para avaliar a tarefa de manutenção.

## **2.6.6 TAREFAS TÉCNICAS DA MANUTENÇÃO DE SOFTWARE**

Em [FOU94], o autor sugere doze tarefas técnicas que são descritas para a manutenção de software.

a) avaliar a solicitação de melhoramento do sistema. Esta tarefa tem por objetivo analisar as solicitações dos usuários quanto a modificações no sistema em produção, com base nas necessidades evolutivas do negócio.

b) avaliar a solicitação de correção de problema do sistema em produção. Onde o objetivo é analisar um problema do sistema ocorrido no ambiente de produção e propor uma solução.

c) aplicar conserto de emergência no programa. Neste momento o objetivo é fazer um conserto de emergência no programa com erro e rapidamente reiniciar a operação do sistema no ambiente de produção.

d) organizar as solicitações de melhoramentos em releases individuais do sistema. Tem por objetivo planejar, priorizar e juntas as solicitações dos usuários de alterações do sistema em releases de manutenção distintos.

e) analisar o release de solicitações de manutenção do sistema. Permite analisar e documentar em detalhes as solicitações associadas a um release específico de manutenção.

f) projetar o release de manutenção do sistema. Objetiva-se nesta tarefa o projeto dos programas e as estruturas físicas de dados do release de manutenção do sistema.

g) codificar e testar o release de manutenção do sistema. Neste momento o objetivo é desenvolver e testar os programas do atual release de manutenção.

h) implementar o release de manutenção do sistema. Esta fase tem por objetivo transferir o release de manutenção do sistema para a produção.

i) executar manutenção preventiva. Onde o principal objetivo é monitorar o comportamento do sistema para garantir que ele permaneça eficiente e produza resultados precisos.

j) treinar a equipe. O objetivo é treinar regularmente o pessoal novo ou retreinar o pessoal existente no uso efetivo ou apoio à evolução do sistema.

k) executar avaliações periódicas do sistema. Aqui objetiva-se a avaliação do grau de efetividade e de uso do sistema à medida que ele envelhece.

l) executar revisões pós-implementação. Esta fase tem por objetivo examinar retrospectivamente o projeto e determinar a sua efetividade em satisfazer as necessidades de negócio da organização.



### 3 NORMAS E MODELOS DE QUALIDADE

A Engenharia de Software tornou-se uma das disciplinas mais importantes desta década, devido à crescente disseminação do uso do software como parte integrante dos mais variados produtos. Entretanto, o desenvolvimento e a manutenção de software são relativamente novos se comparados às áreas tradicionais da engenharia. Como resultado, não há na produção de software o mesmo rigor existente nos projetos tradicionais de engenharia. Assim, existe um nível elevado de insatisfação com o software, tanto por parte dos usuários, que não vêem as suas necessidades atendidas, quanto com relação à organização que o desenvolve, por não conseguir torná-lo econômica e comercialmente viável [GRA97].

Dentro deste contexto, a comunidade de informática vem criando normas para regular e orientar a atividade de produção de software. Neste capítulo, será dada uma visão geral de quatro normas e modelos de qualidade enfatizando o processo da manutenção de software.

#### 3.1 NORMA ISO/IEC 12207 – PROCESSOS DE CICLO DE VIDA DE SOFTWARE

Segundo [GRA97], a norma NBR ISO/IEC 12207 – Processos de Ciclo de Vida do Software tem como principal objetivo o estabelecimento de uma estrutura comum para os processos de ciclo de vida de software, para ser utilizada como referência. Além disso, a Norma considera que o desenvolvimento e a manutenção do software devem ser conduzidos da mesma forma que a disciplina de engenharia.

A estrutura descrita na norma é composta de processos, atividades e tarefas a serem aplicados em operações que envolvam, de alguma forma, o software, seja através de aquisição, fornecimento, desenvolvimento, operação ou manutenção. Esta estrutura permite estabelecer ligações claras com o ambiente de engenharia de sistemas, ou seja, aquele que inclui práticas de software, hardware, pessoal e negócios.

Os processos de ciclo de vida são agrupados em três classes: fundamentais, apoio e organizacionais, que representam a sua natureza, descritos a seguir.

### 3.1.1 PROCESSOS FUNDAMENTAIS

Atendem ao início e à execução do desenvolvimento, operação ou manutenção dos produtos de software durante o ciclo de vida de software. São eles:

**a) processo de aquisição:** Define as atividades do adquirente, organização que adquire um sistema ou produto de software. Inicia-se com a definição da necessidade de adquirir um sistema, um produto de software ou um serviço de software. O processo continua com a preparação e emissão de pedido de proposta, seleção de fornecedor e gerência do processo de aquisição através da aceitação do sistema, produto de software ou serviço de software.

**b) processo de fornecimento:** Define as atividades do fornecedor, organização que provê o produto de software ao adquirente. O processo pode ser iniciado tanto por uma decisão de preparar uma proposta para responder a um pedido de proposta de um adquirente quanto pela assinatura e celebração de um contrato com o adquirente para fornecer o sistema, produto de software ou serviço de software. O processo continua com a determinação dos procedimentos e recursos necessários para gerenciar e garantir o projeto, incluindo o desenvolvimento e a execução dos planos de projeto até a entrega do sistema , produto de software ou serviço de software para o adquirente.

**c) processo de desenvolvimento:** Define as atividades do desenvolvedor, organização que define e desenvolve o produto de software. O processo contém as atividades para análise de requisitos, projeto, codificação, integração, testes, instalação e aceitação relacionada aos produtos de software.

**d) processos de operação:** Define as atividades do operador, organização que provê os serviços de operação de um sistema computacional no seu ambiente de funcionamento para seus usuários. O processo cobre a operação do produto de software e o suporte operacional aos usuários.

**e) processo de manutenção:** Define as atividades do mantenedor, organização que provê os serviços de manutenção do software, isto é, gerenciamento de modificações no software para mantê-lo atualizado e em perfeita operação. Este processo á ativado quando o produto de software é submetido a modificações no código e na documentação associada

devido a um problema, ou à necessidade de melhoria ou adaptação. O objetivo é modificar um produto de software existente, preservando a sua integridade.

### 3.1.2 PROCESSOS DE APOIO

Auxiliam um outro processo e contribuem para o sucesso e qualidade do projeto de software. Um processo de apoio é empregado e executado, quando necessário, por outro processo. São eles:

**a) processo de documentação:** Define as atividades para registrar informações produzidas por um processo ou atividades do ciclo de vida. O processo contém o conjunto de atividades que planeja, projeta, desenvolve, produz, edita, distribui e mantém aqueles documentos necessários a todos os interessados, tais como gerentes, engenheiros e usuários do sistema ou produto de software.

**b) processo de gerência de configuração:** Define as atividades para a aplicação de procedimentos administrativos e técnicos, por todo o ciclo de vida de software, destinado a: identificar e definir os itens de software em um sistema além de estabelecer suas linhas básicas (*baseline*); controlar as modificações e liberações dos itens; registrar e apresentar a situação dos itens e dos pedidos de modificação; garantir completa, a consistência e a correção dos itens; e controlar o armazenamento, a manipulação e a distribuição dos itens.

**c) processo de garantia da qualidade:** Define as atividades para fornecer a garantia adequada de que os processos e produtos de software, no ciclo de vida do projeto, estejam em conformidade com seus requisitos especificados e sejam aderentes aos planos estabelecidos. A abrangência do processo inclui questões como garantia de qualidade do produto (NBR 13596 NBR13596 (1996) que corresponde a ISO/IEC 9126), do processo e do sistema de qualidade (NBR 9001 NBR 9001 (1994) e NBR 9000-3 NBR ISO 900-3 (1993)).

**d) processo de verificação:** Define as atividades para verificação dos produtos de software. É um processo para determinar se os produtos de software de uma atividade atendem completamente aos requisitos ou condições impostas a eles.

**e) processo de validação:** Define as atividades para validação dos produtos produzidos pelo projeto de software. É um processo para determinar se os requisitos e o produto final (sistema ou software) atendem ao uso específico proposto.

**f) processo de revisão conjunta:** Define as atividades para avaliar a situação e produtos de uma atividade de um projeto, se apropriado. As revisões conjuntas são feitas tanto nos níveis de gerenciamento do projeto como nos níveis técnicos e são executadas durante a vigência do contrato.

**g) processo de auditoria:** Define as atividades para determinar adequação aos requisitos, planos e contrato, quando apropriado. Este processo pode ser empregado por quaisquer das duas partes, onde uma parte (parte auditora) faz a auditoria nos produtos de software ou nas atividades da outra parte (parte auditada).

**h) processo de resolução de problemas:** Define um processo para analisar e resolver os problemas (incluindo não-conformidades), de qualquer natureza ou fonte, que são descobertos durante a execução do desenvolvimento, operação, manutenção ou outro processo. O objetivo é prover os meios em tempo adequado e de forma responsável e documentada para garantir que todos os problemas encontrados sejam analisados e resolvidos e tendências sejam identificadas.

### 3.1.3 PROCESSOS ORGANIZACIONAIS

São empregados por uma organização para estabelecer e implementar uma estrutura constituída de processos de ciclo de vida e pessoal associados, melhorando continuamente a estrutura e os processos. Eles são tipicamente empregados fora do domínio de projetos e contratos específicos. Entretanto, ensinamentos destes projetos e contratos contribuem para a melhoria da organização. São eles:

**a) processo de gerência:** Define as atividades genéricas que podem ser empregadas por quaisquer das partes que têm que gerenciar seu(s) respectivo(s) processo(s). O gerente é responsável pelo gerenciamento de produto, gerenciamento de projeto e gerenciamento de tarefa do(s) processo(s) aplicável(eis), tais como aquisição, fornecimento, desenvolvimento, operação, manutenção ou processo de apoio.

**b) processo de infra-estrutura:** Define as atividades para estabelecer e manter a infra-estrutura necessária para qualquer outro processo. A infra-estrutura pode incluir hardware,

software, ferramentas, técnicas, padrões e recursos para o desenvolvimento, operação ou manutenção.

**c) processo de melhoria:** Define as atividades básicas que uma organização (isto é, adquirente, fornecedor, desenvolvedor, operador, mantenedor, ou o gerente de outro processo) executa para estabelecer, avaliar, medir, controlar e melhorar um processo de ciclo de vida de software.

**d) processo de treinamento:** Define as atividades para prover e manter pessoal treinado. A aquisição, o fornecimento, o desenvolvimento, a operação ou a manutenção de produtos de software é extremamente dependente de pessoal com conhecimento e qualificação. Portanto é essencial que o treinamento de pessoal seja planejado e implementado com antecedência para que o pessoal treinado esteja disponível quando o produto de software for adquirido, fornecido, desenvolvido, operado ou mantido.

## **3.2 MANUTENÇÃO DE SOFTWARE SEGUNDO A NORMA ISO/IEC 12207**

Segundo [SCU98], o processo de manutenção contém as atividades e tarefas do responsável em manter o sistema. O processo é ativado quando o produto de software é submetido a modificações no código e na documentação associada devido a um problema, ou necessidade de melhoria ou adaptação. Com o objetivo de modificar um produto de software existente preservando a sua integridade. Este processo inclui a migração e a descontinuação de produto de software, onde termina este processo.

O responsável em manter o sistema gerencia o processo de manutenção no nível de projeto, estabelece uma infra estrutura sobre o processo, adapta o processo de projeto e gerencia o processo organizacional e o processo de melhoria quando o responsável em manter o sistema faz o serviço de manutenção.

As atividades previstas para este processo segundo a norma são: Implementação do Processo de Manutenção, Análise do Problema e da Modificação, Implementação da Modificação, Revisão / Aceitação da Manutenção, Migração e Descontinuação do Software.

### **3.2.1 IMPLEMENTAÇÃO DO PROCESSO MANUTENÇÃO**

O responsável em manter o sistema deve desenvolver, documentar e executar planos e procedimentos para a condução das atividades e tarefas do processo de manutenção.

Deve estabelecer também os procedimentos para receber, registrar e rastrear relatórios de problemas e pedidos de modificação dos usuários, e prover realimentação (feedback) para os usuários. Sempre que problemas forem encontrados, eles devem ser registrados e incluídos no processo (ou estabelecer interface organizacional com) a gerência de configuração para gerenciar as modificações nos sistemas existentes.

### **3.2.2 ANÁLISE DO PROBLEMA E DA MODIFICAÇÃO**

O responsável em manter o sistema deve analisar o relatório de problema foi pedido de modificação segundo o seu impacto na organização, no sistema existente e nos sistemas com os quais interage, observando o seguinte:

- a) tipo de manutenção isto implica em verificar se a manutenção é corretiva, aperfeiçoativa ou adaptativa para um novo ambiente;
- b) escopo da manutenção implica em verificar o tamanho da modificação, o custo envolvido e o prazo para modificar;
- c) criticidade da manutenção isto implica em verificar qual o impacto no desempenho, proteção ou segurança;

O responsável em manter o sistema deve reproduzir ou verificar o problema e baseado na análise, deve desenvolver alternativas para a implementação da modificação.

Deve também documentar o problema ou pedido de modificação, os resultados da análise e as alternativas de implementação.

Após obter aprovação para a alternativa de modificação selecionada, conforme especificado.

### **3.2.3 IMPLEMENTAÇÃO DA MODIFICAÇÃO**

O responsável em manter o sistema deve conduzir a análise e determinar que documentos, módulos de software e versões destas necessitam ser modificadas (estas análises devem ser registradas).

Deve-se utilizar os procedimentos de desenvolvimento para implementar as modificações. Os requisitos devem complementar o seguinte:

- a) Definir e documentar os critérios de teste e de avaliação para testar e avaliar as partes modificadas e as não modificadas do sistema (unidades de software, componentes e itens de configuração);
- b) Garantir a implementação completa e correta dos requisitos novos e dos modificados. Também deve ser garantido que os requisitos originais não modificados não foram afetados. Os resultados dos testes devem ser documentados.

### **3.2.4 REVISÃO / ACEITAÇÃO DA MANUTENÇÃO**

O responsável em manter o sistema deve conduzir revisões com a organização que autorizou a modificação para determinar a integridade do sistema modificado e obter aprovação para conclusão satisfatória da modificação, conforme especificado.

### **3.2.5 MIGRAÇÃO**

Se um sistema ou produto de software (incluindo dados), migrado de um ambiente de operação antigo para um novo, deve ser assegurado que qualquer programa do sistema ou dados produzidos / modificados durante a migração estejam de acordo com a especificação.

Um plano de migração deve ser desenvolvido, documentado e executado. As atividades de planejamento devem incluir os usuários.

Os itens incluídos no plano devem incluir o seguinte:

- a) Análise e definição dos requisitos de migração;
- b) Desenvolvimento de ferramentas de migração;
- c) Conversão de produto de software e dados;
- d) Execução da migração;
- e) Verificação da migração;
- f) Suporte para o Ambiente Antigo.

Os usuários devem receber notificação dos planos e atividades de migração. As notificações devem conter o seguinte:

- a) Explicação do porque o ambiente antigo não ser mais suportado;
- b) Descrição do novo ambiente com sua data de disponibilização;
- c) Descrição de outras opções de suporte disponíveis, se existirem, uma vez que o suporte para o ambiente antigo seja descontinuado.

As operações paralelas dos ambientes antigos e novos podem ser conduzidas para a transição gradual ao novo ambiente. Durante este período, devem ser provido o treinamento necessário, conforme especificação.



Quando a migração programada ocorrer, devem ser enviadas notificações a todos os interessados toda documentação, históricos (logs) e código associado ao ambiente antigo deveriam ser arquivados.

Após a migração, uma revisão deve ser executada para avaliar o impacto da mudança para o novo ambiente. Os resultados da revisão devem ser enviados as autoridades apropriadas para informação, orientação e providências.

Dados utilizados ou associados com o ambiente antigo devem estar acessíveis, de acordo com os requisitos do contrato para preservação e auditoria dos dados.

### **3.2.6 DESCONTINUAÇÃO DO SOFTWARE**

O sistema deverá ser descontinuado a pedido do proprietário. Um plano de descontinuação, para remover o suporte ativo pelas organizações responsáveis pela operação e manutenção, deve ser desenvolvido e documentado. As atividades de planejamento devem incluir os usuários. O plano deve conter:

- a) Cessação total ou parcial de suporte após um certo período de tempo;
- b) Arquivamento do produto de software e sua documentação associada;
- c) Responsabilidade por quaisquer questões futuras de suporte residual;
- d) Transição para o novo produto de software, se aplicável;
- e) Disponibilidade de cópias e arquivos de dados.

O usuário deve receber notificação dos planos e atividades de descontinuação. As notificações devem incluir:

- a) Descrição da substituição ou atualização com sua data de disponibilidade;
- b) Explicação do porquê o produto de software não receberá mais suporte;

c) Descrição de outras opções de suporte disponíveis, uma vez que o suporte seja descontinuado.

As operações paralelas do produto de software em descontinuação e de novo deveriam ser conduzidas para transição gradual ao novo sistema. Durante este período, deve ser provido treinamento de usuário, conforme especificado no contrato.

Quando a descontinuação programada ocorrer, devem ser enviadas notificações a todos os interessados. Toda documentação, históricos (logs) e código associados ao desenvolvimento deveriam ser arquivados, quando apropriado.

Os dados utilizados os associados com o produto de software descontinuado devem ser acessíveis, de acordo com os requisitos do contrato para preservação e auditoria dos dados.

### **3.2.7 ROTEIRO DE MANUTENÇÃO**

A seguir é demonstrado um roteiro proposto para a manutenção de software segundo a norma ISO/IEC 12207, que é demonstrado em [FRA98].

Figura 4 – Roteiro de Manutenção

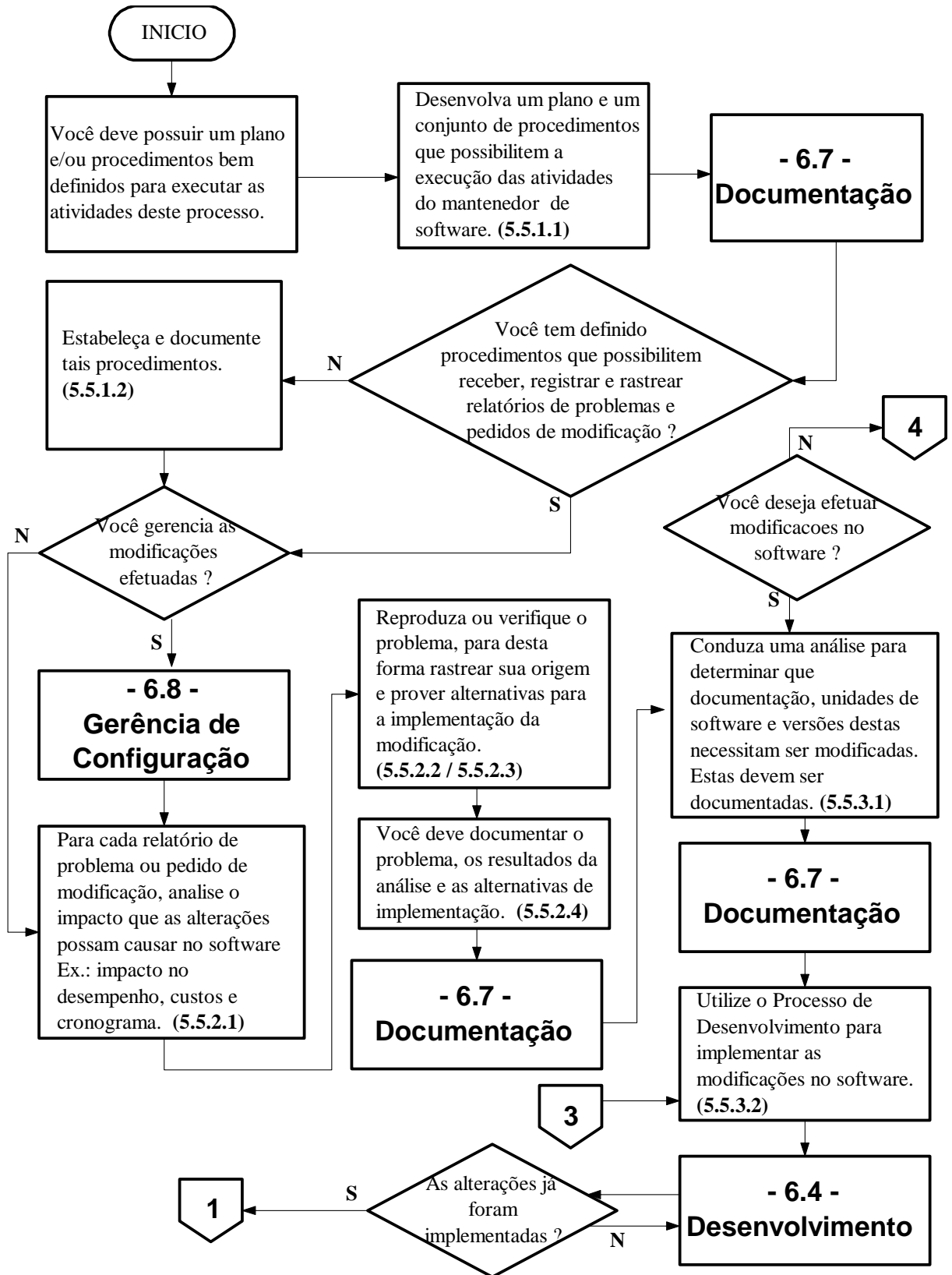


Figura 4 – Roteiro de Manutenção (Continuação)

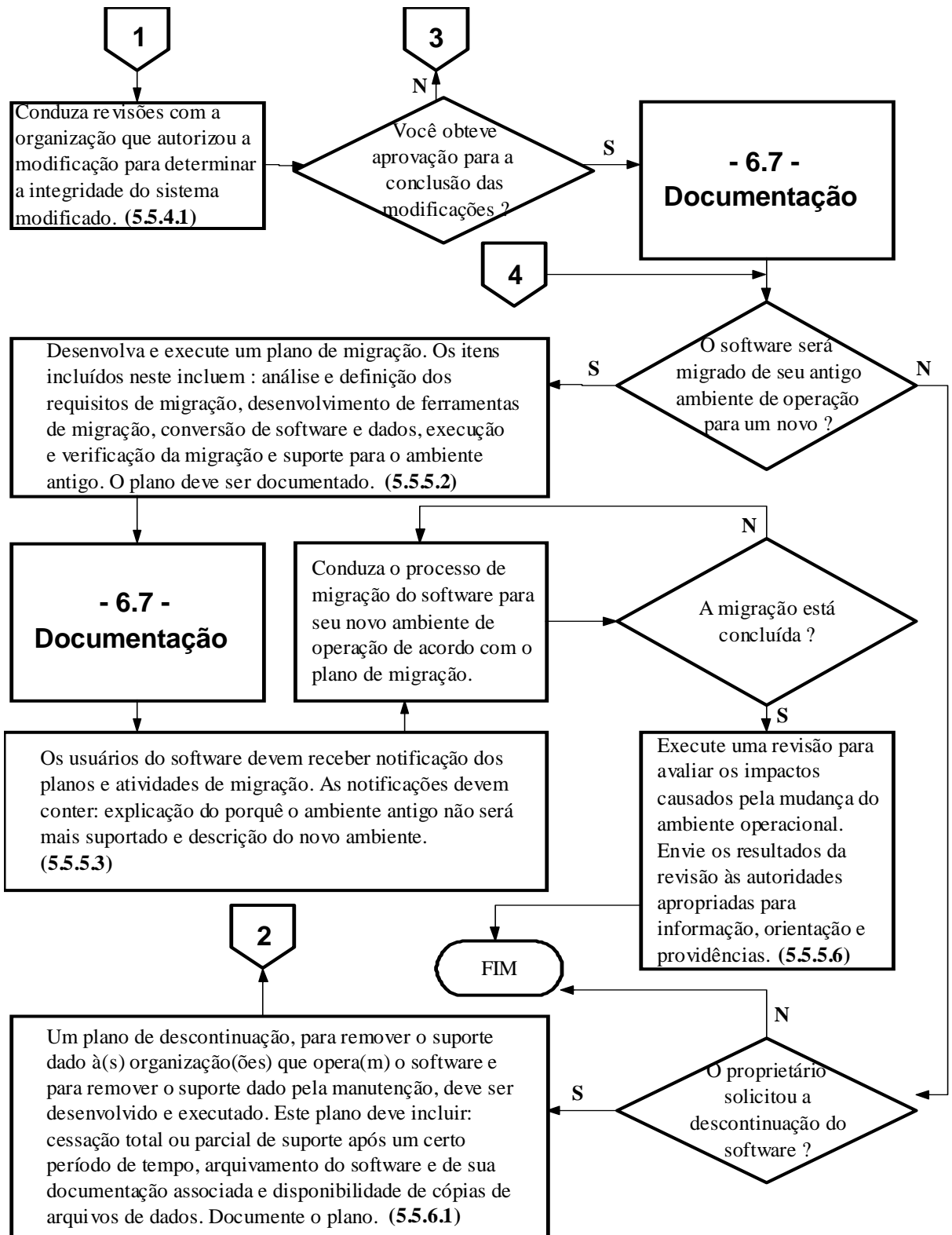
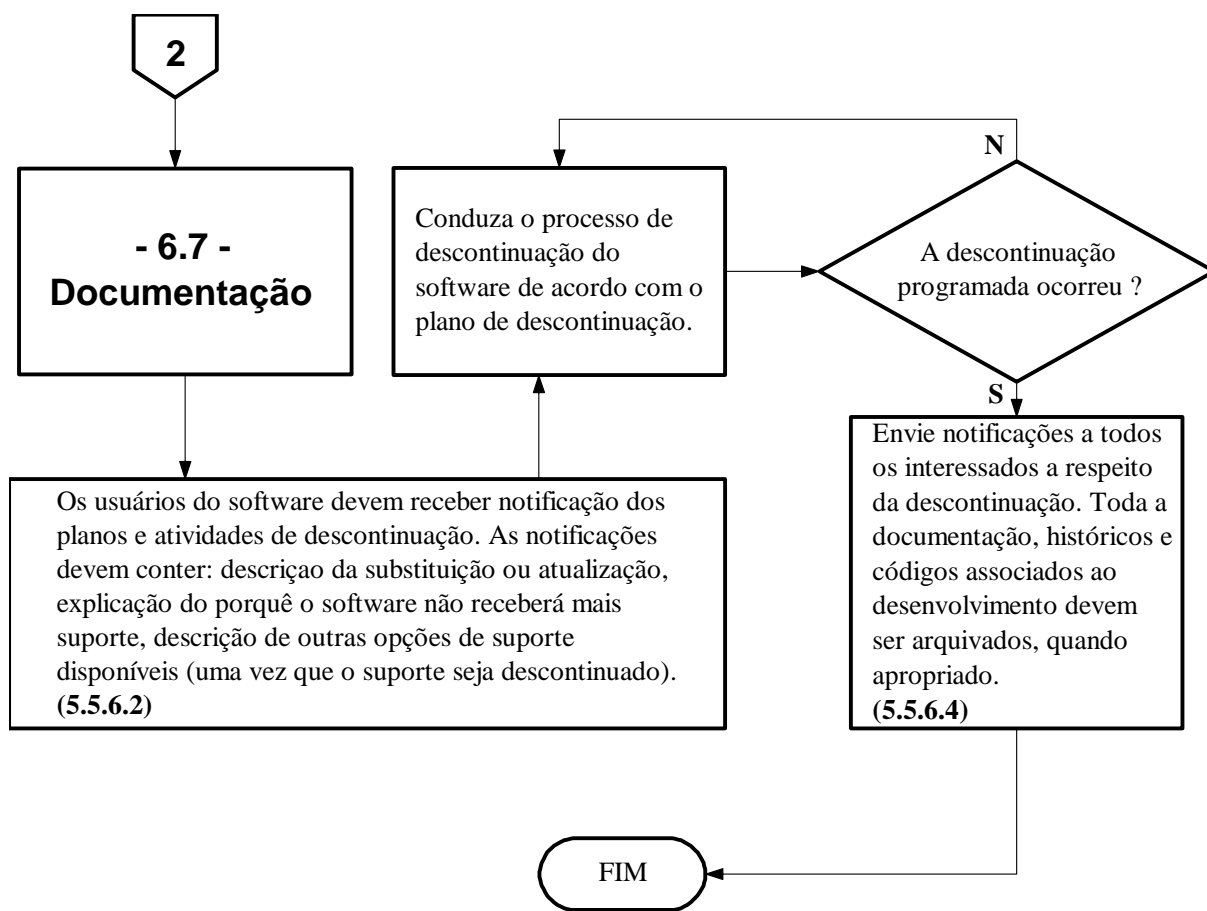


Figura 4 – Roteiro de Manutenção (Continuação)



### 3.3 NORMA ISO 9000-3

A série ISO 9000 é um conjunto de normas que define padrões para garantia e gerenciamento da qualidade. Algumas destas normas seguem na tabela 1.

Tabela 1 – A série ISO 9000

<b>Norma</b>	<b>Objetivo</b>
<b>ISO 9001</b>	Modelo para garantia da qualidade em projeto, desenvolvimento, produção, instalação e assistência técnica.
<b>ISO 9002</b>	Modelo para garantia da qualidade em produção e instalação.
<b>ISO 9003</b>	Modelo para garantia da qualidade em inspeção e ensaio finais.
<b>ISO 9000-1</b>	Diretrizes para escolher entre as normas ISO 9001, 9002 e 9003.
<b>ISO 9000-3</b>	Orientação para a aplicação da ISO 9001, em software.

Segundo [OLI95], a ISO 9000-3 é um guia para interpretação da ISO 9001, específico para o desenvolvimento, fornecimento e manutenção de software. As diretrizes propostas cobrem questões como o entendimento comum para as partes (contratante e contratado) de requisitos funcionais e o uso de metodologias consistentes para o desenvolvimento de software e gerenciamento de projeto como um todo, da concepção até a manutenção.

Esta NBR ISO 9000-3 define diretrizes para facilitar a aplicação da NBR 19001 a organização que desenvolvem, fornecem e mantêm “software”.

Destina-se a fornecer orientação quando um contrato entre duas partes exigir a demonstração da capacidade de um fornecedor em desenvolver, fornecer e manter produtos de “software”.

Estas diretrizes destinam-se a descrever os controles e métodos sugeridos para a produção de “software” que atendam aos requisitos do comprador, evitando-se não conformidade em todos os estágios, desde o desenvolvimento até a manutenção.

As diretrizes desta NBR ISO 9000-3 são aplicáveis em situações contratuais para produtos de “software”, quando:

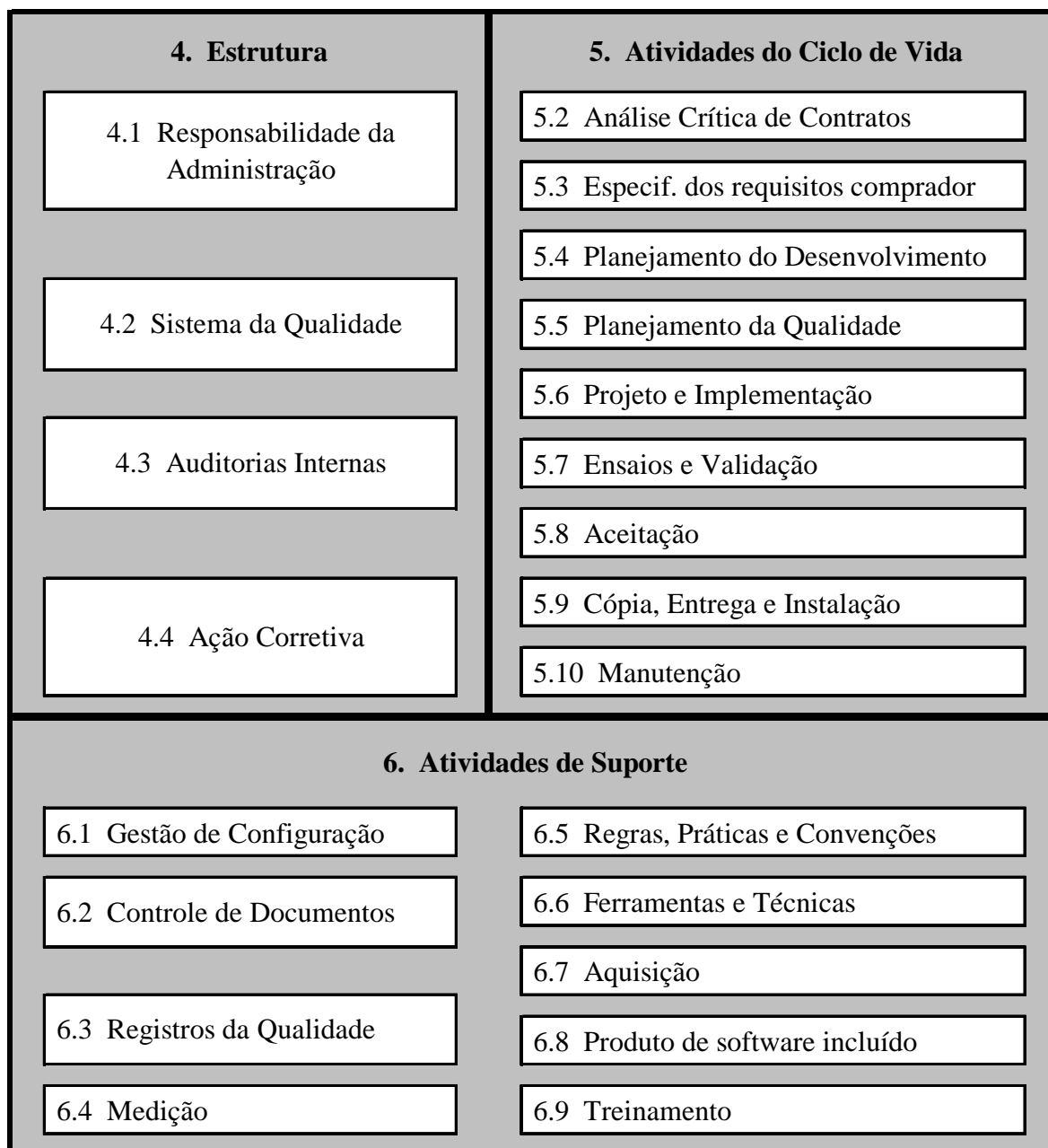
- a) o contrato exigir, especificamente, esforço de projeto, e os requisitos do produto forme indicados principalmente em termos de desempenho, ou precisarem ser estabelecidos;
- b) a confiança no produto puder ser obtida através da demonstração adequada da capacidade de desenvolvimento, fornecimento e manutenção de um determinado fornecedor.

Segundo [ASS93], a ISO 9000-3 divide-se em três partes principais:

- a) Sistema da Qualidade – Estrutura;
- b) Sistema da Qualidade – Atividades do ciclo de vida;
- c) Sistema da Qualidade – Atividades de suporte.

A figura 5 demonstra as três partes:

Figura 5 - Estrutura da norma ISO 9000-3



### 3.3.1 ESTRUTURA DO SISTEMA DA QUALIDADE

São quatro os pontos cobertos por este capítulo da norma [ASS93]:

a) **responsabilidade da administração:** Abrange a definição de uma política interna da qualidade, onde a administração do fornecedor deve definir e documentar sua política e objetivos para a qualidade e assumir o compromisso com estes.

A organização e estrutura da empresa, bem como uma matriz de responsabilidades e autoridades das pessoas que influenciam a Qualidade, cujo o objetivo é definir particularmente as responsabilidades do pessoal que necessita de autonomia organizacional.

Também deve haver recursos materiais e humanos para verificação das atividades da qualidade, e um representante da alta administração para fazer funcionar o sistema da qualidade adotado.

É preciso manter registros das análises críticas do sistema da qualidade, que devem ser realizadas pela alta administração.

b) **sistema da qualidade:** Abrange a documentação do sistema da qualidade, definindo a sua estrutura e um plano da Qualidade. Define uma matriz de responsabilidades e autoridades para o pessoal que executa trabalhos que influem na qualidade.

c) **auditorias internas do sistema da qualidade:** Abrange o planejamento das auditorias, procedimentos para a execução da auditoria, relatório e acompanhamento dos resultados da auditoria e ações corretivas.

d) **ação corretiva:** Abrange responsabilidades, avaliação e investigação de produtos não-conformes, análise de problemas, controle do processo, implantação de melhorias e ações preventivas.

### 3.3.2 ATIVIDADES DO CICLO DE VIDA DO SOFTWARE

Independente do modelo de ciclo de vida definido pela empresa, a norma prevê que as atividades do ciclo podem ser agrupadas em nove grandes categorias [ASS93]:

a) **análise crítica de contrato:** Abrange o registro em contrato dos requisitos do software, definição de responsabilidades (fornecedor e cliente) e cessão de direitos de uso do software.



b) **especificação dos requisitos do comprador:** Abrange a definição dos requisitos do software, retroalimentação proveniente do mercado e cooperação mútua.

c) **planejamento do desenvolvimento:** Abrange a definição de um plano de desenvolvimento do software, bem como as fases de desenvolvimento. O plano deve incluir a definição do projeto, organização dos recursos, cronograma e planos de teste. O plano deve incluir também formas de controle de entradas e saídas para cada fase do ciclo de vida e um método de monitorar e verificar o progresso.

d) **planejamento da qualidade:** Abrange a preparação de um plano da qualidade. Este plano contém exame de ensaio, verificação e validação; objetivos da qualidade e definição de responsabilidades específicas.

e) **projeto e implementação:** Abrange a definição de um projeto de software disciplinado, onde comprador e fornecedor concordam, previamente, sobre o conjunto de informações do projeto (do fornecedor) que serão fornecidas ao comprador. O desenvolvedor deve utilizar uma metodologia sistemática. O projeto deve levar em consideração as futuras atividades de manutenção e aderir a regras e convenções de programação.

f) **ensaios e validação:** Enfatiza a necessidade de teste do software em vários níveis. Um plano de ensaios deve ser sugerido, cobrindo alguns fatores como: ambiente, documentação, casos de teste e dados. A validação do sistema completo e ensaios de campo devem também ser abordados pelo plano de ensaios.

g) **aceitação:** Cobre os critérios acordados previamente pelo comprador, planejamento e execução dos ensaios de aceitação.

h) **cópia, entrega e instalação:** Trata do registro de considerações relativas ao número de cópias, tipo de meio físico utilizado, direitos autorais e licenças, critérios de envio e obrigações do fornecedor e do comprador ligadas à instalação.

i) **manutenção:** Abrange a definição de um plano de manutenção, organização de suporte, atividades de manutenção, registros e relatório de manutenção e procedimentos de liberação.

### 3.3.3 ATIVIDADES DE SUPORTE DO SISTEMA DA QUALIDADE

Não se encontram atreladas a uma determinada fase do ciclo de vida do software. As atividades de suporte compreendem nove itens, os quais devem ser desenvolvidos pelo fornecedor do software [ASS93]:

a) **sistema de gestão de configuração:** Abrange a definição de um mecanismo para controle e rastreabilidade do software, de modo que seja possível: identificar de forma única cada versão; controlar a atualização simultânea do software por mais de uma pessoa; identificar e seguir todas as alterações resultantes de uma solicitação de alteração e assim por diante.

É preciso registrar, administrar e relatar a situação de itens de software.

b) **controle de documentos:** Abrange a determinação dos documentos que devem ser controlados, definição da aprovação e emissão de procedimentos e sistemática de alterações em documentos.

c) **registros da qualidade:** Abrange a definição de um procedimento para identificar, coletar, indexar, arquivar, armazenar, manter e dispor os registros da qualidade, de forma que sejam prontamente recuperáveis.

d) **medição:** Trata das métricas e das técnicas de medição estabelecidas para realizar medições nos produtos e nos processos desde o desenvolvimento até a expedição.

e) **regras, práticas e convenções:** Abrange a definição de regras, práticas e convenções de modo a tornar efetivo o sistema da qualidade especificado na NBR ISO 9000-3.

f) **ferramentas e técnicas:** Determina como responsabilidade do fornecedor, a utilização ferramentas, recursos e técnicas que garantam a efetividade das diretrizes do sistema da qualidade.

g) **aquisição:** Abrange a avaliação, seleção e qualificação de fornecedores e produtos; ensaios de aceitação e recebimento (validação dos produtos adquiridos) e exigência da documentação de aquisição.

h) **produto de software incluído:** Trata basicamente dos cuidados relativos à qualidade para emprego de partes do software do próprio comprador, que serão integradas para compor o software contratado.

i) **treinamento:** Abrange a identificação das necessidades de treinamento interno, visando a qualificação do próprio pessoal que executa tarefas que influem na qualidade. É preciso manter registros do treinamento / experiências.

## 3.4 MANUTENÇÃO DE SOFTWARE SEGUNDO A NORMA

### ISO 9000-3

Segundo [NBR93] e [HOR98] a manutenção deve ser estipulada no contrato, após a entrega e a instalação inicial, quando requerida pelo comprador. O fornecedor deve estabelecer e manter procedimentos para realizar atividades de manutenção e verificar se estas atividades atendem aos requisitos especificados para a manutenção. Pode-se classificar em:

- a) resolução de problemas;
- b) modificação e interface;
- c) expansão funcional ou aprimoramento do desempenho.

#### 3.4.1 Plano de Manutenção

O plano de manutenção deve incluir: objetivo da manutenção, identificação da situação inicial do produto, organizações de suporte, atividades de manutenção, registros e os relatórios de manutenção.

a) **Identificação da situação inicial do produto:** Deve ser definida, documentada e acordada entre fornecedor e comprador.

b) **Organização de Suporte:** Como as atividades da etapa de manutenção não podem ser sempre realizadas de acordo com uma programação, esta organização deve ser bastante flexível para lidar com a ocorrência de problemas inesperados. Também pode ser necessário identificar instalações e recursos a serem usados para as atividades de manutenção.

c) **Tipos de atividades de manutenção:** Todas as alterações no “software” devem ser documentadas de acordo com os procedimentos para controle de documentos:

- Resolução de problemas: envolve a detecção, a análise e a correção de não conformidade de “software” que possam causar problemas operacionais
- Modificações de interface: podem ser requeridas, quando acréscimos ou alterações são feitos no sistema de “hardware”, ou nos componentes de “software”.
- Expansão funcional ou aprimoramento do desempenho pode ser requerido pelo comprador durante a etapa de manutenção.

d) **Registros e relatórios de manutenção:** Os registros de manutenção devem incluir, para cada item de “software” em que se realiza a manutenção, os seguintes tópicos:

- lista de solicitações de assistência ou de relatórios de problemas que tiverem sido recebidos, e a situação de cada um deles;
- organização responsável pelo atendimento às requisições para assistência ou implementação das ações corretivas adequadas;
- resultados das ações corretivas.

### 3.4.2 PROCEDIMENTOS DE LIBERAÇÃO

O fornecedor e o comprador devem entrar em acordo e documentar os procedimentos para a incorporação de alterações em um produto de “software”, deve-se incluir em tais procedimentos:

a) regras básicas para determinar onde as correções (“patches”) localizadas podem ser incorporadas, ou quando é necessária a liberação de uma cópia completa atualizada do produto de “software”;

b) descrições dos tipos de liberações, dependendo de sua frequência e/ou do impacto sobre as operações e a capacidade do comprador de implementar alterações a qualquer momento;

- c) métodos pelos quais o comprador deve ser alterado sobre alterações atuais ou planejadas para o futuro;
- d) métodos para confirmar que as alterações implementadas não introduzem outros problemas;
- e) requisitos para os registros, indicando quais alterações foram implementadas e em que locais, para os vários produtos e instalações.

### 3.5 SPICE

Segundo [BAR97], a ISO/IEC 15504 é uma norma em elaboração conjunta pela (*International Organization for Standardization*) ISO e pelo (*International Electrotechnical Commission*) IEC e atualmente é conhecida como projeto SPICE (*Software Process Improvement and Capability dEtermination*). Ela constitui-se de um padrão para a avaliação do processo de software, visando determinar a capacitação de uma organização. A norma visa ainda orientar a organização para uma melhoria contínua do processo. Ela cobre todos os aspectos da Qualidade do Processo de Software.

Um grupo de estudos da (Associação Brasileira de Normas Técnicas) ABNT está participando do processo de desenvolvimento, além de trabalhar na tradução das versões preliminares da norma para o português.

O projeto SPICE baseia-se nas melhores características de vários modelos de avaliação de processos existentes hoje, tal como: SW-CMM, Trillium, Software Technology Diagnostic (STD), Bootstrap e ISO 9001/9000-3. O resultado deste projeto será transformado na norma ISO/IEC 15504 (Tecnologia de Informação – Avaliação de Processos de Software) até o ano 2001.

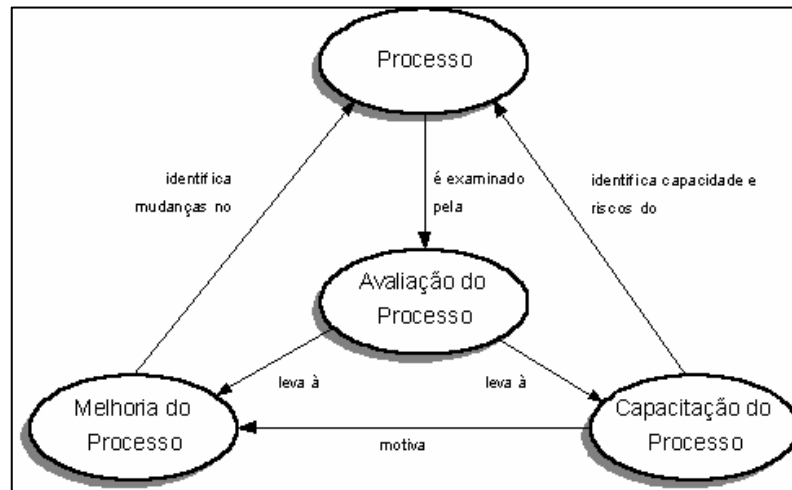
Segundo [TSU97], dentro da visão do SPICE (Figura 6), a avaliação de processos de software tem como propósito:

- a) entender o estado dos processos de uma organização para a melhoria destes processos;

b) determinar a adequação dos processos de uma organização para um requisito particular ou uma classe de requisitos;

c) determinar a adequação dos processos de uma outra organização para um determinado contrato ou para uma classe de contratos.

Figura 6 – Avaliação de Processo de Software – SPICE.



Fonte: [TSU97]

Dentro do contexto da melhoria de processos, a avaliação significa a caracterização das práticas correntes de uma organização, unidade organizacional ou projeto em termos da capacidade dos processos selecionados. A análise dos resultados é feita em relação às necessidades de negócio da organização, identificando os aspectos positivos e negativos, e os riscos associados aos processos. Isto leva a determinar se os processos estão atingindo efetivamente seus objetivos e identificar causas da baixa qualidade, alto custo ou tempo excessivo, indicando a priorização na melhoria dos processos.

A determinação da capacidade dos processos de uma organização é feita através da comparação das capacidades de suas práticas contra um modelo para gerenciamento de processos, onde engloba atividades que se acredita serem fundamentais para uma boa engenharia de software. Essas atividades são estruturadas de modo a proporcionar um modelo lógico do processo de software, identificando práticas que permitem o gerenciamento e melhoria de qualquer parte do processo ou do processo como um todo.

A complexidade do software moderno tem dificultado as companhias que desenvolvem, ou adquirem o software a identificar riscos, para controle de custos, melhoria da eficiência e da qualidade. Gerentes necessitam entender a capacidade de uma organização

em desenvolver sistemas de software. Este entendimento, que o SPICE fornece, deve ser profundo o suficiente para habilitar a identificação dos riscos e promover melhorias na maneira que a organização produz software.

O SPICE inclui um modelo de referência, que serve de base para o processo de avaliação. Este modelo é um conjunto padronizado de processos fundamentais, que orientam para uma boa engenharia de software. Este modelo é dividido em cinco grandes categorias de processo: Cliente-Fornecedor, Engenharia, Suporte, Gerência e Organização. Cada uma destas categorias é detalhada em processos mais específicos. Tudo isso é descrito em detalhes pela norma ISO/IEC 15504.

Além dos processos, o SPICE define também os 6 níveis de capacitação de cada processo, que podem ser incompleto, executado, gerenciado, estabelecido, previsível e otimização. O resultado de uma avaliação, portanto, identifica um perfil da instituição em forma de matriz, onde tem os processos nas linhas e os níveis nas colunas.

### 3.5.1 Cliente-Fornecedor

A categoria CLIENTE-FORNECEDOR consiste de processos que impactam diretamente com o cliente, tais como desenvolvimento de suporte, transição do software para o cliente e fornecimento de assistência/consultoria com relação à operação e uso do produto de software e/ou serviço. Na Tabela 2 estão demonstrados os processos desta categoria.

Tabela 2 – Processos Cliente/Fornecedor.

CUS – Cliente-Fornecedor	
Processo	Descrição
CUS.1	Processo de Aquisição : Obtenção do produto de software e/ou serviço que satisfaça as necessidades expressas pelo cliente. O processo inicia com a identificação das necessidades do consumidor e finaliza com a aceitação do produto e/ou serviço solicitado pelo cliente.

CUS.1.1	Preparação do Processo de Aquisição : Estabelece as necessidades e metas da aquisição.
CUS.1.2	Processo de Seleção do Fornecedor : Identifica a organização que irá se responsabilizar pelo projeto definido em CUS.1.1
CUS.1.3	Monitoramento do Processo de Fornecimento : Monitora o fornecedor durante o processo de desenvolvimento do produto de software e/ou serviço.
CUS.1.4	Processo de Aceitação do Cliente : Aprovação do fornecedor que satisfaz todas as condições impostas pelo cliente.
CUS.2	Processo de Fornecimento : Fornece o software ao cliente conforme os requisitos concordados.
CUS.3	Processo de Elicitação de Requisitos : Localiza problemas que venham interferir na vida do produto de software e/ou serviço, também para estabelecer uma linha que servirá para identificar a linha base do produto de software e/ou serviço.
CUS.4	Processo de Operação : Define o processo de operação e entendimento do produto de software e fornecimento de suporte ao cliente.
CUS.4.1	Processo de Uso Operacional : Assegura a correta e eficiente operação do produto de software pela duração de entendimento de seu uso e seu ambiente.
CUS.4.2	Processo de Suporte ao Cliente : Estabelece e mantém um nível de suporte, assistência e consultoria aceitável ao cliente.

Fonte: [IAH99]

### 3.5.2 Engenharia

Esta categoria consiste de processos que especificam, implementam ou mantêm o produto de software com relação ao sistema e documentação ao cliente. Seus processos estão descritos na tabela 3.

Tabela 3 – Processos de Engenharia de Software.

ENG – Engenharia	
Processo	Descrição
ENG.1	Processo de Desenvolvimento : Transforma os requisitos de software anteriormente acordado em um produto de software funcional.
ENG.1.1	Processo de Análise dos Requisitos e Projeto do Sistema : Estabelece os requisitos e arquitetura do sistema, identifica quais requisitos deverão ser alocados para quais elementos dos sistema.



ENG.1.2	Processo de Análise dos Requisitos : Estabelece os requisitos dos componentes do sistema.
ENG.1.3	Processo de Projeto de Software : Define um modelo que acomode os requisitos do software.
ENG.1.4	Processo de Construção do Software : Produz módulos executáveis de software e verifica se os mesmos refletem o projeto do sistema.
ENG.1.5	Processo de Integração de Software : Junta os módulos executáveis de forma que integrem os itens conforme o projeto do sistema.
ENG.1.6	Processo de Teste de Software : Testa a integridade do sistema conforme seus requisitos.
ENG.1.7	Processo de Teste e Integração do Sistema : Integra os vários componentes do software, produzindo um sistema completo que satisfaça as expectativas expressas nos requisitos do sistema, incluindo manuais e hardware.
ENG.2	Processo de Manutenção de Software e Sistema : Modificações, migrações e exclusões de componentes do sistema, são realizadas neste processo.

Fonte: [IAH99]

### 3.5.3 Suporte

Esta categoria consiste de processos que podem ser empregados por qualquer um dos outros processos. Estes processos estão demonstrados na tabela 4.

Tabela 4 – Processos de Suporte de Software.

SUP – Suporte	
Processo	Descrição
SUP.1	Processo de Documentação : Desenvolve e mantém documentos que registram informações produzidas por processos ou atividades.
SUP.2	Processo de Gerenciamento de Configurações : Estabelece e mantém a integridade de todos os produtos ou projetos.
SUP.3	Processo de Garantia da Qualidade : Assegura que os produtos e atividades de um processo ou projeto esta conforme todos os padrões aplicáveis.
SUP.4	Processo de Verificação : Confirma que cada produto de software e/ou serviço de um processo ou projeto estão de acordo com os requisitos.
SUP.5	Processo de Validação : Confirma que os requisitos especificados estão de acordo com o software que foi trabalhado.

SUP.6	Processo de Revisão Conjunta : Mantém um entendimento comum com o cliente do progresso do projeto contra os objetivos do contrato e que deve ser continuado para assegurar e desenvolvimento de um produto que satisfaça o cliente.
SUP.7	Processo de Auditoria : Confirma independentemente que os produtos e processos empregados estão conforme os requisitos acordados.
SUP.8	Processo de Resolução de Problemas : Assegura que todos os problemas descobertos são analisados e resolvidos.
SUP.9	Processo de Medição : Coleta e analisa dados relativos aos produtos desenvolvidos e processos implementados dentro da unidade organizacional e demonstra objetivamente a qualidade dos produtos.
SUP.10	Processo de Reuso : Promove e facilita o reuso de software.

Fonte: [IAH99]

### 3.5.4 Gerência

Processos que contém práticas de natureza genérica que podem ser usadas por quem gerencia projetos ou processos dentro de um ciclo de vida de software. Estes processos são apresentados na tabela 5.

Tabela 5 – Processos de Gerência de Software.

MAN – Gerência	
MAN.1	Gerenciar o projeto: Define os processos necessários para estabelecer, coordenar e gerenciar um projeto e seus recursos necessários para a produção do produto.
MAN.2	Gerenciar a qualidade: Gerencia a qualidade dos serviços e dos produtos do projeto e assegura que os mesmos satisfaçam as necessidades do cliente.
MAN.3	Gerenciar riscos: Identifica continuamente e alivia os riscos do projeto, desde de seu início e durante o ciclo de vida do mesmo.
MAN.4	Gerenciar subcontratantes: Seleciona os subcontratados qualificados e gerencia a eficiência dos mesmos.

Fonte: [IAH99]

### 3.5.5 Organização

Os processos da tabela 6 estabelecem os objetivos de negócios da organização.

Tabela 6 – Processos de Organização.

ORG – Organização	
ORG.1	Construir o negócio: Suprir as pessoas na organização e projetos com uma visão e cultura que capacitem os mesmos para uma função eficiente.
ORG.2	Definir o processo: Construir e reutilizar bibliotecas de definições de processos (incluindo padrões, procedimentos e modelos)
ORG.3	Melhorar o processo: Melhoria contínua, efetiva e eficaz dos processos usados pela organização que estão em linha com as necessidades do negócio da mesma.
ORG.4	Prover recursos de treinamento: Suprir a organização e os projetos com pessoas capazes e eficientes, fornecendo a elas treinamento.
ORG.5	Prover infra-estrutura organizacional: Sustentar um ambiente estável e confiante, integrando métodos de desenvolvimento com ferramentas que auxiliem nos processos da organização.

Fonte: [IAH99]

### 3.5.6 Níveis de Capacitação

O SPICE, entretanto, não se limita a listar categorias e processos. Seu principal objetivo, na realidade, é avaliar a capacitação da organização em cada processo e permitir a sua melhoria. O modelo de referência do SPICE inclui seis níveis de capacitação. Cada um dos processos mencionados acima deve ser classificado nestes níveis. Os níveis são descritos na tabela 7.

Tabela 7 – Níveis de Capacitação

Nível	Nome	Descrição
0	Incompleto	Há uma falha geral em realizar o objetivo do processo. Não existem produtos de trabalho nem saídas do processo facilmente identificáveis.
1	Realizado	O objetivo do processo em geral é atingido, embora não necessariamente de forma planejada e controlada. Há um consenso na organização de que as ações devem ser realizadas e quando são necessárias. Existem produtos de trabalho para o processo e eles são utilizados para atestar o atendimento dos objetivos.
2	Gerenciado	O processo produz os produtos de trabalho com qualidade aceitável e dentro do prazo. Isto é feito de forma planejada e controlada. Os produtos de trabalho estão de acordo com padrões e requisitos.
3	Estabelecido	O processo é realizado e gerenciado usando um processo definido, baseado em princípios de Engenharia de Software. As pessoas que implementam o processo

		usam processos aprovados, que são versões adaptadas do processo padrão docum..
4	Predizível	O processo é realizado de forma consistente, dentro dos limites de controle, para atingir os objetivos. Medidas da realização do processo são coletadas e analisadas. Isto leva a um entendimento quantitativo da capacitação do processo a uma habilidade de prever a realização.
5	Otimização	A realização do processo é otimizada para atender às necessidades atuais e futuras do negócio. O processo atinge seus objetivos de negócio e consegue ser repetido. São estabelecidos objetivos quantitativos de eficácia e eficiência para o processo, segundo os objetivos da organização. A monitoração constante do processo segundo estes objetivos é conseguida obtendo <i>feedback</i> quantitativo e o melhoramento é conseguido pela análise dos resultados. A otimização do processo envolve o uso piloto de idéias e tecnologias inovadoras, além da mudança de processos ineficientes para atingir os objetivos definidos.

Fonte: [IAH99]

### 3.6 MANUTENÇÃO DE SOFTWARE SEGUNDO O SPICE

Segundo [ISO97], a etapa de manutenção de software no modelo SPICE, está encubado no processo ENG.2 “*processos para manutenção de softwares e sistemas*”. O propósito deste processo é o gerenciamento, ou administração das modificações, migrações, ou retirada de componentes do sistema em resposta aos pedidos de usuários. A origem dos pedidos pode ser a descoberta de um problema ou a necessidade de melhoria ou ainda alguma adaptação. O objetivo é modificar e/ou retirar sistemas e softwares existentes preservando a integridade operacional da organização.

Para um resultado bem-sucedido da implementação do processo é necessário:

- Desenvolver uma estratégia de manutenção para administrar a modificação, migração e retirada de componentes do sistema de acordo com a estratégia de liberação.
- Definir o impacto na organização, operações e interfaces do sistema existente em operação.
- Existência de especificação, desenvolver documentos e testes estratégicos para a atualização.

- Desenvolver a modificação dos componentes de sistema com testes associados de modo a demonstrar que os requisitos atuais do sistema não foram afetados.
- Modelar as atualizações de softwares e sistemas para o ambiente do usuário.
- Padronizar as requisições de sistemas e softwares de modo a minimizar os distúrbios para os usuários e facilitar o controle.

Este processo ainda interage com outros processos que são: CUS4 – Processo de Operação, onde seu propósito é operar o produto de software no ambiente do cliente e prover suporte ao mesmo; CUS4.2 – Processo de suporte ao cliente, onde seu propósito é estabelecer e manter um nível aceitável de serviços para o cliente no que diz respeito ao suporte efetivo do produto de software; e SUP8 – Processo de Resolução de Problema, onde seu propósito é assegurar a descoberta de problemas, analisar e resolvê-los.

### 3.7 COMPARATIVO ENTRE NORMAS E MODELOS

O comparativo do processo de manutenção, perante as normas e modelos de qualidade utilizou o seguinte critério: conforme demonstra a tabela 8, foi extraído da literatura pesquisada, (conforme capítulo 2) os pontos relevantes que deve conter um processo de manutenção. Em seguida foi analisado cada modelo e norma de qualidade para verificar se as mesmas atendiam as características coletadas e o que mais as normas acrescentam. Por fim, foram analisadas as características que todas as normas atendiam. Desta forma, estas características já estariam propostas, após verificou-se as características divergentes entre as normas para então definir o que deveria também fazer parte do processo proposto. Por último verificou-se itens que também seriam importantes estar na norma.

Tabela 8 – Comparativo entre as normas de Qualidade

<b>Tabela comparativa entre as normas e modelos de qualidade</b>	
Aspectos	<b>Normas e modelos de qualidade</b>

	ISO/IEC 12207	ISO 9000-3	SPICE
Tipos de manutenção	Análise do problema da manutenção	Tipos de atividades de manutenção	Definir a estratégia...
Tarefas da manutenção			
Organização para manutenção	Implementação do proc. de manutenção	Organização de suporte	Definir a estratégia...
Relatórios	Implementação da modificação	Registro e Relatórios de manutenção	Desenvolver documentação
Tabela 8 – Continuação			
Fluxo dos eventos	Implementação do proc. de manutenção	Organização de suporte	Definir a estratégia...
Conservação dos registros	Implementação do proc. de manutenção		Definir o impacto na organização
Avaliação	Revisão/Aceitação		Testes estratégicos
Tarefas técnicas da manutenção			
Avaliar a solicitação de melhoramento do sistema	Análise do problema da manutenção	Identificação da situação inicial	Definir o impacto na organização
Avaliar a solicitação de correção do sistema	Análise do problema da manutenção	Registro e relatório de manutenção	Definir o impacto na organização
Aplicar conserto de emergência no programa	Implementação da modificação		
Organizar as solicitações de melhoramento em releases	Análise do problema e da	Registro e relatório de	Desenvolver uma estratégia ...

individuais	modificação	manutenção	
Analisar o release de solicitação de manutenção do sistema	Implementação da modificação	Tipos de atividades de manutenção	Testes estratégicos
Projetar o release de manutenção do sistema	Implementação da modificação	Tipos de atividades de manutenção	Desenvolver a modificação
Codificar e testar o release de manutenção do sistema	Implementação da modificação	Tipos de atividades de manutenção	Desenvolver a modificação
Implementar o release de Tabela 8 – Continuação	Implementação da modificação	Atividades de manutenção	Desenvolver a modificação
Executar a manutenção preventiva			
Treinar a equipe			
Executar avaliações periódicas no sistema	Revisão		Testes estratégicos
Executar avaliações pós-implementações	Revisão/Aceitação		Testes estratégicos

*No documento [ISO97], existe um mapeamento da norma SPICE em relação a ISO/IEC 12207 onde é colocado que a norma SPICE atende diretamente a todos os requisitos da ISO/IEC 12207. Então no quadro acima apenas detalhado qual item do SPICE, atende ao requisito de manutenção exposto.*

### **3.7.1 EXPLICAÇÃO DO COMPARATIVO**

Utilizando como base [FOU94], [KRA98], [NBR93], [PRE95], [PAR90] e [SCU98], segue uma breve descrição do resultado do comparativo entre a teoria de manutenção de sistemas e as normas de qualidade:

Tipos de manutenção: é um item incluso em todas as normas pois é necessário saber de que tipo de manutenção está se tratando, pois muitas vezes dependendo do tipo de manutenção a mesma tem mais ou menos prioridade.

Organização para manutenção: A partir desta etapa que toda a manutenção toma forma, se tivermos uma organização falha com toda certeza num futuro não muito distante teremos problemas em administrar e manter nossos softwares.

Relatórios: Todas as normas referenciam os documentos e relatórios de manutenção, é a partir destes relatórios que todas as pessoas envolvidas no processo podem tirar dúvidas, corrigir falhas e a até mesmo propor melhorias estruturais no sistema. Os relatórios são uma fonte muito grande de conhecimento e por isto podem solucionar dezenas de problemas da manutenção, um software que possui poucos ou nenhum relatório de manutenção é um software sujeito a dificuldades inúmeras para a equipe de manutenção no momento de uma solicitação.

Fluxo dos eventos: Como várias outras etapas o fluxo em que as tarefas são realizadas precisa ser definido, para um processo de manutenção conciso e eficiente, é interessante que a equipe que vai tratar da manutenção tenha a suas mãos procedimentos de fácil e rápida compreensão e é claro adequados a todas as tarefas que a mesma terá que realizar, devido a isto todas as normas também consideram que um bom processo de manutenção necessita de planejamento para que possa chegar ao objetivo.

Conservação dos registros: Esta etapa é a primeira da nossa análise que não tem uma unanimidade entre as normas, pois devido a grande capacidade de armazenamento e a técnicas novas são poucos os problemas relacionados ao armazenamento de alguns dados desnecessários. É claro que nada pode chegar ao “absurdo”, ou seja devemos ter técnicas para saber quais os dados importantes e não criar ambigüidades, porém não é nenhum crime armazenar algum dados que não é de extrema importância para o funcionamento do software .

Avaliação: Apesar de nem todas as normas utilizarem a avaliação do software, esta etapa poderá ajudar um software que está operacional no mercado, a avaliação quando feita de forma profissional e muito bem sustentada pela parte técnica pode auxiliar a diminuir dezenas de problemas da empresa na hora de executar a manutenção, problemas que já poderiam ter sido resolvidos quando a equipe estava ociosa por exemplo e que agora



necessitam de muitas pessoas para executar em tempo hábil é um dos motivos para que se execute em um tempo determinado uma avaliação do software. Além deste aspecto uma avaliação da manutenção de software quando executada em um sistema pode também trazer dezenas de conhecimentos e noções de melhoramentos e técnicas para cada vez mais atingir metas de produtividade e qualidade acima das expectativas.

**Avaliar a solicitação de melhoramento do sistema:** Toda solicitação deverá ser muito bem analisada pois existem solicitações que podem ferir ao aspecto estrutural e a finalidade do software em questão e estas devem ser excluídas das pendências. Este item é também um item que está incluso em todas as normas de qualidade.

**Avaliar a solicitação de correção do sistema:** É necessário que o problema que ocasionou a solicitação seja extremamente pesquisado para que se possa chegar a conclusão correta do que ocasionou o problema e poder então aplicar o conserto correto. Este item e também levado muito em consideração pelas normas.

**Aplicar conserto de emergência no programa:** Em um conserto de emergência muitos dos controles e análises que deviriam ser executadas são deixadas de lado o que pode num futuro não muito distante ocasionar outros tipos de problema, por este motivo não é um consenso geral que deve-se fazer o conserto de emergência, mais seria muito adequado dizer que nem tudo que deveria ser feito acontece então, a norma deverá trazer subsídios para que mesmo que em pequenas porcentagens haja a possibilidade de se fazer um conserto de emergência.

**Organizar as solicitações de melhoramento em releases individuais:** Para que se possa ter um resultado e um controle melhor sobre as manutenção executadas num software, é necessário que todas as manutenções sejam encaradas individualmente, pois, se a equipe começar a acumular muitas manutenções com o intuito de economizar tempo poderá ter vários problemas para organizar as manutenções como começar a introduzir erros com as manutenções em vez de tira-los.

Analisar o release de solicitação de manutenção do sistema: Uma tarefa em que as normas também demonstram ser de bastante utilidade pois é nela que é preparada a solicitação para o projeto.

Projetar o release de manutenção do sistema: É neste ponto que a solicitação passa a tomar forma, toda a estrutura e funções são descritas nesta etapa e para que se possa ter segurança e eficiência na manutenção esta tarefa deverá ser bem documentada e bem implementada.

Codificar e testar o release de manutenção do sistema: esta etapa é realmente a concretização de todo o serviço da manutenção de sistemas, conforme já visto todas as normas falam sobre os aspectos de codificar e testar o software, e é muito importante que esta etapa seja realizada de forma organizada, estruturada e adequada as necessidades expostas.

Implementar o release de manutenção do sistema: Neste momento a manutenção executada entra em processo de produção, todos os passos anteriores e o modo como o software será colocado em operação devem ser adequados para que a organização não tenha muitos problemas com paralisações.

Executar a manutenção preventiva: Apesar de poder ser uma tarefa que possa trazer algumas facilidades para a equipe de manutenção nenhuma das normas explicita a manutenção preventiva.

Treinar a equipe: Outra tarefa que é negligenciada pelas normas é o treinamento, porém acredito que esta tarefa possa ser de muito grande benfeitoria para a empresa pois capacita as pessoas para a utilização do software.

Executar avaliações periódicas no sistema: Esta etapa é utilizada para que o sistema que está em funcionamento receba avaliações para verificar se o mesmo está atendendo todos os requisitos para que foi desenvolvido. Algumas das normas não contemplam esta atividade.

Executar avaliações pós-implementações: Igualmente a primeira esta avaliação apesar de não ser consenso poderia trazer melhorias para o funcionamento de softwares, porém a sua implementação se daria quando a equipe ficasse parada o que não é muito comum.

## 4 PROPOSTA DE PROCESSO DE MANUTENÇÃO

*Segundo[SCU98], fazer a manutenção de sistemas em tempo hábil e com custo e benefícios adequados é um desafio para muitos atualmente. Conhecer os sistemas antigos, os desenvolvidos mais recentemente, e a automação dos processos são objetivos que muitas organizações têm como meta.*

*A manutenção de software quando segue uma especificação anteriormente proposta é uma tarefa que apesar de bastante trabalhosa não é considerada complexa pela equipe de desenvolvimento. O que é realmente necessário é a utilização de um processo bem definido e acima de tudo seguido ao “pé da letra”.*

Espera-se a seguir apresentar uma proposta de processo de manutenção que tenha uma visão mais simples e clara para a utilização pelos usuários de desenvolvimento de softwares. Para o desenvolvimento da proposta de processo de manutenção utilizou-se como base a norma ISO/IEC 12207, devido ao fato desta norma apresentar-se de forma melhor detalhada.

O processo proposto para manutenção de sistemas contém as fases e as tarefas que o responsável ou a equipe de manutenção devem executar no sistema. São englobados no processo desde a alteração no software devido algum problema, como também a melhoria ou adaptação de qualquer parte do sistema preservando-se sua integridade.

Para a especificação do processo procurou-se seguir o comparativo entre as etapas e conceitos de manutenção de software estudadas na bibliografia e as normas e modelos de qualidade.

As fases apresentadas no quadro 1 “Ciclo de vida do processo de manutenção” são específicas para o processo proposto de manutenção de sistemas e devem ser observadas pelo responsável pela manutenção de sistemas. Para facilitar o entendimento, cada fase será descrita separadamente, onde as setas demonstram o fluxo dos eventos e tarefas e os números são utilizados para sub-dividir os eventos.

Nos quadros a seguir utilizou-se das letras “a” e “b” o que significam respectivamente tratar-se da solicitação de correção e modificação. Os símbolos “S” e “N” são utilizados para demonstrar qual será o fluxo dos eventos quando uma pergunta deve ser respondida, caso o questionamento seja positivo deve-se seguir o caminho indicado por “S”, caso contrário utiliza-se caminho indicado por “N”.

O processo é sub-dividido da seguinte forma:

- 1 Implantação do Processo
  - 1.1 Registrar os problemas e modificações
  - 1.2 Receber os problemas e modificações
  - 1.3 Analisar a autenticidade do pedido
  - 1.4 Registrar pendências
- 2 Análise do problema/modificação
  - 2.1 Receber registro de pendência
  - 2.2 Priorizar solicitações
  - 2.3 Analisar o problema e modificações
  - 2.4 Registrar causas do problema
  - 2.5 Registrar alterações de modificação
  - 2.6 Aceitação e homologação da solução
  - 2.7 Encaminhar para responsável
- 3 Implementação da Solução
  - 3.1 Executar priorização
  - 3.2 Verificar coerências
  - 3.3 Modificar programas
  - 3.4 Modificar documentação
  - 3.5 Registrar alterações
  - 3.6 Executar testes preliminares
  - 3.7 Avaliar partes não modificadas
  - 3.8 Encaminhar para testes
  - 3.9 Registrar Envio
- 4 Revisão/Testes
  - 4.1 Receber Testes
  - 4.2 Examinar documentação

#### 4.3 Efetuar plano de teste

4.4 Executar plano de testes

4.5 Validar resultado

4.6 Documentar testes

4.7 Aprovar testes

4.8 Registrar liberação

4.9 Registrar problemas

#### - 5 Qualificação da Equipe

5.1 Organizar pessoal

5.2 Enviar relatórios

5.3 Registrar envio

#### - 6 Verificação Periódica

6.1 Organizar as manutenções

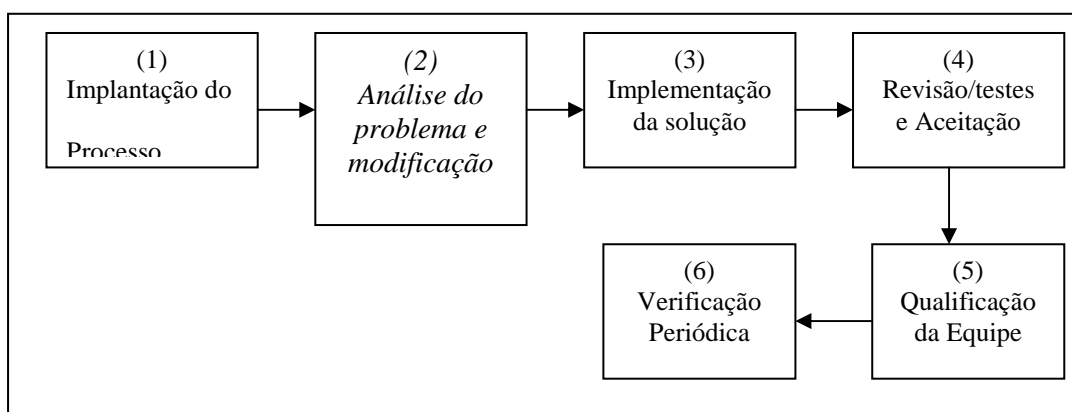
6.2 Verificar com testes

6.3 Verificar os usuários

6.4 Registrar testes

6.5 Encaminhar solicitações

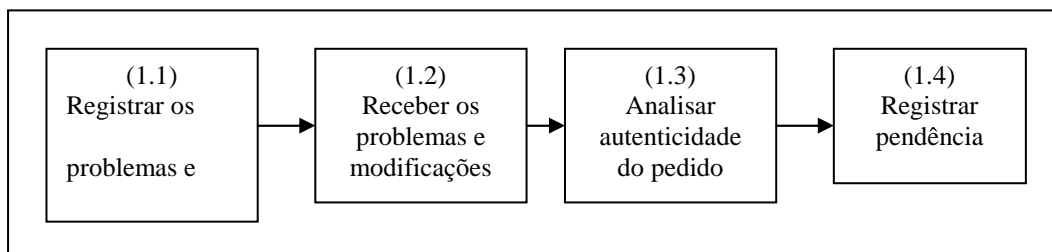
**Quadro 01 – Ciclo de Vida do Processo Proposto de Manutenção**



## 4.1 IMPLANTAÇÃO DO PROCESSO

O objetivo desta fase é estabelecer as tarefas para se iniciar o processo de manutenção, nesta etapa conforme o Quadro 2 estão dispostas as tarefas para registrar e homologar as solicitações de correção e também de modificação do sistema.

**Quadro 02 – Plano de Implantação do Processo de Manutenção**

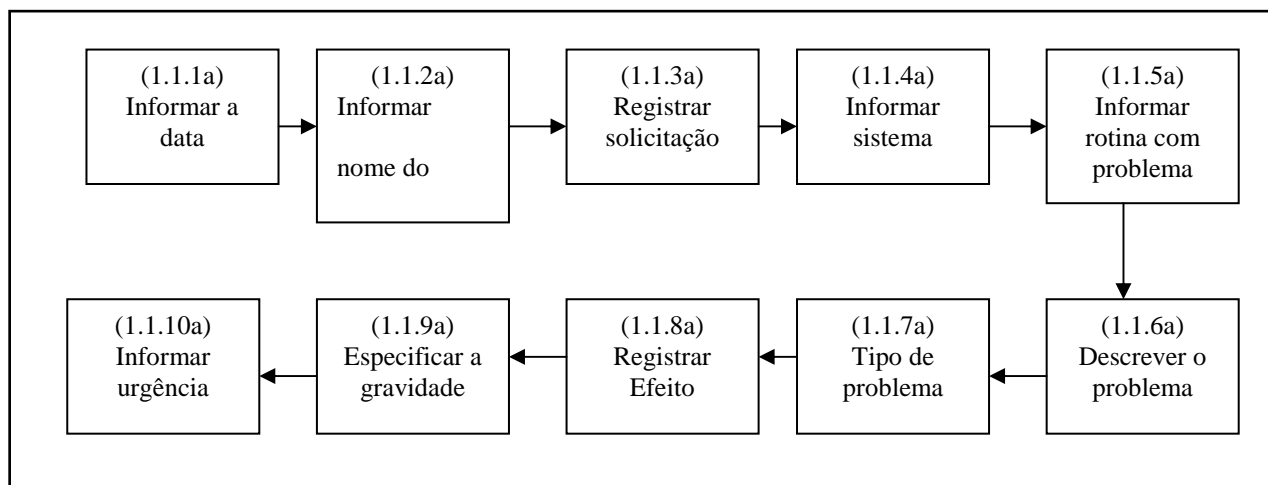


#### 4.1.1 REGISTRAR OS PROBLEMAS/MODIFICAÇÕES DO SOFTWARE

O registro de problemas no sistema é o mecanismo que inicia o processo de correção do sistema. É a partir dele que os usuários ou até mesmo os desenvolvedores informam que existe algum problema com o software. Este registro é a melhor forma de proporcionar à equipe de manutenção *feedback* dos problemas e das soluções das ocorrências de seus sistemas.

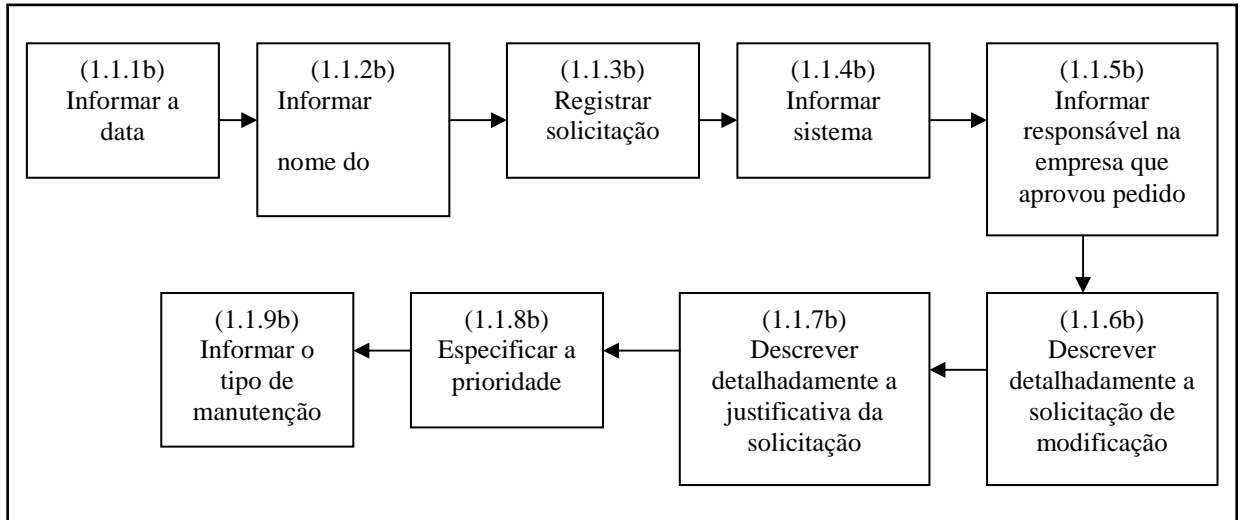
O quadro 03 demonstra esta etapa. É a partir deste momento que a solicitação será analisada e após especificada uma solução para o problema.

**Quadro 03 – Tarefas para registro de problema**



Sempre que uma solicitação ocorre conforme o quadro 04, ela é registrada para então em seguida ser analisada sua coerência, viabilidade e também calculando quando possível e necessário o custo/benefício da modificação.

**Quadro 04 – Tarefas para registro de modificação**

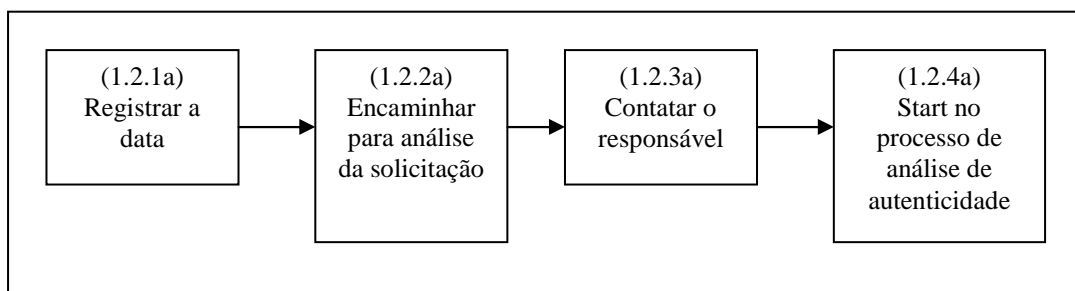


## 4.1.2 RECEBER OS REGISTROS DE PROBLEMA / MODIFICAÇÕES

Logo após que o pedido de correção é conhecido o mesmo deve ser protocolado e encaminhado, dependendo da urgência o mais rápido possível, para a próxima etapa pois dependendo do problema os usuários poderão estar com o sistema parado.

O Quadro 05 demonstra os passos que o responsável por esta etapa deve seguir.

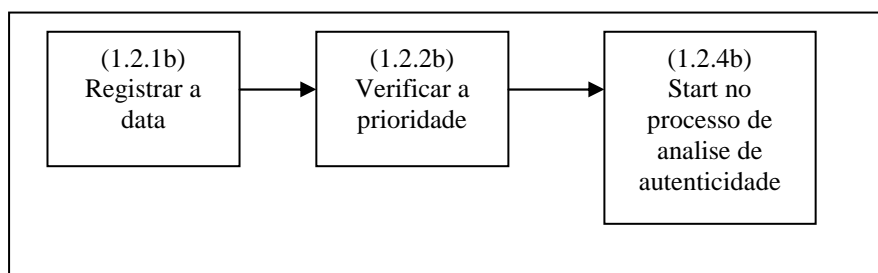
**Quadro 05 – Tarefas do recebimento do registro de problema**



O recebimento do registro de modificação segue praticamente a mesma seqüência de tarefas do recebimento de correção, conforme quadro 06. A diferença é que quando é recebido

este registro é informado para a equipe qual a prioridade em que o usuário necessita desta modificação, considerando evidentemente que esta prioridade não seja igual a prioridade que a equipe de desenvolvimento vai dar para esta solicitação e sim apenas para demonstrar à equipe a urgência dos usuários.

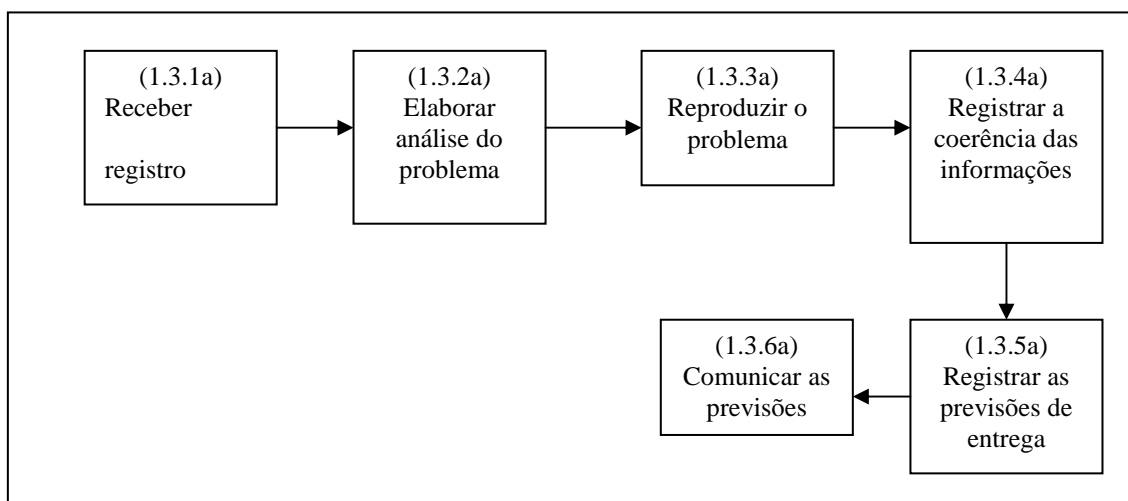
**Quadro 06 – Tarefas do recebimento do registro de modificação**



### 4.1.3 ANALISAR AUTENTICIDADE DO PEDIDO

Esta etapa conforme quadro 07 verifica se a comunicação do problema é realmente correta, pois, em muitas vezes o problema ocorre por alguma falha do equipamento do usuário e não por defeito no sistema. Esta análise visa principalmente em reproduzir o problema para que não se tenha dúvidas de que o sistema necessita de conserto, caso não seja verificado o problema a decisão deverá ser comunicada ao usuário e é claro tentar auxiliá-lo na busca da solução do problema.

**Quadro 07 – Análise de Autenticidade do Pedido de correção**

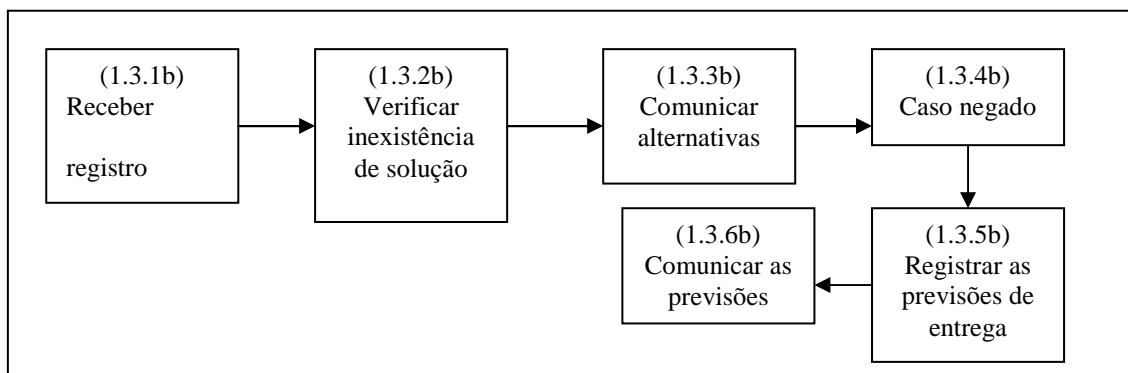


Já para a solicitação de modificação do software, esta etapa visa em principio, conforme quadro 08, verificar se já não existe uma solução no sistema sem que se tenha que



alterar o software existente. Caso exista uma alternativa esta deverá ser comunicada ao usuário para que ele então analise se esta solução atende a todos os requisitos que o mesmo necessita, caso a resposta for afirmativa é cancelada a solicitação de modificação.

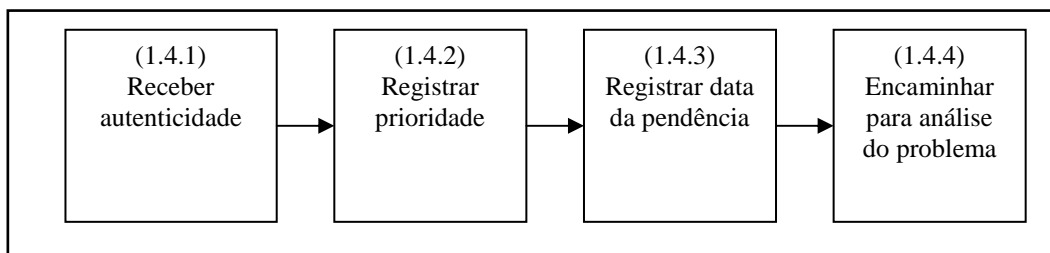
**Quadro 08 – Análise de autenticidade de modificação do software**



#### 4.1.4 REGISTRAR A PENDÊNCIA

Após a solicitação ser aprovada em todas as tarefas anteriores ela é registrada como um pendência para a equipe de manutenção, que deve em seguida priorizá-las e então executar os trabalhos necessários para a sua mais imediata solução. As tarefas desta fase são demonstradas no quadro 09.

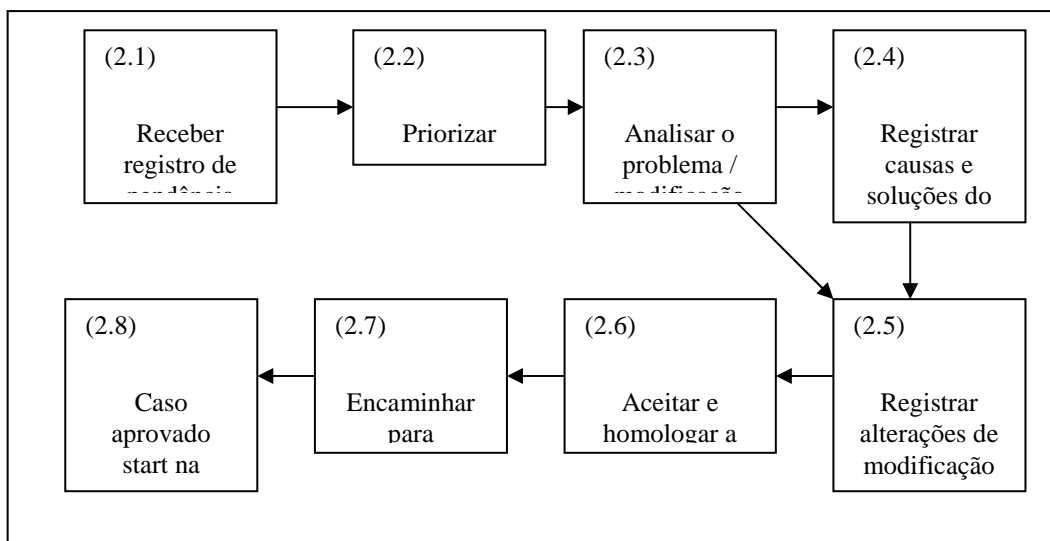
**Quadro 09 – Tarefas de Registro de Pendência**



## 4.2 ANÁLISE DO PROBLEMA E MODIFICAÇÃO

É nesta fase como demonstra o quadro 10 que a manutenção começa a acontecer. Este plano de análise prioriza, verifica, registra e estipula a solução para o problema e/ou modificação. O analista mantenedor deve analisar o pedido de correção e modificação, sempre lembrar de priorizar as solicitações, registrar quais as soluções e desenvolver então quais as alterações e modificações o software deve sofrer.

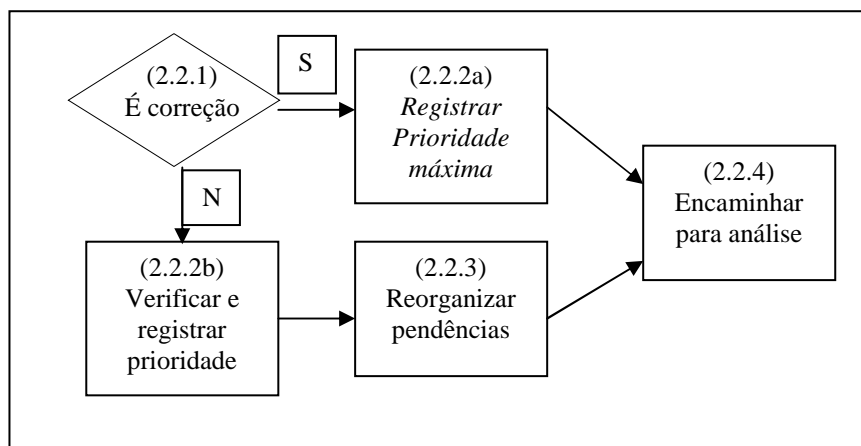
Quadro 10 – Plano de Análise do Problema e Modificação



## 4.2.1 PRIORIZAR AS SOLICITAÇÕES

As solicitações devem primeiramente, conforme demonstra o quadro 11, serem priorizadas em correções e modificações dando-se prioridade máxima para as correções no sistema. Esta etapa é importante para que haja uma seqüência de pendências a resolver e também para que as solicitações mais urgentes não fiquem paradas porque a equipe está resolvendo problemas que tem menos prioridade.

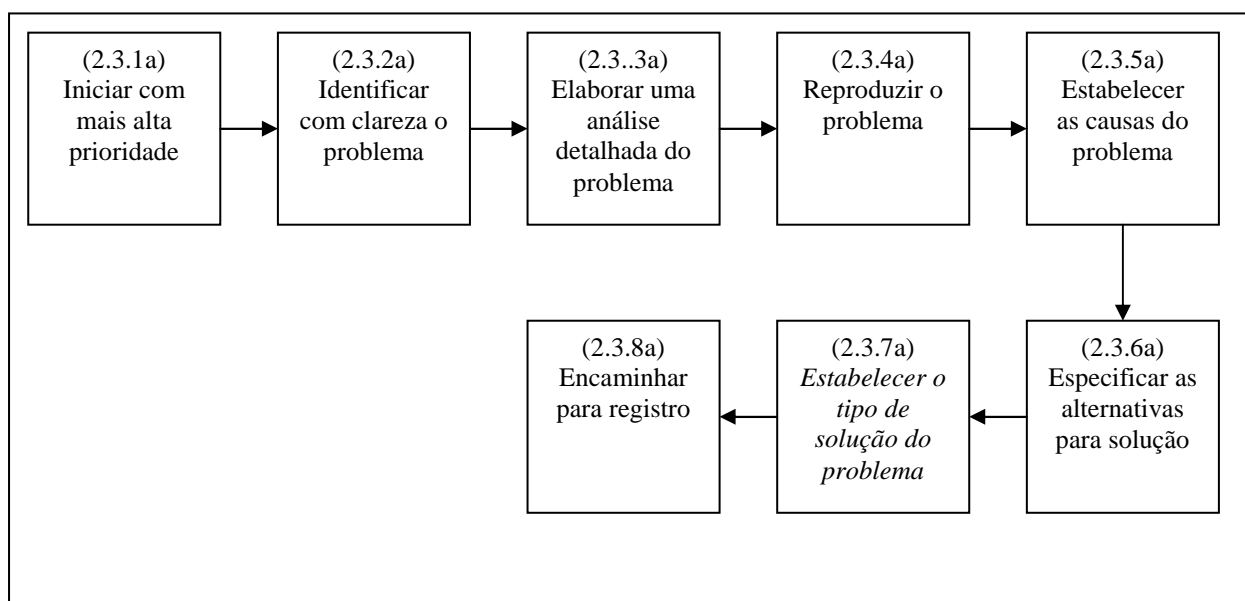
Quadro 11 – Tarefas de Priorização de Solicitações



## 4.2.2 ANÁLISE DO PROBLEMA E MODIFICAÇÃO

Conforme quadro 12, esta tarefa visa chegar a uma solução teórica ao problema, ou seja, primeiramente tentar novamente reproduzir o problema e verificar quais as causas do mesmo. Em seguida deve-se registrar estas causas e também especificar as alternativas de solução encontradas para sanar o problema. Uma vez esta tarefa concluída deve-se ainda verificar se a solução que se encontrou é definitiva para o problema ou ela irá funcionar apenas como uma *solução intermediária* para resolver o problema por um determinado tempo.

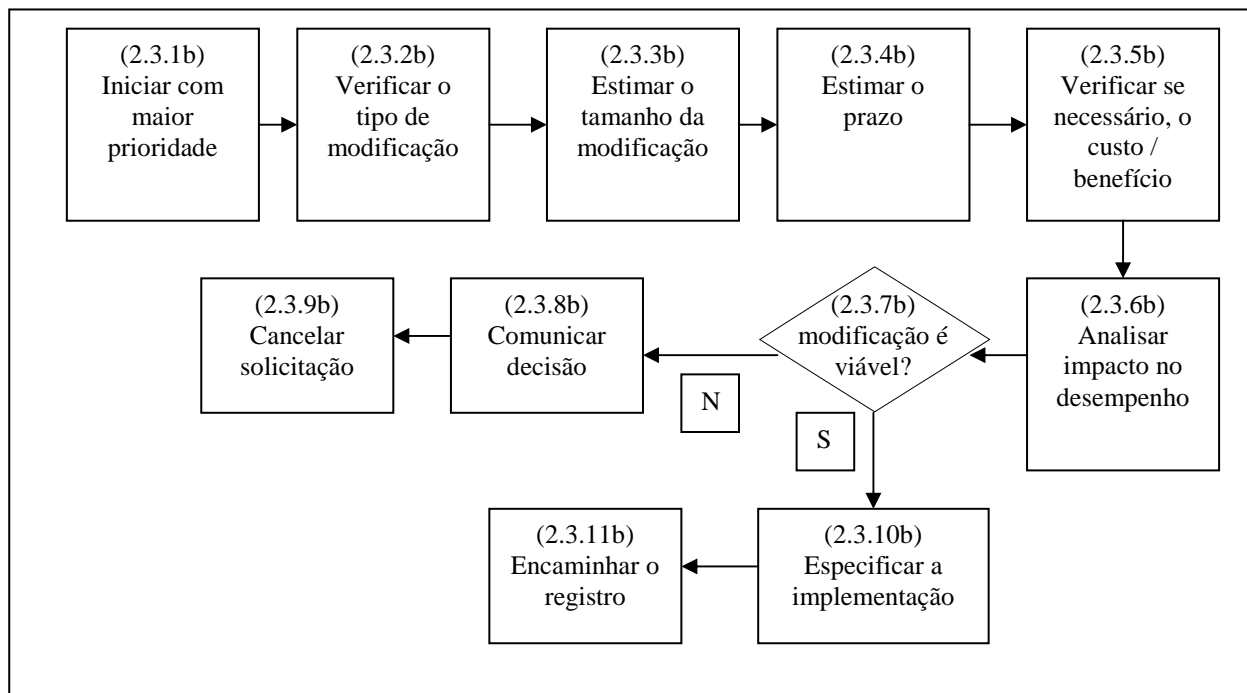
**Quadro 12 – Tarefas da Análise do Problema**



Para a análise da modificação do software (Quadro 13), é analisado entre outras tarefas se a modificação é viável, ou seja, se o custo/benefício ou impacto no desempenho do software desta modificação torna aceitável a realização da mesma. Muito importante para o mantenedor verificar o tipo de manutenção e estimar um prazo para que tanto o usuário quanto a equipe possam programar-se de acordo.

Caso esta modificação implique em custos para o usuário, está é a hora para informá-lo dos mesmos, pois antes de iniciar o processo de implementação deve se ter o aval total do usuário para que não ocorram problemas posteriores a entrega da modificação.

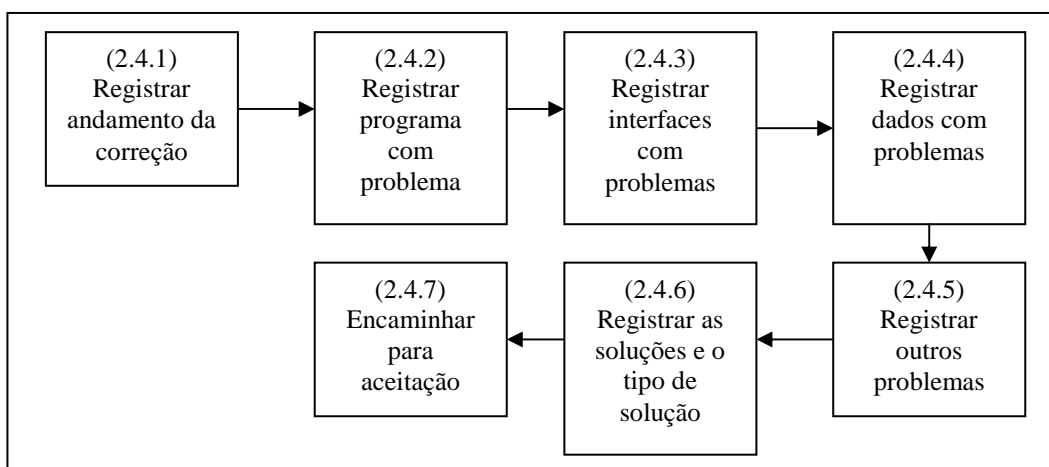
Quadro 13 - Tarefas da Análise da Modificação do Software



## 4.2.3 REGISTRAR CAUSAS DO PROBLEMA

Nesta etapa (Quadro 14) são registradas todas as causas dos problemas cujo o usuário informou. Este registro é muito importante, pois além da equipe ter um controle dos problemas solucionados, dá a equipe um grande recurso de no futuro encontrar e solucionar os problemas muito mais rapidamente, pois ela terá registrado todos os problemas anteriores e/ou um problema atual poderá ser resolvido com um *feedback* de um problema passado.

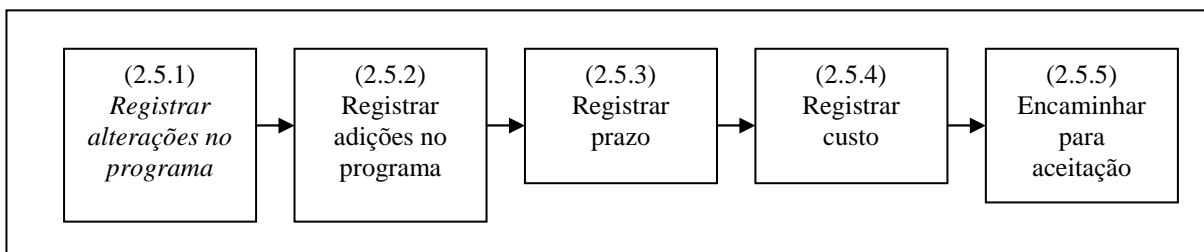
Quadro 14 – Tarefas para Registrar Causas do Problema



## 4.2.4 REGISTRAR ALTERAÇÕES DA MODIFICAÇÃO

O responsável pela manutenção deverá conduzir as atividades de registro da modificação conforme estabelecido no quadro 15. Este registro é importante pois caso o mesmo não seja feito, todo o controle sobre o software ficará prejudicado, ou seja, se alguma destas alterações não for registrada, o software poderá no futuro não ser mais legível pelos documentos do mesmo, o que trará sérios problemas para a equipe de manutenção.

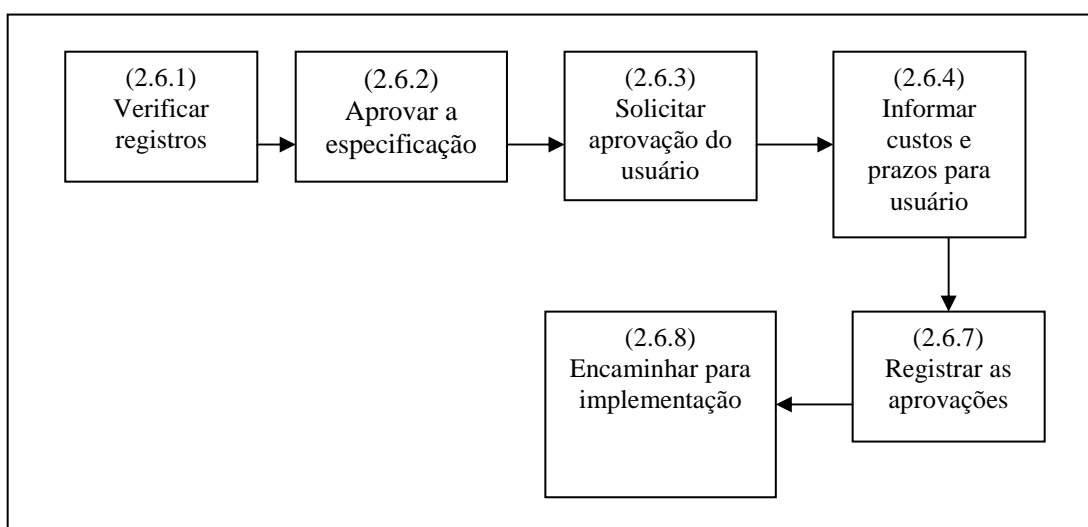
**Quadro 15 – Tarefas de Registro de Alterações da Modificação**



## 4.2.5 ACEITAÇÃO E HOMOLOGAÇÃO DA SOLUÇÃO

Esta é a última etapa da análise do problema e modificação. Nesta etapa como demonstra o quadro 16, as tarefas visam o consentimento da chefia de manutenção com o projeto especificado, ou seja, após todo o processo de análise é necessário que o mesmo seja aprovado pela chefia e também em seguida comunicar a solução para o usuário e verificar se o mesmo concorda quando cabível com a solução. Poderão ainda nesta etapa ser informados os custos e prazos para a entrega da solicitação.

**Quadro 16 – Tarefas da Aceitação e Homologação da Solução**

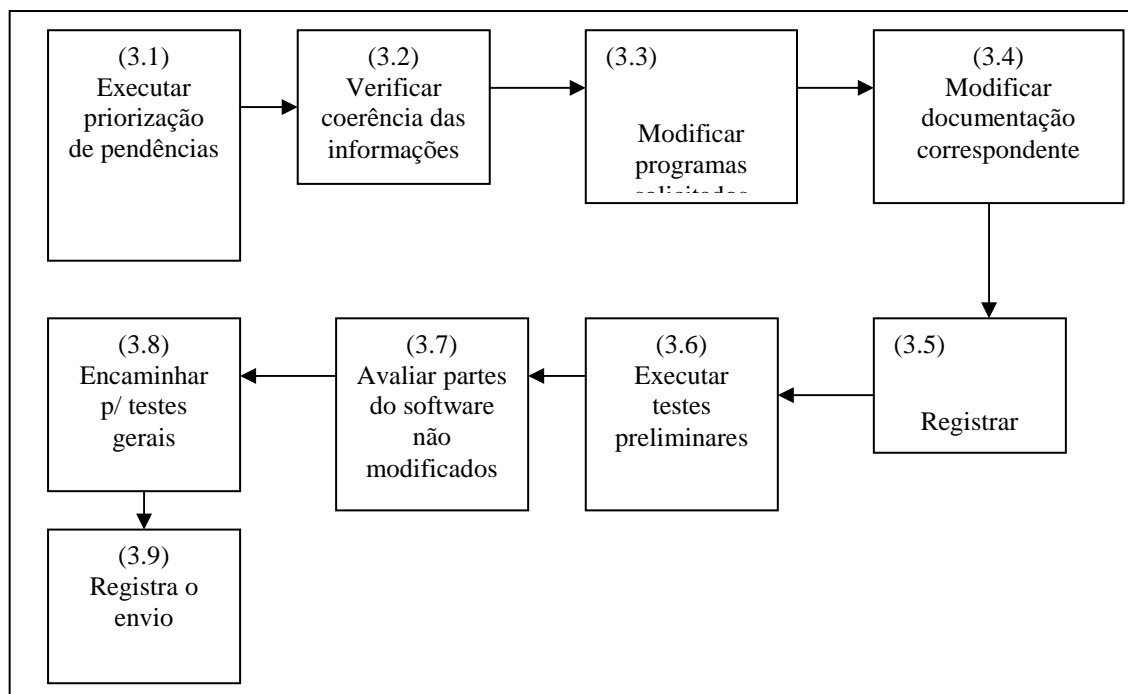


## 4.3 IMPLANTAÇÃO DA SOLUÇÃO

A implantação da solução é o plano do processo de manutenção que vai efetivamente fazer a modificação, alteração e inclusão das rotinas solicitadas pelo usuário. Como já dito anteriormente esta implementação seguirá a ordem de prioridade estipulada e também seguirá com máxima atenção para o projeto especificado na análise.

Nesta implementação como é visto no quadro 17, acontece ainda um teste preliminar do software. Este teste não é aquele que liberara o sistema para utilização, mas apenas fará com que a versão encaminhada para os testes finais seja uma versão que tenha uma boa coerência, o que facilitará muito a equipe ou responsável pelos testes.

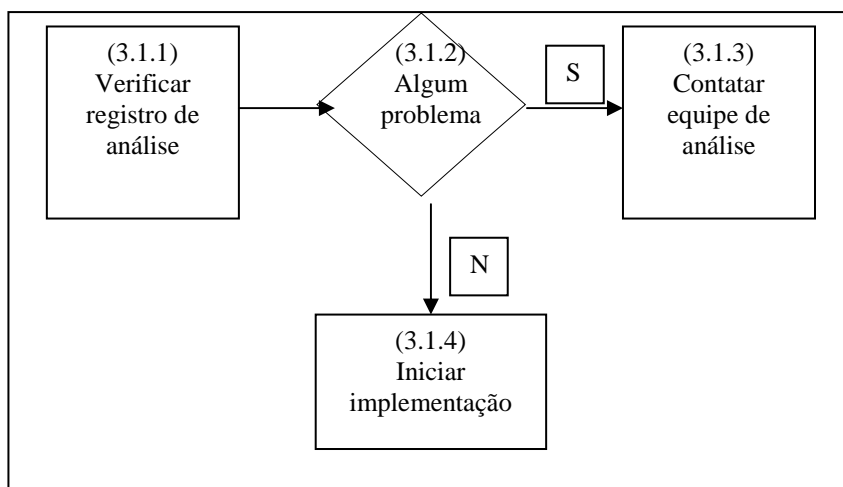
Quadro 17 – Plano de Implementação da Solução



#### 4.3.1 VERIFICAR COERÊNCIA DAS INFORMAÇÕES

O responsável pela manutenção antes de iniciar as alterações deverá proceder uma análise superficial sobre a análise de projeto, conforme quadro 18, para que ele possa, se houver, detectar alguma falha no projeto, pois, dependendo do tamanho da equipe de manutenção, problemas e incoerências poderão ocorrer e quanto mais cedo estes problemas forem solucionados mais rápida e mais eficaz é a solução.

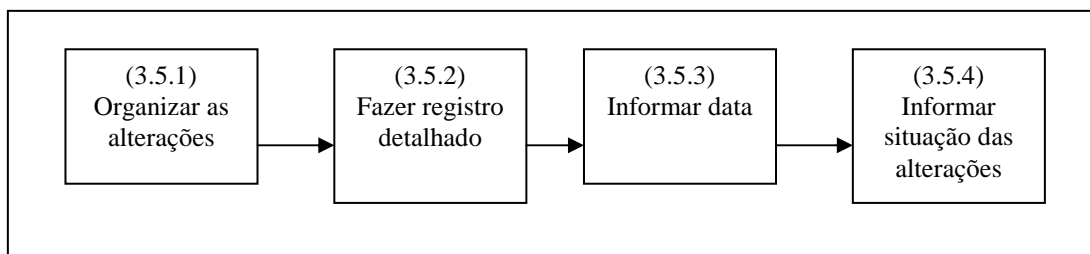
Quadro 18 – Tarefas da Verificação de Coerência da Análise



### 4.3.2 REGISTRAR AS ALTERAÇÕES

Após todas as alterações serem feitas é necessário que o responsável pela manutenção registre (Quadro 19), as alterações que realmente vieram a ser feitas.

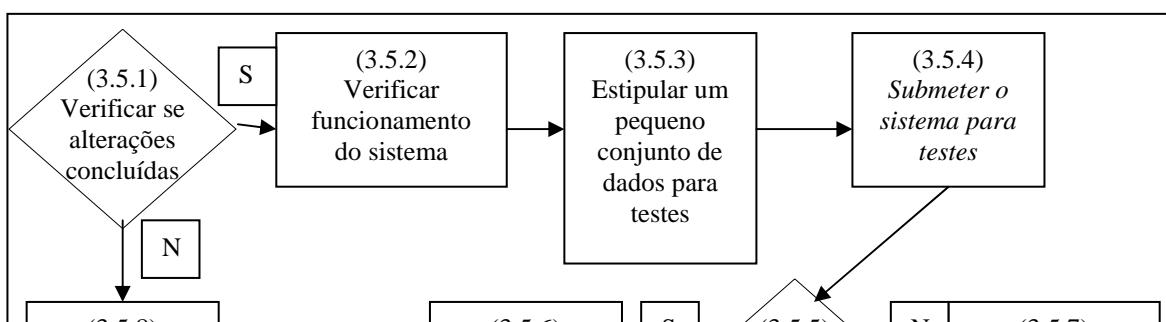
Quadro 19 – Tarefas de Registro de Alterações



### 4.3.3 TESTES PRELIMINARES

Nos testes preliminares (Quadro 20), o responsável pela manutenção executa pequenas seqüências de testes para verificar se a princípio o sistema funciona normalmente. Para isto ele utiliza de um pequeno número de dados para o teste, o que não significa que caso o sistema seja aprovado nestes testes não possa ser reprovado nos testes gerais.

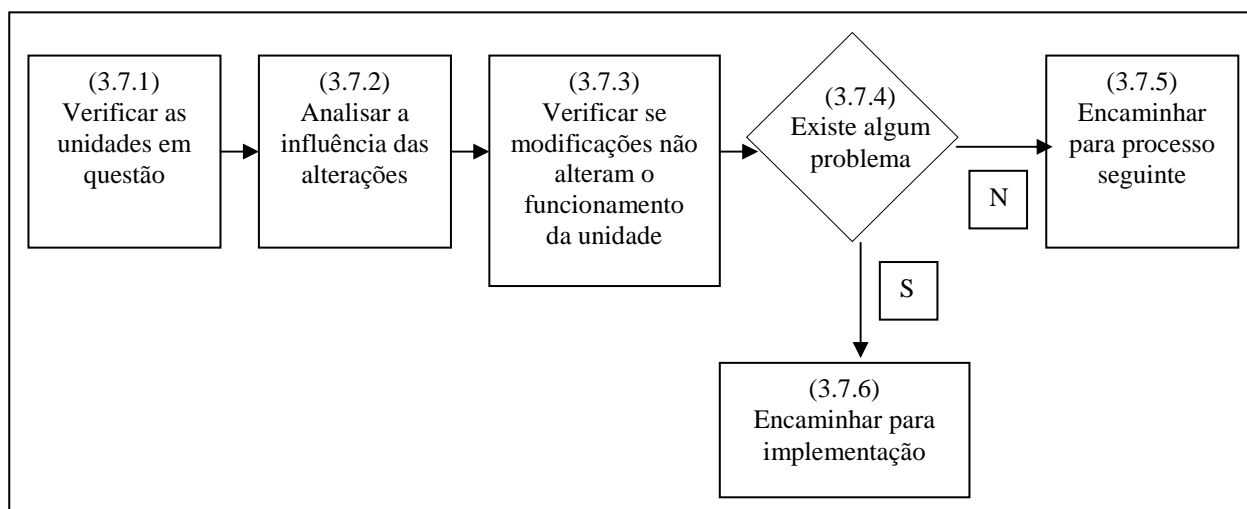
Quadro 20 – Tarefas dos Testes Preliminares



#### 4.3.4 AVALIAR PARTES DO SOFTWARE NÃO MODIFICADOS

Muito freqüentemente quando se é feito uma modificação ou correção no software, esta mesma modificação poderá trazer alguns problemas para outras partes do sistema. Para que isto seja sanado antes mesmo do usuário perceber é necessário que o responsável pela manutenção do sistema faça uma avaliação das partes do software que não foram alteradas (Quadro21).

Quadro 21 – Tarefas da Avaliação das partes do software não modificadas



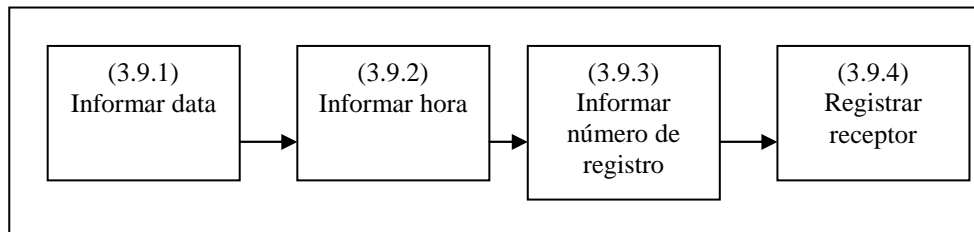
#### 4.3.5 REGISTRO DE ENVIO PARA TESTES FINAIS

É muito importante que seja registrado quando o sistema saiu da fase de implementação e passou para os testes finais, pois, além do controle normal da equipe de



manutenção o usuário pode ser informado para que ele saiba se o prazo de entrega continuara rigorosamente em dia. O quadro 22 mostra quais as tarefas desta etapa.

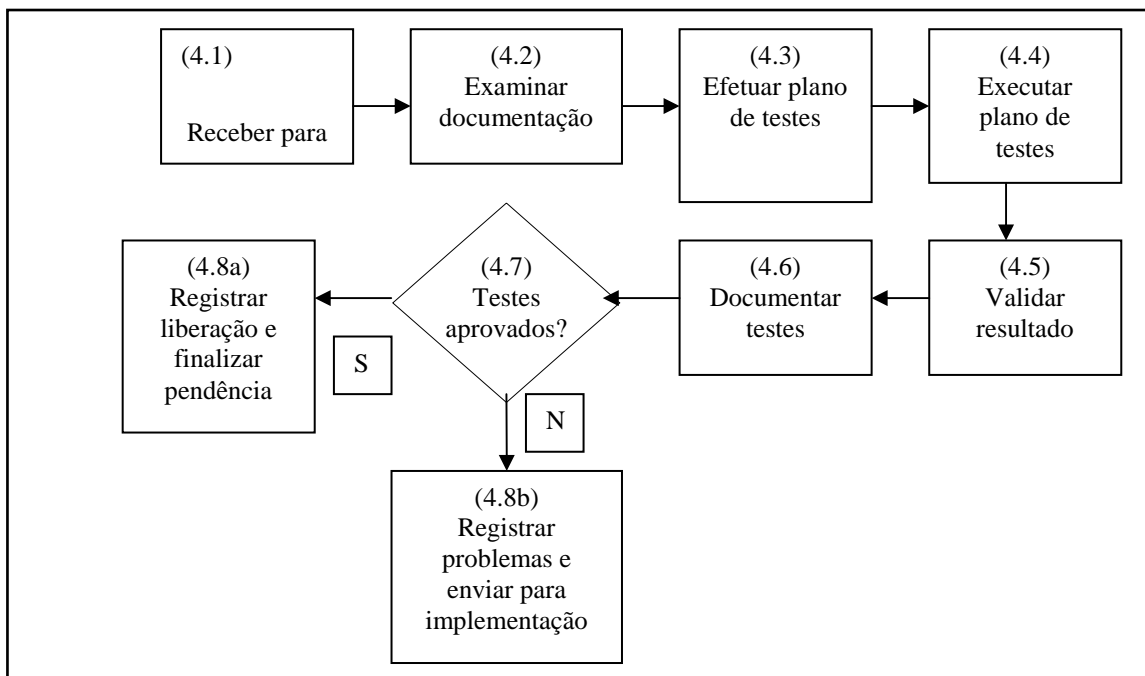
**Quadro 22 – Tarefas do Registro de Envio para Testes Finais**



## 4.4 REVISÃO / TESTES E ACEITAÇÃO

Conforme mostra o quadro 23, a tarefa de revisão e testes é a última em que se mexe diretamente com o sistema alterado. O responsável pela manutenção deverá no final desta fase garantir acima de nenhuma suspeita que o software está de acordo com o projeto e que não tem nenhum problema com o que diz respeito à modificação implantada.

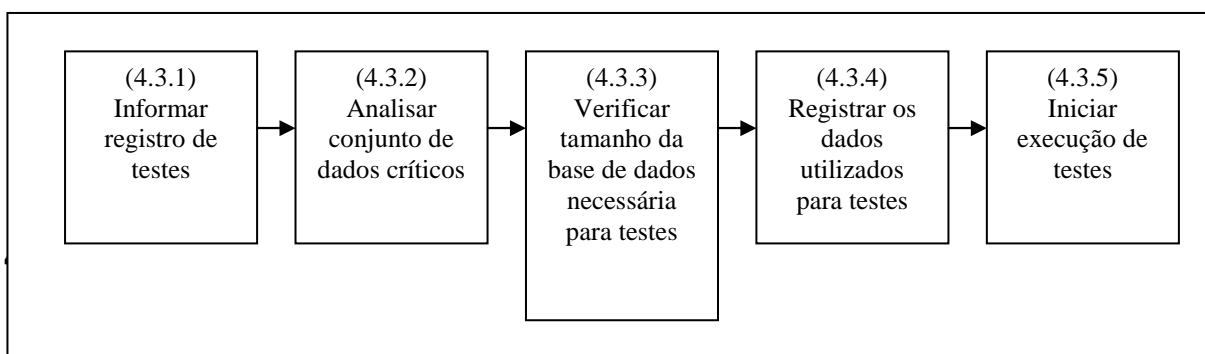
**Quadro 23 – Tarefas de Revisão, Testes e Aceitação**



### 4.4.1 EFETUAR PLANO DE TESTES

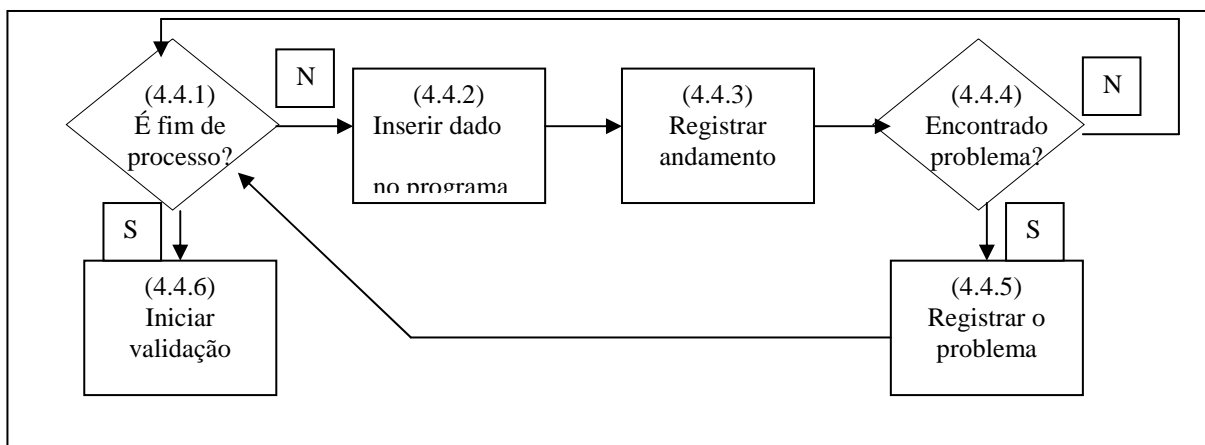
Para que o software seja validado é necessário que os dados utilizados para testá-lo possam suprir os mais diversos problemas que possam ocorrer. Para isto o responsável pela manutenção, segundo o quadro 24, deverá efetuar um plano de testes, onde serão contemplados todos os tipos de adversidade imagináveis.

**Quadro 24 – Tarefas do Plano de Testes**



tarefa de testes do sistema como demonstra o quadro 25. É apreciável que estes testes sejam registrados para futuras consultas pela equipe de desenvolvimento, como também para demonstrar que o sistema realmente funciona.

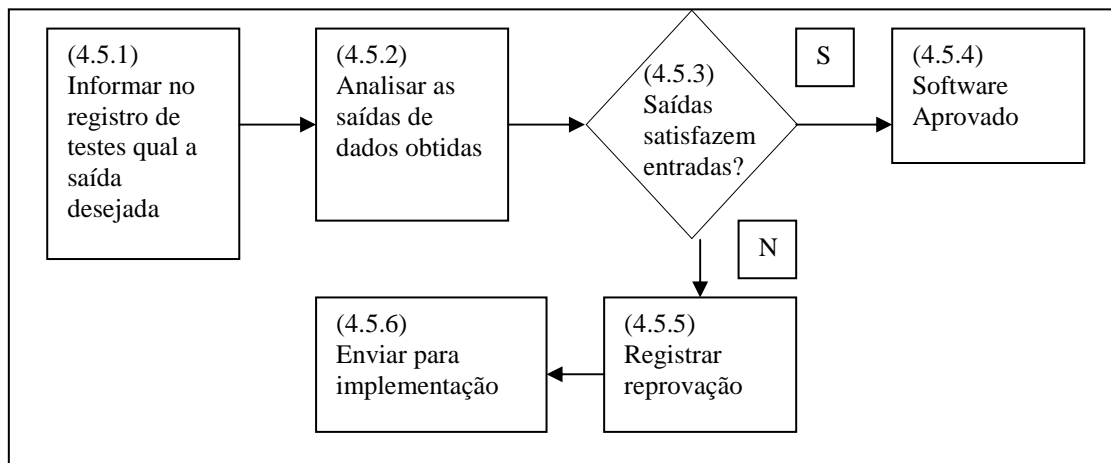
**Quadro 25 – Tarefas da Execução dos testes**



## 4.4.3 VALIDAR OS RESULTADOS

O responsável pela manutenção deverá criar um registro para validar os dados que foram inseridos para teste, nesta validação o mantenedor precisará, conforme quadro 26, criar um registro das saídas desejadas após a inserção dos dados no sistema, para que então a partir disto realmente comprovar que o sistema executa a função para qual foi projetada.

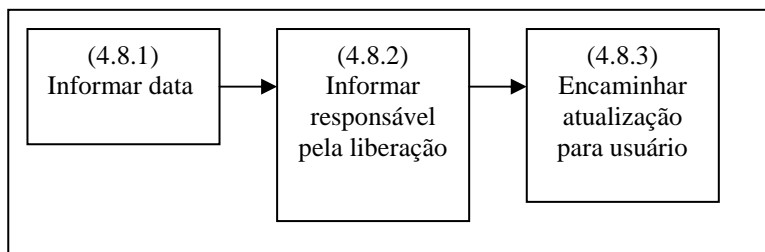
Quadro 26 – Tarefas da Validação dos testes



#### 4.4.4 REGISTRAR LIBERAÇÃO

Após a aprovação do software o responsável deverá registrar a saída do sistema do processo de manutenção como mostra o quadro 27. Por conseguinte o software poderá ser liberado para os usuários.

Quadro 27 – Tarefas da Liberação do Software

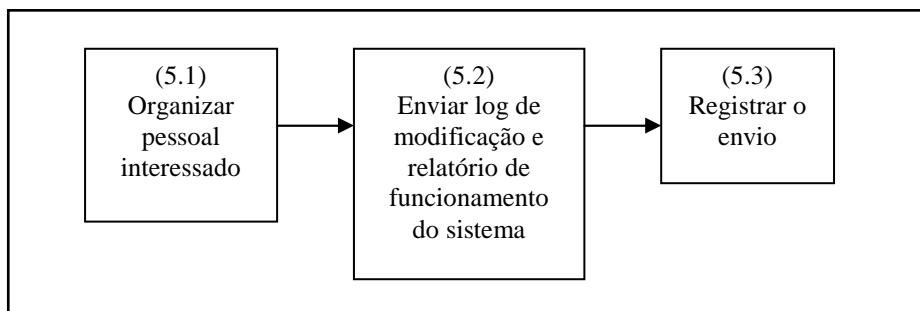


#### 4.5 QUALIFICAÇÃO DA EQUIPE

Em muitas empresas as equipes de manutenção podem estar sub-divididas dependendo de quão grande é esta necessidade. Então é fundamental que todas as pessoas que possam vir a modificar ou até mesmo demonstrar o funcionamento do sistema, conheçam todas as modificações e consertos que o mesmo sofreu. Pensando nesta finalidade desenvolveu-se a qualificação da equipe, cuja a tarefa é muito simples, apenas manter informada qualquer pessoa que esteja direta ou indiretamente ligada ao software. Isto se dá pelo responsável pela manutenção após o termino do processo, enviar ou disponibilizar para os interessados registros de como procedeu a manutenção.

Este ciclo é demonstrado no quadro 28.

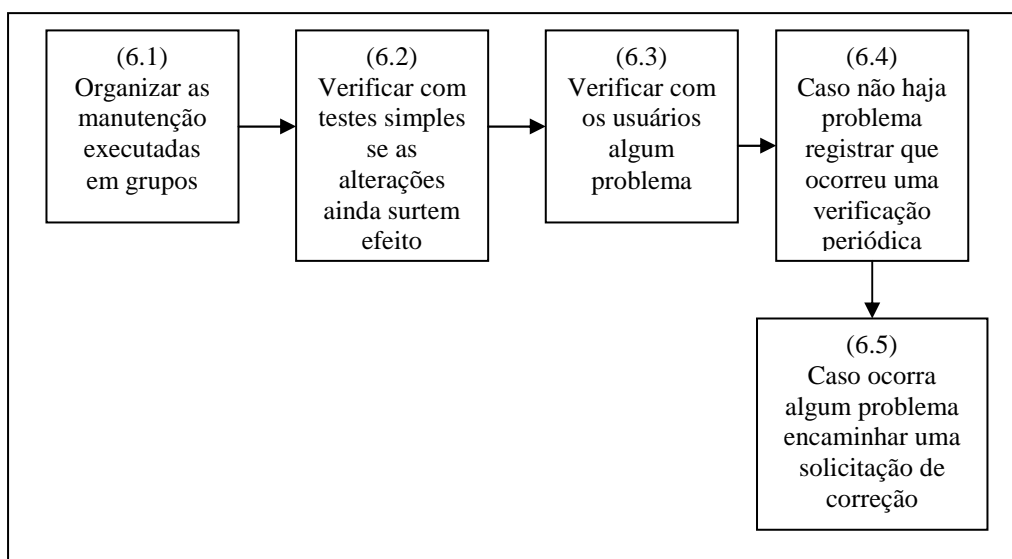
**Quadro 28 – Plano de qualificação de equipe**



## 4.6 VERIFICAÇÃO PERIÓDICA

É também recomendado que o software que sofre muita manutenção tenha de tempos em tempo uma análise se todas as modificações não estejam num geral causando algum problema. Para isto esta etapa visa periodicamente realizar, além de uma consulta aos usuários, uma determinação de que o sistema funciona a contento. Possíveis manutenções poderão vir a ser encaminhadas após esta análise, o que representa uma melhor organização da equipe para realizar a manutenção pois a mesma não será solicitada por um usuário com pressa e sim por um desenvolvedor que está analisando o sistema. O quadro 29 demonstra este processo.

**Quadro 29 – Tarefas da Verificação Periódica**



## 4.7 FORMULÁRIOS PROPOSTOS

Após todo este estudo do processo de manutenção, alguns formulários serão sugeridos para o registro de todo o processo e cada etapa em que a equipe de manutenção finalizou. Todos estes formulários serão de grande ajuda no momento que a equipe necessitar de *feedback* de informações para resolver algum problema.

Os formulários demonstrados nos quadros 29 e 30 tem por objetivo registrar e receber as solicitações de correção e modificação respectivamente, a sua função de informar para a equipe de manutenção qual o problema que está ocorrendo e qual a modificação que o usuário necessitar que seja implantada. Estes formulários atendem as tarefas (1.1) e (1.2).

**Quadro 30 – Formulário de Solicitação de Correção**

Solicitação de Correção	
Numero:	<u>Número para ordenação de arquivo</u>
Data:	<u>Data da Ocorrência</u>
Solicitante:	<u>Pessoa que comunicou o problema</u>
Sistema:	<u>Sistema com Defeito</u>
Rotina:	<u>Qual a Rotina em que o problema foi detectado</u>
Descrição do Problema:	<u>Relatar detalhadamente o problema que está ocorrendo</u>
	_____
	_____
	_____
Efeito do Problema:	<u>Relatar quais os efeitos após ocorrer o problema</u>
	_____
	_____
	_____
Urgência:	<u>Especificar qual a urgência da correção</u>
Gravidade:	<u>Especificar qual grave é o problema</u>
Data Solicitada:	<u>Informar a Data Desejada para Resposta</u>

**Quadro 31 – Formulário de Solicitação de Modificação**

Solicitação de Modificação	
Numero:	<u>Número para ordenação de arquivo</u>
Data:	<u>Data da Ocorrência</u>
Solicitante:	<u>Pessoa que comunicou o problema</u>
Sistema:	<u>Sistema com Defeito</u>
Responsável:	<u>Informar que da empresa autorizou o pedido</u>
Solicitação	<u>Relatar detalhadamente qual a alteração desejada</u>
	_____
	_____
	_____
Justificativa	<u>Justificar o pedido</u>
	_____
	_____
Prioridade:	<u>Especificar qual a prioridade que se deseja</u>

O formulário, demonstrado no quadro 32, é utilizado para fazer a análise da solicitação do problema, ou seja, neste formulário são descritos dados referente a solicitação de correção anteriormente mencionada, para então registrar se a mesma é realmente autêntica e se o sistema realmente necessita de uma correção. Este formulários juntamente com o formulário da figura 33 atendem a etapa 2.

**Quadro 32 – Formulário de Análise da Solicitação de Problema**

Análise da Solicitação de Problema	
Número:	<u>Informar o número de arquivo</u>
Data:	<u>Data da análise</u>
Responsável:	<u>Responsável pela análise</u>
Veracidade das Informações:	<u>Informar se as informações estão em ordem</u>
Verificado o problema:	<u>Verificar se o problema realmente acontece</u>
Observações:	<u>Descrever informações que achar necessário</u>
Tipo de Problema:	<u>informar tipo de problema</u>
Solicitação Aprovada:	<u>informar se Aprovada ou não</u>
Gravidade:	<u>Informar a gravidade</u>
Encaminhada Para:	<u>Pessoa que receber a análise</u>
Previsão de Resposta:	<u>Data para resposta</u>

Já o quadro 33, demonstra a análise da solicitação de modificação, que é utilizada para registrar dados referente a solicitação de modificação e se a mesma tem condições operacionais de ser desenvolvida.

**Quadro 33 – Formulário de Análise de Solicitação de Modificação**

Análise da Solicitação de Modificação	
Número:	<u>Informar o número de arquivo</u>
Data:	<u>Data da análise</u>
Responsável:	<u>Responsável pela análise</u>
Veracidade das Informações:	<u>Informar se as informações estão em ordem</u>
Viabilidade :	<u>Verificar se a modificação é viável</u>
Observações:	<u>Descrever informações que achar necessário</u>
Tipo de modificação:	<u>informar tipo de modificação</u>
Solicitação Aprovada:	<u>informar se Aprovada ou não</u>
Prioridade:	<u>Informar a prioridade</u>
Encaminhada Para:	<u>Pessoa que receber a análise</u>
Previsão de Resposta:	<u>Data para resposta</u>

No quadro 34 é demonstrado o formulário de registro de manutenções, que tem por objetivo registrar todos os acontecimentos relacionados com a manutenção de um software, seja esta manutenção corretiva, adaptativa ou aperfeiçoante. Este formulário funciona como um histórico do sistema em questão, pois todas as ocorrências relacionadas ao sistema estarão descritas neste formulário. Este formulário atende a etapa 3.5.

**Quadro 34 – Formulário de Registro de Manutenção**

Registro de Manutenção
Número: <u>Número de Controle</u>
Número da Solicitação: <u>Número da Solicitação</u>
Responsável: <u>Responsável pela Manutenção</u>
Data e hora de início: <u>Data e Hora de Início</u>
Descrição das tarefas: <u>Descrever detalhadamente componente que sofreu alteração , qual o tipo de alteração e a sua descrição</u>
Encaminhado para teste: <u>Data de Envio para teste</u>
Para: <u>Pessoa que realizará os testes</u>
Avaliação dos Testes: <u>Descrever Qual foi a avaliação dos testes executados</u>
Manutenção Liberada em : <u>Data da Liberação</u>
Por: <u>Nome do responsável pela liberação</u>

## 5 SOFTWARE DE APOIO A MANUTEÇÃO DE SISTEMAS

*O software a seguir foi desenvolvido segundo a proposta de processo de manutenção demonstrada no capítulo anterior. Para a especificação do software utilizou-se a ferramenta Power Designer 6.1. Para o desenvolvimento do software utilizou-se o ambiente de programação Delphi 3.0, com a base de dados baseado nos arquivos Paradox.*

*O software pretende contemplar os processos de solicitação de correção e modificação, análise da solicitação e registro, além de relatórios para arquivamento.*

### 5.1 DIAGRAMA DE CONTEXTO E DFD

*O diagrama de contexto do software está representado na figura 7. Na figura 8 é representado o DFD de nível 1, demonstrando os processos básicos do software. Utilizou-se aqui a notação de YOURDON disponível pela ferramenta CASE POWER DESIGNER.*

Figura 7 – Diagrama de Contexto

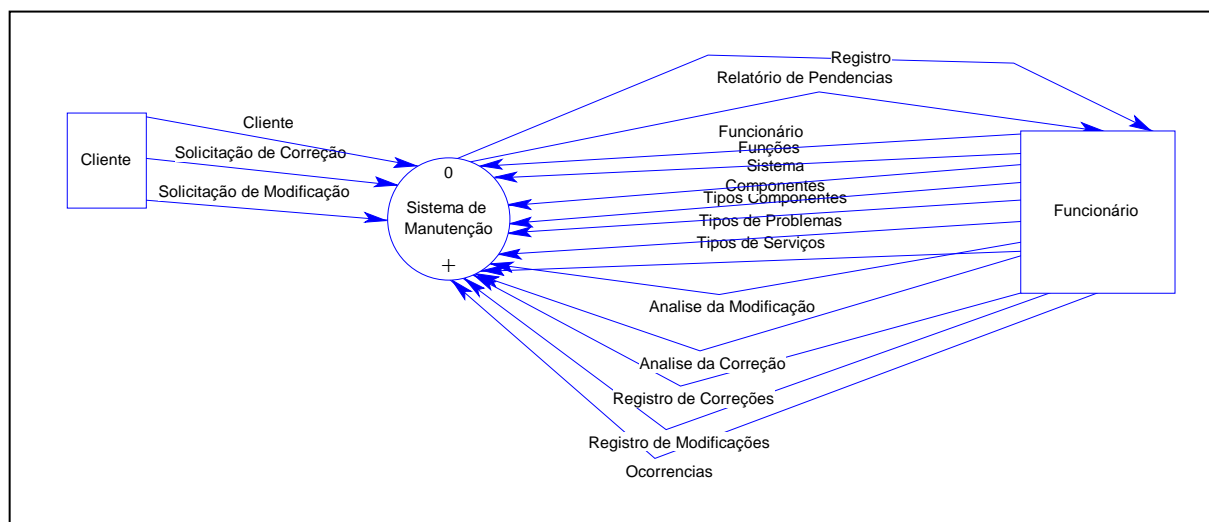
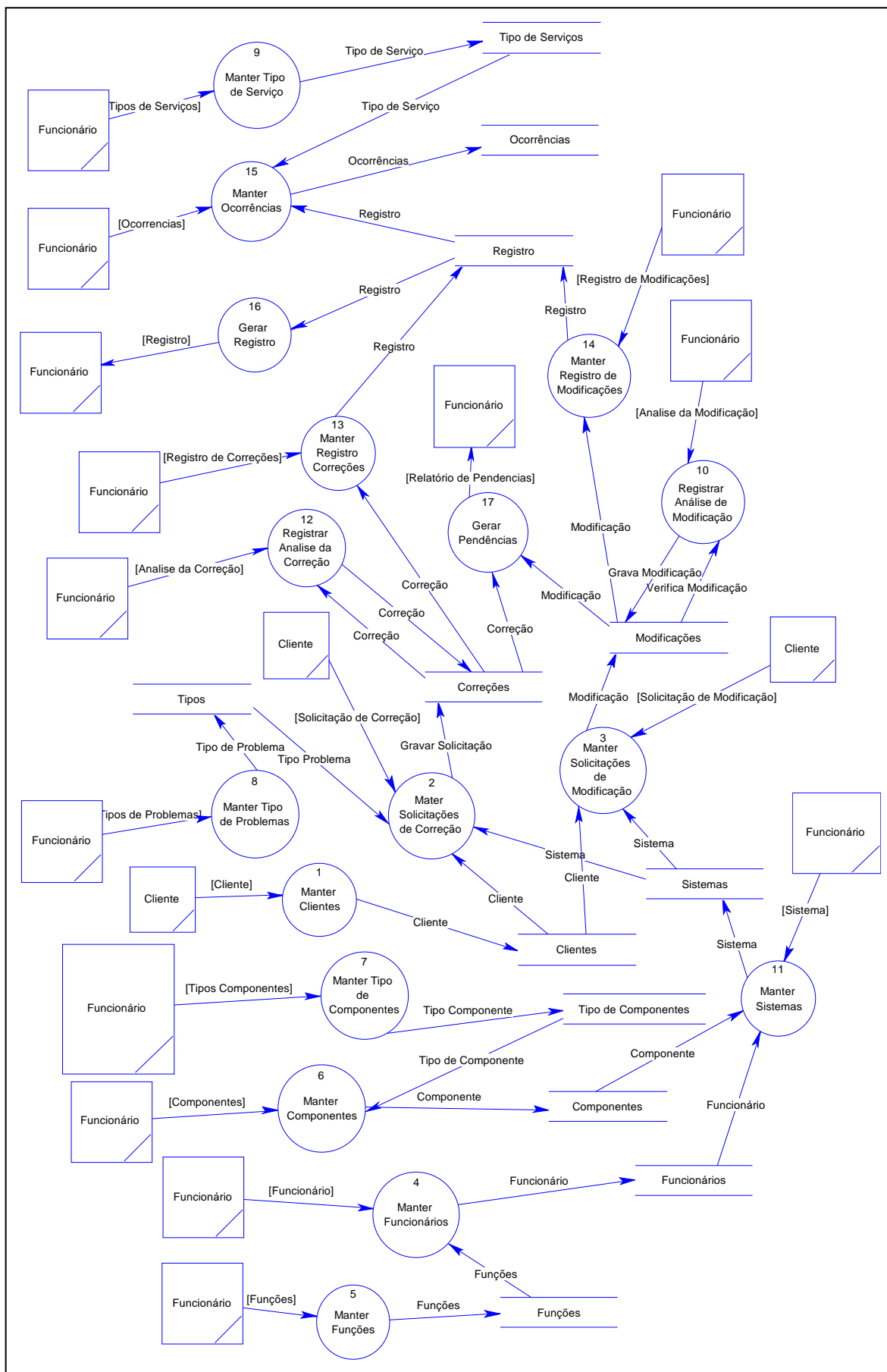




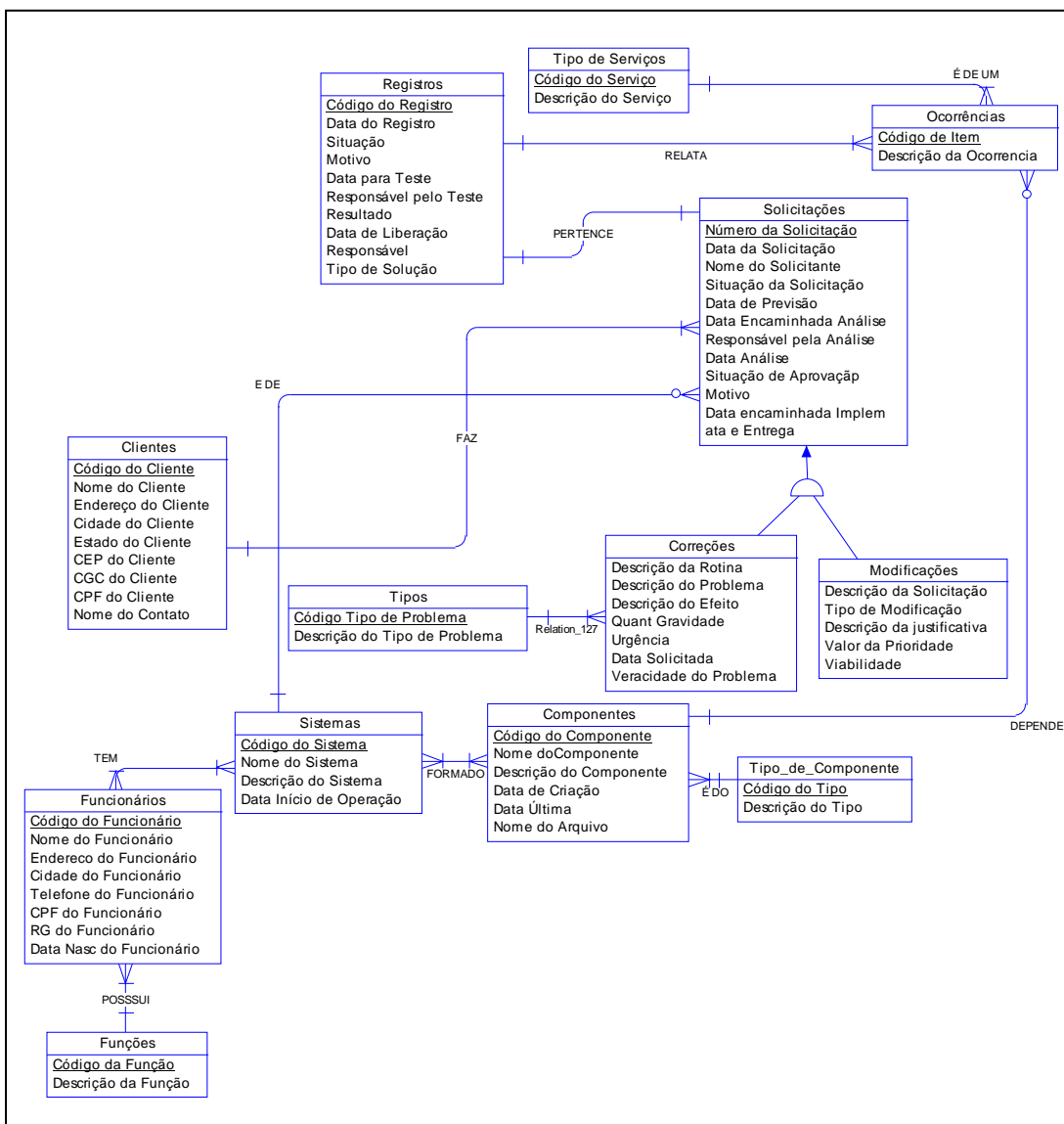
Figura 8 - DFD



## 5.2 MODELO ENTIDADE RELACIONAMENTO

O modelo entidade relacionamento (MER) foi primeiramente desenvolvido a nível lógico como demonstra a figura 9. A notação utilizada foi a de “Martin”. Nota-se que a obrigatoriedade é ilustrada com um “risco” vertical no relacionamento da entidade e para demonstrar a chave primária é sublinhado o atributo.

Figura 9 – Modelo Entidade Relacionamento - Lógico



## 5.3 DICIONÁRIO DE DADOS

Na Figura 10, é demonstrado o dicionário de dados do software. Cada entidade apresenta a lista de seus atributos com o nome descritivo, nome código, tipo do atributo, definição de chave primária(I) e obrigatoriedade (M). Este dicionário foi gerado pela ferramenta CASE.

Figura 10 – Dicionário de Dados

<b>Clientes</b>				
<b>Attribute List</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>I</b>	<b>M</b>
Código do Cliente	CDCLIENTE	I	Yes	Yes
Nome do Cliente	NMCLIENTE	A50	No	Yes
Endereço do Cliente	EDCLIENTE	A40	No	Yes
Cidade do Cliente	CIDCLIENTE	A30	No	Yes
Estado do Cliente	ESTCLIENTE	A30	No	Yes
CEP do Cliente	NRCEPCLIENTE	A8	No	Yes
CGC do Cliente	NRCGCCLIENTE	A18	No	No
CPF do Cliente	NRCPFCLIENTE	A14	No	No
Nome do Contato	NMCONTATO	A30	No	Yes

Figura 10 – Dicionário de Dados - Continuação

<b>Funcionários</b>				
<b>Attribute List</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>I</b>	<b>M</b>
Código do Funcionário	CDFUNCIONARIO	I	Yes	Yes
Nome do Funcionário	NMFUNCIONARIO	A50	No	Yes
Endereco do Funcionário	EDFUNCIONARIO	A50	No	Yes
Cidade do Funcionário	CIDFUNCIONARIO	A30	No	Yes
Telefone do Funcionário	NRTELFUNCIONARIO	A18	No	Yes
CPF do Funcionário	NRCPPFUNCIONARIO	A14	No	Yes
RG do Funcionário	NRRGFUNCIONARIO	A10	No	Yes
Data Nasc do Funcionário	DTNASCFUNCIONARIO	D	No	Yes
<b>Funções</b>				
<b>Attribute List</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>I</b>	<b>M</b>
Código da Função	CDFUNCAO	I	Yes	Yes

Figura 10 – Dicionário de Dados - Continuação

<b>Registros</b>				
<b>Attribute List</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>I</b>	<b>M</b>
Código do Registro	CDREGISTRO	I	Yes	Yes
Data do Registro	DTREGISTRO	D	No	Yes
Situação	SITUACAO	A1	No	Yes
Motivo	MOTIVO	A100	No	Yes
Data para Teste	DTTESTE	D	No	Yes
Responsável pelo Teste	RESPTESTE	A30	No	Yes
Resultado	RESULTADO	BL	No	Yes
Data de Liberação	DTLIBERACAO	D	No	Yes
Responsável	RESP	A30	No	Yes
Tipo de Solução	TPSOLUCAO	A1	No	Yes
<b>Sistemas</b>				
<b>Attribute List</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>I</b>	<b>M</b>
Código do Sistema	CDSISTEMA	I	Yes	Yes
Nome do Sistema	NMSISTEMA	A50	No	Yes
Descrição do Sistema	DSSISTEMA	A50	No	Yes
Data Início de Operação	DTINICIOOPER	D	No	Yes

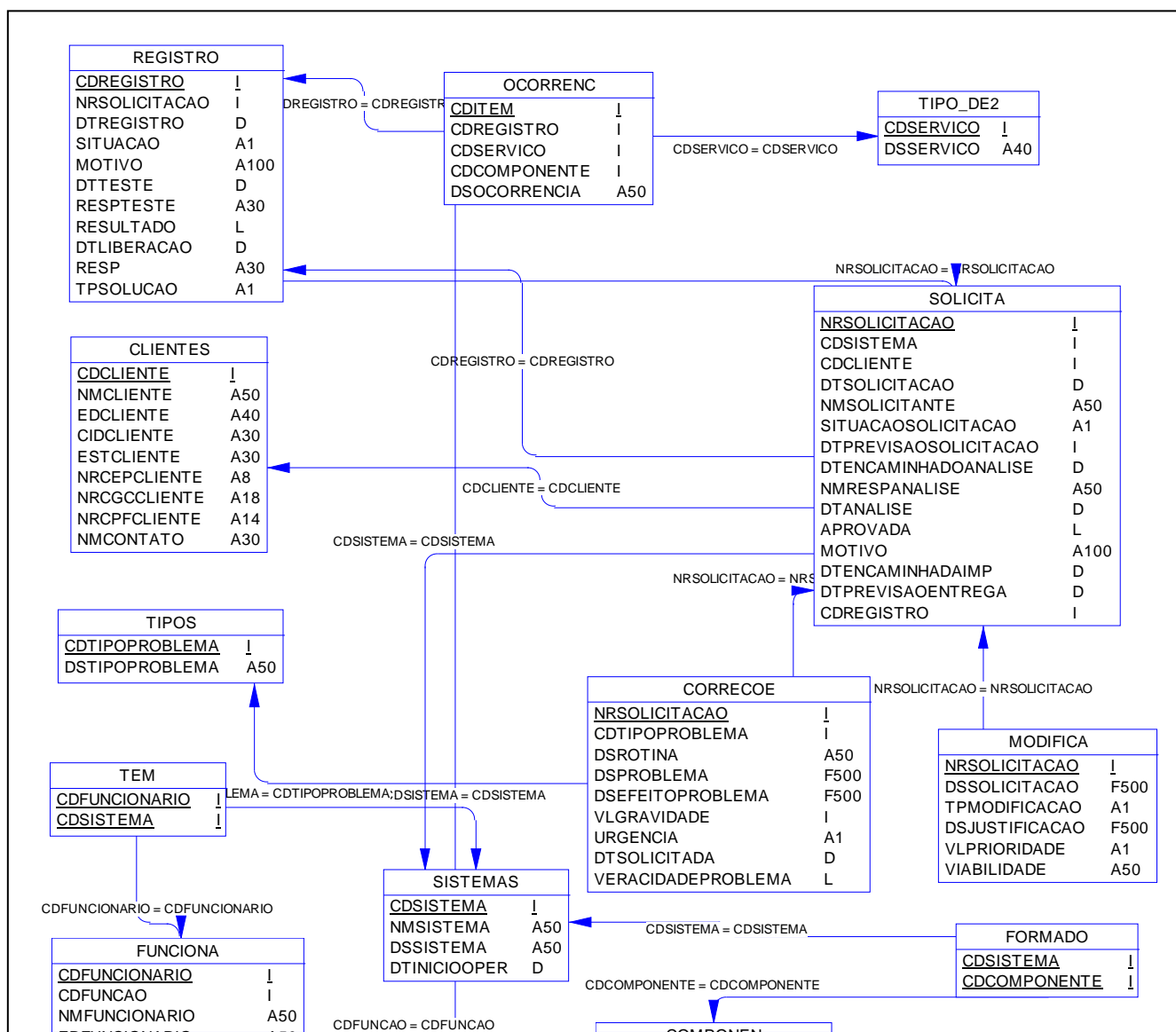
Figura 10 – Dicionário de Dados - Continuação

<b>Tipo de Serviços</b>				
<b>Attribute List</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>I</b>	<b>M</b>
Código do Serviço	CDSERVICO	I	Yes	Yes
Descrição do Serviço	DSSERVICO	A40	No	Yes
<b>Tipo_de_Componente</b>				
<b>Attribute List</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>I</b>	<b>M</b>
Código do Tipo	CDTIPO	I	Yes	Yes
Descrição do Tipo	DSTIPO	A35	No	Yes
<b>Tipos</b>				
<b>Attribute List</b>				
<b>Name</b>	<b>Code</b>	<b>Type</b>	<b>I</b>	<b>M</b>
Código Tipo de Problema	CDTIPOPROBLEMA	I	Yes	Yes
Descrição do Tipo de Problema	DSTIPOPROBLEMA	A50	No	Yes

No modelo físico como é demonstrado na figura 11, são criados alguns campos dentro das entidades que são o relacionamento entre as mesmas e além disto são criados algumas novas tabelas como a de com o nome de “Formado”, o que significa que um sistema é formado por um ou vários componentes e que um componente poderá ser utilizado em um os vários sistemas. Para isto cria-se uma tabela com o código do sistema e o código do componente.

A seguir encontra-se uma tabela com o nome “tem”, que significa que um funcionário trabalha em um ou vários sistemas e também que um sistema poderá ter um ou vários funcionários na equipe de trabalho. Para isto também foi criada uma tabela com o código do funcionário e o código do sistema.

Figura 11 – Modelo Entidade Relacionamento - Físico



## **5.4 IMPLEMENTAÇÃO DO SOFTWARE**

*Como já dito anteriormente o software foi desenvolvido no ambiente de programação Delphi 3.0. O sistema é composto de 4 itens no menu principal: cadastro, pendências, registros e relatórios.*

*No menu cadastro são feitos os cadastros de sistemas, funcionários, funções, componentes, tipos de componentes, tipos de problemas, tipos de serviços e clientes. Estes cadastros serão utilizados mais tarde quando ocorrer uma solicitação de correção ou modificação, a análise das solicitações o registro das manutenções e nos relatórios.*

*O segundo menu comporta a opção de pendências subdividindo-se correções e modificações, as quais são utilizadas quando um cliente entra em contato com a equipe de manutenção para comunicar um problema ou fazer um pedido de modificação do sistema. Dentro de cada opção desta área está também a análise das solicitações, ou seja, assim que o cliente faz a solicitação ela já poderá ser analisada pela equipe e os dados desta análise também serão guardados nesta opção.*

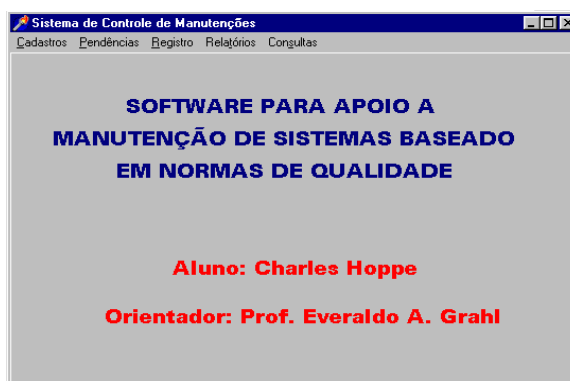
*O terceiro menu abriga a opção de registro da correção e modificação. Nesta opção é registrado tudo que ocorreu com o software durante uma manutenção, ou seja, é aqui que todas as alterações, exclusões, e inclusões no sistemas são descritas.*



*A quarta e última opção é a de relatórios. Nesta opção é possível tirar alguns relatórios para os funcionários como relatório de pendências para verificar quais as solicitações que ainda não foram atendidas e de pendências por data de entrega para saber quais as pendências que estão com o prazo mais próximos de esgotarem.*

## **5.5 PRINCIPAIS TELAS DO SOFTWARE**

Figura 12 – Tela de Abertura



### **5.5.1 TELAS DE CADASTRO**

*A seguir são demonstradas algumas das telas de cadastros do sistema.*

Na figura 13 demonstra-se o cadastro de sistemas que a empresa desenvolve. No início estão os dados gerais, logo abaixo os funcionários que podem ser um ou vários e por fim os componentes do sistema.

Figura 13 – Tela de Manutenção de Sistemas

CDFUNCONARIO	Nome
1	Charles Hoppe
3	Joao Alves

CDCOMPONENTE	Descricao
4	Frente de Caixa
5	Relatorio de Vendas

A figura 14 mostra o cadastro de componentes do sistema onde são cadastrados todos os componentes que a equipe desenvolveu, neste cadastro todos os dados referente a criação, nome do arquivo e descrição são arquivados.

Figura 14 – Tela de Manutenção de Componentes

Código	4	
Tipo	2	Biblioteca
Nome	Frente de Caixa	
Descrição	Executa as operações de frente de caixa	
Data Criação	14/12/1999	
Data de Alteração	14/12/1999	
Arquivo	FrenteCaix	

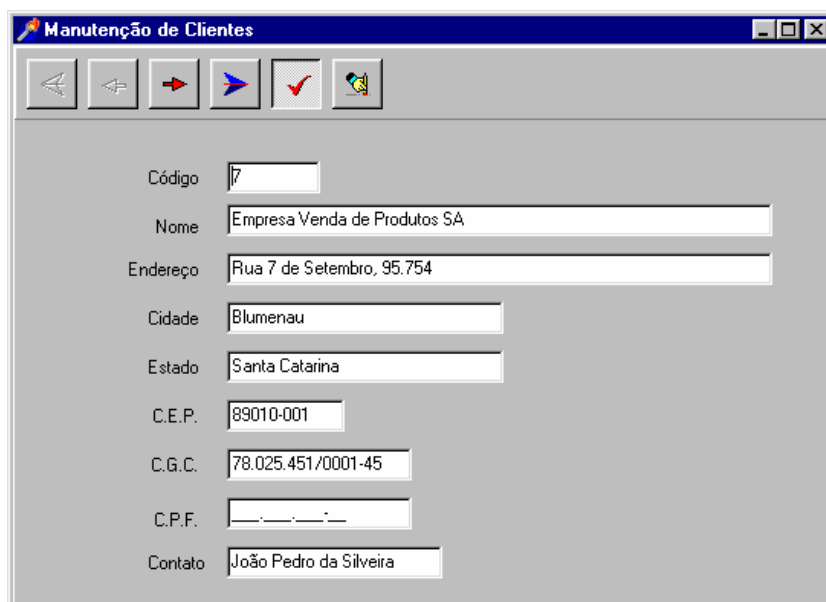
A figura 15 demonstra a tela de manutenção de funcionários que cadastra os funcionários da equipe com os dados pessoais, este funcionários aqui cadastrados serão utilizados posteriormente no cadastro de sistemas.

Figura 15 – Tela de Manutenção de Funcionários

Código	6	
Função	1	Administrador
Nome	Charles Hoppe	
Endereço	Rua Antonio da Veiga, 999	
Cidade	Blumenau	
Telefone	(02147) 326-6666	
C.P.F.	956.157.515-45	
R.G.	3C/645.784.454	
Data Nascimento	31/12/1977	

Na figura 16 é demonstrado o cadastro de clientes, nela são cadastrados todos os dados pessoais e jurídicos dos clientes, estes clientes aqui cadastrados serão utilizados nas solicitações de correção e modificação.

Figura 16 – Tela de Manutenção de Clientes



A imagem mostra uma janela de software intitulada "Manutenção de Clientes". No topo, há uma barra de ferramentas com ícones para navegação e ações. Abaixo, há um formulário com os seguintes campos:

Código	7
Nome	Empresa Venda de Produtos SA
Endereço	Rua 7 de Setembro, 95.754
Cidade	Blumenau
Estado	Santa Catarina
C.E.P.	89010-001
C.G.C.	78.025.451/0001-45
C.P.F.	_____
Contato	João Pedro da Silveira

Os outros cadastros existentes são os de tipos de componentes, tipos de serviços, funções e tipos de problemas.

### 5.5.2 TELAS DE PENDÊNCIAS E ANÁLISE

As telas a seguir são as de manutenção de solicitações de correções e modificações, bem como as de análise das solicitações

As telas de solicitações de correções (figura 17) e solicitações de modificações (figura 18) fazem o cadastro das solicitações tanto de correção quanto de modificação, cada qual com as suas características próprias, ou seja, seguindo aos processo de manutenção proposto. Estas opções contemplam os processos descritos nos quadros 03,04 e 05 do processo proposto.

Figura 17 – Tela de Pendência de Correção

**Manutenção de Pendências de Correção**

Número: 9      Data: 20/12/1999

Cliente: 6      Empresa Venda de Produtos

Solicitante: José da Silva

Sistema: 7      Sistema FiscalPrint      Rotina: Frente de Caixa

Tipo de Problema: 2      Erro de Leitura

Situação da Solicitação:  
 Recebida  
 Em Análise  
 Aprovada  
 Reprovada

Descrição do Problema:  
 Exibe mensagem de erro quando se tenta digitar o código do produto.  
 MENSAGEM: "Erro de Leitura: CADPROD.DB"

Efeito do Problema:  
 Aborta o programa

Gravidade para o Cliente:  
 Alta  
 Média  
 Baixa

Urgência:  
 Alta  
 Média  
 Baixa

Data Solicitada: 20/12/1999      Data de Resposta: 20/12/1999  
 Data Encaminha para análise: 20/12/1999     

Figura 18 – Tela de Pendências de Modificação

**Manutenção de Pendências de Modificação**

Número: 4      Data: 05/01/2000

Cliente: 6      Empresa Venda de Produtos

Solicitante: Augusto de Sá

Sistema: 7      Sistema FiscalPrint

Tipo de Modificação:  
 Adaptativa  
 Aperfeiçoante

Situação da Solicitação:  
 Recebida  
 Em Análise  
 Aprovada  
 Reprovada

Prioridade:  
 Grande  
 Média  
 Baixa

Descrição da Solicitação:  
 Solicita-se fazer rotina para impressão na impressora fiscal Bematech 45-75

Justificativa da Solicitação:  
 Mudou-se de equipamento e o sistema não funciona adequadamente para o novo equipamento

Data Solicitada: 15/01/2000      Data de Resposta: 15/01/2000  
 Data Encaminha para análise: 05/01/2000

Logo a seguir nas figuras 19 e 20 são demonstradas as telas de análise da solicitação de correção e modificação. Note que as mesmas são chamadas a partir das telas de solicitação, no botão de análise e são elas que registram qual a verificação que o analista fez da solicitação. Estas opções são contempladas pelos quadros 07,08,09 e 10 do processo proposto.

**Figura 19 – Tela de Análise da Pendência de Correção**

**Análise da Pendência de Correção**

Número: 9      Data de Análise: 20/12/1999

Responsável: Charles Hoppe

Veracidade do Problema:

- Problema Detectado
- Problema Não Detectado

Urgência Sugerida:

- Alta
- Média
- Baixa

Resultado da Análise:

- Aprovada
- Não Aprovada

Necessita-se alteração porque empresa está está parada

Data Previsão de Entrega: 21/12/1999

Encaminhado para Implementação:

Em: 20/12/1999

Para: João Alves

Confirma

**Figura 20 – Tela de Análise da Pendência de Modificação**

**Análise da Pendência da Modificação**

Número: 4      Data de Análise: 05/01/2000

Responsável: Charles Hoppe

Viabilidade:

- Alta
- Média
- Baixa
- Nenhuma

Motivo:

Além de ser a necessidade de um cliente, poderá ser um atrativo para outros clientes.

Resultado da Análise:

- Aprovada
- Não Aprovada

Modificação sem restrições

Data Previsão de Entrega: 15/01/2000

Encaminhado para Implementação:

Em: 05/01/2000

Para: João Alves

Confirma

### 5.5.3 TELA DE REGISTRO

A tela de registro (figura 21) faz o controle dos acontecimentos durante a implementação, ou seja, todas as alterações, exclusões e inclusões devem ser relatadas neste arquivo para que possam ser utilizadas para consultas posteriores. Esta opção está contemplando os quadros 15,16,19,22 e 27 do processo de manutenção proposto.

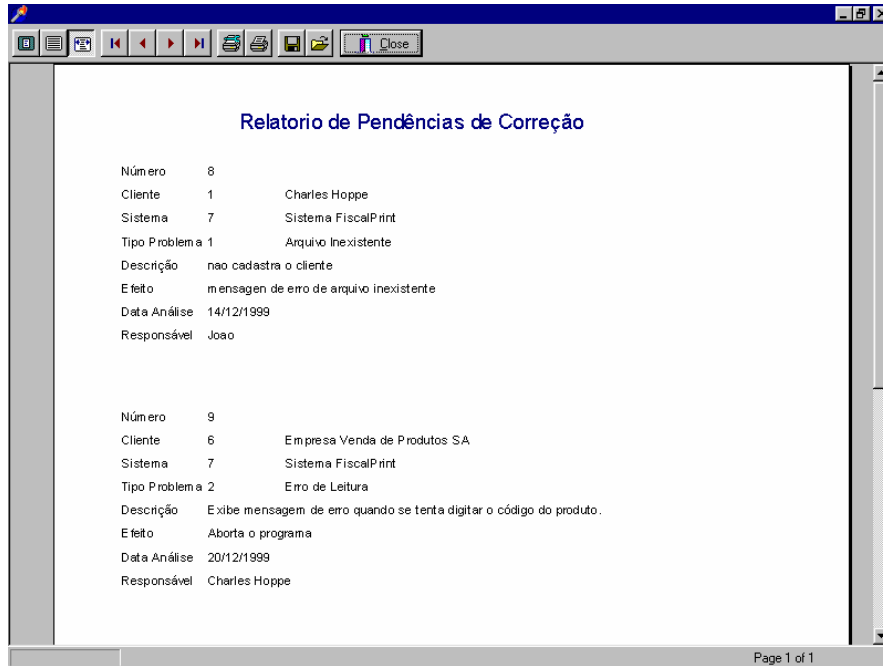
Figura 21 – Tela de Registro de Manutenção

CDCOMPONENTE	DescComp	CDSERVICO	DescServico	DSOCCORRENCIA
4	Frente de Caixa	1	Inclusão de Arquivos	Arquivo não estava sendo criado

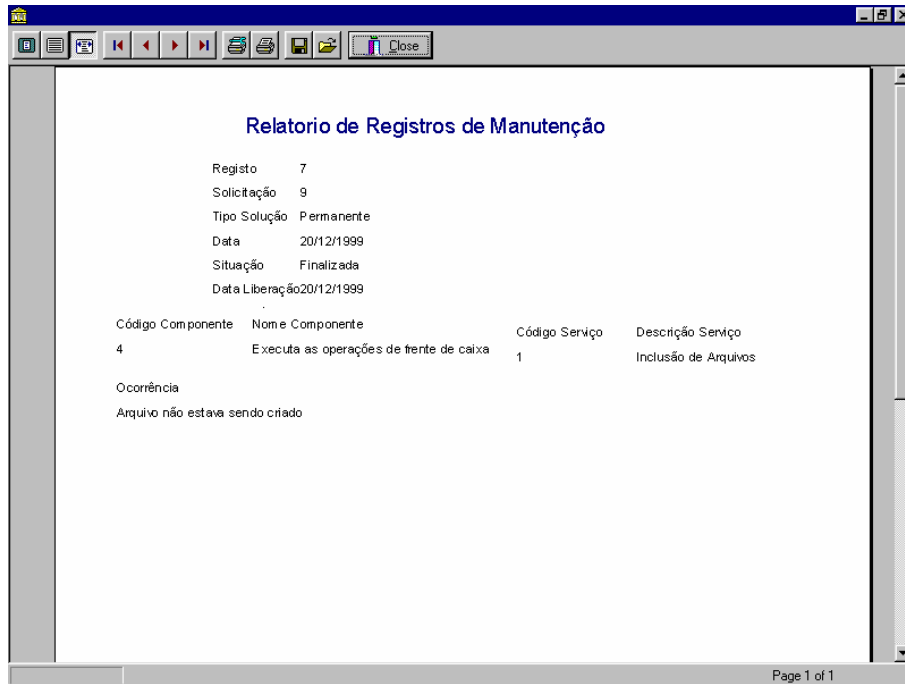
### 5.5.4 RELATÓRIOS

Serão demonstrados dois relatórios. O relatório de pendências (Figura 22) e o relatório de registro (Figura 23). O primeiro emite um relatório das solicitações que ainda estão em aberto, ou seja, que receberão o tratamento adequado, isto facilita a equipe de manutenção caso um determinado sistema receba muitos pedidos de manutenção. O segundo emite o registro das manutenções executadas, isto para que se possa repassar as informações para as pessoas interessadas

**Figura 22 – Relatório de Pendências de Correção**



**Figura 23 – Relatório de Registro de Manutenção**





## 6 CONCLUSÕES

*O software especificado e implementado mostrou-se eficaz no auxílio ao processo de manutenção de sistemas, nas etapas de solicitação e análise das pendências, bem como no registro das manutenções do sistema. Este tipo de controle ajuda a equipe de manutenção de sistemas e até mesmo os desenvolvedores de novas aplicações com feedback de problemas passados para auxiliá-los em dificuldades futuras.*

*A norma ISO/IEC 12207 demonstrou ser abrangente no aspecto da manutenção de sistemas, como também apresentou uma estrutura de fácil compreensão. O processo proposto tomou como base a norma ISO/IEC 12207 e ampliou e inseriu novas etapas como o treinamento da equipe e a verificação periódica. O processo proposto auxilia a empresa que tiver uma equipe de manutenção sub-dividida e que muitas vezes não tem contato entre os elementos da equipe. Desta forma o processo criou etapas de registro que possam facilitar consultas. As outras normas estudadas ISO 9000-3 e SPICE foram então utilizadas apenas no comparativo pelo motivo de que todos os seus processos são contemplados pela norma ISO/IEC 12207.*

*O processo proposto não contemplou as etapas de migração e descontinuação do software, previstas na norma ISO/IEC 12207, por tratarem-se de etapas mais complexas e podendo ser alvo de trabalhos futuros.*

*Uma das principais dificuldades deste trabalho foi a definição do processo de manutenção a partir do estudo das normas. Isto porque as normas não são ainda muito difundidas e exigem um conhecimento bom em engenharia de software.*

*O trabalho pode ser usado para o ensino de manutenção de software em disciplinas de engenharia de software.*

*Para trabalhos futuros pode-se sugerir a criação de um software para o controle e auxílio dos testes de sistemas após feita uma manutenção. Além disso, poderia-se estudar a norma específica para manutenção de software (sub-divisão da ISO/IEC 12207).*



## REFERÊNCIAS BIBLIOGRÁFICAS

- [ANT95] ANTONIONI, J. A. e Rosa, N. B. **Qualidade em software - manual de aplicação da ISO 9000**. São Paulo : Makron Books, 1995.
- [ASS93] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Normas de gestão da qualidade e garantia da qualidade – Diretrizes para a aplicação da NBR 19001 ao desenvolvimento, fornecimento e manutenção de software**, NBR ISO 9000-3. 1993.
- [BAR97] BARRETO, José Carlos. **Qualidade de software**. Endereço Eletrônico : [www.barreto.com.br/qualidade](http://www.barreto.com.br/qualidade), 1997.
- [BEL93] BELLIN, David. **Manutenção de software**. São Paulo : Makron Books, 1993.
- [FOU94] FOURNIER, Roger. **Guia prático para desenvolvimento e manutenção de sistemas estruturados**. São Paulo : Makron Books, 1994.
- [FRA98] FRARE, Alexandre. **Proposta de roteiro de implantação da norma internacional ISO/IEC 12207 – processos do ciclo de vida de software**. Blumenau, 1998. Monografia (Bacharelado em Ciências da Computação) Centro de Ciências Exatas e Naturias. FURB.
- [GRA97] GRAHL, Everaldo A. **Processo de Ciclo de Vida do Software- Norma ISO/IEC 12207**. Blumenau: Seminco – Anais, 1997.
- [HOR98] HORT, Glederson. **Protótipo de auxílio à implementação da norma NBR ISO 9000-3 apoiada pela NBR ISO/IEC 12207**. Blumenau, 1999. Monografia (Bacharelado em Ciências da Computação) Centro de Ciências Exatas e Naturias. FURB.
- [IAH99] IAHN, Anísio. **Avaliação de processos de software utilizando a norma ISO/IEC 15504**. Blumenau, 1999. Monografia (Bacharelado em Ciências da Computação) Centro de Ciências Exatas e Naturias. FURB.

- [ISO97] ISO/IEC JTC1/SC7 N1739. **DTR 15504-2: Information Technology – Software Process Assessment Part: 2 A Reference Model for Processes And Process Capability.** Canada 1997.
- [KRA98] KRAUSE, Conrad. **Análise de uma metodologia de sistemas baseado na norma ISO/IEC 12207.** Blumenau, 1998. Monografia (Bacharelado em Ciências da Computação) Centro de Ciências Exatas e Naturais, FURB.
- [NBR93] NBR ISO 9000-3. **Normas de gestão da qualidade e garantia da qualidade – Parte 3, Diretrizes para aplicação da NBR 9001 ao desenvolvimento, fornecimento e manutenção de “software”.**1993.
- [OLI95] OLIVEIRA, Álvaro Luiz Mendonça de; RÊGO, Claudete M.; COSTA, Márcia C. de Carvalho; JINO, Mário; CARVALHO, Mário BENTO de; COLOMBO, Regina M. T.. **Avaliação dos processos de software: modelos e o TAQA-PROC.** In: Anais do Workshop de Qualidade de Software. Recife – PE : Outubro de 1995.
- [PAR90] PARIKH, Girish. **Reengenharia de software.** Rio de Janeiro : LTC Livros Técnicos e Científicos, 1990.
- [PRE95] PRESSMAN, Roger S. **Engenharia de software.** São Paulo : Makron Books,1995.
- [SCU98] SCUSSIATO, Edéio. **Processo de manutenção de sistemas baseado no norma ISO/IEC 12207.** Blumenau, 1998. Monografia (Pós-Graduação a nível de especialização em tecnologia de desenvolvimento de sistemas) FURB.
- [SWA76] SWANSON, E. B. **The Dimensions of Maintenance.**, Proc Intl. Conf. Software Engineering, IEEE. Outubro,1976.
- [TSU97] TSUKUMO, Claudete M. Rêgo, et. **Qualidade de Software – Visões de Produto e Processo de Software.** Anais da II Escola Regional de Informática da Sociedade Brasileira de Computação Regional de São Paulo – II ERI da SBC. Piracicaba, 1997.

- [WEB95] WEBER, K.C. et al. Melhoria da qualidade dos processos de software: uso do método SEI/CMM em complemento à ISO 9000-3. Recife : 1995.

