

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

**AVALIAÇÃO DA QUALIDADE DO PROCESSO DE  
MANUTENÇÃO DE SOFTWARE UTILIZANDO A NORMA  
NBR ISO/IEC 12207**

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

**MARCELO PAULO DE SOUSA**

BLUMENAU, JUNHO/2002

2002/2-50

# **AVALIAÇÃO DA QUALIDADE DO PROCESSO DE MANUTENÇÃO DE SOFTWARE UTILIZANDO A NORMA NBR ISO/IEC 12207**

**MARCELO PAULO DE SOUSA**

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. Carlos Eduardo Negrão Bizzotto

---

Prof. Evaristo Baptista

# DEDICATÓRIA

Dedico este trabalho à minha esposa, pelo apoio recebido não apenas neste momento, mas durante toda a minha vida acadêmica.

# AGRADECIMENTOS

Ao meu orientador o Prof. Everaldo Artur Grahl pela orientação e apoio no preparo deste trabalho. Pelos seus estímulos e conselhos profissionais, que me auxiliaram a enfrentar os desafios.

A meus pais, Vicente de Paula Sousa e Isolde de Sousa, por toda a força e incentivo que me deram ao longo de minha vida.

A todos os colegas e amigos do Curso de Ciências da Computação, e principalmente aos meus colegas de trabalho. Com certeza foi através do apoio destes que encontrei motivação para seguir em frente ao longo do curso.

A todos aqueles que direta ou indiretamente contribuíram para a realização deste trabalho.

# SUMÁRIO

LISTA DE FIGURAS .....	VIII
LISTA DE QUADROS .....	IX
LISTA DE ABREVIATURAS.....	X
RESUMO .....	XI
ABSTRACT .....	XII
1 INTRODUÇÃO.....	1
1.1 OBJETIVOS.....	3
1.2 ESTRUTURA.....	3
2 MANUTENÇÃO DE SOFTWARE.....	4
2.1 CARACTERÍSTICAS DA MANUTENÇÃO .....	4
2.2 MANUTENIBILIDADE .....	9
2.3 TAREFAS DE MANUTENÇÃO.....	10
2.4 AVALIAÇÃO DA MANUTENÇÃO .....	12
3 QUALIDADE DE SOFTWARE.....	14
3.1 MEDIÇÃO DE SOFTWARE.....	15
3.2 ABORDAGEM GQM .....	17
3.3 NORMAS E MODELOS DE QUALIDADE .....	20
3.3.1 MANUTENÇÃO DE SOFTWARE NA NORMA ISO 9000-3 .....	21
3.3.2 MANUTENÇÃO DE SOFTWARE NA NORMA NBR ISO/IEC 12207 .....	22
3.3.3 MANUTENÇÃO DE SOFTWARE SEGUNDO O SPICE .....	25
3.3.4 BREVE COMPARATIVO DOS PROCESSOS DE MANUTENÇÃO SEGUNDO AS NORMAS DE QUALIDADE .....	27
4 DESENVOLVIMENTO DO TRABALHO .....	32

4.1	ANÁLISE DOS REQUISITOS.....	32
4.1.1	DESENVOLVIMENTO DO PLANO GQM .....	33
4.1.2	COLETA DE DADOS .....	34
4.2	ESPECIFICAÇÃO DO SOFTWARE.....	39
4.2.1	DIAGRAMA DE CASOS DE USO.....	40
4.2.2	DIAGRAMA ENTIDADE RELACIONAMENTO.....	42
4.2.3	DICIONÁRIO DE DADOS .....	44
4.3	IMPLEMENTAÇÃO .....	48
4.3.1	MENU PRINCIPAL.....	48
4.3.2	CADASTRO DE ORGANIZAÇÕES .....	49
4.3.3	CADASTRO DE PROJETOS .....	50
4.3.4	CADASTRO DE METAS.....	51
4.3.5	CADASTRO DE QUESTÕES.....	53
4.3.6	CADASTRO DE MÉTRICAS .....	54
4.3.7	CADASTRO DE REGRAS DE PONTUAÇÃO .....	56
4.3.8	CADASTRO DE CLASSIFICAÇÕES .....	57
4.3.9	MENU AVALIAÇÃO.....	58
4.3.10	PROCESSO DA AVALIAÇÃO .....	59
4.3.11	RELATÓRIO DA AVALIAÇÃO DE SOFTWARE.....	60
4.3.12	RELATÓRIO QUESTIONÁRIO DE AVALIAÇÃO.....	61
4.3.13	MENU SOBRE .....	61
4.4	RESULTADOS PRINCIPAIS .....	61
4.5	CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO.....	62
5	CONCLUSÕES .....	64
5.1	SUGESTÕES .....	65

REFERÊNCIAS BIBLIOGRÁFICAS .....	<b>ERRO! INDICADOR NÃO DEFINIDO.</b>
ANEXO I – RELATÓRIO DE AVALIAÇÃO DO PROCESSO MANUTENÇÃO.....	66
ANEXO II – QUESTIONÁRIO DE AVALIAÇÃO DO PROCESSO MANUTENÇÃO.....	69

## LISTA DE FIGURAS

FIGURA 3-1- Preocupação com Qualidade de Software.....	15
FIGURA 3-2 - Medir Qualidade. ....	16
FIGURA 3-3 - Razões para medir software. ....	17
FIGURA 3-4 - abordagem GQM.....	18
FIGURA 4-1 – Diagrama de casos de uso .....	41
FIGURA 4-2 – Cardinalidade da notação IDEF1X .....	43
FIGURA 4-3 - Diagrama entidade relacionamento.....	44
FIGURA 4-4 – Menu Principal do Protótipo. ....	48
FIGURA 4-5 – Tela de Cadastro da Organização.....	50
FIGURA 4-6 – Tela de Cadastro de Projetos.....	51
FIGURA 4-7 – Tela de Cadastro de Metas. ....	53
FIGURA 4-8 – Tela de Cadastro de Questões. ....	54
FIGURA 4-9 – Tela de Cadastro de Métricas.....	55
FIGURA 4-10 – Tela de Cadastro de Regras.....	57
FIGURA 4-11 – Tela de Cadastro de Classificações.....	58
FIGURA 4-12 – Tela de Avaliação.....	59
FIGURA 4-13 – Tela do Processo de Avaliação.....	60
FIGURA 4-14 – Tela Sobre .....	61



## LISTA DE QUADROS

QUADRO 2-1 – Categorias de Manutenção de Software.....	6
QUADRO 3-1 - Folha de abstração. ....	19
QUADRO 3-2 – Comparativo do processo de manutenção.....	27
QUADRO 4-1 – Meta G1.....	35
QUADRO 4-2 – Meta G2.....	37
QUADRO 4-3 – Meta G3.....	38
QUADRO 4-4 – Dicionário de Dados.....	45

## LISTA DE ABREVIATURAS

ABNT	Associação Brasileira de Normas Técnicas
CASE	<i>Computer Aided Systems Engineering</i>
DER	Diagrama Entidade Relacionamento
IEC	<i>International Electrotechnical Committee</i>
ISO	<i>International Organization for Standardization</i>
GQM	<i>Goal Question Metric</i>
SQL	<i>Structured Query Language</i>

# RESUMO

Este trabalho demonstra uma forma de avaliação do processo de manutenção de software em empresas produtoras de software. O trabalho usa como base a norma NBR ISO/IEC 12207 e a abordagem *Goal Question Metric* (GQM) que dá suporte à definição de métricas para avaliação. Como forma de demonstrar a avaliação foi desenvolvido um protótipo que fornece suporte ao processo de avaliação proposto.

## **ABSTRACT**

This work demonstrates a form of evaluation of the process of software maintenance in companies producing of software. The work uses as base the NBR ISO/IEC 12207 norm and the Goal Question Metric approach (GQM) that gives support to the definition of metric for evaluation. As form of demonstrating the evaluation was developed a prototype that supplies support to the proposed evaluation process.

# 1 INTRODUÇÃO

A intensificação da competição entre empresas e a caracterização da tecnologia como área estratégica, delineia um cenário em que o software se apresenta como elemento crítico. Visando eliminar este gargalo, definido como “A Crise do Software” desde fins da década de sessenta, várias iniciativas tem sido tomadas no sentido de definir e aplicar métodos e mecanismos que viabilizem um controle efetivo sobre o desenvolvimento, bem como um aumento na qualidade e produtividade (Parikh, 1990).

Segundo Rocha (2001), o desenvolvimento de um sistema termina quando o produto é liberado para o cliente e o software é instalado para uso operacional. Daí em diante, deve-se garantir que esse sistema continue sendo útil e atendendo às necessidades do usuário. Começa então a fase de manutenção. Segundo Parikh (1990), a manutenção eficaz, eficiente e econômica é crucial para o sucesso organizacional e pessoal. Até mesmo a própria sobrevivência de uma organização pode depender da manutenção oportuna e confiável. As estatísticas geralmente citadas são alarmantes. A maioria das empresas aloca aproximadamente 50% do seu orçamento para a manutenção contínua. A maioria dos programadores gasta mais de 50% de seu tempo na manutenção e, embora sendo discutível, 67% do ciclo de vida do software é manutenção, enquanto apenas 33% são novo desenvolvimento.

A globalização da economia tem influenciado as empresas produtoras e prestadoras de serviços de software a alcançar o patamar de qualidade e produtividade internacional para enfrentarem a competitividade cada vez maior. A norma NBR ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de Software é usada como referência em muitos países, inclusive no Brasil, para alcançar esse diferencial competitivo. Ela tem por objetivo auxiliar os envolvidos na produção de software a definir seus papéis, por meio de processos bem definidos e assim proporcionar às organizações que a utilizam um melhor entendimento das atividades a serem executadas nas operações que envolvem, de alguma forma o software (Rocha, 2001).

A qualidade de software é um tema que atualmente está em evidência devido à preocupação com os custos na manutenção de software e a baixa conformidade nos requisitos dos mesmos (Koscianski, 1999). A qualidade representa um fator essencial no desenvolvimento de um software, fato que contribui para o surgimento de diversas

abordagens de avaliação de qualidade. Segundo Wangenheim (1999) e Fernandes (1995), com a utilização da abordagem *Goal Question Metric* (GQM), cada vez mais será possível ter a compreensão dos processos de desenvolvimento de software, fazer a gerência de projetos de software e focar em um melhoramento contínuo dos sistemas. A abordagem GQM é o mecanismo usado para definir e avaliar um conjunto de objetivos operacionais usando métricas. Essa abordagem representa uma sistemática para ajuste e integração de objetivos com modelos de processos, produtos e perspectivas de qualidade de software, baseadas em necessidades específicas do projeto e da organização.

O objetivo principal de uma organização é desenvolver um produto de alta qualidade dentro de um cronograma e orçamento aceitáveis. Por causa dos requisitos permanentemente crescentes no desenvolvimento de software com respeito ao cliente ou novas tecnologias, o melhoramento sistemático e contínuo precisa ser parte integrada ao processo de software. O melhoramento do produto é tipicamente atingido através do melhoramento dos processos usados para produzir o produto. O melhoramento dos processos pode ser atingido pela modificação dos processos gerenciais ou técnicos, ou pela introdução das novas tecnologias. Em todo caso, mensuração de software é uma tecnologia chave de qualquer programa de melhoria, enfocando no melhoramento da qualidade, os pontos fortes e fracos, que podem ser identificados pelo programa de mensuração inicial (Wangenheim, 2000).

Segundo Barros (1998), uma avaliação efetiva das atividades de manutenção de software é normalmente difícil ou complicada pela falta de dados concretos, e somente através da conservação dos registros é possível obter-se medidas de desempenho para avaliar o processo de manutenção. A manutenção de software, até muito recentemente, era a fase negligenciada do processo de desenvolvimento de software. A literatura sobre manutenção contém poucos lançamentos quando comparada com as atividades de desenvolvimento. Relativamente pouca pesquisa ou dados de produção tem sido compilado sobre o assunto, e poucas abordagens ou métodos técnicos têm sido propostos para se realizar a avaliação da eficácia das atividades de manutenção (Pressmann, 1995). Esta dificuldade de mensuração do processo de manutenção motivou a elaboração de um conjunto de métricas para a avaliação da qualidade do processo de manutenção de software. Para isso são relevantes o estudo das normas de qualidade e o domínio das atividades envolvidas no processo de manutenção.

## 1.1 OBJETIVOS

O objetivo principal deste trabalho foi a elaboração de um conjunto de métricas apropriadas para a avaliação do processo de manutenção de software.

Os objetivos específicos do trabalho foram:

- a) utilização do método GQM para definição de métricas;
- b) análise do processo de manutenção nas normas ISO 9000-3, NBR ISO/IEC 12207 e SPICE;
- c) desenvolvimento de um protótipo de software de apoio ao processo de avaliação elaborado.

## 1.2 ESTRUTURA

O primeiro capítulo apresenta uma introdução ao trabalho, iniciando com alguns conceitos empregados em sua elaboração. São apresentados ainda, os objetivos e a organização do texto.

O segundo capítulo apresenta conceitos referentes a manutenção de software.

O terceiro capítulo apresenta conceitos referentes a medição e normas de qualidade de software.

O quarto capítulo apresenta o desenvolvimento do trabalho, com a especificação do protótipo. Além disso é apresentado o funcionamento do protótipo de software de apoio à avaliação do processo de manutenção de software.

O quinto capítulo apresenta as conclusões e sugestões do trabalho.

## 2 MANUTENÇÃO DE SOFTWARE

Conforme Barros (1998), o conceito de manutenção de software é visto como o processo de modificação de um ou mais programas depois que o software foi entregue ao cliente e está sendo usado (em produção). A manutenção de software é bem mais do que simplesmente consertar erros.

Segundo Lima (2001), a atividade de Manutenção do Sistema de Software desenvolvida deve ser entendida como um processo de modificação do sistema de software ou de seus componentes, a fim de corrigir defeitos, melhorar desempenhos ou outros atributos, ou ainda, adaptá-lo a um ambiente em constante transformação. Para Pressmann (1995) a manutenção de software também é definida como a realização das atividades necessárias para manter um sistema operacional e reativo a seus usuários após ter sido aceito e colocado em produção.

A manutenção de sistemas de computador deve ser encarada como uma atividade de extrema importância e necessária para adequar os produtos de software às necessidades da organização. A manutenção de sistemas não pode ser considerada só como uma fase ou etapa do ciclo de vida de software na organização (Scussiato, 1998).

### 2.1 CARACTERÍSTICAS DA MANUTENÇÃO

Tipicamente, a vida útil de um sistema grande e complexo varia de cinco a dez anos. Todavia, não é difícil encontrar sistemas que tenham entre 15 e 20 anos. Por essa razão, os peritos em informática vêem a entrega do sistema para seus usuários como o início de sua vida útil. Desse ponto em diante o sistema continuará a evoluir até que seja eliminado e substituído por outro (Fournier, 1994).

O ciclo de vida da engenharia de software deve sempre enfatizar a necessidade de montar sistemas que sejam flexíveis e adaptáveis. O processo de manutenção de software deve ser executado sistemática e estruturadamente para retardar a degradação do sistema que ocorre com o passar do tempo e garantir que continue sendo passível de manutenção.

O conceito tradicional de manutenção (um programador consertando um defeito) não é um quadro preciso do processo de manutenção como um todo. Na realidade, diversas



pesquisas demonstraram que se gasta mais tempo melhorando o sistema, para que ele atenda às necessidades dos usuários que com qualquer outro tipo de atividade de manutenção. A manutenção de software já foi caracterizada como (ou comparada a) um *iceberg*. O engenheiro de software está sempre esperando (na verdade torcendo) que o trabalho necessário se restrinja ao que é visível no momento. Mas geralmente, um grande número de problemas se esconde sob a superfície (Barros, 1998).

O processo de construção do software se dá por meio da realização de várias atividades e é efetuado por refinamentos sucessivos. Parte-se de uma visão macroscópica e segue-se evoluindo até o detalhamento do código. Cada atividade tem como base para sua realização os produtos gerados na etapa anterior. Depois de construído, o software entra em operação e pode ser necessário efetuar alterações no mesmo (Rocha, 2001). O que pode provocar essas manutenções são principalmente:

- a) falhas de processamento: devido a erros no software/produto;
- b) falhas de desempenho: nesse caso não existem erros no software, mas a falha se manifesta porque o comportamento do mesmo não se mantém adequado à sua especificação funcional;
- c) falhas de implementação: nesse caso não existem erros e o desempenho está adequado, porém existem violações nos padrões de programação ou inconsistência/não-conclusão no projeto detalhado. Essas falhas resultam na diminuição da qualidade do software e só são descobertas quando ocorrem auditorias para assegurar a aderência às práticas de programação. A manutenção efetuada em resposta às falhas desse tipo é a manutenção corretiva;
- d) alteração no ambiente de dados: devido às alterações no formato dos dados ou às alterações na maneira de lidar com os dados como um todo;
- e) alteração no ambiente de processamento: devido à rápida mudança que se processa em todo o campo da computação com o surgimento de novas gerações de hardware, novos sistemas operacionais, novos periféricos e outros elementos que freqüentemente estão sendo atualizados ou modificados. A manutenção efetuada em resposta a essas alterações é a manutenção adaptativa;
- f) inclusão de novas capacidades, modificações em funções existentes ou ampliações gerais: mesmo que um pacote de software seja bem sucedido, ele pode se tornar ineficiente à medida que for usado e, para corrigir isso, o cliente solicita que sejam

realizadas modificações. A manutenção efetuada em resposta a essas modificações/ampliações é denominada manutenção aperfeiçoadora. Esse tipo de manutenção é responsável pela maior parte de todo esforço despendido em manutenção de software;

- g) melhoramento na confiabilidade ou na facilidade de futuras manutenções: mesmo que um software seja construído e documentado de acordo com os padrões estabelecidos (e, portanto, não existam falhas de implementação), ainda é possível melhorá-lo no sentido de torná-lo mais confiável ou mais fácil de ser alterado ou ampliado. Essa manutenção é chamada de manutenção preventiva.

Em contraste com a manutenção corretiva e adaptativa que servem simplesmente para manter um sistema funcionando, as manutenções aperfeiçoadora e preventiva estão direcionadas para manter um sistema funcionando com menos gastos ou funcionando para melhor servir as necessidades de seus usuários (Parikh, 1990).

O quadro 2-1 mostra as categorias genéricas de atividades de manutenção e seus objetivos individuais para um sistema (o que precisa ser feito).

QUADRO 2-1 – CATEGORIAS DE MANUTENÇÃO DE SOFTWARE.

Categoria	Objetivo	Exemplos
Aperfeiçoadora	Melhorar o sistema com mudanças solicitadas pelos usuários para atender às necessidades de negócio em evolução.	Inclusão de novas funções. Inclusão de novos relatórios. Modificação de relatórios existentes.
Adaptativa	Modificar o sistema para acomodar mudanças físicas no próprio ambiente.	Introdução de uma nova versão do sistema operacional. Adição de periféricos novos. Introdução de uma nova versão do compilador.
Corretiva	Modificar o sistema em função de mudanças exigidas pela descoberta de erros ou ambigüidades.	Consertar uma tela que devolve valores inválidos. Corrigir um relatório que mostra totais incorretos.

		Consertar um cancelamento anormal em produção causado por uma transação de entrada inválida.
Preventiva	Modificar o sistema em função de mudanças necessárias para manter sua eficiência e confiabilidade.	Reorganizar o banco de dados para otimizar o desempenho. Aumentar os tamanhos de arquivos/banco de dados de produção. Manutenção regular de equipamentos e recursos de rede.
Estrutural	Modificar a arquitetura interna do sistema com alterações destinadas a facilitar sua manutenção.	Melhorar a documentação existente de programas. Reestruturar o código para melhorar a sua legibilidade. Refazer os programas de aplicações para melhorar a sua modularidade.

Fonte: (Fournier, 1994, p. 269)

Segundo Fournier (1994), embora haja muitos problemas a enfrentar na manutenção de software, três aspectos são centrais:

- a) a dificuldade de executar as atividades de manutenção;
- b) o alto grau de erro associado com a execução de tarefas de manutenção;
- c) a baixa moral dos mantenedores de software.

A principal dificuldade na execução das tarefas de manutenção origina-se do fato de que a complexidade de grandes sistemas aumenta dramaticamente com o tempo, em grande parte à mudança da estrutura do programa. Esta dificuldade é composta pela informação imprecisa, desatualizada e violações flagrantes dos padrões de programação da organização. O alto grau de erro associado com a execução das tarefas de manutenção é geralmente resultado direto do aumento da complexidade dos programas.

Parte do problema com relação ao moral está relacionado com a falta da aplicação de ferramentas de engenharia de software moderna e métodos de controle da qualidade na manutenção de software e percepção dos mantenedores de que estão trabalhando sem suporte

técnico adequado. Há também uma forte pressão durante as atividades de manutenção para produzir quantidade e não qualidade (Parikh, 1990).

Segundo Pressmann (1995), a maioria dos problemas associados à manutenção de software pode remeter-se a deficiências na maneira segundo a qual o software foi planejado e desenvolvido. A falta de controle e disciplina no desenvolvimento da engenharia de software quase sempre se traduz em problemas durante a manutenção de software. Entre os muitos problemas clássicos que podem estar associados à manutenção de software encontram-se os seguintes:

- a) freqüentemente é difícil ou impossível rastrear a evolução do software através de muitas versões ou lançamentos. As mudanças não estão adequadamente documentadas;
- b) freqüentemente é difícil ou impossível rastrear o processo através do qual o software foi criado;
- c) 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;
- d) a “outra pessoa” freqüentemente 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 desenvolvedor de software quando a manutenção for necessária;
- e) a documentação não existe ou é muito ruim. O reconhecimento de que o software deve ser documentado é um primeiro passo, mas a documentação deve ser compreensível e consistente com o código-fonte para ter algum valor;
- 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;
- g) a manutenção não é vista como um trabalho muito glamouroso. Grande parte dessa percepção vem do elevado nível de frustração associado ao trabalho de manutenção.

## 2.2 MANUTENIBILIDADE

Segundo Pressmann (1995), a manutenibilidade pode ser definida qualitativamente como a facilidade com que um software pode ser entendido, corrigido, adaptado e/ou aumentado.

A manutenibilidade final do software é afetada por muitos fatores. A negligência no projeto, na codificação e no teste tem um impacto negativo óbvio sobre nossa capacidade de manter o software resultante. Uma configuração de software ruim pode causar um impacto negativo semelhante, mesmo quando os passos técnicos anteriormente mencionados foram realizados com cuidado.

Além dos fatores que podem estar associados à metodologia de desenvolvimento, existe uma série de fatores que pode estar relacionada ao ambiente de desenvolvimento:

- a) disponibilidade de um pessoal de software qualificado;
- b) estrutura de sistema compreensível;
- c) facilidade de manuseio do sistema;
- d) uso de linguagens de programação padronizadas;
- e) uso de sistemas operacionais padronizados;
- f) estrutura de documentação padronizada;
- g) disponibilidade de casos de teste;
- h) facilidades de depuração embutidas;
- i) disponibilidade de um computador adequado para realizar a manutenção;
- j) disponibilidade da pessoa ou grupo que originalmente desenvolveu o software.

A manutenibilidade de software, assim como a qualidade e a confiabilidade, são aspectos difíceis de quantificar. Porém, pode-se avaliar a manutenibilidade indiretamente, ao considerar os atributos da atividade de manutenção que podem ser medidos. Uma série de métricas da manutenibilidade que se relaciona ao esforço despendido durante a manutenção podem ser citados (Pressmann, 1995):

- a) tempo de reconhecimento do problema;
- b) tempo de retardo administrativo;
- c) tempo de coleta de ferramentas de manutenção;
- d) tempo de análise do problema;

- e) tempo de especificação das mudanças;
- f) tempo de correção (ou modificação) ativa;
- g) tempo de testes locais;
- h) tempo de testes globais;
- i) tempo de revisão de manutenção;
- j) tempo de recuperação total.

Além dessas medidas orientadas para o tempo, a manutenibilidade pode ser medida indiretamente, ao considerar-se as medidas da estrutura do projeto e as métricas da complexidade do software.

## **2.3 TAREFAS DE MANUTENÇÃO**

As tarefas associadas a 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 estabelecida, procedimentos de avaliação e relatórios devem ser descritos e uma seqüência de eventos padronizada deve ser definida para cada pedido de manutenção. Além disso, um sistema de registro das atividades de manutenção deve ser estabelecido e critérios de avaliação e revisão devem ser definidos (Pressmann, 1995).

No caso da manutenção raramente existem organizações formais (notáveis exceções são grandes empresas desenvolvedoras de software), e a manutenção freqüentemente é executada informalmente. Embora não seja necessário estabelecer uma organização de manutenção formal, uma delegação de responsabilidades informal é absolutamente essencial até mesmo para os pequenos desenvolvedores de software. Deve-se estabelecer uma área de responsabilidade pela manutenção com um fluxo de eventos pré-definido iniciado pelo pedido de manutenção.

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 solicitante que deseja a atividade de manutenção. Se um erro for encontrado, uma descrição completa de circunstâncias que levaram ao erro (inclusive dados de entrada, listagens e outros materiais de apoio) deve ser

incluída. Para pedidos de manutenção adaptativa ou perfectiva (aperfeiçoadora), uma breve especificação de mudança é submetida (Fournier, 1994).

A seqüência de eventos que ocorre como resultado de um pedido de manutenção consiste em primeiro determinar o tipo de manutenção que deve ser realizado. Em muitos casos, um usuário/solicitante pode ver um pedido como uma indicação de erro de software (manutenção corretiva) enquanto um desenvolvedor pode ver o mesmo pedido como uma adaptação ou acréscimo. Se existir uma diferença de opinião, um acordo deve ser negociado.

Conforme Pressmann (1995), um fluxo de pedido de manutenção corretiva inicia-se com uma avaliação de gravidade do erro. Se existir um erro grave (por exemplo, um sistema crítico não pode funcionar), é designada uma equipe sob a direção do supervisor de sistemas e a análise do problema inicia-se imediatamente. Para erros menos graves, o pedido de manutenção é avaliado e categorizado, e depois programado em conjunto com outras tarefas que exijam recursos de desenvolvimento de software.

Em certos casos, um erro pode ser tão grave que os controles normais de manutenção devem ser abandonados temporariamente. O código deve ser modificado imediatamente, sem uma correspondente avaliação dos efeitos colaterais e uma apropriada atualização da documentação. Esse modo “apaga incêndio” para a manutenção corretiva é reservado somente para situações de “crise” e deve representar uma porcentagem muito pequena de todas as atividades de manutenção.

Os pedidos de manutenção adaptativa e aperfeiçoadora seguem um caminho diferente. As adaptações são avaliadas e categorizadas (priorizadas) antes de serem colocadas numa fila para sofrer manutenção. Os acréscimos passam pela mesma avaliação. Porém, nem todos os pedidos de acréscimo são levados a efeito. A estratégia comercial, os recursos disponíveis, a tendência dos softwares atuais e futuros e muitas outras questões podem fazer com que um pedido de acréscimo seja rejeitado. Aqueles acréscimos que devem ser feitos também são colocados na fila de manutenção. A prioridade de cada pedido é estabelecida e o trabalho exigido é programado como se fosse outro esforço de desenvolvimento (para todos os fins, ele o é). Se uma prioridade extremamente elevada for estabelecida, o trabalho pode iniciar-se imediatamente (Pressmann, 1995).

Entre as atividades de manutenção de software incluem-se ainda as mesmas a modificação do projeto de software, revisões, modificações necessárias do código, testes unitários e de integração (inclusive testes de regressão e casos de teste anteriores) e testes de validação.

## 2.4 AVALIAÇÃO DA MANUTENÇÃO

Segundo Pressmann (1995), historicamente, a conservação de registros para a manutenção de software tem sido inadequada. Por essa razão, freqüentemente não se pode avaliar a efetividade das técnicas de manutenção, determinar a qualidade de um programa de produção e não há como determinar quanto custa realmente a manutenção. O primeiro problema recai sobre quais dados devem ser guardados, sugere-se uma lista abrangente:

- a) identificação do programa;
- b) número de instruções-fonte;
- c) número de instruções de código de máquina;
- d) linguagem de programação usada;
- e) data de instalação do programa;
- f) número de execuções do programa desde a instalação;
- g) número de falhas de processamento;
- h) identificação e nível de mudança do programa;
- i) número de instruções-fonte adicionadas pela mudança no programa;
- j) número de instruções-fonte suprimidas pela mudança no programa;
- k) número de pessoas-hora empregadas por mudança;
- l) data de mudança no programa;
- m) identificação do engenheiro de software;
- n) identificação do formulário do pedido de manutenção;
- o) tipo de manutenção;
- p) datas de início e encerramento da manutenção;
- q) número cumulativo de pessoas/hora empregadas em manutenção;
- r) benefícios líquidos associados à manutenção realizada.

Uma avaliação das atividades de manutenção de software freqüentemente é complicada pela falta de dados concretos. Se uma conservação de registros for iniciada, uma



série de medidas de desempenho de manutenção pode ser desenvolvida. Uma lista abreviada de medidas em potencial pode ser proposta (Pressmann, 1995):

- a) número médio de falhas de processamento por execução do programa;
- b) total de pessoas/hora empregadas em cada categoria de manutenção;
- c) número médio de mudanças de programa feitas por programa, por linguagem e por tipo de manutenção;
- d) número médio de pessoas/hora empregadas por instruções-fonte adicionadas ou suprimidas devido à manutenção;
- e) média de pessoas/hora empregadas por linguagem;
- f) tempo médio de renovação do formulário de pedido de manutenção;
- g) porcentagem de pedidos de manutenção por tipo.

As medidas citadas podem proporcionar uma estrutura quantitativa a partir da qual podem ser tomadas decisões sobre a técnica de desenvolvimento, escolha da linguagem, projeções sobre o esforço de manutenção, alocação de recursos e muitas outras questões (Pressmann, 1995).

### 3 QUALIDADE DE SOFTWARE

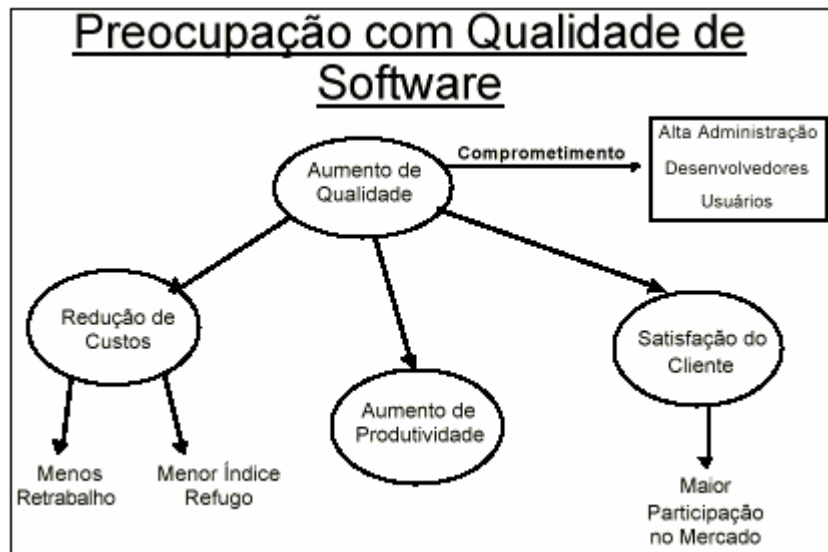
Qualidade de software é a conformidade a requisitos funcionais e desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente desenvolvido (Cunha, 2001). Segundo Fernandes (1995), para se avaliar a qualidade do software devem ser obtidos fatores e requisitos através de pesquisas junto a clientes, grupos de foco, entre outros. Entretanto, o atingimento de níveis aceitáveis de qualidade está subordinado à medição e à comparação dos resultados com padrões já determinados pela organização.

Outra condição básica é a avaliação sistemática da satisfação do cliente/usuário com o produto logo após a sua implantação e a avaliação da satisfação durante o uso mais prolongado, a fim de verificar se os requisitos foram plenamente satisfeitos (são raríssimas as organizações de informática que adotam tal procedimento).

De qualquer forma, a questão a resolver é que muitas vezes os requisitos dos clientes e/ou usuários são conflitantes com o atingimento das metas da qualidade referentes ao projeto/processo e ao produto. O prazo é um exemplo típico. Muito freqüentemente são impostos prazos inexecutáveis, cuja busca de atingimento pela equipe de projeto impacta severamente nos fatores de risco do processo, principalmente aqueles relacionados com os custos da má qualidade.

Não existem, infelizmente, soluções de curto prazo para resolver este problema. Somente através de medições relativas ao projeto, processo e produto, aliadas à permanente educação do cliente/usuário e dos desenvolvedores de software, é que se pode resolver a questão e isto, naturalmente, requer tempo, paciência, persistência e comprometimento. Essa preocupação com a qualidade de software nas organizações de informática pode ser vista na fig. 3-1.

FIGURA 3-1- PREOCUPAÇÃO COM QUALIDADE DE SOFTWARE.



Fonte: (Cunha, 2001, p. 14)

### 3.1 MEDIÇÃO DE SOFTWARE

Segundo Fernandes (1995), a Gestão de Projetos e de produtos de software somente atinge determinado nível de eficácia e exatidão se houverem medidas que possibilitem gerenciar através de fatos e o que é mais importante, gerenciar os aspectos econômicos do software, que geralmente são negligenciados em organizações de desenvolvimento.

Medidas, em Engenharia de Software, são denominadas de métricas, que podem ser definidas como métodos de determinar, quantitativamente, a extensão em que o projeto, o processo e o produto de software têm certos atributos. Isto inclui a fórmula para determinar o valor da métrica como também a sua forma de apresentação e as diretrizes de utilização e interpretação dos resultados obtidos no contexto do ambiente de desenvolvimento do software.

Segundo Cunha (2001), as métricas são linhas de conduta e não regras: dão uma indicação de qualidade e do progresso que um projeto tem. Um conceito para métrica de software seria de uma função cuja entrada são dados e cuja saída é um valor que pode ser

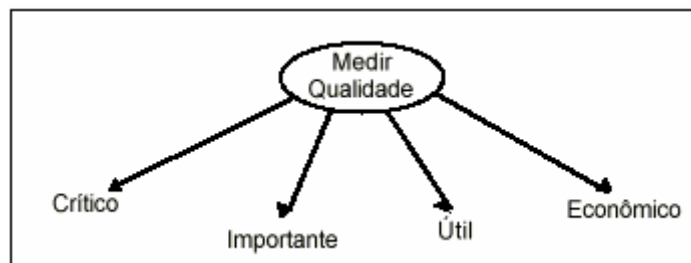
interpretado como o grau em que o software possui um dado atributo que afeta a sua qualidade.

Há várias características importantes associadas com o emprego das métricas de software. Sua escolha, para fins de gerenciar através de fatos e dados, requer alguns pré-requisitos importantes:

- a) os objetivos que se pretende atingir com a utilização de métricas;
- b) as métricas devem ser simples de entender;
- c) as métricas devem ser objetivas;
- d) as métricas devem ser efetivas no custo;
- e) as métricas devem ser informativas.

As métricas de software fornecem insumos para a gerência orientar-se em relação à saúde do projeto e identificam a necessidade de correções em toda a atividade de desenvolvimento de software. A fig. 3-2 mostra os vários benefícios de medição da qualidade do software.

FIGURA 3-2 - MEDIR QUALIDADE.



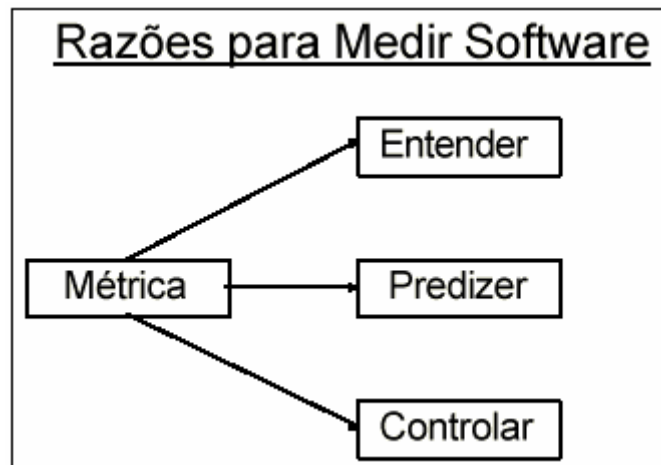
Fonte: (Cunha, 2001, p. 33)

As vantagens que a medição traz ao processo e ao produto de software são:

- a) quantificar a confiabilidade e o desempenho do produto;
- b) isolar atributos do processo e do produto que impactam em confiabilidade e desempenho;
- c) demonstrar como as mudanças de processo e de produto impactam nestes atributos.

As vantagens podem ser resumidas na fig. 3-3 em três razões para medir software.

FIGURA 3-3 - RAZÕES PARA MEDIR SOFTWARE.



Fonte: (Cunha, 2001, p. 35)

A caracterização das métricas pode ser realizada de três formas:

- a) através de exemplos;
- b) através de critérios pessoais;
- c) de acordo com experiências.

A utilização de métricas de software quantifica o grau de presença de determinadas características no produto de software. As principais dificuldades encontradas para o uso de métricas de software são a falta de experimentos para validação e falta de ferramentas para apoio.

Em Engenharia de Software as propriedades que podem ser avaliadas através de métricas são:

- a) diretamente mensuráveis (tamanho, esforços, avaliação de cronograma);
- b) indiretamente mensuráveis (qualidade, desempenho, confiabilidade, complexidade, manutenibilidade, usabilidade, portabilidade).

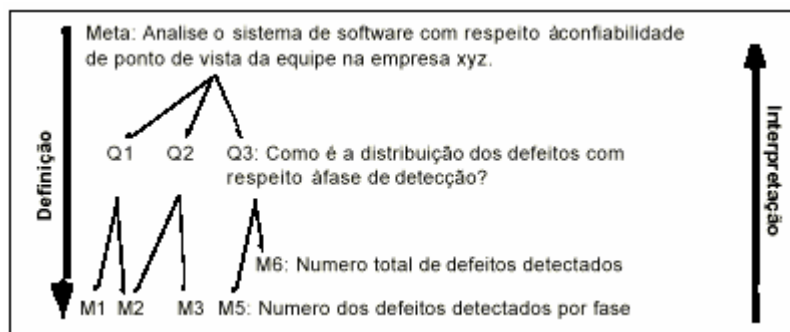
## 3.2 ABORDAGEM GQM

Mensuração é uma tecnologia chave de todo programa de melhoramento. O paradigma *Goal/Question/Metric* (GQM) é uma abordagem orientada a metas para a mensuração de produtos e processos de software, suportando a definição *top-down* de um programa de

mensuração e a análise e interpretação *bottom-up* dos dados de mensuração. Ela foi utilizada com sucesso em diversas empresas, como NASA-SEL (EUA), Robert Bosch GmbH (Alemanha), Allianz Lebensversicherungs-AG (Alemanha), Digital SPA (Itália), Motorola e Schlumberger (Holanda).

O paradigma GQM é baseado no requisito de que a mensuração deveria ser orientada a metas, por exemplo, toda coleta dos dados deve ser baseada num fundamento lógico, que é documentado explicitamente. Essa abordagem tem várias vantagens: ela suporta a identificação das métricas úteis e relevantes tanto quanto suporta a análise e interpretação dos dados coletados. Ela permite um assessoramento da validade das conclusões a que se chegou e evita a resistência contra programas de mensuração (Wangenheim, 2000). Na fig. 3-4 pode ser vista a abordagem GQM.

FIGURA 3-4 - ABORDAGEM GQM.



Fonte: (Wangenheim, 2000, p. 9)

O GQM é um método para realização de métricas a partir do qual são estipuladas metas (*goal*). Com o intuito de atender a estas metas, são elaboradas perguntas (*question*) e, a partir destas perguntas é identificado o que deve ser medido (*metric*). O refinamento das metas em métricas permite interpretar as métricas dentro de um contexto, em uma abordagem *top-down* (do mais geral para o mais específico). Outro aspecto interessante é que essa estrutura hierárquica definida no GQM previne a interpretação errônea dos dados, pois fazem sentido no contexto elaborado ao qual se aplicam as métricas (Cunha, 2001).

Segundo Emam (1998), a elaboração do Plano GQM compreende as seguintes atividades:

- a) preparação da Folha de Abstração: esta é uma técnica constante do GQM, cuja

preparação requer a definição do foco da qualidade (definido no quadrante superior esquerdo da folha de abstração, e exprime o conteúdo principal do objetivo da avaliação), fatores de variação (definidos no quadrante superior direito da folha de abstração e compreendem itens que podem interferir nos resultados da avaliação, como tipo de técnica utilizada, experiência e motivação dos participantes, etc), hipóteses básicas (ocupam o quadrante inferior esquerdo da folha de abstração e representam resultados que se espera identificar através da avaliação) e impactos sobre as hipóteses básicas (descritos no quadrante inferior direito da folha de abstração e definem o modo como os fatores de variação afetam o foco de qualidade). Um modelo de folha de abstração é mostrado no quadro 3-1;

- b) definição das questões: de acordo com o GQM, compreende a definição das questões, que devem ser compatíveis com os objetivos previamente definidos para a avaliação. Sugere-se que tais questões sejam formuladas nesta seqüência: primeiro definem-se as questões relativas ao foco de qualidade; depois, as questões referentes aos fatores de variação;
- c) definição das métricas: as métricas aqui definidas visam quantificar as questões formuladas. Cada métrica deve estar relacionada a pelo menos uma questão, e cada questão devem ter pelo menos uma métrica a ela associada.

QUADRO 3-1 - FOLHA DE ABSTRAÇÃO.

Objetivo	Objeto	Propósito	Foco de Qualidade	Ponto de Vista	Ambiente
Foco de Qualidade			Fatores de Variação		
Descreve o foco de qualidade			Quais os fatores têm impacto no foco de qualidade?		
Hipótese Básica			Impacto na Hipótese Básica		
Qual o estímulo é o estado corrente em relação ao foco de qualidade?			Como os fatores de variação influenciam no foco de qualidade?		

Fonte: (Gross, 2001)

No GQM a partir das metas de qualidade estipuladas são elaboradas perguntas que, para serem respondidas, necessitam que sejam realizadas medições.

O método utilizado no GQM envolve:

- a) a realização de um pré-estudo: coleta das informações relevantes para iniciar o processo de GQM (p. ex.: processos alvo para medição);
- b) a identificação das metas a atingir e a elaboração de um plano de GQM: determinar as metas do programa de métricas com base nas metas organizacionais e do projeto onde o GQM será aplicado;
- c) a elaboração de um plano de métricas: a fim de saber, para o desenvolvimento de software, tudo o que deve ser medido;
- d) a coleta de dados: responsabilidade da equipe do projeto, considerando que o programa de métricas deve gerar o mínimo de esforço possível ao dia-a-dia dos projetos;
- e) a análise e interpretação dos dados coletados;
- f) o armazenamento das experiências.

### **3.3 NORMAS E MODELOS DE QUALIDADE**

O principal objetivo da engenharia de software é, sem dúvida, melhorar a qualidade do software. A qualidade de produtos de software, entretanto, está fortemente relacionada à qualidade do processo de software. Para muitos engenheiros de software, a qualidade do processo de software é tão importante quanto a qualidade do produto. Assim, na década de 90 houve uma grande preocupação com a modelagem e melhorias no processo de software (Rocha, 2001). Abordagens importantes como as normas ISO 9000-3 (NBR ISO 9000-3, 1993), a NBR ISO/IEC 12207 (ISO/IEC 12207, 2000) e o SPICE (*Software Process Improvement and Capability dEtermination*) (Emam, 1998) sugerem que melhorando o processo de software, pode-se melhorar a qualidade dos produtos.

Segundo Hoppe (1999), a norma ISO 9000-3 fornece diretrizes para a aplicação da ISO 9001 ao desenvolvimento, fornecimento e manutenção de software.

A norma NBR ISO/IEC 12207 (ISO/IEC 12207, 2000) tem como principal objetivo o estabelecimento de uma estrutura comum para os processos de ciclo de vida de software, para ser utilizada como referência (Hoppe, 1999).



Segundo Emam (1998), a ISO/IEC 15504 (*SPICE*) presta-se à realização de avaliações de processos de software com dois objetivos: a melhoria dos processos e a determinação da capacidade de processos de uma organização. Se o objetivo for a melhoria dos processos, a organização pode realizar a avaliação gerando um perfil dos processos que serão usados para a elaboração de um plano de melhorias. A organização deve definir os objetivos e o contexto, bem como escolher o modelo e o método para a avaliação e definir os objetivos de melhoria.

Na próxima seção são descritos os processos de manutenção segundo as normas de qualidade citadas.

### **3.3.1 MANUTENÇÃO DE SOFTWARE NA NORMA ISO 9000-3**

Segundo Hort (1999), a ISO 9000-3 determina especificamente para a manutenção de software que o fornecedor deve estabelecer e manter procedimentos para realizar atividades e verificar se estas atendem aos requisitos especificados para a manutenção. A mesma deve estar especificada no contrato, após a entrega. Para a ISO 9000-3 a manutenção pode descrever uma:

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

Além disso, a norma ISO 9000-3 determina para a manutenção procedimentos de liberação e um plano funcional para a mesma com base nos itens acima.

Para incorporar alterações ao produto, deve-se seguir alguns procedimentos para liberação:

- a) existência de regras básicas para determinar onde as correções localizadas podem ser incorporadas, ou quando é necessária a liberação de uma cópia completa atualizada do produto de software;
- b) descrever os tipos de liberações, dependendo de sua frequência e/ou impacto sobre as suas operações e a capacidade do comprador de implementar alterações a qualquer momento;
- c) métodos pelos quais o comprador deve ser informado 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 as alterações foram implementadas e em que locais, para os vários produtos e instalações.

### **3.3.2 MANUTENÇÃO DE SOFTWARE NA NORMA NBR ISO/IEC 12207**

Segundo Scussiato (1998), o processo de manutenção 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 ainda 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, Análise do Problema e da Modificação, Implementação da Modificação, Revisão / Aceitação da Manutenção, Migração e Descontinuação do Software (Rocha, 2001).

#### **3.3.2.1 IMPLEMENTAÇÃO DO PROCESSO**

Nesta atividade, deve-se desenvolver, documentar e executar planos e procedimentos para conduzir as atividades e tarefas relativas ao processo de manutenção. Delegar responsabilidades é absolutamente essencial para que o fluxo de atividades de manutenção possa ser melhorado e a confusão reduzida. Deve-se estabelecer procedimentos para receber, armazenar e acompanhar os pedidos de modificação e os relatórios de problemas dos usuários e fornecer um retorno para os mesmos. Uma interface organizacional com o processo de gestão de configuração também deve ser feita para que as modificações realizadas no sistema possam ser gerenciadas.

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

A seqüência de eventos que ocorre como resultado de um pedido de manutenção deve ser iniciada com uma análise do pedido de modificação ou do relatório do problema, para verificar o impacto do mesmo sobre a organização, sobre o sistema existente e sobre os sistemas que fazem interface com o que se deseja modificar. Essa análise deve considerar o tipo de manutenção (corretiva, adaptativa, perfectiva ou preventiva), o alcance da alteração (tamanho da modificação, custo envolvido e tempo para modificar) e as conseqüências da modificação (impacto no desempenho, na proteção ou na segurança do sistema). Após a análise, deve-se desenvolver opções para implementar a modificação. Em certos casos, por exemplo, um erro pode ser tão grave que os controles normais de manutenção devem ser abandonados temporariamente ou um pedido de ampliação pode ser rejeitado devido à estratégia comercial ou à inadequação dos recursos disponíveis. O pedido de modificação ou relatório de problemas, juntamente com os resultados das análises e opções de implementação, deve ser documentado. A opção de modificação selecionada só será efetivamente realizada após a aprovação e considerando-se o especificado no contrato.

### **3.3.2.3 IMPLEMENTAÇÃO DA MODIFICAÇÃO**

Independentemente do tipo de manutenção, as mesmas tarefas técnicas são realizadas. O mantenedor deve, primeiramente, determinar e documentar as unidades, versões e documentações relacionadas do software que precisaram ser modificadas e então implementar a modificação reaplicando as atividades de engenharia de software realizadas para o desenvolvimento. Quando os módulos e estruturas já estiverem modificados, a não ocorrência de erros no software deve ser revalidada. O mantenedor deve demonstrar que não somente a nova lógica está correta, mas também que as partes do software que não foram modificadas permanecem intactas e que o software como um todo ainda funciona corretamente. Isso envolve testar as novas partes e efetuar testes de regressão nas partes já existentes. A revalidação deve assemelhar-se ao processo original de validação do software, usando os mesmos (ou similares) casos e dados de testes usados quando o programa foi originariamente desenvolvido. Como os efeitos colaterais podem acontecer não só no código que está sendo alterado, mas também nos dados e na documentação do software, faz-se necessário também identificar possíveis problemas com a integridade da configuração de software.

### **3.3.2.4 REVISÃO/ACEITAÇÃO DA MANUTENÇÃO**

O mantenedor deve conduzir revisões com a organização para assegurar a integridade do sistema modificado e então obter a concordância da mesma de que a tarefa foi realizada satisfatoriamente e de acordo com o especificado no contrato.

### **3.3.2.5 MIGRAÇÃO**

Se um sistema ou produto de software (incluindo os dados) for migrado de um antigo para um novo ambiente operacional, deve-se assegurar de que qualquer produto de software ou dado produzido ou modificado durante a migração esteja de acordo com o padrão NBR ISO/IEC 12207. Para isso, são necessários o desenvolvimento, a documentação e a execução de um plano de migração. Os itens contidos nesse plano envolvem:

- a) análise e definição dos requisitos de migração;
- b) desenvolvimento de ferramentas de migração;
- c) conversão dos dados e produtos de software;
- d) execução da migração;
- e) verificação da migração;
- f) apoio para o ambiente antigo no futuro.

Como o plano também envolve o usuário, o mesmo deve ser notificado do plano e das atividades de migração. A notificação inclui as características do sistema antigo que não mais serão suportadas, a descrição do novo ambiente e outras características de apoio disponíveis, se houver, quando a característica do sistema antigo for removida. Para que a transição seja mais suave, podem ser conduzidas operações paralelas do antigo e do novo ambiente. Durante esse período deve ser fornecido, de acordo com o contrato, o treinamento necessário para a utilização do novo ambiente. Na época agendada para a migração, todos os envolvidos devem ser notificados e toda a documentação, logs e códigos relacionados ao ambiente antigo devem ser armazenados em arquivos para posteriores consultas. Ao término da operação de migração, devem ser realizadas revisões para avaliar o impacto da mudança para o novo ambiente. Os resultados dessas revisões devem ser enviados às autoridades apropriadas para informação, orientação e ação. Os dados usados ou associados ao ambiente antigo devem estar acessíveis de acordo com os requisitos de contrato para proteção de dados e auditorias aplicáveis aos mesmos.

### 3.3.2.6 DESCONTINUAÇÃO DO SOFTWARE

Essa atividade envolve o desenvolvimento, a documentação e a execução de um plano de descontinuação para remover o apoio ativo dado ao sistema pelas organizações de operação e manutenção. Os itens contidos no plano envolvem:

- a) interrupção total ou parcial do apoio depois de um determinado período de tempo;
- b) arquivamento do produto de software e da documentação associada;
- c) responsabilidade por quaisquer questões residuais futuras de apoio;
- d) transição para o novo produto de software, se aplicável;
- e) acessibilidade às cópias de dados arquivadas.

Como o plano de descontinuação também envolve o usuário, o mesmo deve ser notificado do plano e das atividades de descontinuação. A notificação inclui a descrição da substituição ou do upgrade com seus dados de disponibilidade, explicações quanto a por que o produto de software não será mais apoiado e a descrição de outras opções de apoio disponíveis, uma vez que o apoio foi removido. Para que haja uma descontinuação mais suave, podem ser conduzidas operações paralelas do antigo e do novo ambiente. Durante esse período deve ser fornecido, de acordo com o contrato, o treinamento necessário para a utilização do novo ambiente. Na época agendada para a descontinuação todos os envolvidos devem ser notificados e toda a documentação, *logs* e códigos relacionados ao ambiente antigo devem ser armazenados em arquivos para posteriores consultas. Os dados usados ou associados ao produto de software descontinuado devem estar acessíveis de acordo com os requisitos de contrato para a proteção de dados e auditorias aplicáveis aos mesmos.

Uma parte fundamental inicia ou executa o desenvolvimento, a operação ou a manutenção de produtos de software. Como a operação do produto de software está integrada à operação do sistema, as atividades e tarefas desse processo se referem ao sistema (Rocha, 2001).

### 3.3.3 MANUTENÇÃO DE SOFTWARE SEGUNDO O SPICE

O propósito da manutenção do software segundo o item da categoria de processo engenharia ENG.7 (manutenção) é modificar o sistema, seu hardware, o sistema de rede,

software, e documentação associada em resposta ao usuário para preservar a integridade e o conceito do sistema (Emam, 1998).

Há várias fontes que criam a necessidade por modificar o sistema ou software:

- a) descoberta de erros;
- b) deficiências;
- c) problemas na operação do sistema ou software;
- d) melhoria particular ou modificação do sistema ou requisito do software ou pedido do cliente (externo ou interno).

Os processos que envolvem o ENG.7 do SPICE podem ser divididos em:

- a) determinar requisitos de manutenção (elementos e modificações necessárias) (ENG.7.1) – neste processo devem ser determinados o sistema/software e elementos que precisam ser mantidos e os custos exigidos;
- b) analisar problemas e requisições do usuário (avaliar impactos) (ENG.7.2) – neste processo devem ser analisados os problemas de usuário e custos. Analisar os problemas de usuário e pedidos e os esforços/custos exigidos, avaliando o possível impacto de opções diferentes por modificar o sistema operacional e software, interfaces de sistema, e exigências;
- c) determinar modificações para próximo upgrade (ENG.7.3) – neste processo deve ser determinado com base nas análises das modificações, quais deveriam ser aplicadas no próximo sistema ou versão aperfeiçoada de software e deveriam ser documentadas que unidades de software e outros elementos de sistemas e qual documentação precisarão ser mudadas e quais testes precisarão ser executados;
- d) implementar e testar modificações (ENG.7.4) – neste processo deve-se utilizar os outros processos de engenharia, como destinar, implementar e testar as modificações selecionadas e demonstrar que as funções não alteradas no sistema não foram danificadas pela versão aperfeiçoada;
- e) fazer *upgrade* do sistema (ENG.7.5) – neste processo deve-se migrar o sistema melhorado e software com as modificações para o ambiente do usuário, provendo para a mesma a operação paralela do anterior e do novo upgrade, treinamento de usuário adicional, opções de suporte e descontinuação do sistema anterior.

### 3.3.4 BREVE COMPARATIVO DOS PROCESSOS DE MANUTENÇÃO SEGUNDO AS NORMAS DE QUALIDADE

Para facilitar o entendimento dos processos de manutenção segundo as normas NBR ISO/IEC 12207, ISO/IEC 15504 (SPICE) e ISO 9000-3, com base no comparativo elaborado por Hoppe (1999) mostram-se o quadro 3-2 contendo o que determina cada norma referente ao ciclo de processo da manutenção.

A norma NBR ISO/IEC 12207 foi escolhida como parâmetro de comparação por possuir um maior detalhamento de cada atividade do processo de manutenção.

QUADRO 3-2 – COMPARATIVO DO PROCESSO DE MANUTENÇÃO.

<b>NORMAS E MODELOS DE QUALIDADE</b>			
<b>ASPECTOS</b>	ISO/IEC 12207	ISO 9000-3	SPICE
<b>Tipos de manutenção</b>	Análise do problema da manutenção	Tipos de atividades de manutenção	Definir a estratégia...
<b>Tarefas da manutenção</b>			
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
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
<b>Tarefas técnicas da manutenção</b>			
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 melhoria em <i>releases</i> individuais	Análise do problema e da modificação	Registro e relatório de manutenção	Desenvolver uma estratégia...
Analisar o <i>release</i> de solicitação de manutenção do sistema	Implementação da modificação	Tipos de atividades de manutenção	Testes estratégicos
Projetar o <i>release</i> de manutenção do sistema	Implementação da modificação	Tipos de atividades de manutenção	Desenvolver a modificação
Codificar e testar o <i>release</i> de manutenção do sistema	Implementação da modificação	Tipos de atividades de manutenção	Desenvolver a modificação
Implementar o <i>release</i> de manutenção do sistema	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

Conforme NBR ISO/IEC 12207 (ISO/IEC 12207, 2000), 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 3.2 apenas é detalhado qual item do SPICE, atende ao requisito de manutenção exposto.

Utilizando como base Hoppe (1999), segue uma breve descrição do resultado do comparativo entre a teoria de manutenção de sistemas e as normas de qualidade:

- a) 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;



- b) organização para manutenção: A partir desta etapa que toda a manutenção toma forma, se a organização for falha com toda certeza num futuro não muito distante esta terá problemas em administrar e manter seus softwares;
- c) 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;
- d) 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;
- e) conservação dos registros: Esta etapa é a primeira da 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, deve-se ter técnicas para saber quais os dados importantes e não criar ambigüidades, porém não é nenhum crime armazenar algum dado que não é de extrema importância para o funcionamento do software;
- f) 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;

- g) avaliar a solicitação de melhoramento do sistema: Toda solicitação deverá ser 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;
- h) 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 é também levado muito em consideração pelas normas;
- i) aplicar conserto de emergência no programa: Em um conserto de emergência muitos 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 se deve 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;
- j) organizar as solicitações de melhoramento em *releases* individuais: Para que se possa ter um resultado e um controle melhor sobre as manutenções 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 tirá-los;
- k) 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;
- l) 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;

- m) 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 a necessidade exposta;
- n) 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;
- o) 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;
- p) treinar a equipe: Outra tarefa que é negligenciada pelas normas é o treinamento, porém acredita-se que esta tarefa possa ser de muito grande benfeitoria para a empresa, pois capacita as pessoas para a utilização do software;
- q) 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;
- r) 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 DESENVOLVIMENTO DO TRABALHO

Neste capítulo serão descritas as atividades desempenhadas para a construção de um modelo de avaliação para o processo de manutenção no desenvolvimento de software. Também serão relacionados e discutidos os resultados obtidos a partir do mesmo. As atividades descritas a seguir definem como foram realizadas a análise dos requisitos, especificação e implementação do protótipo proposto.

### 4.1 ANÁLISE DOS REQUISITOS

Nesta seção serão apresentados os principais requisitos do trabalho, através de uma contextualização do cenário que originou seu desenvolvimento, bem como as funcionalidades propostas e características gerais.

O presente trabalho caracteriza-se na montagem de um conjunto de métricas de avaliação do processo de manutenção, o qual poderá ser apoiado pelo protótipo para automatizar o processo de avaliação da manutenção utilizando como base o processo de manutenção da Empresa Mult Sistemas.

A Mult Sistemas é uma empresa do ramo de desenvolvimento de sistemas de informação, voltada principalmente à área de gestão empresarial e aplicativos comerciais. Sediada em Santa Catarina, na cidade de Blumenau, a empresa conta atualmente com 45 colaboradores que atendem a mais de 50 revendedores distribuídos em 20 estados e uma carteira de aproximadamente 2000 empresas clientes em todo o Brasil. Fundada em 1990, a empresa encontra-se consolidada no mercado nacional e é uma das 5 maiores produtoras de sistemas de gestão empresarial de Blumenau.

A metodologia de desenvolvimento de software adotada pela empresa é a análise e projeto orientados a objeto. Atualmente, a linguagem de programação utilizada para codificação de seus produtos é o *Object Pascal*.

Identificou-se a necessidade da definição de um conjunto de métricas quando se necessita avaliar o processo de manutenção com requisitos pré-definidos, isto é, melhorar a qualidade dos processos baseando-se nas metas da organização para os mesmos.

Com base na abordagem GQM foram definidas metas de qualidade. A partir das metas, foram definidas questões/métricas para determinar a medida de atingimento das metas e em que classificação está enquadrada a organização através do resultado da avaliação das métricas.

A fase de análise de requisitos foi realizada através de reuniões com profissionais de diversas áreas ligadas ao assunto, como relatado mais adiante nesta mesma seção.

Através do detalhamento da abordagem GQM e das normas NBR ISO/IEC 12207 e ISO/IEC 15504 tornou-se possível o desenvolvimento do protótipo para avaliação do processo de manutenção de software, conforme definido no objetivo do trabalho.

Esta seção abordará a forma em que o plano GQM foi montado, a participação dos envolvidos e o plano GQM dividido por objetivo.

#### **4.1.1 DESENVOLVIMENTO DO PLANO GQM**

Para que se pudesse ter o resultado mais próximo possível da realidade atual em relação aos processos de manutenção de uma produtora de software, utilizou-se para a montagem do plano GQM, que consistiu da montagem dos objetivos, questões e métricas e a coleta de dados baseada nas experiências de profissionais envolvidos em um processo de manutenção. Realizaram-se reuniões informais com profissionais de áreas específicas, selecionadas com objetivo de explorar o conhecimento destes profissionais e extrair um plano bastante depurado. Os profissionais escolhidos foram os seguintes:

- a) gerente de projetos – o profissional desta área possui visão para o controle, ele sempre está acompanhando o desempenho de uma equipe de desenvolvimento, custos e tecnologias envolvidas, pois ele participa da definição dos projetos de desenvolvimento e está sempre em contato com outros profissionais desta área. Este profissional possui uma visão gerencial das fases do projeto, facilitando o entendimento de alguns pontos fundamentais do plano desenvolvido e acompanha o processo como um todo através de uma visão gerencial;

- b) analista de sistemas – este profissional tem a visão voltada para a parte técnica, para detalhes fundamentais relacionados ao desenvolvimento, prototipação e estruturação do software. Este profissional contribui muito no que tange aos conceitos corretos do sistema, pois além da visão técnica que é fundamental para sua área, ele possui visão de mercado, o que possibilita o entendimento para desenvolvimento das rotinas selecionadas e acompanha a fase de especificação e análise de modificações realizadas no processo através de uma visão técnica;
- c) consultor técnico – o profissional de consultoria possui uma espécie de conhecimento geral, pois deve possuir a sensibilidade de entender o usuário e as limitações da parte técnica também. É com este fator que este profissional contribui, pois tanto o profissional de análise como o de gerência de projeto são voltados com mais intensidade para suas áreas, faltando uma espécie de meio campo capaz de entender ambos os lados. Este profissional ainda contribui muitas vezes com sua experiência em várias soluções, o que o habilita com visões diferentes para um mesmo assunto e acompanha a fase de identificação das necessidades do usuário em relação ao produto final (em manutenção), geralmente estabelece os problemas/não-conformidades;

#### **4.1.2 COLETA DE DADOS**

Para a coleta de dados, definiu-se uma entrevista com cada profissional envolvido que tinha como objetivo explicar o trabalho e fazer o desenvolvimento do plano GQM.

Realizaram-se entrevistas com duração média de uma hora para cada profissional que foi caracterizada pelo grande número de discussões em relação a cada objetivo e questão. Foi investido em média para a montagem das métricas cerca de oito horas no total, já que se necessitava definir para cada questão, ao menos uma métrica.

Para facilitar a definição dos objetivos, questões e métricas para avaliação do processo de manutenção, utilizou-se como base a folha de abstração. Na montagem da folha de abstração procurou-se buscar as fases do processo de manutenção sugeridas pelo ISO/IEC 12207 citadas em 3.5.2 e também pelo SPICE em 3.5.3, além de utilizar como base o processo de manutenção da Mult Sistemas.

Os objetivos foram distribuídos em três grandes grupos. A seguir têm-se comentários sobre cada um deles, bem como uma visão das perguntas e métricas que se criou em função de cada objetivo. No caso da Mult Sistemas, as informações são extraídas através da base de dados do Sistec – Sistema de Controle Técnico (informações relativas ao processo de manutenção), através de *queries* e relatórios e são digitadas as respostas das métricas no protótipo avaliador.

- a) G1 – Aderência a norma NBR ISO/IEC 12207: criou-se esta meta com intuito de avaliar o processo de manutenção segundo uma norma de qualidade. Isto é, se pretende traçar um paralelo entre as atividades de manutenção prevista na norma e os processos em uso da produtora de software. Este objetivo está exemplificado e descrito no quadro 4-1 - Meta G1;

QUADRO 4-1 – META G1

<b>Folha de Abstração</b>	
<b>Meta:</b> <b>Objetivo:</b> <b>Objeto:</b> <b>Enfoque de qualidade:</b> <b>Ponto de vista:</b> <b>Contexto:</b>	<b>G1</b> <b>Caracterizar</b> <b>O processo de manutenção de software</b> <b>Com respeito à aderência a NBR ISO/IEC 12207</b> <b>Do ponto de vista do gerente de software</b> <b>No contexto da organização</b>
<b>Enfoque de qualidade</b>	<b>Fatores de variação</b>
- qtde. de pedidos de manutenção em conformidade; - qtde. de problemas rastreados; - qtde. de modificações de acordo com o contrato; - qtde. dos testes realizados documentados pelo mantenedor em relação ao total de alterações; - qtde. das revisões realizadas com os clientes;	- desvio padrão da análise de problemas/modificações; - qualidade das notificações referentes ao software aos clientes;
<b>Hipótese de base</b>	<b>Impacto na hipótese de base</b>

<ul style="list-style-type: none"> <li>- médio nível de detalhamento dos pedidos de manutenção;</li> <li>- tem forma de rastrear os problemas;</li> <li>- baixo grau de problemas rastreados;</li> <li>- baixo nível de análises acima do limite de tempo;</li> <li>- baixo grau de modificações de acordo com as especificações do contrato;</li> <li>- médio nível de testes realizados pelo mantenedor;</li> <li>- baixo nível de revisões realizadas com os clientes;</li> </ul>		<ul style="list-style-type: none"> <li>- quanto maior o nível de detalhamento dos pedidos de manutenção, maior o grau de problemas rastreados;</li> <li>- quanto maior o rastreamento dos problemas, menor o tempo de análise do mesmo;</li> <li>- quanto maior o grau de modificações de acordo com o contrato, menor o nível de revisões realizadas com os clientes;</li> </ul>
Questão	Q11	Os dados de um pedido de manutenção no software são detalhados de forma clara e concisa?
Métricas	M1	Qtde. de pedidos devolvidos pela manutenção por falta de informação nos pedidos de manutenção (mês)
Questão	Q12	É realizada a rastreabilidade das <i>releases</i> do software, para descobrir quando o problema foi originado?
Métricas	M2	Qtde. de pedidos que foram rastreados (mês)
Questão	Q13	Qual o tempo utilizado para realizar a análise do problema / modificação a ser passado para a codificação?
Métricas	M3	Média de tempo da análise
	M4	Desvio Padrão
	M5	Qtde. de casos acima do limite superior (20 dias)
Questão	Q14	As modificações realizadas no software consideram as especificações do contrato de manutenção com o cliente?
	M6	Qtde. de modificações aperfeiçoadoras de acordo com o contrato de manutenção (mês)
	M7	% de exceções identificadas (qtde. exc. / qtde. total)
Questão	Q15	O mantenedor após a codificação da alteração realiza testes na nova lógica implementada e nas partes não modificadas para que o software como um todo ainda funcione corretamente?
	M8	Qtde. de testes realizados documentados em relação ao total de alterações (%)
Questão	Q16	São realizadas revisões com os clientes contratados para a garantia da qualidade das tarefas realizadas em um determinada <i>release</i> ?
	M9	Qtde. de alterações revisadas em conjunto com o cliente em liberação de <i>releases</i> betas (mês)
Questão	Q17	Os usuários (clientes) são notificados Quando da necessidade de migração para um novo ambiente operacional?
	M10	Qtde. de usuários (clientes) notificados quando da migração em relação ao total da organização (%)
Questão	Q18	Existe uma política de descontinuação do software e sua conseqüente



		substituição ou upgrade, de forma que os dados antigos fiquem armazenados para realização de auditorias?
	M11	Qtde. de clientes que tiveram problemas de perda de dados em migrações por não possuírem backup's de suas informações

- b) G2 – Procedimentos e resultados: criou-se esta meta com intuito de avaliar o processo de manutenção segundo os procedimentos e resultados obtidos dos mesmos, conforme descrito no quadro 4-2 - Meta G2;

QUADRO 4-2 – META G2

<b>Folha de Abstração</b>		
<b>Meta:</b>		<b>G2</b>
<b>Objetivo:</b>		<b>Avaliar e melhorar</b>
<b>Objeto:</b>		<b>O processo de manutenção de software</b>
<b>Enfoque de qualidade:</b>		<b>Com respeito aos procedimentos e resultados obtidos</b>
<b>Ponto de vista:</b>		<b>Do ponto de vista do gerente de software</b>
<b>Contexto:</b>		<b>No contexto da organização</b>
<b>Enfoque de qualidade</b>		<b>Fatores de variação</b>
- nível de cobertura dos procedimentos em relação a entrada das solicitações;		- documentações realizadas
<b>Hipótese de base</b>		<b>Impacto na hipótese de base</b>
- alto nível de cobertura dos procedimentos de manutenção; - fácil visualização dos resultados;		- quanto maior a cobertura dos procedimentos, menor o retrabalho da equipe de manutenção.
Questão	Q21	Existem procedimentos para registro das solicitações de problemas / não-conformidades detectadas?
Métricas	M12	% de solicitações registradas na ferramenta em relação ao total de solicitações (mês)
Questão	Q22	Existem procedimentos bem definidos para avaliar as solicitações de melhoramento do sistema sem ferir o aspecto estrutural e a finalidade do software em questão?
Métricas	M13	% de solicitações de melhoramentos analisadas pelo analista de sistemas em relação ao total (mês)
Questão	Q23	Existem registros históricos de ações corretivas realizadas no software para avaliar solicitações de problemas?
Métricas	M14	Qtde. de ações corretivas documentadas (Mês)
Questão	Q24	A organização possui procedimentos padrões que garantam a qualidade das ações corretivas de emergência no software?
	M15	% ações corretivas testadas para liberação (Qtde./mês) em relação ao

		total liberado (Qtde./mês)
Questão	Q25	Existe um plano de liberação de <i>releases</i> para correção de problemas / não-conformidades no software?
	M16	% satisfação da equipe de manutenção em relação ao plano
Questão	Q26	Qual a quantidade de ocorrências de manutenção corretivas por mês?
	M17	Qtde. de manutenções corretivas (mês)
Questão	Q27	Qual a quantidade de ocorrências de melhoramentos no software por mês?
	M18	Qtde. de manutenções aperfeiçoadoras (mês)
Questão	Q28	Qual a causa (origem) dos erros encontrados no software?
	M19	% de erros causados na codificação do software (Mês)
	M20	% de erros causados pela especificação do software (Mês)
	M21	% de erros causados em manutenções anteriores (Mês)
	M22	% de erros causados por requerimentos do usuário (Mês)
Questão	Q29	Qual a quantidade de verificações periódicas de validação (testes) gerais no software são efetuadas?
	M23	% de testes gerais realizados (mês) em relação ao nr. Manutenções

- c) G3 – Produtividade: criou-se esta meta com intuito de avaliar a o processo de manutenção segundo a produtividade do processo de manutenção, conforme descrito no quadro 4-3 -Meta G3;

QUADRO 4-3 – META G3

<b>Folha de Abstração</b>	
<b>Meta:</b>	<b>G3</b>
<b>Objeto:</b>	<b>Analisar</b>
<b>Objetivo:</b>	<b>O processo de manutenção de software</b>
<b>Enfoque de qualidade:</b>	<b>Com respeito a produtividade do processo</b>
<b>Ponto de vista:</b>	<b>Do ponto de vista do gerente de software</b>
<b>Contexto:</b>	<b>No contexto da organização</b>
<b>Enfoque de qualidade</b>	<b>Fatores de variação</b>

<ul style="list-style-type: none"> <li>- nível de erro nas estimativas para tarefas de manutenção;</li> <li>- experiência da equipe de manutenção;</li> <li>- esforço utilizado pela equipe de manutenção em manutenções corretivas e melhoramentos;</li> </ul>		<ul style="list-style-type: none"> <li>- desvio padrão em estimativas</li> <li>- nível da equipe de manutenção</li> </ul>
<b>Hipótese de base</b>		<b>Impacto na hipótese de base</b>
<ul style="list-style-type: none"> <li>- médio nível de erros em estimativas;</li> <li>- pouca experiência da equipe de manutenção;</li> <li>- grande esforço utilizado em tarefas corretivas;</li> </ul>		<ul style="list-style-type: none"> <li>- quanto menor a experiência da equipe, menor a qualidade das manutenções efetuadas.</li> </ul>
Questão	Q31	Qual a variação média de erro nas estimativas de horas em uma tarefa de manutenção?
Métricas	M24	% de erros de estimativas na realização de manutenções
Questão	Q32	Qual o tempo médio para realização das tarefas de manutenção?
Métricas	M25	Tempo médio das solicitações realizadas pela manutenção
Questão	Q33	Qual o conhecimento da equipe de manutenção em relação a funcionalidade do software a ser mantido?
Métricas	M26	Grau de conhecimento da equipe de manutenção em relação ao negócio que o software trata
	M27	Grau de conhecimento das ferramentas e utilitários do software a ser mantido
Questão	Q34	Qual o esforço utilizado (horas/mês) para as manutenções corretivas?
	M28	Qtde. de horas/mês utilizadas em manutenções corretivas
Questão	Q35	Qual o esforço utilizado (horas/mês) para melhoramentos no software?
	M29	Qtde. de horas/mês utilizadas em melhoramentos do software
Questão	Q36	Qual a influência de uma equipe de validação no resultado das manutenções corretivas efetuadas?
	M30	% de erros retornados a codificação pela equipe de validação em relação ao nr. Solicitações realizadas
Questão	Q37	Qual a influência dos registros (documentação/históricos) efetuados das manutenções para a eficiência de manutenções futuras?
	M31	Qtde. de manutenções que após serem feitas retornaram para nova correção (Mês)

## 4.2 ESPECIFICAÇÃO DO SOFTWARE

Este tópico caracteriza-se por especificar um protótipo capaz de apoiar as métricas definidas para a avaliação da qualidade dos processos de manutenção existentes em

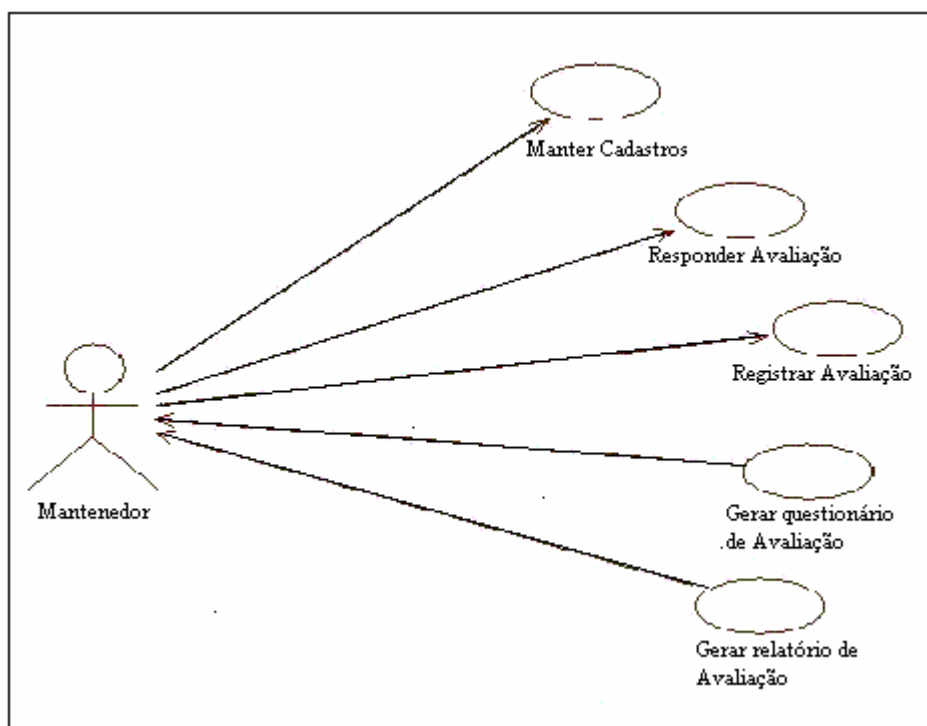
produtoras de software. Identificou-se a necessidade deste protótipo quando se necessita comparar e avaliar diversos processos de manutenção com requisitos pré-definidos. Portanto, desde o início de sua construção o protótipo objetivou tornar-se efetivamente um produto de auxílio ao processo de avaliação, através do qual podemos automatizar e conservar os registros destas avaliações.

Na fase inicial de especificação do protótipo definiram-se quais as ferramentas seriam utilizadas para a construção dos diagramas, modelos e estruturas de dados que constituem na representação lógica do trabalho. Para a montagem do *Use-Case* utilizou-se a ferramenta CASE Rational Rose, produzida pela *Rational Rose Corporation*. Utilizou-se para a especificação do protótipo ( DER e Dicionários de Dados) a ferramenta ERwin que é uma ferramenta CASE de modelagem de dados, produzida pela empresa *Computer Associates (CA)*, que suporta a manutenção de bancos de dados em vários ambientes. O ERwin permite a elaboração do DERs lógico e físico, suportando as notações IDEF1X (*Integration DEFINition for Information Modeling*), IE (*Information Engineering*) e DM (*Dimensional Modeling*), sendo a última aplicada somente à visão lógica do diagrama. A ferramenta Erwin permite a geração automática do dicionário de dados a partir dos atributos definidos no DER e permite gerar relatórios do mesmo.

#### **4.2.1 DIAGRAMA DE CASOS DE USO**

O caso de uso identifica e esquematiza graficamente e descreve o comportamento do sistema. Estes diagramas apresentam uma visão geral nivelada de como o sistema é usado, na perspectiva de um ator. Este diagrama pode descrever alguns ou todos os casos de uso de um sistema. Na fig. 4-1 estão representados os casos de uso implementado no protótipo do trabalho.

FIGURA 4-1 – DIAGRAMA DE CASOS DE USO



O principal ator é referenciado como mantenedor, ele é que fará as entradas e receberá os resultados em forma de relatório. Para que seja possível atender todas as necessidades do usuário, foram criados os seguintes casos de uso:

- a) Manter cadastros: este caso trata das informações cadastrais do aplicativo, através destes cadastros será possível partir para os próximos passos do aplicativo. Os cadastros utilizados para este caso são: Organização, Projeto, Meta, Questão, Métrica, Classificação e Regra.
- b) Responder avaliação: neste caso é feita a movimentação da avaliação propriamente dita, aqui é feito o questionamento ao ator sobre o processo a ser avaliado no caso o de manutenção. Neste ponto são utilizadas todas as tabelas cadastrais citadas no item manter cadastros, além da criação de uma nova tabela utilizada para armazenagem de dados de movimentação da avaliação.
- c) Registrar avaliação: este caso trata da entrada inicial para a avaliação de software, nela será informado o código da avaliação, código da organização, código do projeto e data em que se realizou a avaliação, esta tabela também armazena informações relativas à soma de pontos da respectiva avaliação e classificação

obtida pela avaliação.

- d) Gerar questionário de avaliação: este caso de uso utilizará somente as tabelas cadastrais, metas, questões e métricas e tem por objetivo fornecer a impressão do modelo de avaliação proposto para outras empresas ou outra pessoa interessada em fazer a avaliação para que depois sejam inseridos os dados no protótipo.
- e) Gerar relatório da avaliação: caso de uso que gera o relatório da avaliação do processo de manutenção com os resultados obtidos pela organização e como a mesma está classificada.

## 4.2.2 DIAGRAMA ENTIDADE RELACIONAMENTO

O diagrama entidade relacionamento (DER), enfatiza os principais objetivos ou entidades do sistema. Detalha as associações existentes entre as entidades de dados utilizando componentes semânticos próprios. A ferramenta CASE CA ErWin armazena as definições de seus diagramas entidade x relacionamento de duas formas: em arquivos binários com formato não divulgado pelo fabricante, ou através de um mecanismo que transfere as definições para um banco de dados. A ferramenta ErWin possibilita a definição da metodologia a ser utilizada nos diagramas, tanto para notação física quanto para a notação lógica, sendo dois tipos: *Integration Definition for Information Modeling (IDEF1X)* e *Information Engineering (IE)*. Na notação IDEF1X, os relacionamentos N para N são representados por uma linha sólida com pontos em ambos os fins, por exemplo. Na fig. 4-2 pode ser vista a cardinalidade usada pela notação IDEF1X. Maiores informações sobre a notação podem ser vistas em Logic (1997).

FIGURA 4-2 – CARDINALIDADE DA NOTAÇÃO IDEF1X

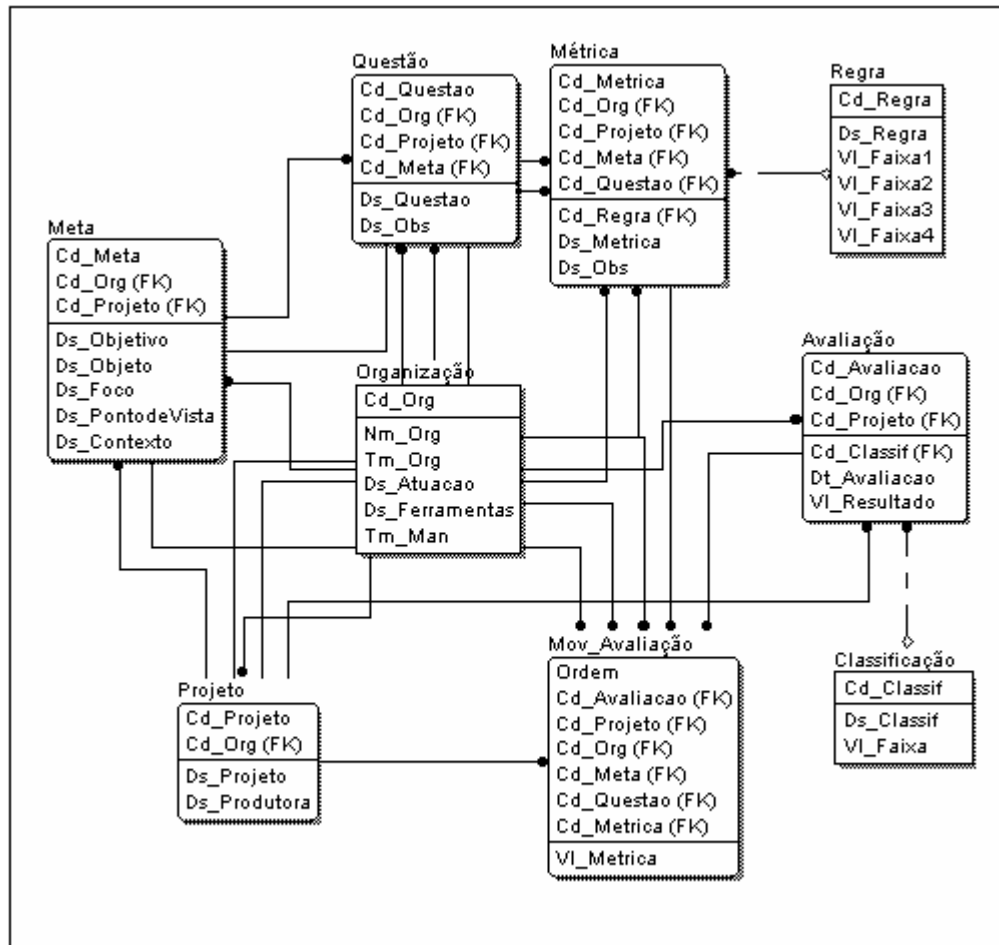
Cardinalidade	Notação IDEF1X		Notação IDE	
	Identificado	N-Identif.	Identificado	N-Ident.if.
Um para zero, um, ou mais				
Um para um ou mais				
Um para zero ou um				
Zero ou um para zero, um, ou mais (somente não-identificado)				
Zero ou um para zero ou um (somente não-identificado)				

Fonte: (Logic, 1997, p. 48)

O Diagrama Entidade Relacionamento é apresentado na fig. 4-3, seu modelo lógico e físico foram gerados a partir da ferramenta ErWin. Como se pode observar, todas as entidades possuem obrigatoriedade com relação à entidade principal, permitindo um relacionamento de 1 para N ou 1 para 1, ou seja, torna-se obrigatório existir um registro pai para que se cadastre um ou mais registros na entidade filho.

A avaliação é realizada sobre projetos ligados a alguma produtora de software (organização), desta forma, é necessário que haja uma organização cadastrada para que se informe qual projeto será avaliado. Outra obrigatoriedade é o cadastramento completo do plano GQM, que constitui das metas, questões e métricas, que ficam disponibilizadas para qualquer avaliação que se deseja fazer. Com relação às métricas, ainda existe uma ligação com uma regra, que será relacionada para cada métrica, para que seja possível enquadrar a resposta dada.

FIGURA 4-3 - DIAGRAMA ENTIDADE RELACIONAMENTO.



A configuração da avaliação constitui na seleção das metas, questões e métricas que se deseja aplicar para aquela avaliação, desta forma torna-se obrigatório informar uma organização e um projeto a ser avaliado, bem como as metas e questões. Para este movimento fica armazenada a resposta da métrica na estrutura da avaliação.

Por fim, na própria avaliação será montado o resultado da avaliação, sendo que para cada avaliação estará associado um resultado ou classificação.

### 4.2.3 DICIONÁRIO DE DADOS

O dicionário de dados contém a definição de todos os dados mencionados no DER, as entidades e seus atributos, incluindo detalhes do formato físico, como: tipo, tamanho, descrição do atributo, se o atributo é chave primária (PK-Primary Key) e se o atributo é chave



estrangeira (FK-*Foreign Key*). Gerou-se um relatório na ferramenta CASE ErWin, utilizada para o desenvolvimento da modelagem. A seguir no quadro 4-4 estão descritos as tabelas de cada entidade com seus respectivos atributos.

**Quadro 4-4 – Dicionário de Dados**

<b>Dicionário de Dados</b>				
<b>Entidade: Avaliação</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Avaliação	Cd_Avaliacao	Number	Yes	No
Código da Classificação	Cd_Classif	Number	No	Yes
Código da Organização	Cd_Org	Alpha(15)	Yes	Yes
Código do Projeto	Cd_Projeto	Alpha(15)	Yes	Yes
Data da Avaliação	Dt_Avaliacao	Date	No	No
Valor do Resultado	Vl_Resultado	Number	No	No
<b>Entidade: Classificação</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Classificação	Cd_Classif	Number	Yes	No
Descrição da Classificação	Ds_Classif	Alpha(50)	No	No
Valor da Faixa	Vl_Faixa	Number	No	No
<b>Entidade: Meta</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Meta	Cd_Meta	Number	Yes	No
Código da Organização	Cd_Org	Alpha(15)	Yes	Yes
Código do Projeto	Cd_Projeto	Alpha(15)	Yes	Yes
Descrição do Contexto	Ds_Contexto	Alpha(80)	No	No
Descrição do Foco	Ds_Foco	Alpha(80)	No	No
Descrição do Objetivo	Ds_Objetivo	Alpha(80)	No	No
Descrição do Objeto	Ds_Objeto	Alpha(80)	No	No
Descrição do Ponto de Vista	Ds_PontodeVista	Alpha(80)	No	No

<b>Entidade: Mov_Avaliação</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Avaliação	Cd_Avaliacao	Number	Yes	Yes
Código da Meta	Cd_Meta	Number	Yes	Yes
Código da Métrica	Cd_Metrica	Number	Yes	Yes
Código da Organização	Cd_Org	Alpha(15)	Yes	Yes
Código da Questão	Cd_Questao	Number	Yes	Yes
Código do Projeto	Cd_Projeto	Alpha(15)	Yes	Yes
Ordem do Lançamento	Ordem	Autoincrement	Yes	No
Valor da Métrica	Vl_Metrica	Number	No	No
<b>Entidade: Métrica</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Meta	Cd_Meta	Number	Yes	Yes
Código da Métrica	Cd_Metrica	Number	Yes	No
Código da Organização	Cd_Org	Alpha(15)	Yes	Yes
Código da Questão	Cd_Questao	Number	Yes	Yes
Código da Regra	Cd_Regra	Alpha(18)	No	Yes
Código do Projeto	Cd_Projeto	Alpha(15)	Yes	Yes
Descrição da Métrica	Ds_Metrica	Alpha(200)	No	No
Observação da Métrica	Ds_Obs	Alpha(200)	No	No
<b>Entidade: Organização</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Organização	Cd_Org	Alpha(15)	Yes	No
Descrição da Área de Atuação	Ds_Atuoacao	Alpha(50)	No	No
Descrição Ferramentas Utilizadas	Ds_Ferramentas	Alpha(50)	No	No
Nome da Organização	Nm_Org	Alpha(50)	No	No
Nr. Funcionários da Organização	Tm_Org	Number	No	No

Nr. Funcionários Manutenção	Tm_Man	Number	No	No
<b>Entidade: Projeto</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Organização	Cd_Org	Alpha(15)	Yes	Yes
Código do Projeto	Cd_Projeto	Alpha(15)	Yes	No
Descrição da Produtora	Ds_Produtora	Alpha(50)	No	No
Descrição do Projeto	Ds_Projeto	Alpha(80)	No	No
<b>Entidade: Questão</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Meta	Cd_Meta	Number	Yes	Yes
Código da Organização	Cd_Org	Alpha(15)	Yes	Yes
Código da Questão	Cd_Questao	Number	Yes	No
Código do Projeto	Cd_Projeto	Alpha(15)	Yes	Yes
Descrição da Questão	Ds_Questao	Alpha(200)	No	No
Observação da Questão	Ds_Obs	Alpha(200)	No	No
<b>Entidade: Regra</b>				
<b>Descrição do Atributo</b>	<b>Atributo</b>	<b>Tipo</b>	<b>PK</b>	<b>FK</b>
Código da Regra	Cd_Regra	Number	Yes	No
Descrição da Regra	Ds_Regra	Alpha(50)	No	No
Valor Faixa 1	Vl_Faixa1	Number	No	No
Valor Faixa 2	Vl_Faixa2	Number	No	No
Valor Faixa 3	Vl_Faixa3	Number	No	No
Valor Faixa 4	Vl_Faixa4	Number	No	No

## 4.3 IMPLEMENTAÇÃO

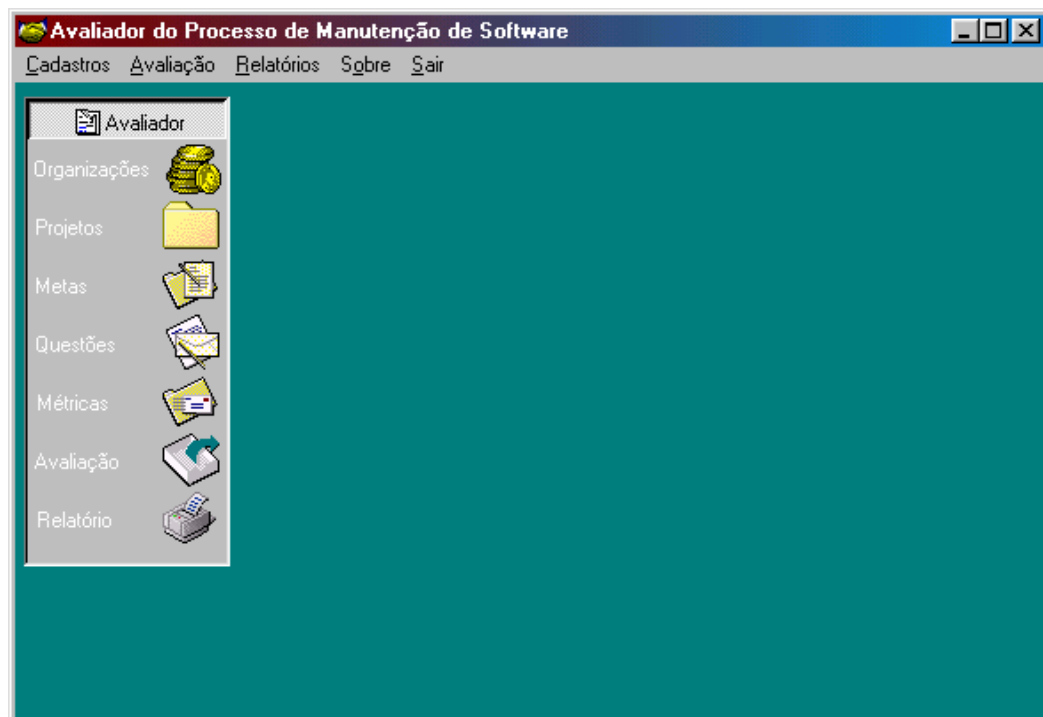
A seguir são apresentadas passo a passo as principais telas disponíveis para utilização do protótipo. Com intuito de facilitar a demonstração e compreensão, será realizada uma avaliação utilizando informações referentes ao processo de manutenção da empresa Mult Sistemas.

O processo de manutenção a cada dia vai se tornando responsável por uma fatia percentualmente maior no processo de uma produtora de software justamente pela criação de novas necessidades ou pela otimização de aplicativos já existentes. Este é um dos motivos que motivaram a criação do protótipo que será apresentado a seguir, como forma de demonstrar a avaliação deste tipo de processo.

### 4.3.1 MENU PRINCIPAL

Ao executar o aplicativo, será apresentada a tela principal do protótipo, disponibilizando acesso aos demais recursos do protótipo, conforme fig. 4-4.

FIGURA 4-4 – MENU PRINCIPAL DO PROTÓTIPO.



O primeiro passo para se utilizar o sistema, é efetuar o cadastramento das tabelas. Para o mesmo deverá acessar o primeiro item do menu, chamado “Cadastros”. Nele estão às telas de manutenção dos cadastros do sistema: organizações, projetos, metas, questões, métricas, classificações e regras. É nessa opção que o usuário irá armazenar os dados relevantes para a utilização da avaliação do processo de manutenção.

Após o cadastramento das tabelas, para que o usuário possa trabalhar com a movimentação de avaliações, deverá ser acessado o segundo item de menu, “Avaliação”, disponibiliza-se a avaliação propriamente dita, nele será configurada a avaliação e o usuário entrará com respostas solicitadas pelo protótipo.

Para a emissão de relatórios, o usuário encontra no terceiro item de menu, “Relatórios”, as principais saídas do protótipo, que são: Relatório da avaliação e questionário.

No quarto item de menu, disponibiliza-se a opção “Sobre”, onde há um breve descritivo sobre o objetivo do protótipo.

No quinto item de menu, disponibiliza-se a opção “Sair”, para efetuar o término das operações do aplicativo. Cada um desses itens será mostrado em detalhes mais adiante.

Quando se deseja fazer avaliação de um determinado processo, é necessário que o avaliador ou a pessoa interessada em fazer a avaliação saiba os requisitos e objetivos básicos desejados para a avaliação. No capítulo 4.1.2 é proposto o desenvolvimento de um modelo de avaliação, seguindo as principais características do processo de manutenção da Mult Sistemas.

### **4.3.2 CADASTRO DE ORGANIZAÇÕES**

Na fig. 4-5 está representada a tela de cadastramento de organizações. Na organização estão contidas informações referentes a organização que será avaliada, dando flexibilidade para que processos de manutenção de organizações diferentes possam ser avaliados.

Os atributos para este cadastramento são código, nome, número de funcionários da organização, número de funcionários da equipe de manutenção, área de atuação e métodos e ferramentas utilizados para garantia da qualidade. Na fig. 4-5, mostra-se o cadastramento da

organização Mult Sistemas, utilizou-se como código da organização um campo alfanumérico no nosso exemplo MULT.

FIGURA 4-5 – TELA DE CADASTRO DA ORGANIZAÇÃO.

Código	Nome	Área Atuação	Nr. Funcion
MULT	Mult Sistemas Ltda.	Sistema de Gestão Integrada e Aut...	

Código:   
 Nome:   
 Número de Funcionários da Organização: 
                         
 Número de Funcionários da Equipe de Manutenção:   
 Área de Atuação:   
 Métodos e Ferramentas Utilizadas para a Garantia da qualidade:

**Fechar**

### 4.3.3 CADASTRO DE PROJETOS

Na fig. 4-6 está representada a tela de cadastramento de projetos. O projeto é utilizado para identificar processos de manutenção em módulos específicos por exemplo, nele estão contidas informações referentes ao projeto que será avaliado, sendo que para uma mesma organização poderá ser cadastrado mais de um projeto, facilitando a avaliação do mesmo processo com metas diferenciadas.

Os atributos para este cadastramento são código e descrição do projeto, a organização e nome da produtora (podendo não ser a organização), todos utilizados posteriormente na avaliação ou em relatórios.

Na fig. 4-6, mostra-se o cadastramento de um projeto para a organização Mult Sistemas, utilizou-se como código do projeto um campo alfanumérico e a descrição do projeto é a identificação do projeto pelo avaliador ou o nome do processo daquela determinada organização, que no exemplo chama-se de Manutenção do Módulo Estoque e Faturamento.

FIGURA 4-6 – TELA DE CADASTRO DE PROJETOS.

A imagem mostra a janela 'Cadastro de Projetos' com uma tabela e campos de formulário. A tabela contém as seguintes informações:

Código	Organização	Descrição
MANUT. ECV	MULT	Manutenção do Módulo de Estoque e Faturamento
MANUT. ECV	SPICE	

Os campos de formulário abaixo da tabela contêm os seguintes valores:

- Código: MANUT. ECV
- Organização: MULT
- Descrição: Manutenção do Módulo de Estoque e Faturamento
- Produtora: Mult Sistemas Ltda

Um botão 'Fechar' está localizado no canto inferior direito da janela.

#### 4.3.4 CADASTRO DE METAS

Toda avaliação deverá possuir pelo menos 1 (uma) meta definida, desta forma, torna-se obrigatório o seu cadastramento. Nesta fase, além do cadastramento propriamente dito, é feita a definição do projeto de avaliação do processo de manutenção, tornando-se indispensável o conhecimento detalhado do que se deseja avaliar. O ideal é seguir o tópico 4.1.2, para que seja possível dar um rumo adequado para a avaliação. A montagem das metas propostas para a avaliação do processo de manutenção deve tomar como base os objetivos da

organização em relação ao foco de qualidade que se pretende atingir. Através de metas direcionadas a realidade da empresa e que consideram padrões e normas de qualidade já existentes torna-se mais fácil determinar os pontos positivos e negativos do processo em funcionamento.

Na fig. 4-7 mostra-se a tela para cadastro de metas, ela é iniciada pela identificação do código da meta, código da organização e código do projeto. O atributo objetivo destina-se a receber a ação para que se destina esta avaliação. Os atributos de objeto e foco de qualidade objeto recebem informações das características daquela avaliação ou enfoque que se gostaria de dar para este objetivo, podendo ser relativo à qualidade, benfeitorias e até mesmo de valores. O atributo ponto de vista sugere a informação do solicitante daquela avaliação ou do solicitante do projeto como um todo, demonstra que tipo de pessoa ou usuário está interessado em avaliar o software. Para o exemplo, está sendo avaliado sob ponto de vista de um gerente de software. O atributo contexto define em que ambiente está inserida a meta a ser avaliada.



FIGURA 4-7 – TELA DE CADASTRO DE METAS.

Código	Organização	Projeto	Objetivo	Objeto	Foco de
1	MULT	MANUT. ECV	Caracterizar	O processo de manutenção de	Com res
2	MULT	MANUT. ECV	Avaliar e	O processo de manutenção de	Com res
3	MULT	MANUT. ECV	Analisar	O processo de manutenção de	Com res

Código:   
 Organização:    
 Projeto:    
 Objetivo:   
 Objeto:   
 Foco de qualidade:   
 Ponto de Vista:   
 Contexto:

### 4.3.5 CADASTRO DE QUESTÕES

A montagem das questões é influenciada diretamente pelas metas, já que a meta é que direciona as questões e consecutivamente as métricas. O número de questões cadastradas para cada meta é de no mínimo 1 (uma), devido à necessidade e estrutura da abordagem GQM. Cada meta poderá ter várias questões, já que uma meta para ser atendida poderá requerer mais de uma questão. Na fig. 4-8 pode ser vista a tela de cadastro de questões do protótipo.

FIGURA 4-8 – TELA DE CADASTRO DE QUESTÕES.

Organização	Projeto	Meta	Código	Descrição
MULT	MANUT. ECV	1	1	Os dados de um pedido de manutenção no software são detalh
MULT	MANUT. ECV	1	2	É realizada a rastreabilidade das releases do software, para des
MULT	MANUT. ECV	1	3	Qual o tempo utilizado para realizar a análise do problema / mod
MULT	MANUT. ECV	1	4	As modificações realizadas no software consideram as especific
MULT	MANUT. ECV	1	5	O mantenedor após a codificação da alteração, realiza testes n

Código:   
 Organização:    
 Projeto:    
 Meta:    
 Descrição:   
 Observação:

### 4.3.6 CADASTRO DE MÉTRICAS

O cadastro de uma métrica se faz necessário para atender uma questão. Cada questão obrigatoriamente deve possuir uma métrica para respondê-la, seguindo a estrutura GQM. Uma questão poderá ter uma ou inúmeras métricas e uma mesma métrica poderá ser utilizada por várias questões. Na fig. 4-9 exemplifica-se o cadastro da métrica com o código 1, como se verifica no exemplo, o atributo código da métrica é seqüencial, possui também uma descrição e observação para esta métrica. Outros atributos são necessários, o atributo de ligação com a organização, projeto, meta e questão, para indicar a relação com qual processo de manutenção esta métrica estará atrelada. Outro atributo é o código da regra que é utilizado para enquadrar esta métrica em uma determinada faixa de pontuação. Constata-se em detalhes esta pontuação na tela de cadastro de regras.

Na fig. 4-9 pode-se ver algumas métricas elaboradas com base no processo de manutenção de uma produtora de software, no caso a Mult Sistemas. Estas métricas têm como premissa básica determinar através de regras pré-definidas a qualidade e produtividade do

processo de manutenção avaliado. A métrica 3 por exemplo “Média de tempo da análise” tem por objetivo determinar qual o tempo médio necessário para se avaliar uma solicitação encaminhada para a área de manutenção da produtora de software. Outro exemplo pode ser a métrica 2 “Qtde de pedidos que foram rastreados (mês)” onde pretende-se saber quantas solicitações de manutenção tiveram o motivo/origem do problema/modificação rastreado ou seja se existem registros anteriores referentes ao assunto da solicitação. Já com a métrica 1 “Qtde de pedidos devolvidos pela manutenção por falta de informação nos pedidos de manutenção” pretende-se identificar a qtde. de solicitações que não estão descritas de forma clara a fim de se tomar uma ação efetiva em relação a mesma.

FIGURA 4-9 – TELA DE CADASTRO DE MÉTRICAS.

Código	Organização	Projeto	Meta	Questão	Regra	Descrição	Obs
1	MULT	MANUT. ECV	1	1	2	Qtde de pedidos devolvidos pela manutenção por falta de informação nos pedidos de manutenção	Métr
2	MULT	MANUT. ECV	1	2	2	Qtde de pedidos que foram rastreados (mês)	Métr
3	MULT	MANUT. ECV	1	3	2	Média de tempo da análise	Métr
4	MULT	MANUT. ECV	1	3	2	Desvio Padrão	Métr
5	MULT	MANUT. ECV	1	3	2	Qtde de casos acima do limite superior (20 dias)	Métr

Código:   
 Organização:   Projeto:   Meta:    
 Questão:   Regra:    
 Descrição:   
 Descrição:   
 Observação:

### 4.3.7 CADASTRO DE REGRAS DE PONTUAÇÃO

Para cada métrica associa-se uma regra. Na fig. 4-10 estão os atributos para o cadastramento da regra. Ela funcionará como um conceito para as respostas dadas nas métricas. O problema encontrado é que as respostas das métricas divergem muito uma em relação à outra e até mesmo dentro de uma mesma questão, as métricas poderão ter respostas totalmente diferenciadas. Precisou-se encontrar uma forma de igualar estes valores para que na somatória destes, se pudesse ter valores igualados. Criaram-se 4 (quatro) atributos, ótimo, recomendado, parcial e ruim. Estes atributos recebem valores que são considerados nas respostas das métricas, ou seja, dependendo da resposta dada para a métrica, como a métrica esta associada a uma regra, o valor da resposta será comparado com a regra e enquadrado em uma das faixas citadas, formando assim o resultado final da avaliação.

Na fig. 4-10 ilustra-se a regra 2 (dois) que anteriormente associou-se com a métrica 1 (um), quando esta métrica for respondida, será comparado o valor dado com o valor da regra e atribuído valor para a mesma.

Os valores atribuídos para cada faixa da regra foram definidos de acordo com valores entendidos como compatíveis com cada métrica, estudou-se cada métrica e enquadrou-se numa faixa específica. Outro fator importante é a pontuação que cada métrica receberá após ser respondida, definiu-se que quando o valor cair na faixa “ótimo”, receberá 20 pontos, para a faixa “recomendado”, receberá 15 pontos, para a faixa “parcial”, receberá 10 pontos e para a faixa “ruim”, 5 pontos, pois quando a regra é ascendente quanto menor a pontuação, maior o conceito e quando a regra for descendente quanto maior a pontuação menor o conceito.

FIGURA 4-10 – TELA DE CADASTRO DE REGRAS.

Código	Descrição	Faixa 1	Faixa 2	Faixa 3	Faixa 4
2	Ascendente - 1 / 4 / 7 / 10	1	4	7	10
3	Descendente - 20 / 15 / 10 / 5	20	15	10	5
4	Descendente - 10 / 7 / 4 / 1	10	7	4	1
5	Descendente - 100 / 75 / 50 / 25	100	75	50	25
6	Descendente - 30 / 20 / 10 / 5	30	20	10	5
7	Descendente - 50 / 40 / 20 / 10	50	40	20	10
8	Descendente - 6 / 4 / 2 / 0	6	4	2	0
9	Ascendente - 10 / 20 / 30 / 50	10	20	30	50

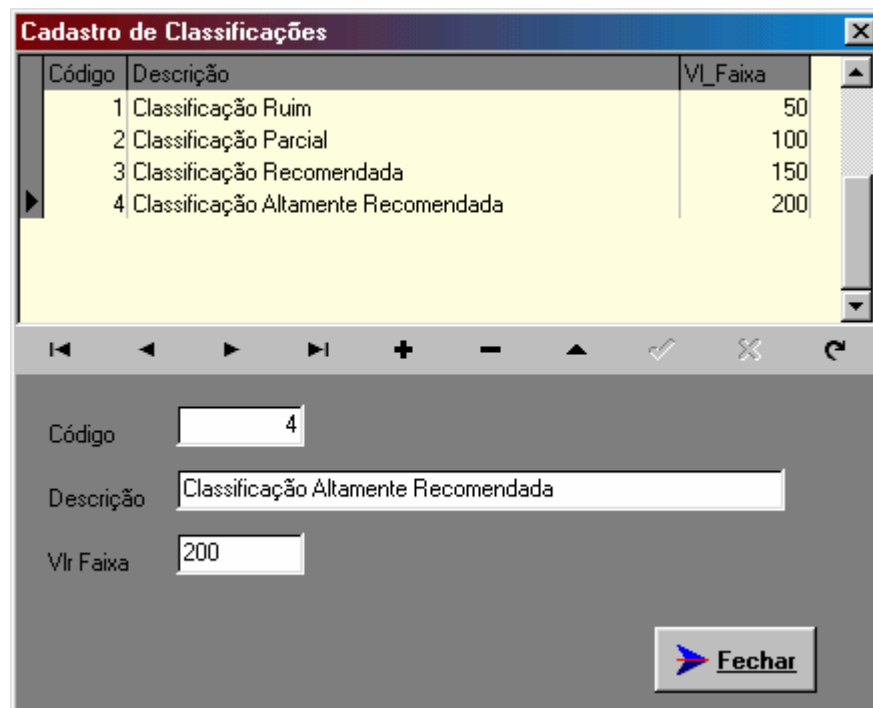
Código:   
 Descrição:   
 Ótimo:  Parcial:   
 Recomendado:  Ruim:

**Fechar**

### 4.3.8 CADASTRO DE CLASSIFICAÇÕES

Na fig. 4-11 mostra-se o cadastro das classificações, elas são utilizadas para retornarem a classificação das avaliações. Após o término da avaliação, a qual será vista em detalhes mais adiante, será feito um cálculo somatório em relação aos valores respondidos nas métricas, este resultado será comparado com as classificações e enquadrado na classificação correspondente.

FIGURA 4-11 – TELA DE CADASTRO DE CLASSIFICAÇÕES.



### 4.3.9 MENU AVALIAÇÃO

Através do menu principal do protótipo, aciona-se a opção de avaliação, que trará a tela de movimentação de avaliações, esta é a tela de entrada para a montagem das avaliações do processo de manutenção. Na fig. 4-12 exemplifica-se a tela citada, ela possui o cadastro da avaliação e na parte inferior da tela, mostra os seguintes botões:

- a) Responder avaliação: a partir desta opção torna-se possível à associação do plano de metas/questões/métricas com uma avaliação. Será vista mais adiante em detalhes a configuração e o funcionamento desta opção, pelo fato de se tratar da avaliação propriamente dita;
- b) Totalizar avaliação: esta opção monta a coluna "Resultado" e a coluna "Classificação" do browser de avaliações. Esta opção calcula de acordo com a avaliação selecionada todo o seu resultado, levando em consideração as respostas dadas para as métricas, à regra usada pela métrica e a faixa em que o valor dado à métrica se encaixa (Ótimo, Recomendado, Parcial ou Ruim).

FIGURA 4-12 – TELA DE AVALIAÇÃO.

Avaliação	Organização	Projeto	Data	Resultado	Classificação
1	MULT	MANUT. ECV	05/06/02	30	1

Código:   
 Organização:    
 Projeto:    
 Data:   
 Descrição:   
 Resultado:       Classificação:

### 4.3.10 PROCESSO DA AVALIAÇÃO

Após clicar no botão “Responder Avaliação”, é mostrada a tela de processo da avaliação, onde se pode selecionar as metas/questões/métricas a serem respondidas pelo usuário para obter posteriormente o resultado da avaliação, sendo obrigatório responder a todas as métricas para que a questão seja totalmente atendida e a avaliação classificada adequadamente. Na fig. 4-13 exemplifica-se a tela de processo de avaliação, nela estão contidas as informações do código da avaliação, código da organização e do projeto a ser avaliado. Selecionando com o mouse a meta através do browse de metas, trará somente as questões relacionadas a meta e selecionando a questão somente serão apresentadas no browse de métricas, as métricas relacionadas a questão selecionada.

É necessário responder a todas as métricas para que a questão seja totalmente atendida e os valores dados neste momento serão armazenados para a movimentação da avaliação e será usado para a totalização da avaliação, como mostrado na tela principal da avaliação. Ao clicar na métrica que se deseja informar o valor será mostrada automaticamente a faixa de enquadramento da regra que a métrica está inserida. Ao informar o valor da métrica deve-se clicar no botão gravar para efetivar a alteração do campo.

FIGURA 4-13 – TELA DO PROCESSO DE AVALIAÇÃO.

**Processo da Avaliação**

Avaliação: 1      Organização: MULT      Projeto: MANUT. ECV

**Meta**

Meta	Objetivo	Objeto	Foco de Qualidade	Ponto de Vista	Contexto
1	Caracterizar	O processo de	Com respeito à	Do ponto de vista	No conte
2	Avaliar e melhorar	O processo de	Com respeito aos	Do ponto de vista	No conte

**Questão**

Questão	Descrição
1	Os dados de um pedido de manutenção no software são detalhados de forma clara e concisa
2	É realizada a rastreabilidade das releases do software, para descobrir quando o problema foi

**Métrica**

Métrica	Descrição
1	Qtde de pedidos devolvidos pela manutenção por falta de informação nos pedidos de manutenção
2	Qtde de pedidos que foram rastreados (mês)

**Resposta**

Entre com o valor da métrica:

Faixas de valores da métrica:

Ótimo	Recomendado	Parcial	Ruim
3	6	10	20

### 4.3.11 RELATÓRIO DA AVALIAÇÃO DE SOFTWARE

No Anexo I, pode ser verificado o primeiro relatório do protótipo, ele é chamado de relatório da avaliação e trata-se de um espelho da avaliação, mostrando de forma estruturada o



que aconteceu naquela avaliação. Traz ordenado pela meta, após as questões que foram respondidas com suas métricas. O relatório também mostra a qual organização / projeto a avaliação pertence, o resultado da avaliação e qual foi a sua classificação final.

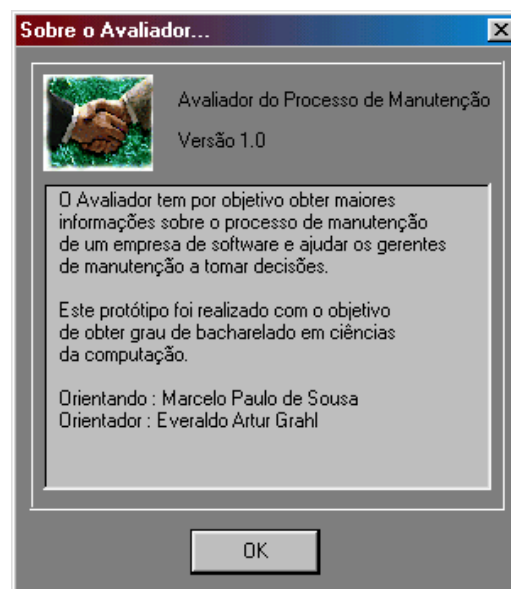
### 4.3.12 RELATÓRIO QUESTIONÁRIO DE AVALIAÇÃO

O relatório questionário de avaliação está exemplificado no Anexo II. A criação deste relatório tem por objetivo disponibilizar para impressão do modelo de avaliação criado no protótipo, desta forma é possível imprimir este relatório e enviar para uma pessoa ou empresa interessada em fazer a avaliação, pedir para que o avaliador responda o questionário e devolva-o para que se possa inserir os dados no sistema sem a necessidade de enviar o protótipo para o avaliador.

### 4.3.13 MENU SOBRE

Na fig. 4-14 pode-se ver a tela “Sobre” que traz um breve descritivo do objetivo do protótipo.

FIGURA 4-14 – TELA SOBRE



## 4.4 RESULTADOS PRINCIPAIS

Os resultados principais conseguidos através do protótipo por meio de visualização das avaliações gravadas ou por intermédio de relatórios se deve principalmente a forma como o

mesmo foi construído, considerando a estrutura da abordagem GQM como base para as tabelas criadas, tornando possível a flexibilização no cadastramento de metas/questões/métricas de acordo com as necessidades da organização.

O relatório de avaliação do processo de manutenção traz as informações de uma determinada avaliação realizada no protótipo, a classificação dada a avaliação é realizada de forma a considerar as regras determinadas para as métricas e considerando as seguintes faixas fixas para pontuar uniformemente as métricas: faixa “ótimo”, receberá 20 pontos, para a faixa “recomendado”, receberá 15 pontos, para a faixa “parcial”, receberá 10 pontos e para a faixa “ruim”, 5 pontos. O valor da métrica apontado no relatório será enquadrado conforme a regra da métrica. Este valor de enquadramento teve que ser determinado para métricas com regras diferentes por exemplo uma métrica que envolve percentual e outra que envolve valor (qtde) mas que no conceito para classificação da avaliação deveria enquadrar-se nas quatro faixas.

O protótipo não possui forma de extração de dados via importação, sendo seus dados informados pelo mantenedor através do questionário disponibilizado pelo protótipo e via entrada de dados, no caso da Mult se busca os dados de fontes como planilhas de acompanhamento da área de manutenção (busca os dados do banco do SISTEC – Sistema de Controle Técnico) ou através de *queries* SQL de consulta que podem ser montadas no mesmo.

O sucesso da avaliação só se concretizará se o modelo proposto atender as necessidades da organização a ser avaliada, através da construção do modelo de métricas verificou-se que quanto menos normatizado o processo a ser avaliado mais complicado se determinar métricas para o mesmo.

## **4.5 CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO**

Para a implementação foram utilizados alguns componentes da linguagem Delphi 5.0 para tratamento da interface, impressão de dados, leitura e gravação. Por tratar-se de uma base de dados padrão, utilizou-se para gravação de arquivos, tabelas paradox.

Utilizou-se para os menus, forms com labels. Para o item de menu Cadastros, foram utilizados DbGrid para visualização dos registros. Os componentes utilizados DBEdit para a entrada dos dados e Labels e GroupBox para trazer a descrição dos respectivos campos.

Usou-se também o `DataModule` no qual estão os componentes relativos às tabelas (`Table`), comandos SQL para leitura, inserção, atualização e exclusão de registros (`Query` e `UpdateSQL`), distribuidor de dados (`DataSource`).

Para desenvolvimento dos relatórios utilizou-se `QuickReport 3.0`, que acompanha a instalação do `Delphi 5.0`. Optou-se pelo seu uso devido à diversidade de funções que o mesmo disponibiliza em tempo de desenvolvimento e de execução, pela facilidade de manipulação do layout dos campos de um relatório.

## 5 CONCLUSÕES

A utilização das normas ISO/IEC 12207 e ISO/IEC 15504 (SPICE) ajudaram a montar a avaliação do processo de manutenção de software. Elas designam etapas a serem cumpridas e disponibilizam ao avaliador passos para determinar a avaliação.

A dificuldade das empresas produtoras de software em manter os produtos existentes, faz com que cada vez mais se busquem alternativas de minimizar os impactos da manutenção de seus produtos no dia-a-dia.

Quanto ao protótipo conclui-se:

- a) uma das vantagens de sua utilização é a criação de base histórica, permitindo assim ao avaliador comparar avaliações anteriores com futuras avaliações;
- b) pode-se criar outras metas/questões/métricas para outras organizações sem ser as propostas neste trabalho. A abordagem GQM facilita a criação da parametrização para qualquer tipo de processo;
- c) uma das principais características do protótipo é sua flexibilidade para montagem de avaliações, onde o avaliador poderá criar sua avaliação de acordo com sua necessidade, utilizando regras pré-definidas;

Ao final deste trabalho, alcançou-se de forma satisfatória os objetivos definidos, através do conjunto de métricas para avaliação da manutenção podem-se obter dados quantitativos e qualitativos a respeito do processo.

O protótipo e o conjunto de métricas construídas permite ao gerente de software flexibilizar a avaliação do processo de manutenção através de regras pré-definidas para as métricas.

Por fim, a utilização de uma abordagem simples, mas, no entanto poderosa, juntamente com normalizações que regem o mercado internacional formam um conjunto muito eficaz de ferramentas que ajudam a avaliar e mostrar as principais diferenças entre processos de manutenção.

## 5.1 SUGESTÕES

Para trabalhos futuros sugere-se o tratamento mais refinado da entrada de dados, podendo se importar as informações de outras aplicações que forneceriam a informação necessária para as respostas, que é o caso da Mult Sistemas, que hoje possui uma ferramenta para alimentação de pendências da manutenção, mas que precisaria de uma integração para fornecer os dados para o protótipo. Também pode-se criar um mecanismo para configurar os pesos de enquadramento geral para a classificação da avaliação, haja visto que hoje se determinou valores fixos para pontuação das métricas na soma geral da avaliação e conseqüente classificação da mesma.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARROS, Roberto S.M.. **Manutenção de Software**, Recife, set. 1998. Disponível em: <<http://www.di.ufpe.br/~roberto/Engsoft1.html>>. Acesso em: 13 abr. 2002.

CUNHA, Gabriela Elisa da. **Métricas de Software**. Projeto rumo ao CMM – Lqs/Unisinos e Softsul, jun. 2001. Disponível em: <<http://www.htm>>. Acesso em: 01 fev. 2002.

EMAM, Khaled El et al.. **SPICE : the theory and practice of software process improvement and capability determination**. Washington, D.C : IEEE Computer Society, 1998.

FERNANDES, Aguinaldo Aragon. **Gerência de software através de métricas**. São Paulo: Atlas S.A, 1995.

FOURNIER, Roger. **Guia Prático para Desenvolvimento e Manutenção de Sistemas Estruturados**. São Paulo: Makron Books, 1994.

GROSS, Jan Charles. **Protótipo de um software de apoio à utilização do GQM (Goal-Question-Metric)**, 2001. 70 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

HOPPE, Charles. **Software de apoio a manutenção de sistemas baseado em normas de qualidade**, 1999. 109 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

HORT, Glederson. **Protótipo de auxílio a implantação da Norma NBR ISO 9000-3**, 1999. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

ISO/IEC 12207. **Tecnologia de informação**. Modelo de qualidade para produto de software, Rio de Janeiro, jan. 2000. Disponível em: <[http://www.abnt.org.br/index body.htm](http://www.abnt.org.br/index_body.htm)>. Acesso em: 01 mai. 2002.

KOSCIANSKI, André. **Guia para utilização de normas sobre qualidade de produto de software ISO/IEC 9126 e ISO/IEC 14598**, Curitiba, maio 1999. Disponível em: <<http://www.pr.gov.br/abntosftware/publica.html>>. Acesso em: 01 fev. 2002.

LIMA, Mirian Fuá de. **Análise do processo de manutenção de software em empresas de Porto União da Vitória**, 2001. 34 f. Monografia de Pós-Graduação em Nível de Especialização em Tecnologia de Desenvolvimento de Sistemas, Universidade Regional de Blumenau, Blumenau.

LOGIC WORKS INC. **Reference Guide**. Princeton, 1997.

NBR ISO 9000-3. **Normas de gestão da qualidade e garantia da qualidade**. Parte 3: Diretrizes para a aplicação da NBR 19001 ao desenvolvimento, fornecimento e manutenção de “software”, nov. 1993. Disponível em: <[http://www.abnt.org.br/index body.htm](http://www.abnt.org.br/index_body.htm)>. Acesso em: 11 jul. 2002.

PARIKH, Girish. **Reengenharia de software: técnicas de manutenção de programas e sistemas**. Rio de Janeiro: LTC, 1990.

PRESSMANN, Roger. **Engenharia de Software**. São Paulo: Makron Books, 1995.

ROCHA, Ana Regina Cavalcanti da, MALDONADO, Jose Carlos, et al.. **Qualidade de software: teoria e prática**. São Paulo : Prentice Hall, 2001.

SCUSSIATO, Edésio. **Processo de manutenção de sistemas baseado na norma ISO-IEC 12207**, 1998. 63 f. Monografia de Pós-Graduação em Nível de Especialização em Tecnologia de Desenvolvimento de Sistemas - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

WANGENHEIM, Christiane Gresse V. **Programa de melhoramento de qualidade e produtividade de software baseado em mensuração**, Florianópolis, jun. 1999. Disponível em: <<http://sts.geness.inf.ufsc.br/html/cursos/qualidade.html>>. Acesso em: 20 jan. 2002.

WANGENHEIM, Christiane Gresse V. **Utilização do GQM no desenvolvimento de software**, Porto Alegre, set. 2000. Disponível em: <<http://www.inf.unisinos.br/instituto/lqs/gqm.html>>. Acesso em: 15 mar. 2002.

# ANEXO I – RELATÓRIO DE AVALIAÇÃO DO PROCESSO MANUTENÇÃO

Metas - Questões - Métricas		Valor Métrica
<b>1 - Caracterizar - O processo de manutenção de software</b>		
<b>1 - Os dados de um pedido de manutenção no software são detalhados de forma clara e concisa?</b>		
1 - Qtde de pedidos devolvidos pela manutenção por falta de informação nos pedidos de manutenção (mês)		20
<b>2 - É realizada a rastreabilidade das releases do software, para descobrir quando o problema foi originado?</b>		
2 - Qtde de pedidos que foram rastreados (mês)		20
<b>3 - Qual o tempo utilizado para realizar a análise do problema /modificação a ser passado para a codificação?</b>		
3 - Média de tempo da análise		20
4 - Desvio Padrão		10
5 - Qtde de casos acima do limite superior (20 dias)		8
<b>4 - As modificações realizadas no software consideram as especificações do contrato de manutenção com o cliente?</b>		
6 - Qtde de modificações aperfeiçoadas de acordo com o contrato de manutenção (mês)		20
7 - % de exceções identificadas (qtde est /qtde total)		20
<b>5 - O mantenedor após a codificação da alteração, realiza testes man, ovlógica implementada e nas partes não modificadas para que o software como um todo ainda funcione corretamente?</b>		
8 - Qtde de modificações aperfeiçoadas de acordo com o contrato de manutenção (mês)		10
<b>6 - São realizadas revisões com os clientes contratados para a garantia da qualidade das tarefas realizadas em uma determinada release?</b>		
9 - Qtde de alterações revisadas em conjunto com o cliente em liberação de releases betas (mês)		10
<b>7 - Os usuários (clientes) são notificados Quando da necessidade de migração para um novo ambiente operacional?</b>		



## ANEXO II – QUESTIONÁRIO DE AVALIAÇÃO DO PROCESSO MANUTENÇÃO

Questionário para avaliação do processo de manutenção 05/06/02 15:52:30

Questão	Resposta
<b>Métrica</b>	
<b>Os dados de um pedido de manutenção no software são detalhados de forma clara e concisa?</b> Qtd de pedidos devolvidos pela manutenção por falta de informação nos pedidos de manutenção (mês)	_____
<b>É realizada a rastreabilidade das releases do software, para descobrir quando o problema foi originado?</b> Qtd de pedidos que foram rastreados (mês)	_____
<b>Qual o tempo utilizado para realizar a análise do problema/modificação a ser passado para a codificação?</b> Média de tempo da análise	_____
Desvio Padrão	_____
Qtd de casos acima do limite superior (20 dias)	_____
<b>As modificações realizadas no software consideram as especificações do contrato de manutenção com o cliente?</b> Qtd de modificações aperfeiçoadoras de acordo com o contrato de manutenção (mês)	_____
% de exceções identificadas (qtd ex / qtd total)	_____
<b>O mantenedor após a codificação da alteração, realiza testes na lógica implementada e nas partes não modificadas para que o software como um todo ainda funcione corretamente?</b> Qtd de modificações aperfeiçoadoras de acordo com o contrato de manutenção (mês)	_____
<b>São realizadas reuniões com os clientes contratados para a garantia da qualidade das tarefas realizadas em um determinado release?</b> Qtd de alterações revisadas em conjunto com o cliente em liberação de releases betas (mês)	_____
<b>Os usuários (clientes) são notificados quando há necessidade de migração para um novo ambiente operacional?</b> Qtd de usuários (clientes) notificados quando da migração em relação ao total da organização (%)	_____
<b>Existe uma política de descontinuação do software e sua consequente substituição ou upgrade, de forma que os dados antigos fiquem armazenados para realização de auditorias?</b> Qtd de clientes que tiveram problemas de perda de dados em migrações por não possuírem backup's de suas informações (Mês)	_____
<b>Existem procedimentos para registro das solicitações de problemas/não-conformidades detectados?</b> % de solicitações registradas na ferramenta em relação ao total de solicitações (mês)	_____
<b>Existem procedimentos bem definidos para avaliar as solicitações de melhoramento do sistema sem ferir o aspecto estrutural e a finalidade do software em questão?</b>	_____

Page 1 of 2