

**Instituto de Pesquisas Tecnológicas do Estado de São Paulo**

**Suelane Garcia Fontes**

**UM PROCESSO DE MEDIÇÃO DE DEFEITOS DO SOFTWARE  
ORIENTADO A TESTES**

**São Paulo  
2008**

**Suelane Garcia Fontes**

**UM PROCESSO DE MEDIÇÃO DE DEFEITOS DO SOFTWARE  
ORIENTADO A TESTES**

Projeto de pesquisa apresentado ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo para a obtenção do título de Mestre em Engenharia de Computação.

Área de concentração: Engenharia de software.

Orientador: Prof. Dr. Kechi Hirama.

**São Paulo  
fevereiro/2008**

## **Agradecimentos**

Agradeço a Deus por iluminar o meu caminho e permitir a conclusão deste sonho.

Aos meus pais, irmãs e sobrinhos, que são a razão da minha vida, pelo amor incondicional, por todo incentivo e compreensão.

Ao orientador Kechi Hirama por todo conhecimento doado, toda disponibilidade, paciência, incentivo e amizade.

A uma amiga, em especial, cuja amizade e a generosidade foram decisivas para conclusão deste trabalho.

Às empresas envolvidas nos estudos de caso, pela oportunidade e, também, por toda disponibilidade e dedicação de seus membros.

Aos amigos e professores que, direta ou indiretamente, ajudaram na realização deste sonho.

## RESUMO

Atualmente, a qualidade de software é considerada a principal característica competitiva entre as organizações. Baixa qualidade de software pode afetar seu desempenho nos negócios. Portanto, é muito importante planejar as atividades de teste de maneira cuidadosa. Os testes mostram a existência de defeitos e a sua análise quantitativa é relevante porque ela pode revelar a qualidade do produto e do processo. Então, a medição de defeitos é o passo fundamental para controlar e melhorar as atividades de teste. O número de defeitos de produto é uma métrica relacionada à percepção dos usuários. Além disso, esta métrica pode ser associada com outros atributos de qualidade como confiabilidade, correção, completeza, eficiência e usabilidade.

Quando o teste de software e o processo de medição trabalham juntos, muitas questões surgem sobre o teste e a medição relevante, bem estruturado e de boa qualidade.

Este trabalho propõe um processo de medição que coleta defeitos da fase de teste de software. Este processo, baseado na técnica chamada *Practical Software Measurement* (PSM), do Departamento de Defesa Norte-Americano, gera resultados que atendem às necessidades de informação dos responsáveis pela fase de teste. Para ilustrar este processo, dois estudos de caso são apresentados.

**Palavras-chave:** Teste de software, Medição de software, Defeitos do software, Qualidade de Software, PSM.

## **ABSTRACT**

### **A Testing Oriented Software Defects Measurement Process**

Currently, the software quality is considered the main competitive feature among the organizations. Low software quality can affect their business performance. Thus, it is much important to plan the testing activities in careful way. The testing shows the defects existence and their quantitative analysis is relevant because it can reveal the product and process quality. Then, the defects measurement is the fundamental step to control and improve the software testing activities. The number of product defects is a metric related on quality of user's perception. Furthermore, this metric can be associated with other quality attributes like reliability, correction, completeness, efficiency and usability.

When the software testing and measurement process work together, many questions emerge about relevant, well structured and good quality testing and measurement.

This work proposes a measurement process that collects defects from software testing phase. This process based on technique named Practical Software Measurement (PSM), of USA Department of Defense, generates results which address information needs of responsible of testing phase. To illustrate this process, two case studies are presented.

**Keywords:** Software testing, Software measurement, Software defects, Quality Software, PSM.

## Lista de Figuras

Figura 1 - Relação entre Erro, Defeito e Falha.....	17
Figura 2 – Relação entre Níveis, Tipos, Técnicas e Estratégias de Teste.....	23
Figura 3 – Relação entre Etapas e Documentos do Processo de Teste.....	25
Figura 4 - GQM.....	32
Figura 5 - <i>Goal - Driven Software Measurement</i> .....	34
Figura 6 - Modelo de Informação – PSM.....	37
Figura 7 - Construção de uma medição pelo PSM.....	38
Figura 8 - Modelo de Processo - PSM .....	39
Figura 9 - Plano de Medição PSM .....	43
Figura 10 - Relação do Processo de Teste com o Processo de Medição.....	47
Figura 11- Processo de Teste .....	49
Figura 12 - Processo de Medição - Contexto .....	51
Figura 13 – Relação entre o Processo de Medição Proposto e o PSM .....	52
Figura 14 - Processo de Medição – Fases.....	53
Figura 15 - Fase de Planejamento da Medição.....	54
Figura 16 - Plano de Medição .....	55
Figura 17 - Plano de Medição – Descrição da Meta.....	56
Figura 18 - Plano de Medição – Papéis e Responsabilidades.....	57
Figura 19 - Plano de Medição - Identificação das Necessidades de Informação.....	58
Figura 20 - Plano de Medição – Seleção da Medição .....	59
Figura 21 – Plano de Medição - Especificação da Medição – Medidas Básicas.....	60
Figura 22 – Plano de Medição - Especificação da Medição – Medidas Derivadas .....	61
Figura 23 – Plano de Medição - Especificação da Medição – Indicadores .....	62
Figura 24 – Construção da medição – Esquema hierárquico.....	63
Figura 25 - Exemplo de Construção da medição – Esquema hierárquico .....	64
Figura 26 – Plano de Medição - Especificação da Medição - Coleta de Dados.....	65
Figura 27 - Fase de Execução da Medição.....	65
Figura 28 - Estudo de Caso - Empresa A - Estratégias de Teste .....	72
Figura 29 - Estudo de Caso - Empresa A - Construção da Medição (Meta 1) .....	77
Figura 30 - Estudo de Caso - Empresa A - Construção da Medição (Meta 2) .....	78
Figura 31 - Estudo de Caso - Empresa A - Gráfico de Defeitos de Inspeção de Código .....	80
Figura 32 - Estudo de Caso - Empresa A - Gráfico de Defeitos do Teste da Área de Qualidade.....	80
Figura 33 - Estudo de Caso - Empresa A - Gráfico de Defeitos do Teste do Analista .....	81

Figura 34 - Estudo de Caso - Empresa A - Gráfico de Distribuição de Defeitos do Teste de Software .....	82
Figura 35 - Estudo de Caso - Empresa A - Gráfico do Esforço do Teste do Programador ...	83
Figura 36 - Estudo de Caso - Empresa A - Gráfico do Esforço da Inspeção de Código .....	84
Figura 37 - Estudo de Caso - Empresa A – Gráfico do Esforço do Teste da Qualidade .....	84
Figura 38 - Estudo de Caso - Empresa A – Gráfico de Esforço do Teste de Software .....	85
Figura 39 - Estudo de Caso - Empresa A - Gráfico de Estimativa de Custo do Teste de Software .....	86
Figura 40 - Estudo de Caso - Empresa B - Estratégias de Teste .....	90
Figura 41 - Estudo de Caso - Empresa B - Construção da medição .....	94
Figura 42 – Estudo de Caso – Empresa B - Estrutura do Projeto <i>Web Site</i> .....	95
Figura 43 - Estudo de Caso - Empresa B – Gráfico de Defeitos por Teste de Unidade .....	97
Figura 44 - Estudo de Caso - Empresa B - Gráfico de Defeitos por Teste de Integração .....	97
Figura 45 - Estudo de Caso - Empresa B – Gráfico de Estimativa de Produtividade do Teste de Software .....	98
Figura 46 - Exemplo de ICOM .....	109

## Lista de Tabelas

Tabela 1 - Modelos de Teste de Software.....	16
Tabela 2 - Detalhamento do Plano de Teste.....	27
Tabela 3 - Processo GQM .....	33
Tabela 4 - Processo <i>Goal - Driven Software Measurement</i> .....	35
Tabela 5 - Exemplo do Plano de Medição - Descrição da Meta .....	57
Tabela 6 - Exemplo do Plano de Medição – Papéis e Responsabilidades .....	57
Tabela 7 - Exemplo do Plano de Medição – Identificação das Necessidades de Informação .....	58
Tabela 8 - Exemplo do Plano de Medição – Seleção da Medição.....	59
Tabela 9 - Exemplo do Plano de Medição - Especificação da Medição – Medidas Básicas .	61
Tabela 10 - Exemplo do Plano de Medição - Especificação da Medição – Medidas Derivadas .....	61
Tabela 11 – Exemplo do Plano de Medição - Especificação da Medição – Indicadores.....	63
Tabela 12 - Exemplo do Plano de Medição - Especificação da Medição - Coleta de Dados	65
Tabela 13 - Estudo de Caso - Empresa A - Descrição das Metas.....	71
Tabela 14 - Estudo de Caso - Empresa A – Papéis e Responsáveis .....	72
Tabela 15 - Estudo de Caso - Empresa A - Identificação das Necessidades de Informação	73
Tabela 16 - Estudo de Caso - Empresa A - Seleção de Medidas.....	74
Tabela 17 - Estudo de Caso - Empresa A - Especificação de Medidas Básicas .....	75
Tabela 18 - Estudo de Caso - Empresa A - Especificação de Medidas Derivadas.....	76
Tabela 19 - Estudo de Caso - Empresa A - Especificação de Indicadores.....	76
Tabela 20 - Estudo de Caso - Empresa A - Especificação da Coleta de Dados.....	78
Tabela 21 - Estudo de Caso – Empresa A – Medidas Básicas (Meta 1) .....	79
Tabela 22 - Estudo de Caso - Empresa A – Medidas Derivadas (Meta 1).....	81
Tabela 23 - Estudo de Caso – Empresa A – Medidas Básicas – Esforço (Meta 2) .....	82
Tabela 24 - Estudo de Caso – Empresa A – Medidas Básicas – Valor Hora (Meta 2).....	83
Tabela 25 - Estudo de Caso - Empresa A - Medidas Derivadas (Meta 2) .....	85
Tabela 26 - Estudo de Caso - Empresa B – Descrição das Metas .....	91
Tabela 27 - Estudo de Caso - Empresa B – Papéis e Responsabilidades .....	91
Tabela 28 - Estudo de Caso - Empresa B - Identificação das Necessidades de Informação	92
Tabela 29 - Estudo de Caso - Empresa B - Seleção de Medidas.....	92
Tabela 30 - Estudo de Caso - Empresa B - Especificação de Medidas Básicas .....	93
Tabela 31 - Estudo de Caso - Empresa B - Especificação de Medidas Derivadas.....	93
Tabela 32 - Estudo de Caso - Empresa B - Especificação de Indicadores.....	93
Tabela 33 - Estudo de Caso - Empresa B - Especificação da Coleta de Dados.....	94



Tabela 34 - Estudo de Caso - Empresa B - Teste de Unidade (programas).....	96
Tabela 35 - Estudo de Caso - Empresa B - Teste de Unidade – Medida Derivada .....	96
Tabela 36 - Estudo de Caso - Empresa B - Teste de Integração – Medida Derivada.....	97
Tabela 37 - Estudo de Caso - Empresa B – Estimativa de Produtividade do Teste .....	98

## **Lista de Abreviaturas e Siglas**

<b>ATM</b>	<i>Automated Teller Machine</i>
<b>GQM</b>	<i>Goal Question Metric</i>
<b>GQIM</b>	<i>Goal Question Indicator Metric</i>
<b>IDEF</b>	<i>Integration Definition for Function Modeling</i>
<b>ICOM</b>	<i>Input Control Output Mechanism</i>
<b>PSM</b>	<i>Practical Software Measurement</i>
<b>SADT</b>	<i>Structured Analysis and Design Technique</i>
<b>TI</b>	<i>Tecnologia da informação</i>
<b>URA</b>	<i>Unidade de Resposta Audível</i>

## Sumário

CAPÍTULO 1 – INTRODUÇÃO .....	12
1.1 Motivação .....	12
1.2 Objetivo.....	13
1.3 Contribuições e Resultados .....	13
1.4 Metodologia .....	14
1.5 Organização do Trabalho.....	15
CAPÍTULO 2 – TESTE DE SOFTWARE.....	16
2.1 Conceitos Básicos .....	16
2.2 Terminologia.....	17
2.3 Técnicas de Teste de Software.....	18
2.3.1 Teste Estrutural (Caixa Branca).....	18
2.3.2 Teste Funcional (Caixa Preta) .....	19
2.4 Estratégias de Teste de Software .....	20
2.4.1 Teste de Unidade .....	20
2.4.2 Teste de Integração.....	20
2.4.3 Teste de Sistema.....	21
2.5 Processo de Teste .....	23
2.6 Norma IEEE 829.....	24
CAPÍTULO 3 - MEDIÇÃO DE SOFTWARE .....	30
3.1 Conceitos Básicos .....	30
3.2 Diretrizes e Recomendações para o Processo de Medição .....	31
3.2.1 GQM.....	31
3.2.2 <i>Goal – Driven Software Measurement</i> .....	34
3.2.3 <i>PSM - Practical Software Measurement</i> .....	36
CAPÍTULO 4 – PROCESSO DE MEDIÇÃO DE DEFEITOS.....	46
4.1. Contexto da Proposta .....	46
4.2. Processo de Medição Proposto .....	51
CAPÍTULO 5 – ESTUDOS DE CASO.....	68
5.1. Estudo de Caso – Empresa A.....	68
5.1.1. Caracterização dos Projetos da Área .....	68
5.1.2. Processo de Teste da Área .....	69
5.1.3. Detalhamento do Estudo de Caso .....	70
5.1.4. Análise da Aplicação do Processo de Medição .....	86
5.2. Estudo de Caso – Empresa B.....	88
5.2.1. Caracterização dos Projetos da Área .....	88
5.2.2. Processo de Teste da Área .....	88

5.2.3. Detalhamento do Estudo de Caso .....	89
5.2.4. Análise da Aplicação do Processo de Medição .....	98
5.3. Conclusões Relativas aos Estudos de Caso .....	99
CAPÍTULO 6 – CONCLUSÕES .....	101
6.1. Contribuições do Trabalho .....	101
6.2. Trabalhos Futuros.....	103
REFERÊNCIAS BIBLIOGRÁFICAS.....	104
GLOSSÁRIO .....	106
APÊNDICE A - IDEF0 (Integration Definition for Function Modeling) .....	108

## **CAPÍTULO 1 – INTRODUÇÃO**

### **1.1 Motivação**

O teste de software surgiu quase ao mesmo tempo em que os primeiros programas de computador e, segundo Hetzel (1984), encontram-se menções sobre teste desde 1950. A atividade de teste de software é considerada um elemento crucial para a garantia da qualidade do software.

Na literatura existe uma grande quantidade de propostas de modelos de teste que sugerem atividades que permitem a prevenção e detecção de defeitos de forma planejada e bem estruturada, como, por exemplo, os modelos de demonstração (WHITE e COHEN, 1980), de destruição (MYERS, 1979), de avaliação (ADRION, BRANSTED e CHERNIAVSKY, 1982) e de prevenção (GELPERIN e HETZEL, 1988).

Com a qualidade do produto software sendo considerada, atualmente, como um fator de competitividade para as organizações, as falhas de software transformaram-se em forças propulsoras para a adoção de atividades de teste mais cuidadosas e bem planejadas. A atividade de teste passou a ser considerada não como uma forma de mostrar a ausência de defeitos, mas como um meio de mostrar que os defeitos de software estão presentes, tornando-se relevante a coleta e análise dos dados sobre os defeitos encontrados nos testes. A principal razão em se coletar esses dados é que essas informações constituem um importante indicador de qualidade dos produtos desenvolvidos. Além de permitir a monitoração da qualidade dos produtos, a medição dos defeitos pode ser útil para identificar possibilidades de melhoria da produtividade das equipes. Como parte do esforço consumido pelos projetos é destinada à correção de defeitos, a incidência desses defeitos influi diretamente no custo de um projeto. Ao controlá-los quantitativamente, é possível identificar os pontos do processo mais problemáticos, ou seja, as iterações e os fluxos que estão inserindo o maior número de defeitos, e trabalhar para reduzi-los.

Segundo Almendra (2003), há várias propostas de medição de software com objetivos variados: avaliar qualidade, estimar tempo de desenvolvimento, avaliar confiabilidade. Tem-se que a utilização de métricas é um passo fundamental para controlar e melhorar os processos relacionados ao software (HUMPHREY, 1995). A

opção pela medição de defeitos como forma de controlar quantitativamente a qualidade dos produtos deve-se ao fato dos defeitos constituírem um dos poucos critérios de qualidade de medição objetiva, da taxa de defeitos influir diretamente na percepção de qualidade do produto pelos usuários e pela contagem de defeitos constituir a base para outros indicadores de qualidade como confiabilidade, correção, completeza, eficiência e usabilidade.

Na literatura existem diversos métodos e técnicas que auxiliam na definição e implementação do processo de medição, dentre os quais destacam-se o GQM (*Goal-Question-Metric*) (BASILI, CALDIERA e ROMBACH, 1994), o *Goal-Driven Software Measurement* (PARK, GOETHERT e FLORAC, 1996) e o PSM (*Practical Software Measurement*) (MCGARRY, 2001). Esses métodos e modelos, em geral, apóiam a seleção de medidas e a estruturação do processo de medição.

Deve-se observar que, apesar da importância do teste do software e da medição dos defeitos resultantes desses testes para a garantia da qualidade do produto, o processo de medição do teste é abordado, geralmente, em separado da atividade de teste o que torna oportuno um estudo que mostre a aplicabilidade conjunta desses conceitos.

## **1.2 Objetivo**

O objetivo desse trabalho é propor um processo de medição de defeitos do software orientado a testes de software, utilizando a técnica do PSM (*Practical Software Measurement*) visando a melhoria do processo de teste.

Essa proposta aborda o planejamento, a definição e avaliação da medição em função dos dados coletados em atividades de teste. Para apoiar a elaboração do processo de medição é utilizado o PSM, uma técnica que auxilia a estruturação da atividade de medição em projetos de software.

## **1.3 Contribuições e Resultados**

Na literatura, geralmente, os conceitos de teste de software e de processo de medição são tratados em separado. O processo proposto neste trabalho apresenta uma aplicação conjunta dos conceitos de teste de software e processo de medição e mostra como as recomendações consolidadas pelo PSM podem contribuir academicamente na elaboração de um processo estruturado de medição.

Considerando que há uma grande dificuldade por parte dos profissionais da área de desenvolvimento de software na elaboração de testes e medições de software, um importante resultado é fornecer um processo que apóie esses profissionais na medição dos resultados dos testes de software.

Estudos de caso são realizados com base nos processos de teste de software de duas empresas, efetuando-se a análise da aplicação do processo proposto e a avaliação de como as informações obtidas com a aplicação desse processo contribuem para a melhoria do processo de teste.

#### **1.4 Metodologia**

A base para elaboração deste trabalho é a análise de diversas referências bibliográficas relacionadas ao teste e medição de software como livros, dissertações e teses, artigos de congressos internacionais e diversos *web sites* da Internet.

No estudo das referências bibliográficas as atividades foram divididas da seguinte forma:

- Revisão e síntese sobre o teste de software, com a finalidade de compreender as técnicas, as estratégias e os modelos de testes existentes na literatura;
- Revisão e síntese sobre a medição de software com o objetivo de entender os conceitos que norteiam a medição, dando ênfase ao que se refere à medição de defeitos;
- Análise e síntese dos modelos e técnicas de medição, GQM, o *Goal – Driven Software Measurement* e o PSM, com a finalidade de apurar as vantagens e desvantagens da adoção do PSM como o modelo a ser utilizado no presente trabalho.

Com base nas referências bibliográficas analisadas foi elaborado e descrito o processo de medição proposto, relacionando os conceitos de teste e medição de software.

Estudos de caso do processo proposto foram realizados, efetuando medições baseadas no processo de teste de duas empresas de desenvolvimento de software e avaliando os resultados obtidos com a aplicação do processo de medição proposto.

## 1.5 Organização do Trabalho

O trabalho está dividido em seis capítulos, cujos resumos são expostos a seguir.

O capítulo 2, Teste de Software, apresenta os conceitos básicos sobre teste de software, abordando as técnicas de teste, as estratégias de teste e o processo de teste.

O capítulo 3, Medição de Software, apresenta os conceitos básicos sobre a medição de software. Também, descreve os modelos e metodologias que apóiam a implementação do processo de medição.

O capítulo 4, Processo de Medição de Defeitos, descreve o processo proposto baseado nos conceitos de teste e medição de software abordados nos capítulos 2 e 3. Compreende o detalhamento do processo de medição de software, apresentando uma forma de utilização conjunta dos conceitos de teste de software e de processo de medição.

O capítulo 5, Estudo de Caso, apresenta dois exemplos práticos da aplicação do processo de medição proposto e a análise dos resultados obtidos com esse estudo. Cada fase da abordagem proposta é avaliada por meio da aplicação do processo em duas empresas de desenvolvimento de software.

O capítulo 6, Conclusões, apresenta a conclusão do trabalho desenvolvido, suas contribuições e sugestões para trabalhos futuros.



## CAPÍTULO 2 – TESTE DE SOFTWARE

Este capítulo apresenta a descrição dos principais conceitos, existentes na literatura, sobre o teste de software fornecendo subsídios para a elaboração do processo de medição proposto.

### 2.1 Conceitos Básicos

O processo de teste de software vem apoiar a engenharia de software, elaborando critérios, métricas, a fim de garantir a execução do teste ao longo do desenvolvimento do software de forma que a cada versão do produto liberada, seu nível de qualidade seja mais elevado (PRESSMAN, 2002).

Os programas executados nos primórdios da computação também tinham que ser testados e encontram-se menções ao teste de programas na literatura desde 1950. Ao longo da história, destacam-se quatro modelos de teste de software (GELPERIN e HETZEL, 1988), como mostrado na tabela 1.

Segundo Carvalho (2004), dentre as fases do ciclo de vida de um projeto definida pela engenharia de software, o teste representa, em média, de 30% a 50% do custo envolvido no desenvolvimento do software. Um defeito detectado nas fases finais do projeto representa um acréscimo de 60% no custo do projeto.

**Tabela 1 - Modelos de Teste de Software**

<b>Período</b>	<b>Modelo</b>	<b>Objetivo</b>
- 1956	Depuração	Todo o processo de identificação e correção de erros era descrito como teste de software.
1957 - 1978	Demonstração	Demonstrar que o software satisfaz suas especificações.
1979 – 1982	Destruição	Executar o software com a intenção de descobrir erros.
1983 – 1987	Avaliação	Detectar erros no conjunto de requisitos, no projeto e na implementação do software. Adição da atividade de teste ao longo de todo o ciclo de desenvolvimento do software.
1988 –	Prevenção	Prevenir erros no conjunto de requisitos, no projeto e na implementação do software.

Fonte: adaptado de Gelperin e Hetzel (1988)

A seguir, são apresentados alguns termos importantes para compreensão deste capítulo e as técnicas e as estratégias de software comumente abordadas na literatura.

## 2.2 Terminologia

Devido à existência de várias discussões no que tange à definição de erro, defeito e falha, é considerada neste trabalho a definição descrita na Norma IEEE 610 (1991), como mostrada na figura 1.

**Erro:** é uma ação humana que resulta em um defeito.

**Defeito:** uma etapa, processo ou definição de dados incorreta, por exemplo, uma instrução incorreta no código do software.

**Falha:** ocorre quando um programa apresenta algum problema. É uma consequência da existência de algum defeito no software que pode prejudicar sua capacidade de funcionar conforme pretendido.



Figura 1 - Relação entre Erro, Defeito e Falha

Outros termos utilizados, no presente trabalho, como: caso de teste, *drivers* e *stubs* também são definidos de acordo com a Norma IEEE 610 (1991).

**Caso de teste:** é um conjunto de dados de entrada, para um software ou componente, e o seu respectivo resultado esperado com a finalidade de exercitar um componente ou um software para verificar a conformidade com os requisitos especificados.

**Driver:** é um programa que aceita dados de casos de teste e transfere esses dados para o módulo a ser testado recebendo as respostas.

**Stubs:** servem para substituir módulos que estejam subordinados ao módulo sob teste. Um *stub* é um programa simulador que implementa a interface do módulo subordinado, valida os dados de entrada recebidos, realiza alguma manipulação mínima dos dados de entrada, gera os dados esperados e retorna para o módulo sob teste.

## 2.3 Técnicas de Teste de Software

As técnicas de teste são utilizadas para apoiar de forma sistemática a construção dos casos de teste, oferecendo um mecanismo de auxílio na revelação de defeitos no software.

Segundo Pressman (2006), “um software pode ser testado de duas maneiras: (1) verificando se as funções do software são operacionais, abordagem que trata as entradas, saídas e princípios funcionais de um módulo, e (2) conhecendo-se o funcionamento interno do software, abordagem que trata da estrutura de código de um módulo de software. A primeira abordagem é chamada de **teste funcional** ou **caixa preta** e a segunda, **teste estrutural** ou **caixa branca**”.

Deve-se observar que essas técnicas de teste não são mutuamente exclusivas, mas, sim, complementares e, portanto, devem ser usadas em conjunto.

### 2.3.1 Teste Estrutural (Caixa Branca)

Os testes estruturais ou de caixa branca referem-se a um minucioso exame dos detalhes internos do software. Usando-se técnicas de teste de caixa branca podem-se derivar casos de teste que assegurem que todos os caminhos de execução existentes no software sejam executados pelo menos uma vez e que exercitem as estruturas de dados internas, todos os laços e todas as condições lógicas para valores verdadeiros e falsos. Os casos de teste são elaborados a partir do conhecimento da estrutura e implementação do software.

No teste estrutural, as técnicas de teste dividem-se em dois grupos: testes de caminho básico e os testes de controle. O teste de **caminho básico** (PRESSMAN, 2006) possibilita medir a complexidade lógica de um programa e utilizar esta medida como referência para definir um conjunto básico de caminhos de execução, garantindo que todas as instruções do programa sejam executadas pelo menos uma vez.

Os testes de estrutura de controle podem ser combinados com o teste de caminho básico. Esses testes são divididos em: **teste de condição** (PRESSMAN, 2006) que se concentra em testar cada condição lógica contida num módulo de programa e **teste de laços** que se concentra exclusivamente na validade das construções de laços (*loops*).

Segundo Humphrey (1995), entre as técnicas de teste, a técnica estrutural tem a mais alta capacidade de detectar defeitos. Um estudo efetuado mostrou que, na média, a técnica estrutural é capaz de detectar 72,9% de todos os defeitos encontrados por usuários de um grande sistema. No entanto, essa técnica possui algumas limitações e desvantagens como o problema de caminhos ausentes. Quando uma funcionalidade não é implementada no programa, não existe um caminho que corresponda àquela funcionalidade e, portanto, nenhum caso de teste será requerido para exercitá-la. Para identificar a ausência dessa funcionalidade é necessário utilizar outra técnica de teste, mais especificamente a técnica de teste funcional.

### 2.3.2 Teste Funcional (Caixa Preta)

Os testes funcionais ou de caixa preta têm a finalidade de avaliar o comportamento do software, testando a cobertura das suas funcionalidades e verificando se cada função opera segundo a definição de requisitos de software sem se preocupar com os detalhes de implementação.

Essa técnica permite a identificação de defeitos como: funções incorretas ou ausentes, defeitos de interface, defeitos na estrutura de dados ou no acesso a dados externos, baixo desempenho e defeitos de iniciação ou finalização.

Os métodos mais conhecidos de teste funcional são particionamento de equivalência e análise do valor limite. O teste de **particionamento de equivalência** (PRESSMAN, 2006) propõe que os dados de entrada do software sejam divididos em classes ou partições de equivalência válidas e inválidas, que possuem características comuns, como por exemplo, números positivos e negativos. Já o teste de **análise do valor limite** (PRESSMAN, 2006), é uma técnica de teste que complementa o particionamento de equivalência. Um exemplo seria uma condição de entrada especificando um intervalo para os dados de entrada: casos de teste devem ser projetados com os valores mínimo e máximo desse intervalo.

Para a definição do nível de teste ou fase do desenvolvimento do software na qual o teste será aplicado, a técnica de teste que será utilizada, o critério de teste a ser adotado e o tipo de teste a ser aplicado no software é necessário elaborar uma estratégia de teste.

## 2.4 Estratégias de Teste de Software

Segundo Pressman (2006), uma estratégia de teste de software integra técnicas de projeto de casos de teste numa série definida de passos, resultando na construção bem sucedida do software. Uma estratégia compreende os seguintes níveis de teste: de unidade, de integração e de sistema.

### 2.4.1 Teste de Unidade

O principal objetivo do teste de unidade é detectar defeitos de implementação nos módulos individuais que constituem o sistema. Os módulos devem ser testados, de forma a assegurar, que todas as funcionalidades e especificações de projeto, implementadas por um determinado módulo, tenham sido implementadas de maneira correta.

Os testes de unidade baseiam-se nos testes de caixa branca, verificando os caminhos básicos por meio das estruturas de controle. Geralmente esses testes, são efetuados pelo programador podendo-se utilizar ferramentas de depuração de código fonte.

Como, freqüentemente, um módulo não é um programa individual, o desenvolvimento de um software *driver* e/ou *stubs* podem ser necessários para a execução dos testes de unidade (CARVALHO, 2004).

### 2.4.2 Teste de Integração

Geralmente, um sistema é constituído de vários módulos que se comunicam por meio de interfaces bem definidas. Os testes de integração baseiam-se na técnica de teste de caixa preta enfatizando as interfaces existentes. Com o objetivo de validar o funcionamento das interfaces de comunicação entre os vários módulos que constituem o sistema, o teste de integração explicita efeitos adversos de um módulo sobre outro, apresentando, assim, uma oportunidade para a avaliação dos mecanismos implementados para o tratamento de defeitos. Esses mecanismos testam todos os dados que chegam de outros módulos do sistema, visando assegurar a robustez do sistema como um todo.

O teste de integração deve ser realizado de forma incremental, testando e integrando os módulos em pequenos grupos facilitando o isolamento das prováveis

causas dos defeitos detectados e, conseqüentemente, a correção desses defeitos (MYERS, 1979). A integração dos módulos pode ser feita sob três tipos de estratégias:

**Teste de integração de Baixo para Cima (*bottom up*):** inicia-se com a aplicação de testes aos módulos de nível inferior passando para os módulos superiores na hierarquia estrutural do software até que os módulos de mais alto nível sejam testados. Quando os módulos de níveis superiores são testados, os módulos subordinados já estão prontos e, portanto, não é necessária a construção de *stubs*.

**Teste de Integração de Cima para Baixo (*top down*):** inicia-se pelo programa principal e uma ou duas rotinas subordinadas. A vantagem dessa abordagem é a possibilidade de iniciar os testes com os módulos principais – sem a necessidade de utilização de *drivers*. A desvantagem é a necessidade de utilizar *stubs* nos testes para simular o funcionamento dos módulos em um nível inferior ao dos módulos sob teste.

**Teste de Integração Misto:** é utilizado, geralmente, quando não é possível aderir adequadamente à abordagem de teste de integração de cima para baixo. Um exemplo seria quando não é possível utilizar *stubs* para simular módulos de níveis inferiores sendo necessário testar, antes de tudo, alguns módulos inferiores críticos para a operação do sistema (ROTTA, 2001).

### 2.4.3 Teste de Sistema

Os testes de sistema envolvem uma série de diferentes testes cuja finalidade é executar por completo o software para validar seus aspectos funcionais (PRESSMAN, 2006). São realizados após a execução do teste de unidade e de integração, onde o software é testado em conjunto com outros sistemas de software e elementos de hardware com os quais implementa a interface. O objetivo principal do teste de sistema é por à prova todo o sistema tal e qual ele será implantado e colocado em operação (ROTTA, 2001). Assim sendo, o teste de sistema constitui-se de várias categorias de teste:

**Teste de Recuperação:** força o software a falhar de diversos modos e verifica se a recuperação é adequadamente realizada (PRESSMAN, 2006).

**Teste de Segurança:** verifica se todos os mecanismos de proteção incorporados em um software realmente o protegem de invasão imprópria (PRESSMAN, 2006).

**Teste de Estresse:** verifica até que ponto pode-se utilizar o software antes que falhe utilizando recursos em quantidade, freqüência ou volumes anormais (PRESSMAN, 2006).

**Teste de Desempenho:** projetado para testar o desempenho do software durante a execução no contexto de um sistema integrado (PRESSMAN, 2006).

Em suma, a elaboração de uma estratégia de teste compreende a definição dos seguintes itens: do nível de teste, que corresponde à definição da fase de desenvolvimento em que será aplicado o teste; da técnica de teste a ser utilizada; do critério de teste a ser adotado e do tipo de teste a ser aplicado no software. Portanto, a elaboração de uma estratégia de teste abrange a definição de quando deve ser efetuado o teste, do que deve ser testado e como deve ser efetuado o teste.

A definição de quando deve ser efetuado o teste envolve a seleção dos níveis de teste de acordo com a fase de desenvolvimento. Um exemplo seria o teste de unidade que é aplicado na fase de codificação. Já a definição do que deve ser testado compreende a escolha dos tipos de teste a serem utilizados como os testes de interface ou de desempenho, por exemplo. Os tipos de teste determinam as características do software que podem ser testadas (SILVA et al., 2004). A definição de como serão efetuados os testes abrange a seleção de técnicas de teste de acordo com o nível de teste adotado. As técnicas de teste são, basicamente, duas: a estrutural, que identifica defeitos na estrutura interna do código e a funcional que tem a finalidade de garantir que o software atende aos requisitos. Ao se adotar técnicas de teste é necessário definir os critérios de teste que serão utilizados. Os critérios de teste correspondem aos elementos ou as características do software que deverão ser executadas no teste (SILVA et al., 2004).

Na figura 2, é mostrada a relação entre os níveis de teste, as técnicas de teste, os tipos de teste e os critérios de teste que podem ser adotados na elaboração de uma estratégia de teste.

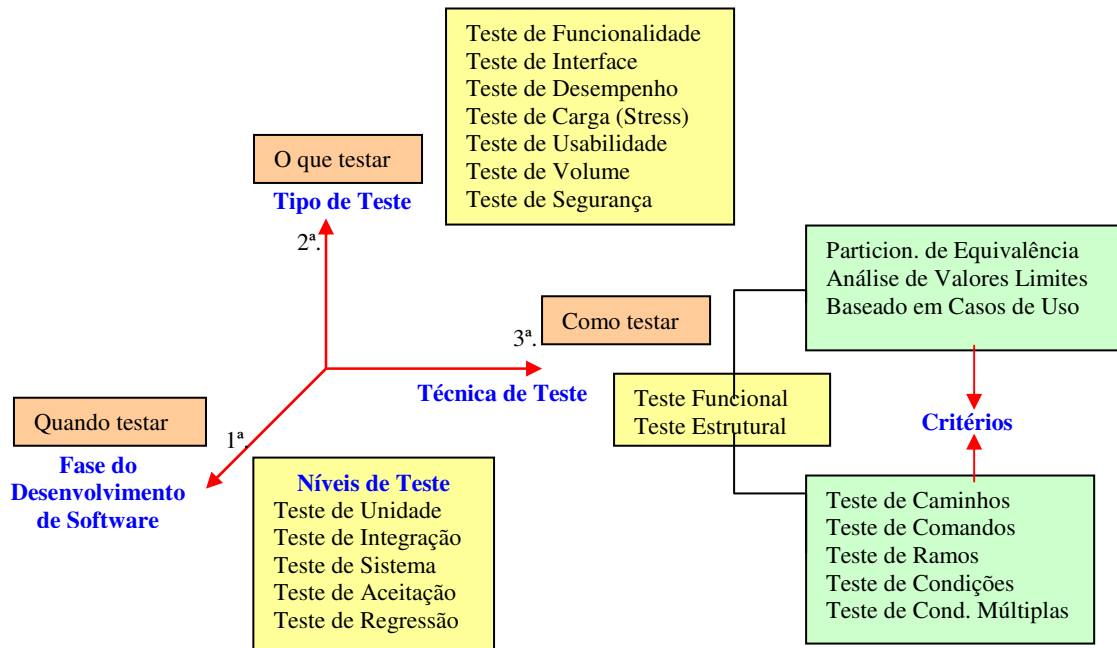


Figura 2 – Relação entre Níveis, Tipos, Técnicas e Estratégias de Teste

Fonte: Silva et al. (2004)

A estratégia de teste deve ser definida na fase de planejamento do processo de teste.

## 2.5 Processo de Teste

O Processo de Teste é caracterizado por um conjunto de atividades de teste que são executadas ao longo de todo o ciclo de desenvolvimento do software. Essas atividades estão agrupadas em etapas bem definidas, que são:

### a) Planejamento

Nesta etapa é descrito o escopo dos testes, identificando métodos que serão empregados, recursos necessários e o cronograma das atividades de teste. Os itens e as funcionalidades a serem testadas, as tarefas a serem realizadas e os riscos associados com a atividade de teste são identificados como partes do Plano de Teste.

### b) Projeto

Nesta etapa, é escolhido um grupo específico de características a serem testadas, descrevendo-se detalhadamente os métodos que serão empregados e os testes que serão efetuados. Essas informações são detalhadas no Projeto de Teste.



### **c) Implementação**

Nesta etapa, são detalhados os itens que serão testados, as entradas utilizadas, as saídas esperadas e os passos necessários para se executar os testes. Como resultado, temos o Caso de Teste e o Procedimento de Teste.

### **d) Execução**

Nesta etapa, a partir do procedimento de teste e dos casos de teste, são realizados os testes. Ao final dessa atividade, um resumo é fornecido referindo os itens testados, sumarizando os resultados obtidos e fornecendo uma avaliação baseada nesses resultados. Nessa etapa são gerados: o Relatório de Envio de Item de Teste, o Diário de Registro de Teste, o Relatório de Incidentes de Teste e o Relatório de Resumo de Teste.

As etapas de planejamento, projeto e implementação correspondem a atividade de preparação dos testes. Com base na documentação gerada nestas etapas são executados os testes, que geram resultados que devem ser registrados. É importante ressaltar a necessidade do registro do teste, em particular dos defeitos encontrados, considerando que a análise quantitativa dessas informações permite o acompanhamento dos projetos e a medição da qualidade do produto.

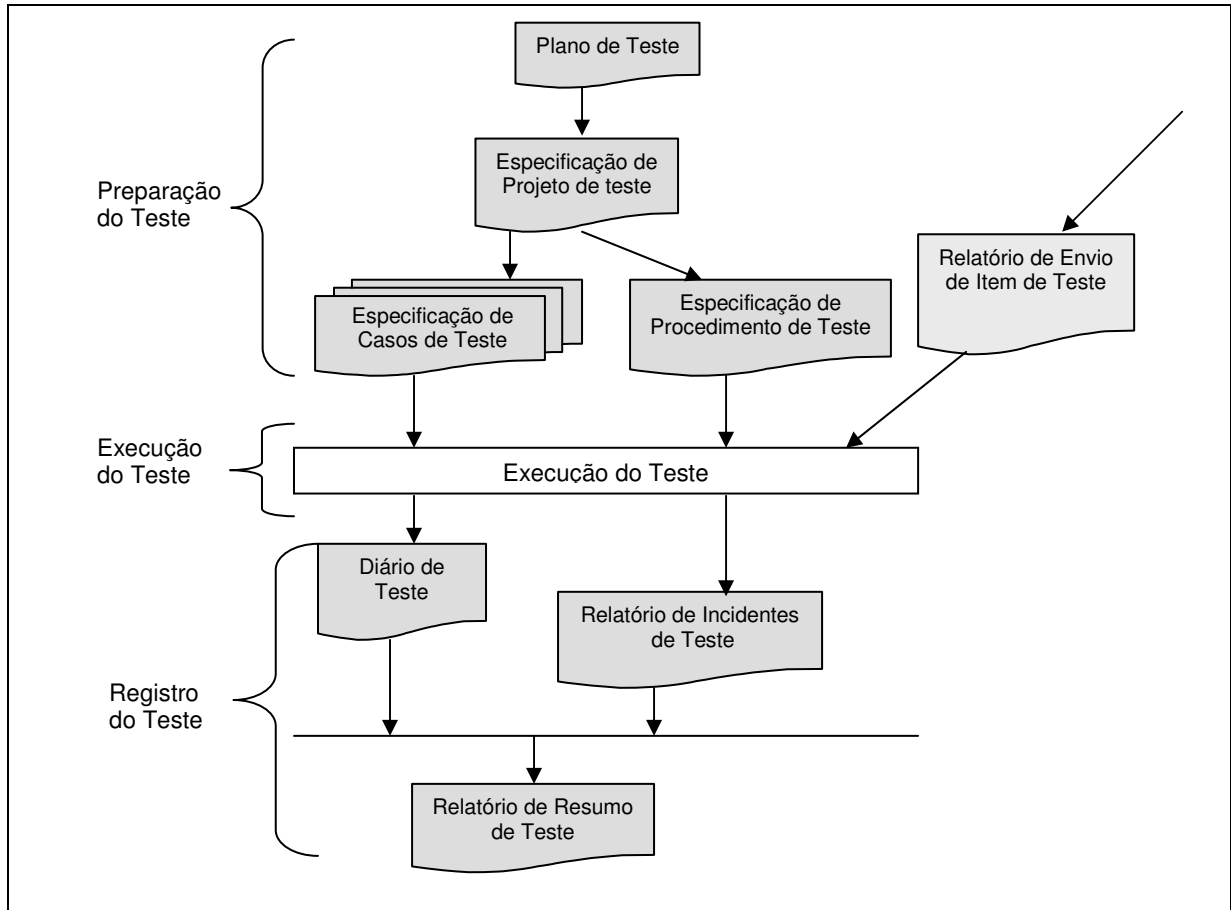
## **2.6 Norma IEEE 829**

A Norma IEEE 829 (1983) descreve um conjunto básico de documentos para serem utilizados nas atividades de teste de software, como mostra a figura 3. Os documentos definidos pela norma cobrem as tarefas de planejamento, especificação e relato dos testes. Estes documentos são agrupados da seguinte forma:

- a) Plano de Teste;
- b) Especificação de Teste;
- c) Relatório de Teste.

O Plano de Teste prescreve o escopo, abordagem, recursos e cronograma das atividades de teste e identifica os itens a serem testados, as características que serão testadas, as atividades de teste a serem executadas, os responsáveis por

cada atividade e os riscos associados ao plano. Esse documento é utilizado na etapa de Planejamento do Teste.



**Figura 3 – Relação entre Etapas e Documentos do Processo de Teste.**

Fonte: adaptado da Norma IEEE 829 (1983)

A Especificação de Teste abrange três tipos de documentos:

a) Especificação do Projeto de Teste - refina a abordagem de teste e identifica os casos de teste, procedimentos de teste e os critérios de sucesso e de falha.

b) Especificação de Casos de Teste – documenta os valores usados para entradas e as respectivas saídas. Um caso de teste pode também identificar as restrições no procedimento de teste, resultado do uso de um caso de teste específico. Casos de teste devem ser separados do projeto de teste, para habilitar o uso em mais de um projeto e permitir o reuso em outras situações.

c) Especificação do Procedimento de Teste – identifica todos os passos requeridos para operar o sistema e exercitar um caso de teste especificado.

O Relatório de Teste abrange quatro tipos de documentos:

a) Relatório de Envio de Item de Teste – identifica os itens de teste que estão sendo enviados para teste.

b) Diário de Registro de Teste – um registro cronológico sobre detalhes relevantes na execução dos testes.

c) Relatório de Incidentes de Teste – descreve qualquer evento ocorrido durante a execução dos testes e que mereça investigação adicional.

d) Relatório de Resumo – descreve as atividades de teste associadas com uma ou mais especificações de projeto.

Segundo a Norma IEEE 829 (1983), o Plano de Teste deve ter a seguinte estrutura:

1. Identificação do plano de teste;
2. Introdução;
3. Itens de teste;
4. Características a serem testadas;
5. Características que não serão testadas;
6. Abordagem;
7. Critério de sucesso/falha;
8. Critério de suspensão/recomeço;
9. Documentos elaborados;
10. Atividades de teste;
11. Necessidades de ambiente;
12. Responsabilidades;
13. Equipe e treinamento necessários;
14. Cronograma;
15. Riscos e contingências;
16. Aprovação.

Na tabela 2, é apresentado o detalhamento dos itens contidos no plano de teste.

Tabela 2 - Detalhamento do Plano de Teste

Item do Plano	Descrição
Identificação do plano de teste; Introdução; Itens de teste; Características a serem testadas; Características que não serão testadas.	Estas seções definem o escopo do sistema que será testado.
Abordagem	Descreve a visão de alto nível da estratégia técnica e do processo de teste. Para cada grupo de funcionalidades principais deve-se especificar uma abordagem para garantir que esses grupos de funções serão adequadamente testados.
Critérios de sucesso / falha	Especifica os critérios que serão usados para determinar se cada item de teste passou ou falhou no teste.
Critérios de suspensão/recomeço	Especifica os critérios que serão usados para suspender total ou parcialmente as atividades de teste e quais atividades devem ser repetidas no caso de suspensão.
Documentos elaborados	Resume e incorpora todas as referências do plano de teste, identificando os documentos.
Atividades de teste	Define a estrutura de trabalho para o projeto de teste.
Necessidades de ambiente	Define os recursos de hardware e software necessários à execução dos testes.
Responsabilidades	Define quem são os responsáveis por cada parte do processo de teste.
Equipe e treinamento necessários.	Especifica os perfis necessários e as necessidades de treinamento para testadores/desenvolvedores.
Cronograma	Data para início e estimativa de prazo para as atividades de teste.
Riscos e contingências	Identifica suposições de alto risco no plano de teste.
Aprovações	Especifica o nome e o cargo de todas as pessoas que deverão aprovar o plano.

Fonte: adaptado da Norma IEEE 829 (1983)

Na literatura há várias referências sobre o teste de software como Pressman (2006), Myers (1979) e Gelperin e Hetzel (1988) que abordam os conceitos sobre o teste no que tange às estratégias, técnicas, processo e modelos de teste.

Cabe destacar o trabalho de Rotta (2001), que propõe um método de teste de software baseado na execução de casos de teste e que segue as recomendações propostas pela Norma IEEE 829 (1983) no que tange à documentação do teste. Apresenta a aplicabilidade dos casos de testes por meio da utilização das estratégias de teste de unidade, de integração e de sistema.

Apresenta, também, os benefícios obtidos com o teste baseado em casos de teste considerando sua eficiência, aplicabilidade e capacidade de identificar os erros mais comuns cometidos pela equipe de desenvolvimento.

O método proposto por Rotta (2001) é dividido em cinco fases: planejamento, projeto, preparação de infra-estrutura, execução e análise do teste.

Cada fase é dividida em atividades específicas. Na fase de planejamento é gerado o plano de teste, contendo: a especificação do teste, a definição da estratégia de teste a ser utilizada, a descrição do ambiente de teste e das funcionalidades e tarefas, *drivers* e *stubs* que serão utilizados.

Na fase de projeto, o plano de teste definido na fase de planejamento é detalhado por meio dos casos e procedimentos de teste, descrevendo as entradas e saídas que serão consideradas na atividade de teste.

Na fase de preparação de infra-estrutura são efetuadas a montagem e instalação do ambiente de teste e o desenvolvimento dos *drivers* e *stubs* necessários. A fase de execução consiste em aplicar os casos de teste e registrar os resultados obtidos que serão analisados na fase de análise do teste.

Esse método de teste é estruturado e segue a documentação recomendada pela Norma IEEE 829 (1983) que fornece informações sobre as estratégias de teste adotadas e prevê o registro dos defeitos obtidos no teste. Portanto, as fases e documentações utilizadas no método de teste, proposto por Rotta (2001), são a base para a construção do processo de teste utilizado como referência no presente trabalho.

Neste capítulo foram apresentados os conceitos, existentes na literatura, sobre o teste de software abordando as técnicas, as estratégias e o processo de teste. Ressalta-se a importância do teste de software como um mecanismo que auxilia na detecção de defeitos no produto por meio do uso de técnicas e estratégias de teste definidas, de acordo com as características e a fase de desenvolvimento do produto. Cabe observar que o registro dos defeitos, encontrados no teste, é de suma importância considerando que a coleta, o armazenamento e a análise quantitativa destes defeitos é crucial para o controle do projeto e para a avaliação da qualidade do produto. Para analisar os defeitos encontrados nos testes, de forma a agregar valor para a organização, é necessário avaliar quantitativamente os resultados obtidos por meio da medição de software.

Nesse ponto o teste e a medição são complementares, o teste fornece os dados sobre o produto que, analisados com base em métricas definidas pelo processo de medição, permitem que a organização faça ajustes no seu processo de desenvolvimento de software para remover as causas de defeitos que

prejudicam a qualidade do produto e do processo. A seguir, são apresentados os conceitos sobre medição de software.

## CAPÍTULO 3 - MEDIÇÃO DE SOFTWARE

Este capítulo apresenta a descrição dos principais conceitos e recomendações, existentes na literatura, sobre a medição de software, fornecendo subsídios para a elaboração do processo de medição proposto.

### 3.1 Conceitos Básicos

Segundo Schneidewind (1992), a medição pode ser vista como o processo de definir, coletar, analisar e agir sobre medidas que possam potencialmente melhorar tanto a qualidade do software desenvolvido por uma organização quanto o próprio processo de desenvolvimento utilizado. A medição produz como resultado um conjunto de medidas. Bush e Fenton (1990, apud ALMENDRA, 2003), definem uma medida como “uma atribuição objetiva empírica de um número (ou símbolo) a uma entidade que caracteriza um atributo<sup>1</sup> particular da entidade”. Na métrica são estabelecidas unidades de medida para convencionar como esses atributos devem ser registrados.

Para medir e prever aspectos dos processos, recursos e dos produtos relevantes para a atividade de engenharia de software são utilizadas métricas de software no processo de medição. As métricas de software envolvem por um lado a caracterização numérica (medidas) de atributos, de processos e de recursos; por outro lado, também envolvem a predição (estimativas) de atributos desses três elementos em uma situação baseada nos dados disponíveis no presente. Métricas de projeto e de processo permitem ao pessoal de software ter noção da eficácia do processo de software e de projetos que são conduzidos usando o processo como base.

As medições podem ser divididas em duas categorias: medição direta e medição indireta. Segundo Bush e Fenton (1990, apud ALMENDRA, 2003), na medição direta o atributo de interesse é medido sem necessidade de recorrer a outros atributos. Um exemplo de medição direta seria a medição do atributo cor que pode ser classificado apenas pela observação do objeto. Já a medição indireta é efetuada recorrendo-se a outros atributos, como por exemplo, a qualidade do

---

<sup>1</sup> Atributo é uma propriedade da entidade como o tamanho ou a funcionalidade (de uma especificação). Entidade pode ser uma pessoa, um objeto (uma especificação) ou um evento (um teste executado sobre um produto).

produto que possui vários fatores que influenciam a sua definição da qualidade do produto.

Segundo Schneidewind (1992), as métricas de software são utilizadas para avaliar, controlar e efetuar previsões de atributos relacionados com software. Essa utilização se dá dentro do contexto de um processo de medição, como mostrado no item 3.2.

### **3.2 Diretrizes e Recomendações para o Processo de Medição**

A definição e implementação de um processo de medição é uma tarefa complexa, requer a seleção de métricas relevantes que atendam as necessidades da organização. Para auxiliar esse trabalho de definição do processo e escolha das métricas encontram-se na literatura algumas técnicas para o processo de medição, tais como, o GQM (*Goal-Question-Metric*), *Goal-Driven Software Measurement* e o PSM (*Practical Software Measurement*).

#### **3.2.1 GQM**

O GQM (*Goal-Question-Metric*) é utilizado para guiar o processo de seleção das medidas. Foi originalmente proposto por Basili, Caldiera e Rombach (1994) para a caracterização e a avaliação de defeitos de projetos desenvolvidos no laboratório de engenharia de software da NASA.

Essa técnica, como mostra a figura 4, propõe uma visão *top-down* para o processo de medição partindo dos objetivos que se deseja atingir com o uso de medição e chegando às métricas. Segundo Borges (2003), o ponto de partida para a escolha das medidas é, portanto, a definição das metas que guiarão o restante do processo. De forma resumida, o GQM descreve três passos que devem ser seguidos:

- Definição de um conjunto de metas (*goal*) (por exemplo, satisfação dos usuários) que se pretende atingir quanto ao processo e aos produtos com as atividades de medição;
- Elaboração de questões (*question*) que, se respondidas, ajudariam a organização a atingir as metas estabelecidas;



- Especificação das métricas (*metric*) que definem as medidas que serão coletadas de forma a obter as informações necessárias para responder as perguntas geradas.

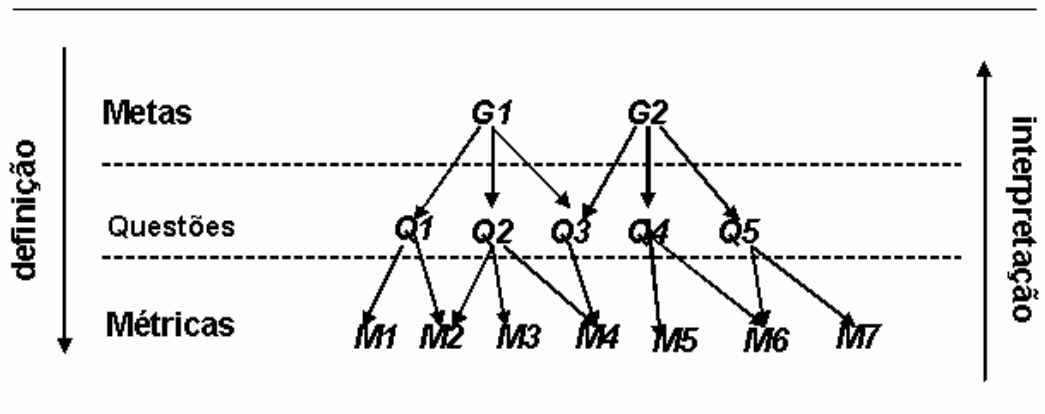


Figura 4 - GQM  
Fonte: Borges (2003)

Quando o método proposto pelo GQM é seguido corretamente, proporciona benefícios como a garantia de que as métricas estão alinhadas com as metas da organização, o motivo da coleta de cada uma das medidas é claramente estabelecido e o rastreamento entre as metas e métricas, se devidamente documentados, serve como base para futuras adaptações e melhorias do processo de medição.

Na tabela 3, é apresentada a estrutura do processo GQM. Esse processo é dividido em três etapas que são: o desenvolvimento do plano GQM, a execução do plano GQM e a preparação dos resultados.

Tabela 3 - Processo GQM

<b>1. Etapa</b> Desenvolvimento do Plano GQM.	<b>1.1. Pré-Estudo</b> Identifica as pré-condições para a introdução do programa de medição.
	<b>1.2. Elaboração do Plano GQM</b> <b>1.2.1 Identificação das metas da medição</b> Essas metas servirão de base para a elaboração do plano GQM. Envolve os seguintes aspectos: a) <u>Objeto de estudo</u> : processo ou produto; b) <u>Propósito</u> : Especifica o objetivo alvo da medição e a razão da coleta. Exemplo: determinar ou caracterizar um aspecto de qualidade do objeto de medida. c) <u>Perspectiva</u> : define o aspecto a ser utilizado como referência (custo, correção e remoção de defeitos são alguns exemplos) e o ponto de vista que será levado em consideração, que pode ser o dos usuários, dos clientes, por exemplo. d) <u>Ambiente</u> : fornece o contexto para a interpretação dos resultados, registrando as características do ambiente em que as medições serão realizadas.
	<b>1.2.2 Preparação do Plano GQM</b> Elaboração de um conjunto de questões, caracterizando o objeto de medição e seus aspectos de qualidade, assim como um conjunto de métricas associadas às questões que visam respondê-las de maneira quantitativa.
	<b>1.2.3 Elaboração do Plano de Medição</b> Definição de quando e como os dados serão coletados e quais os responsáveis pela coleta.
<b>2. Etapa</b> Execução do Plano de Medição.	<b>2.1. Coleta de dados</b> Obter dados de avaliação, por meio do preenchimento do questionário elaborado. Esses dados são validados por meio da verificação da não ocorrência de erros, consistência e completude.
	<b>2.2. Tratamento dos dados</b> Análise e interpretação dos dados para que as informações obtidas durante a coleta sejam utilizadas, de forma a responder às questões definidas. A interpretação dos dados da aplicação do GQM deve estar de acordo com as metas e as métricas definidas anteriormente.
<b>3. Etapa:</b> Preparação dos Resultados	<b>3.1. Preparação da documentação final.</b> É desenvolvido um documento contendo todos os dados coletados durante a medição.
	<b>3.2. Composição da base de experiências.</b> Armazena as experiências adquiridas no programa de medição para reutilizar esse conhecimento em projetos de software e programas de medição futuros.

Fonte: adaptado de Gladcheff, Sanches e Silva (2001).

### 3.2.2 Goal – Driven Software Measurement

O *Goal-Driven Software Measurement* (PARK, GOETHERT e FLORAC, 1996) foi elaborado em 1996 pelo SEI (*Software Engineering Institute*), na Universidade de Carnegie Mellon dos EUA, como uma proposta de extensão do GQM. Essa extensão praticamente transforma os princípios propostos pelo GQM em um processo completo para a definição de uma política de medição. Detalha passo a passo uma seqüência de atividades que produz, ao final, um conjunto de medidas adequado às necessidades de uma organização. As metas de negócio da organização devem representar objetivos estratégicos da organização, e não precisam estar diretamente relacionadas aos aspectos mensuráveis do processo.

No GQM, as medidas necessárias são deduzidas diretamente a partir das questões. No *Goal-Driven Software Measurement*, como mostrado na figura 5, são introduzidos os indicadores, que constituem um nível intermediário entre as questões e as métricas para auxiliar a identificação das métricas mais adequadas. Um indicador é uma representação de uma combinação de medidas em um formato específico (geralmente, listas, tabelas e gráficos), que se construído pode fornecer informações úteis para responder às perguntas derivadas das metas de medição. Essa versão adaptada do GQM é denominada GQ(I)M (*Goal-Question-Indicator-Metric*). Segundo Borges (2003), a utilização de indicadores é a principal diferença entre o GQM e o GQ(I)M utilizado no *Goal-Driven Software Measurement*.

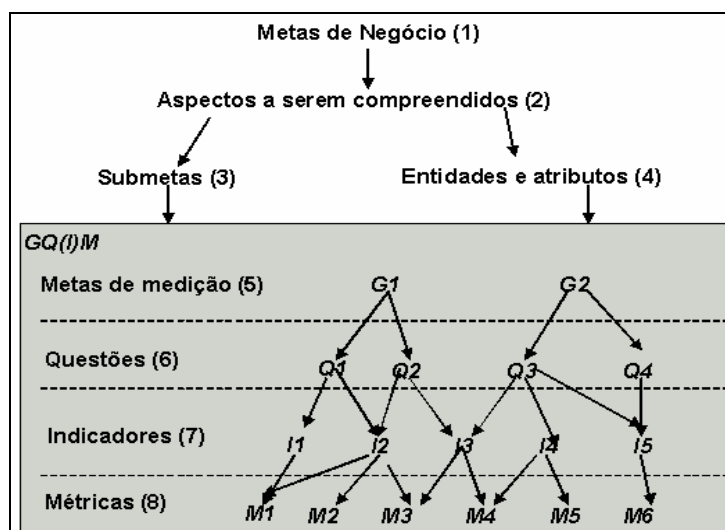


Figura 5 - Goal - Driven Software Measurement

Fonte: Borges (2003)

Como apresentado na tabela 4, o processo *Goal-Driven Software Measurement* é dividido em dez passos. Os quatro primeiros passos têm como objetivo gerar uma estrutura de informação que servirá como base para a definição das metas de medição.

**Tabela 4 - Processo Goal - Driven Software Measurement**

<p><b>1. Identificação de metas de negócio.</b> Metas de negócio devem representar objetivos estratégicos da organização e não precisam estar diretamente relacionados aos aspectos mensuráveis do processo. Exemplo: Aumento da Satisfação do Usuário.</p>
<p><b>2. Identificação do que se deseja aprender.</b> Formulação de questões que definam o que se pretende aprender ou conhecer. Exemplo: Os produtos são entregues no dia? Os produtos possuem níveis aceitáveis de defeitos?</p>
<p><b>3. Identificação de submetas que refinam as metas de negócio.</b> Agrupando em temas as questões, formuladas sobre as metas de negócio, é possível desdobrar a meta de negócio em outras mais específicas. Por exemplo: - Melhorar a confiabilidade e o desempenho do produto final; - Melhorar o acompanhamento do progresso em relação aos compromissos assumidos.</p>
<p><b>4. Identificação de entidades e atributos envolvidos.</b> Entidades podem ser: código-fonte, manual do usuário e defeitos. Atributos mensuráveis podem ser: tamanho, distribuição, frequência, tempo e esforço.</p>
<p><b>5. Formalização de metas de medição.</b> As metas formais são compostas por quatro componentes: a) <u>objeto de interesse</u>: pode ser um produto, processo, recurso, atividade, entre outros. É algo que se espera descrever ou saber mais a respeito; - <u>propósito</u>: pode ser prever, comparar, planejar, controlar, avaliar, entre outros. - <u>perspectiva</u>: identifica quem é interessado no resultado das medidas; - <u>ambiente e restrições</u>: provê um contexto para interpretação dos resultados.</p>
<p><b>6. Identificação de questões quantitativas e indicadores relacionados às metas da medição.</b> Identificação das questões e dos indicadores que serão usados para atingir os objetivos de medição.</p>
<p><b>7. Identificação de elementos de dados a serem coletados para a construção dos indicadores apontados.</b> Identificação das medidas a serem coletadas.</p>
<p><b>8. Definição e padronização das medições a serem realizadas e da forma de coleta dos dados.</b> Definição das medidas a serem utilizadas e da forma de coleta dos dados.</p>
<p><b>9. Identificação das ações necessárias para a implementação do processo de medição.</b> - <u>Análise</u>: envolve identificar as medidas que a organização está utilizando e entender como estão sendo coletados os dados; - <u>Diagnóstico</u>: avaliar os elementos de dados que a organização está coletando atualmente, determinando como se relaciona com as necessidades de suas medidas dirigidas ao objetivo e as ações propostas apropriadas; - <u>Ação</u>: significa traduzir os resultados das análises efetuadas e diagnosticar dentro de planos de implementação.</p>
<p><b>10. Preparação de um plano para a implementação do processo de medição.</b> Preparação de um plano para implementar as medidas.</p>

No quinto, sexto e sétimo passos são aplicados os princípios do GQM para a definição das metas considerando o propósito, a perspectiva e o ambiente. Os três passos restantes correspondem à definição das medidas, da forma de coleta dos dados e o planejamento da implantação do processo de medição na organização.

### **3.2.3 PSM - *Practical Software Measurement***

O PSM foi criado pelo Departamento de Defesa Norte-Americano, com o objetivo de estabelecer um conjunto de práticas, ferramentas e serviços para auxiliar os gerentes de projetos a obter informações objetivas sobre os projetos em andamento, para que estes atinjam suas metas de prazo, custo e qualidade (DOD, 2003).

A sua estratégia foi reunir as práticas de maior sucesso na área de medição de software para compor um processo único que possa ser utilizado como guia para implantação de uma política de medição em organizações.

O PSM foi publicado em 1997 e os seus princípios foram formalizados em um padrão ISO/IEC 15939 (2001).

Segundo Carvalho (2004), o PSM considera que as atividades de medição de software, para obterem sucesso, devem possuir duas características:

- Um alinhamento direto das atividades de coleta, análise e divulgação de dados medidos com as necessidades de informação dos responsáveis pela tomada de decisão do projeto. Nesse ponto, os princípios do PSM se assemelham aos do GQM, ressaltando a importância de se estabelecer de antemão os objetivos que se pretende atingir com a medição;
- A existência de um processo de medição, estruturado e bem documentado, que defina com precisão as atividades de medição.

A estrutura do PSM é baseada em dois modelos, cada um voltado para a obtenção de uma dessas características:

- Modelo de Informação: determina como as medidas a serem utilizadas devem ser especificadas e estruturadas para atender às necessidades de informação;
- Modelo de Processo: especifica como o processo de medição deve ser conduzido.

O Modelo de Informação é um mecanismo para mapear as necessidades de informação em termos dos atributos do produto ou processo que devem ser medidos. Segundo o PSM, a construção de uma medição deve descrever como os atributos serão quantificados e combinados para formar indicadores que forneçam a base para a tomada de decisão. Para isso, devem ser definidas em três níveis diferentes: medidas básicas (diretas), medidas derivadas (indiretas) e Indicadores.

O Modelo de Informação, como mostra a figura 6, constitui-se de alguns passos para formar o plano de medição, são eles:

- Necessidades de informação: identificação da informação a ser utilizada para a tomada de decisão;
- Conceito mensurável: identificação das entidades a serem medidas para satisfação da necessidade de informação.
- Entidades: pode ser uma pessoa, um objeto (uma especificação) ou um evento (um teste executado sobre um produto).
- Atributos: identificação das propriedades ou características das entidades que serão analisadas na medição. Atributos podem ser, por exemplo, o tamanho ou a funcionalidade de uma especificação.
- Construção mensurável: formalização de um conceito mensurável e especificação do que será medido e de como os dados serão trabalhados para satisfazer a necessidade de informação;
- Produto de informação: são os indicadores gerados a partir da medição efetuada.

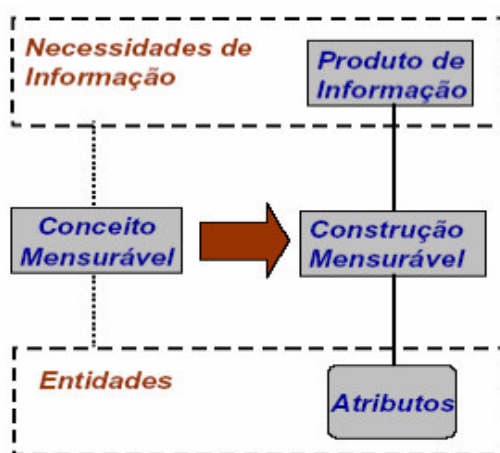
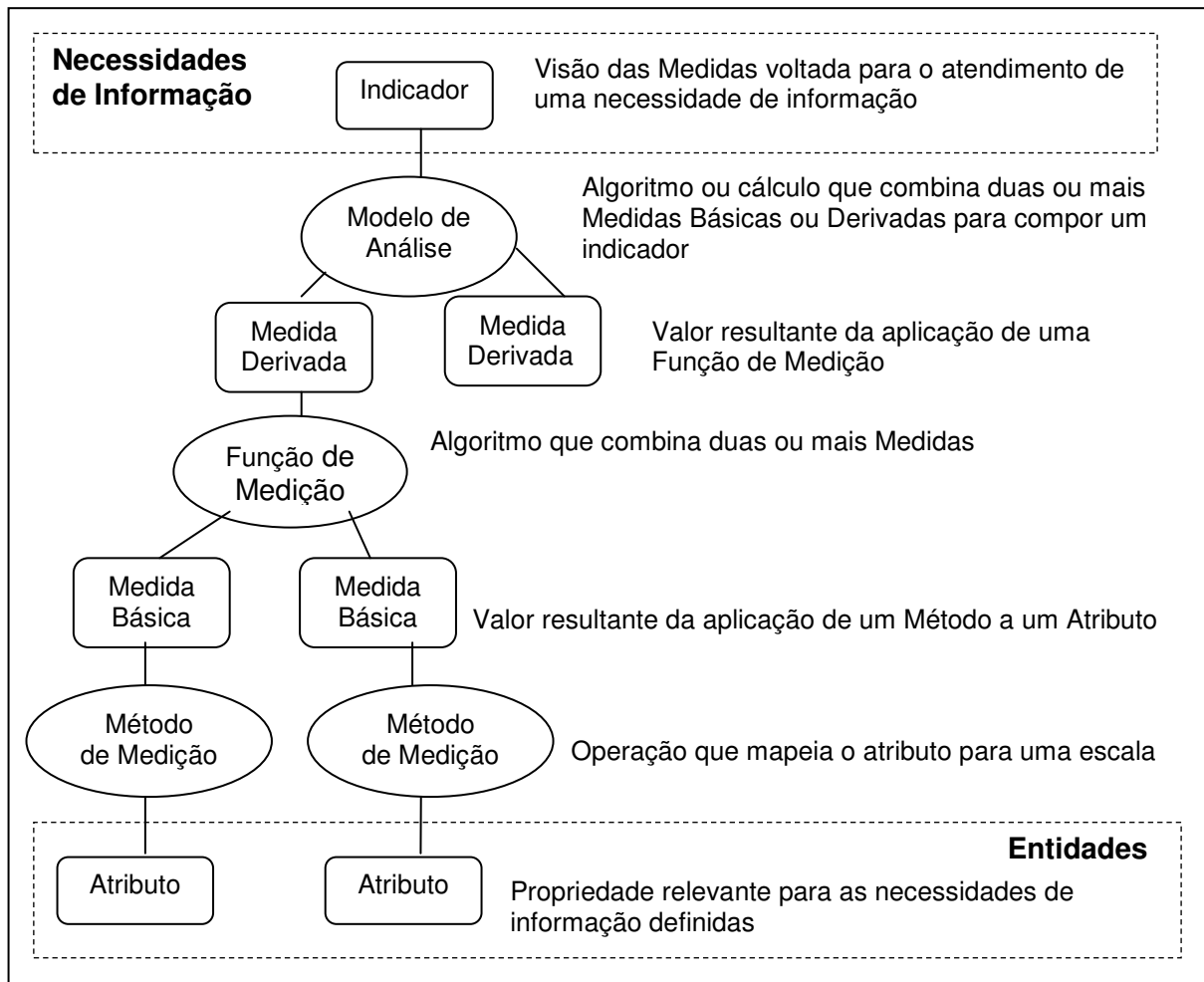


Figura 6 - Modelo de Informação – PSM

Fonte: McGarry (2001)

O detalhamento da construção de uma medição, com base no modelo de informação, é apresentado na figura 7. De acordo com as necessidades de informação identificadas são definidos os objetivos de medição. Para cada objetivo são estabelecidas questões que ao serem respondidas ajudarão no atendimento das necessidades de informação.

Para responder a essas questões são identificadas as entidades a serem medidas e as características ou atributos dessas entidades que serão considerados no momento da medição. Para medir cada atributo, são definidos os métodos de medição, que indicam como o atributo será medido. Com a aplicação do método de medição é obtido como resultado uma medida básica. Ao combinar duas ou mais medidas básicas por meio de uma função de medição obtêm-se uma medida derivada.



**Figura 7 - Construção de uma medição pelo PSM**

Fonte: McGarry (2001)

Com a aplicação de um modelo de medição, algoritmo ou cálculo, que combine duas ou mais medidas derivadas tem-se como resultado um indicador. O indicador é o produto da informação e ajudará no atendimento das necessidades de informação.

Enquanto o Modelo de Informação fornece uma estrutura para relacionar as necessidades de informação a um conjunto de medidas, o Modelo de Processo apresenta uma estrutura para a implementação da medição em um projeto, descrevendo as principais atividades que devem ser realizadas.

O Modelo de Processo é baseado em quatro atividades, sendo estas agrupadas em atividades centrais (previstas para obter a medida, que é o produto central do processo) e atividades não-centrais (mas indispensáveis à implantação do processo). Como mostra a figura 8, as atividades centrais são: planejar a medição e executar a medição. As atividades não-centrais são: estabelecer e sustentar comprometimento e avaliar a medição. Além dessas atividades há os processos técnicos e gerenciais que não estão no escopo do PSM, apenas indicam que este modelo de medição deve relacionar-se aos processos existentes nas organizações, viabilizando sua implantação.

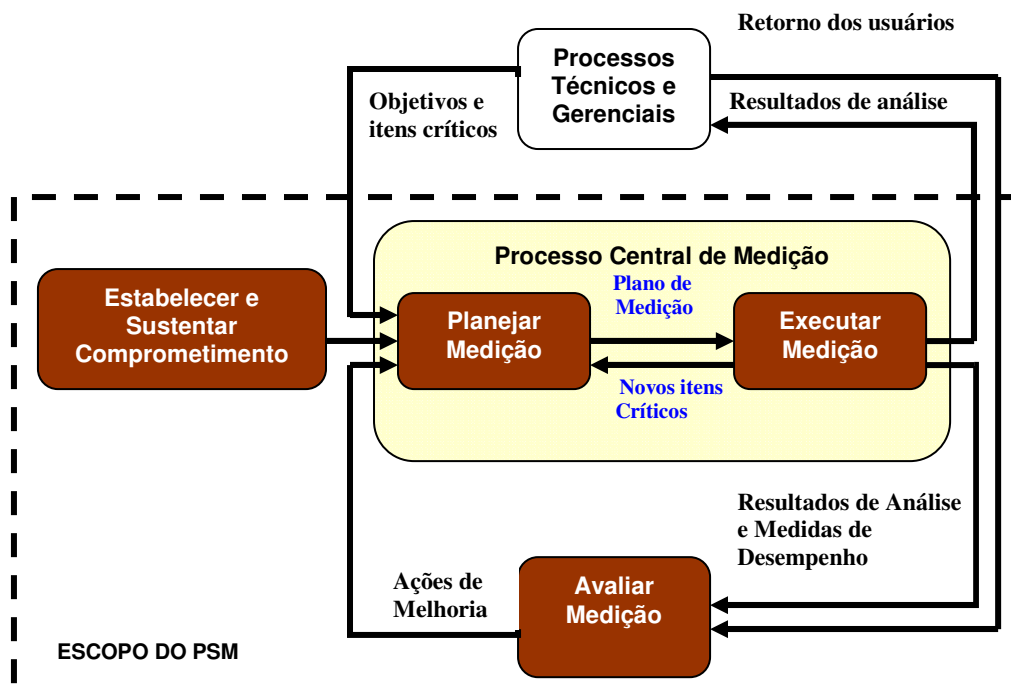


Figura 8 - Modelo de Processo - PSM

Fonte: McGarry (2001)



### **a) Estabelecer e sustentar comprometimento**

Nesta atividade define-se o fornecimento de recursos e infra-estrutura necessários à implementação e manutenção do programa de medição, garantindo o apoio organizacional ao processo.

- Obter comprometimento organizacional;
- Definir responsabilidades;
- Prover recursos;
- Rever progresso do programa de medição.

### **b) Planejar a medição**

Nesta atividade aplica-se o Modelo de Informação que envolve a identificação das necessidades de informação, a seleção das medidas mais adequadas para atender as necessidades, formalização dos procedimentos de coleta, análise e divulgação de dados, a definição de regras e convenções sobre a condução do processo de medição e o planejamento dos recursos e tecnologias a serem disponibilizadas para medição. Essa atividade é dividida nas seguintes sub-atividades:

- **Identificar e priorizar as necessidades de informação**

Nesta atividade são identificadas as informações a serem obtidas com a medição, categorizando e priorizando essas informações de acordo com o projeto. Consiste em:

- Identificar a necessidade de informação: as possíveis fontes para se identificar necessidades de informação de um projeto são avaliação de risco, suposições e restrições do projeto, utilização de tecnologias específicas, critérios de aceitação do produto, requisitos externos e experiência.
- Mapear as necessidades para as categorias definidas pelo PSM;
- Priorizar a necessidade de informação: é efetuada a análise do impacto que a falta da informação pode causar sobre a probabilidade de uma informação ser necessária para o projeto.

- **Selecionar e especificar medidas**

Nesta atividade são identificadas e descritas as medidas aplicáveis ao projeto, definidas com base nas necessidades de informação estabelecidas.

Consiste em:

- Identificar as categorias de medidas: identificação dos conceitos mensuráveis relacionados às necessidades e às categorias da informação. Normalmente, cada necessidade deverá gerar ao menos um conceito mensurável.<sup>2</sup>
- Mapear as áreas comuns do PSM com as categorias e medidas;
- Selecionar as categorias de medidas: formulação de questões que podem ser respondidas com base nas medidas das categorias;
- Selecionar as medidas aplicáveis;
- Especificar as medidas;
- Selecionar e especificar as medidas para projetos existentes.

- **Integrar a medição aos processos do projeto**

Nesta atividade é analisado o contexto onde será implantado o processo de medição, formalizando os procedimentos de coleta, análise e divulgação de dados, definindo regras e convenções para a condução do processo de medição e planejando os recursos e tecnologias a serem disponibilizadas para medição.

Consiste em:

- Caracterizar o ambiente;
- Identificar oportunidades de medição: análise da medição efetuada e dados históricos existentes;
- Especificar requisitos de implementação de medidas: definição de medidas, escopo de medição, coleta de dados, análise dos dados, comunicação dos resultados e avaliação da medição.

---

<sup>2</sup> Conceito mensurável – pode ser qualquer atividade ou produto de onde se pode obter elementos possuindo atributos e podendo ser medidos. Exemplo: Esforço do Pessoal (MCGARRY, 2001)

### **c) Executar medição**

Nesta atividade implementam-se os processos de coleta e análise das métricas identificadas na atividade de planejar a medição.

Consiste em:

- Coletar e processar dados: coleta de dados, preparação para a análise e armazenamento em local acessível para que possam ser analisados;
- Analisar dados: transformação das medidas básicas em indicadores;
- Produzir recomendações: avaliação global do projeto, identificação de problemas específicos e produção de recomendações.

### **d) Avaliar a medição**

Nesta atividade identificam-se as possibilidades de melhoria no processo de medição, para garantir que ele seja continuamente atualizado para suprir novas necessidades de informação.

Consiste em:

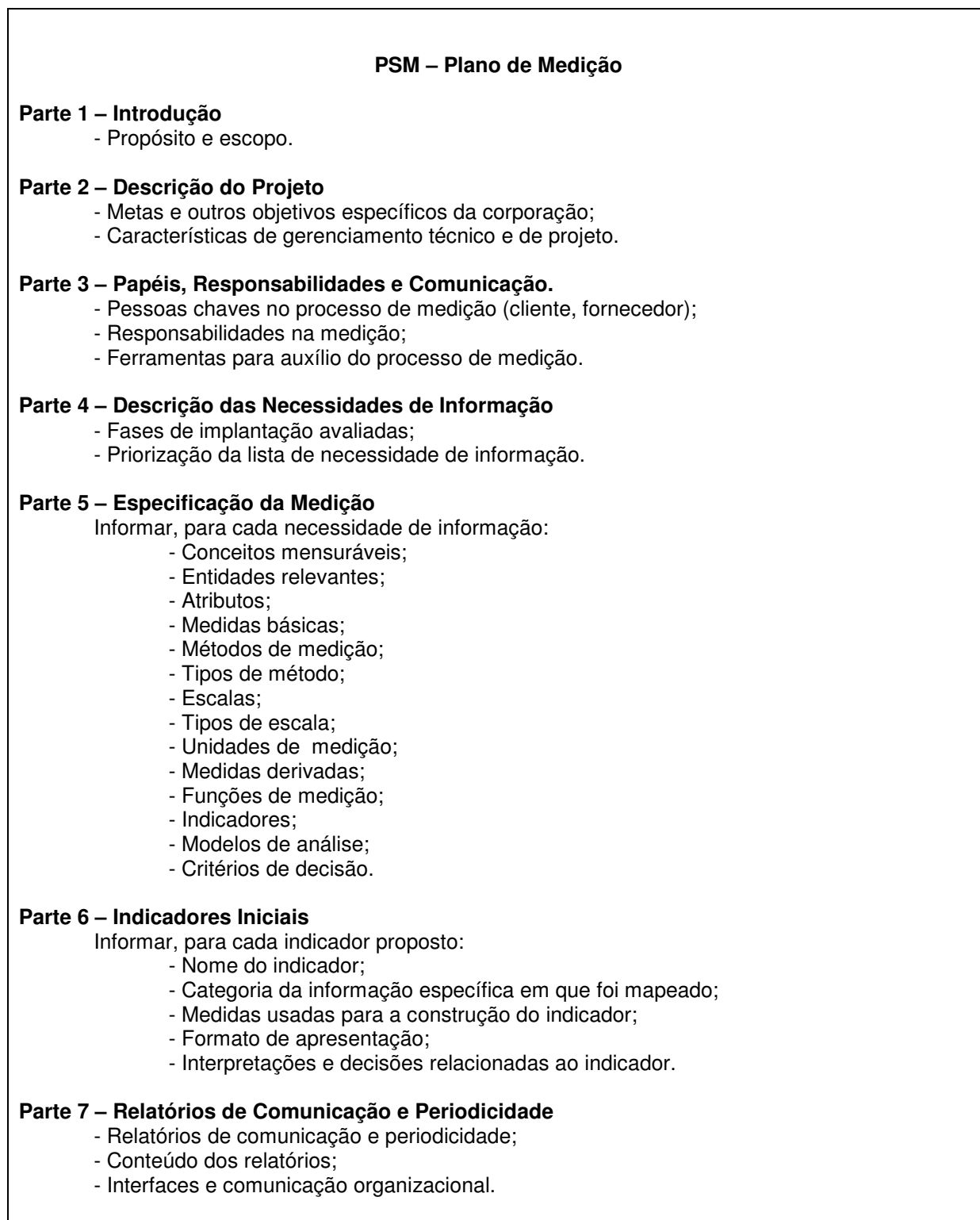
- Avaliar medidas;
- Avaliar o processo de medição;
- Atualizar a base de experiência;
- Identificar e implementar melhorias.

Como apresentado na figura 9, o plano de medição é o principal produto gerado por meio da execução do PSM, todas as informações da medição devem ser registradas nesse documento.

O plano de medição está dividido em sete partes:

- Introdução: descreve o objetivo e o escopo da medição;
- Descrição do Projeto: apresenta uma visão geral sobre o projeto, descrevendo as metas e objetivos específicos do projeto.
- Papéis, Responsabilidades e Comunicação: define os papéis e responsabilidades dos envolvidos no processo de medição e as ferramentas a serem utilizadas.
- Descrição das Necessidades de Informação: descreve as informações que devem ser obtidas com a medição.

- Especificação da Medição: com base nas necessidades de informação são identificados as entidades e atributos a serem analisados, obtendo-se e descrevendo-se as medidas consideradas para o projeto.



**Figura 9 - Plano de Medição PSM**

**Fonte: McGarry (2001)**

- Indicadores Iniciais: De acordo com as medidas adotadas para o projeto são definidos os indicadores, determinando a forma que esses indicadores devem ser interpretados.
- Relatórios de Comunicação e Periodicidade: Descreve o formato dos relatórios de comunicação dos resultados da medição, bem como a periodicidade de comunicação desses resultados.

Neste capítulo foram abordados os conceitos sobre medição de software como medida, métrica, indicadores e processo de medição.

A medição de software permite avaliar, conhecer e controlar um produto ou processo de forma quantitativa permitindo que ações sejam tomadas que ocasionem melhorias no produto e no processo.

Na literatura há vários autores que abordam a medição de software, como Almendra (2003) que apresenta um modelo de avaliação que permite diagnosticar a presença ou ausência de problemas que limitam ou anulam a utilidade de métricas de software, utilizando questionários a serem respondidos com as informações disponíveis sobre a métrica. Tem como base o GQM e efetua a ligação entre o problema e as questões utilizando o modelo probabilístico de redes Bayesianas. Esse modelo permite integrar as evidências oferecidas pelas respostas das diversas questões em uma única probabilidade associada ao problema.

O trabalho de Almendra (2003) contribui, para o presente trabalho, fornecendo as definições sobre medição, métricas e processo de medição. Outra contribuição é a apresentação do uso do GQM como modelo padrão para a seleção das métricas.

Como a definição de um processo de medição não é uma tarefa fácil, há vários modelos e metodologias, na literatura, que apóiam a seleção de medidas e a estruturação do processo de medição como: o GQM (BASILI, CALDIERA e ROMBACH, 1994), o *Goal-Driven Software Measurement* (PARK, GOETHERT e FLORAC, 1996) e o PSM (MCGARRY, 2001). Borges (2003), em seu trabalho propõe um modelo de medição focado no processo de desenvolvimento de software, onde em cada fase do desenvolvimento é apresentado um conjunto de métricas relevantes para atender a necessidade da fase em questão.

O trabalho de Borges (2003) propõe formulários para coleta e armazenamento destas medidas e formas de avaliação dos resultados coletados. Contribui, para o presente trabalho, fornecendo o embasamento teórico sobre defeito, medição de defeito, GQM, *Goal-Driven Software Measurement* e PSM e propondo uma forma de coleta e armazenamento dos resultados dos testes. Também contribui apresentando a aplicação do modelo de medição proposto com base nos princípios do *Goal-Driven Software Measurement*.

Como o objetivo deste trabalho é unir os conceitos de teste de software e de processo de medição, na elaboração do processo proposto são utilizados os conceitos sobre teste de software, principalmente, no que corresponde às fases do processo de teste e nas documentações geradas. Também, os conceitos de medição de software considerando que o teste fornece os dados sobre o produto que ao serem avaliados quantitativamente permitem avaliar, controlar e propor melhorias para o produto e para o próprio processo.

Para a concepção e implementação do processo de medição são utilizados os conceitos básicos de medida, indicador e métricas e é adotado o PSM como base para a estruturação do processo de medição por ser um modelo consolidado de medição de software e, principalmente, por ser uma abordagem mais abrangente que o GQM e o *Goal-Driven Software Measurement*, contemplando tanto a seleção de medidas e indicadores, quanto a implementação do processo de medição. A seguir é apresentado o processo de medição proposto.

## **CAPÍTULO 4 – PROCESSO DE MEDIÇÃO DE DEFEITOS**

Com a competitividade existente nos dias de hoje, o investimento das empresas em mecanismos que permitam conhecer, avaliar e controlar a qualidade de seus processos e produtos tornou-se um fator de sobrevivência. A medição, como um desses mecanismos, passou a ser considerada fundamental por fornecer aos gestores um conjunto de dados úteis e tangíveis para dimensionar, estimar, planejar e controlar processos, projetos e produtos.

Como o teste de software representa a última revisão da especificação, projeto e codificação e as organizações, geralmente, possuem um processo de teste vigente e, por falta de conhecimento, tempo ou recursos, acabam desperdiçando as informações obtidas nos testes. A idéia é elaborar um processo que permita medir os defeitos resultantes das atividades de teste, de forma objetiva e estruturada, fornecendo informações relevantes que contribuam para melhoria do processo de teste de software e do produto.

Os processos apresentados nesse capítulo são descritos utilizando a notação gráfica IDEF0 (NIST, 1993) que mostra o processo e as suas atividades permitindo visualizar as entradas, saídas, informações de controle e informações de recursos que compõem essas atividades. Mais informações, sobre esta notação gráfica, podem ser consultadas no apêndice A. Já os termos utilizados nas figuras desse capítulo estão descritos no glossário.

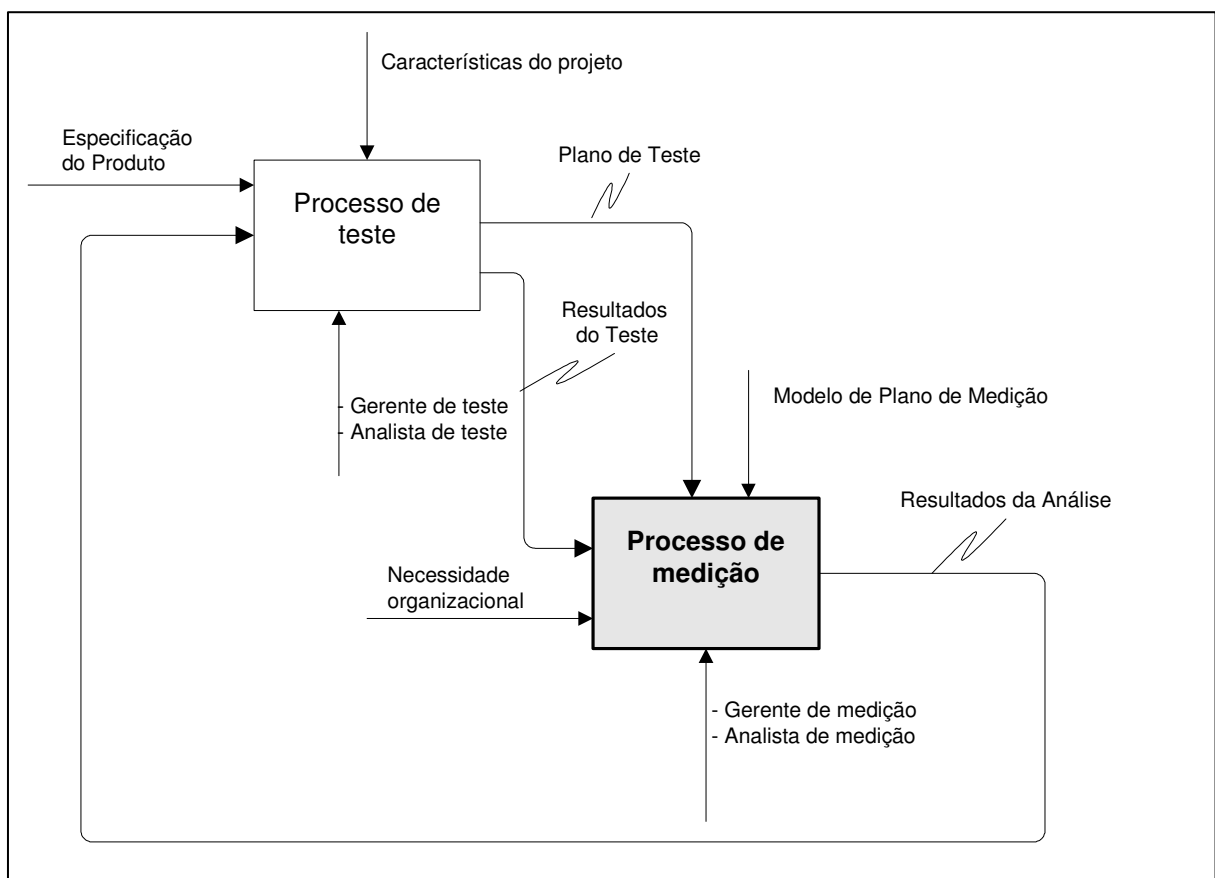
### **4.1. Contexto da Proposta**

Como esse trabalho tem como foco o teste de software, o processo de medição tem uma relação direta com o processo de teste da organização onde será implantada a medição. A proposta é não interferir na definição do processo de teste adotado, cabe à organização definir as estratégias, técnicas e recursos a serem utilizados em suas atividades de teste. No entanto, alguns elementos básicos precisam estar presentes no processo de teste da organização para que haja a relação entre esse processo e o processo de medição. O mínimo necessário é que a organização execute atividades de teste de software, tenha o registro das estratégias de teste utilizadas, por exemplo, um plano de teste, e armazene os resultados obtidos nos testes de forma acessível.

Como premissa para a aplicação do processo de medição é considerado que as fases do desenvolvimento de software, como a especificação de requisitos e as revisões, efetuadas anteriormente ao teste de software, foram executadas.

Considerando que uma organização possua um processo de teste que esteja sendo efetivamente aplicado, a idéia é relacionar esse processo de teste com o processo de medição. Como mostra a figura 10, para estabelecer a relação entre esses processos, dois elementos são cruciais: o Plano de Teste e os Resultados do Teste.

O Plano de Teste por descrever o escopo e a abordagem utilizada nos testes permitindo identificar as estratégias de teste que serão utilizadas. As estratégias de teste serão a base para a elaboração do planejamento de medição. De acordo com as estratégias de teste adotadas podem variar as metas e as medidas a serem utilizadas. Já os Resultados do Teste são a fonte de onde os defeitos identificados nas atividades de teste são extraídos.



**Figura 10 - Relação do Processo de Teste com o Processo de Medição**



Outra relação entre esses dois processos é que, ao término da medição, é gerado um relatório contendo os resultados e as recomendações que servirão de entrada para o processo de teste podendo influenciar o novo planejamento do teste.

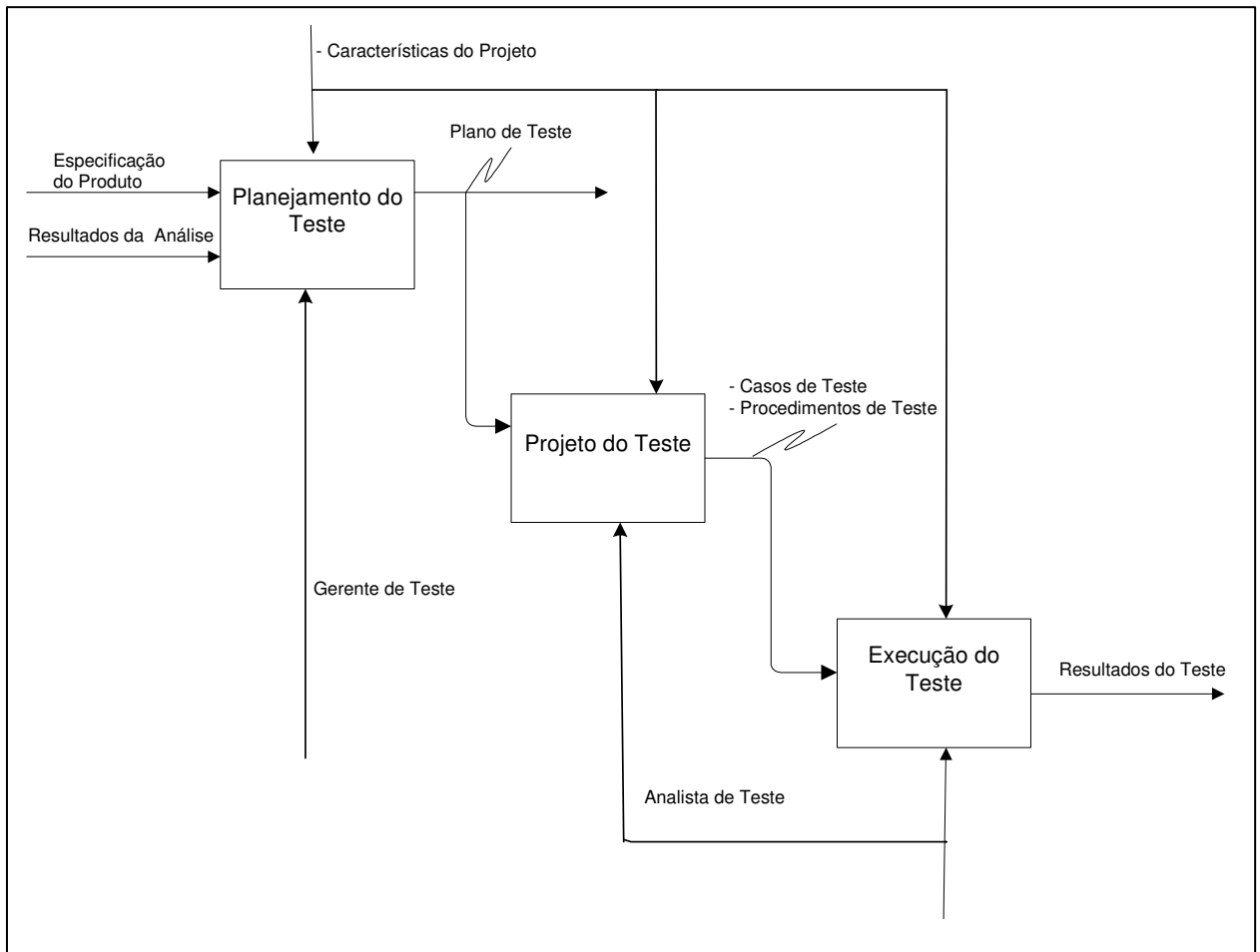
Como o processo de medição proposto está relacionado ao teste de software e na literatura são encontradas várias propostas de processos de teste, a seguir é apresentado como referência um processo de teste de software baseado na proposta de Rotta (2001), utilizando a documentação recomendada pela Norma IEEE 829 (1983). O processo de Rotta (2001) fornece os elementos necessários para a implantação do processo de medição proposto.

### **Processo de Teste**

Na figura 11, é apresentado o processo de teste de software utilizado como referência neste trabalho. Suas etapas foram criadas com base nas fases propostas por Rotta (2001), que são: planejamento, projeto, preparação da infra-estrutura, execução e análise do teste. No entanto, para facilitar a compreensão do fluxo de informações entre as etapas do teste, no processo de teste referenciado, são consideradas somente três etapas: planejamento, projeto e execução do teste. As fases de preparação de infra-estrutura e análise do teste, propostas por Rotta (2001), não são consideradas como etapas no processo de teste referenciado. No entanto, as atividades contidas nessas etapas estão sendo mantidas. As atividades da etapa de preparação de infra-estrutura (ROTTA, 2001), que são: a definição e a preparação do ambiente de teste estão sendo efetuadas na fase de projeto do teste e a análise dos resultados obtidos no teste, efetuada na etapa de análise do teste (ROTTA, 2001), estão sendo consideradas na fase de execução do teste.

Os documentos gerados nas etapas do processo de teste proposto como o plano de teste, os casos de teste, os procedimentos de teste e os resultados dos testes estão de acordo com as recomendações da Norma IEEE 829 (1983).

Para a aplicação do processo de teste sugerido foram definidos dois papéis: o do Gerente de Teste e do Analista de Teste. O Gerente de Teste é responsável pelo planejamento do teste. O Analista de Teste é responsável pelo projeto dos testes, execução dos testes definidos e registro dos resultados obtidos. As etapas do processo de teste são detalhadas a seguir.



**Figura 11- Processo de Teste**

Fonte: adaptado de Rotta (2001)

### a) Planejamento do Teste

O Plano de Teste é elaborado pelo Gerente de Teste de acordo com a especificação do produto e as características específicas de cada projeto (tempo, custo, recursos, entre outros). Nessa fase são descritos o escopo do teste, os recursos necessários e o cronograma de atividades de teste e são identificados os métodos que serão empregados. Os itens e as funcionalidades a serem testadas, as tarefas a serem realizadas e os riscos associados às atividades de teste são identificados no documento Plano de Teste.

Nesse plano é descrita a abordagem que será utilizada no teste, que compreende a definição dos níveis de teste que serão utilizados (teste de unidade, integração e sistema, por exemplo). Apesar da Norma IEEE 829 (1983) adotar o termo abordagem para se referir à definição dos níveis de teste, no presente trabalho, é convencionalizado o uso do termo estratégia de teste para se referir aos

níveis de teste definidos no plano de teste. A estratégia definida é utilizada como base para o planejamento da medição.

Como há uma relação direta proposta entre os processos de teste e de medição os resultados e as recomendações gerados na execução da medição são considerados como entradas para o processo de teste, indicando a necessidade de melhorias nesse processo.

## **b) Projeto do Teste**

Na fase de Projeto do Teste, o Analista de Teste seleciona as características a serem testadas e descreve, detalhadamente, os métodos que serão empregados e os testes que serão efetuados com base nas definições contidas no plano de teste. São detalhados os itens que serão testados, com as entradas utilizadas, saídas esperadas e os passos necessários para se executar os testes. Como resultado são gerados os Casos de Teste e os Procedimentos de Teste. O primeiro apresenta os dados de entrada, resultados esperados, ações e condições gerais para a execução do teste. O segundo especifica os passos para a execução de um conjunto de casos de teste.

Nessa fase também é preparada a infra-estrutura de hardware e software necessária para execução do teste e são gerados os *drivers* e *stubs* se necessário.

## **c) Execução do Teste**

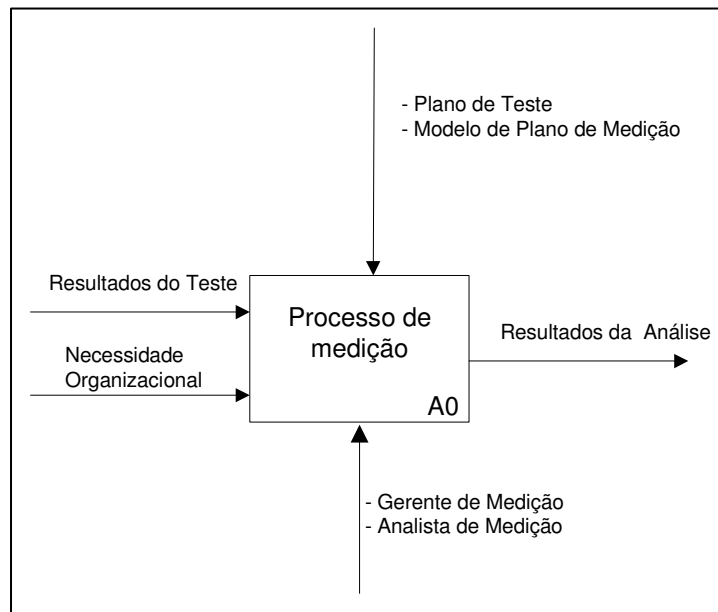
Nessa fase os Procedimentos e Casos de Teste são colocados em prática pelo Analista de Teste. Os resultados obtidos denominados Resultados do Teste são descritos em relatórios como: o Relatório de Envio de Item de Teste, o Diário de Registro de Teste, o Relatório de Incidentes de Teste e o Relatório de Resumo de Teste. O Diário de Registro de Teste apresenta registros cronológicos dos detalhes relevantes relacionados à execução do teste. O Relatório de Incidentes de Teste documenta qualquer evento que ocorra durante a atividade de teste e que mereça ser investigado. O Relatório de Resumo de Teste apresenta de forma resumida os resultados das atividades de teste associadas com uma ou mais especificações de projeto de teste e provê avaliações baseadas nesses resultados.

A definição da documentação de teste pode sofrer alterações dependendo das características de cada organização e de seus projetos e produtos.

A seguir, é mostrada a estrutura do processo de medição, tendo como base os princípios recomendados pelo PSM, que foram adaptados para apoiar as características particulares do teste de software.

#### 4.2. Processo de Medição Proposto

Na figura 12, é mostrada uma visão macro do processo de medição proposto que possui como entradas os resultados do teste e as necessidades da organização. Os resultados do teste correspondem aos dados resultantes da execução do teste de software e as necessidades organizacionais são informações que os tomadores de decisão precisam obter por meio da medição para direcionar suas ações que não estão diretamente ligadas aos dados obtidos no teste.



**Figura 12 - Processo de Medição - Contexto**

De acordo com a entrada fornecida, as necessidades organizacionais e os resultados dos testes, o processo de medição é aplicado. Para guiar o planejamento e a execução da medição são consideradas as estratégias de teste definidas no plano de teste e o modelo de plano de medição que é preenchido ao longo das fases do processo e serve como um roteiro para os gerentes e analistas responsáveis pela medição. Como resultado da aplicação do processo de medição é gerado como

saída os Resultados da Análise com recomendações que poderão servir como entrada para o processo de teste.

Na elaboração do processo de medição foi utilizado como guia as práticas recomendadas pelo PSM. No entanto, como este trabalho é focado em processo de teste de software, algumas adaptações tiveram que ser realizadas para atender ao contexto dessa proposta. Na figura 13, é apresentada a relação entre o processo de teste e o processo de medição com base na estrutura do modelo de processo do PSM.

Do modelo de processo do PSM foram adotadas somente as fases contidas no processo central de medição que são: planejar e executar a medição. Foi considerado que um processo de medição, basicamente, tem que possuir uma fase de planejamento na qual é efetuada a análise do que se deseja conhecer ou entender, definindo as metas e objetivos a serem obtidos com a medição. Portanto, antes de realizar a medição efetiva é necessário ter bem definido o que se deseja obter com essa medição. Com o planejamento concluído é preciso colocar em prática, na fase de execução da medição, o que foi planejado.

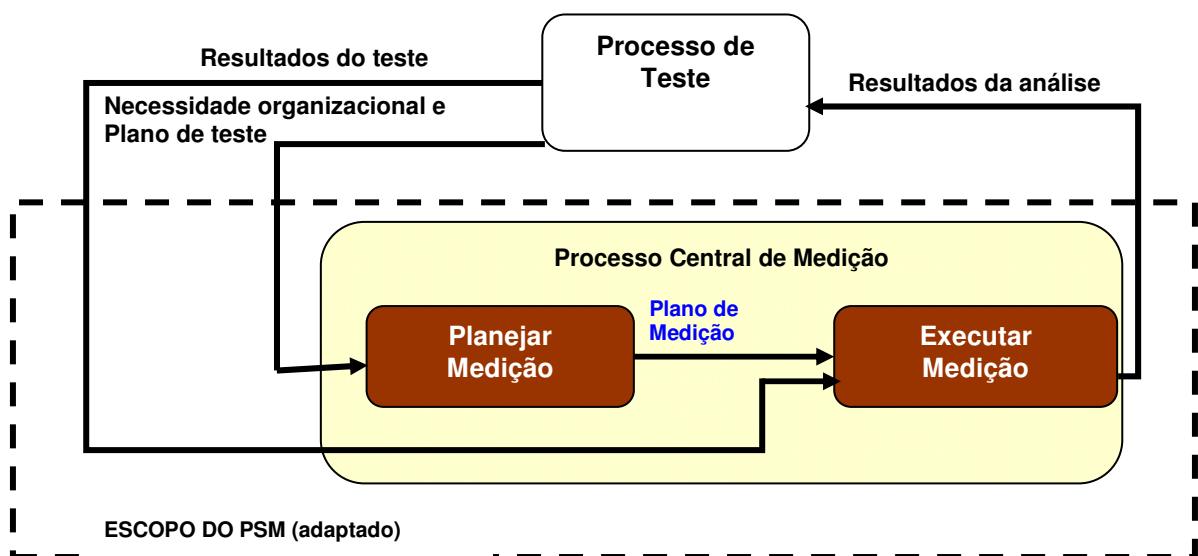


Figura 13 – Relação entre o Processo de Medição Proposto e o PSM

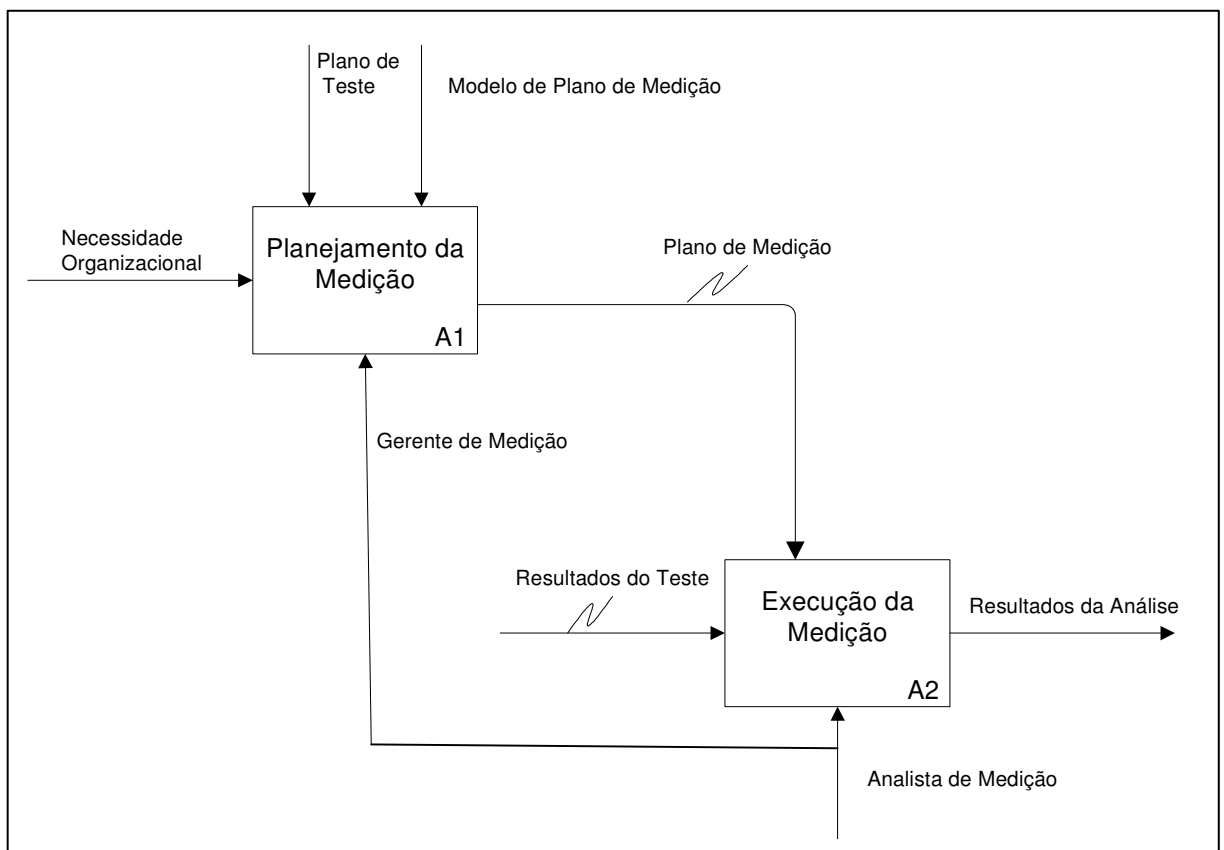
Fonte: adaptado de McGarry (2001)

As atividades não-centrais do PSM como “Estabelecer e sustentar comprometimento” e “Avaliar a medição” não são abordadas no processo de medição proposto. Essas atividades abrangem, respectivamente, a manutenção e a

melhoria do processo de medição o que excede o escopo do presente trabalho, considerando que seria muito precipitado propor atividades de manutenção e melhoria do processo quando a estrutura básica desse processo, o planejamento e a execução da medição, ainda está sendo proposta. Já a atividade “Processos técnicos e gerenciais”, corresponde ao processo de teste que é um processo técnico vigente na organização e possui uma relação direta com o processo de medição.

Na aplicação do processo de medição são considerados dois papéis: o do Gerente de Medição, que efetua o planejamento da medição e que, na fase de execução da medição, produz os resultados da medição e o do Analista de Medição que é responsável pela coleta, processamento e análise dos dados.

Como pode ser observado na figura 14, o processo proposto considera duas fases para a medição dos defeitos encontrados nos testes: o Planejamento da Medição e a Execução da Medição.



**Figura 14 - Processo de Medição – Fases**

## A1) Planejamento da Medição

Nessa fase, como mostra a figura 15, o Gerente de Medição efetua uma análise do que se deseja conhecer ou entender. A medição não pode ser aplicada se não há uma definição clara da meta a ser alcançada com essa medição. Portanto, na fase de planejamento são identificadas as necessidades de informação dos tomadores de decisão no que tange o teste de software. A organização possui a necessidade de obter alguma informação para atingir uma determinada meta. Então, o primeiro passo é definir as metas a serem alcançadas e os objetivos específicos que devem ser atingidos com relação às estratégias de teste utilizadas. Com base nesses objetivos são definidas questões que servirão de apoio para a seleção das medidas necessárias para responder a esses objetivos.

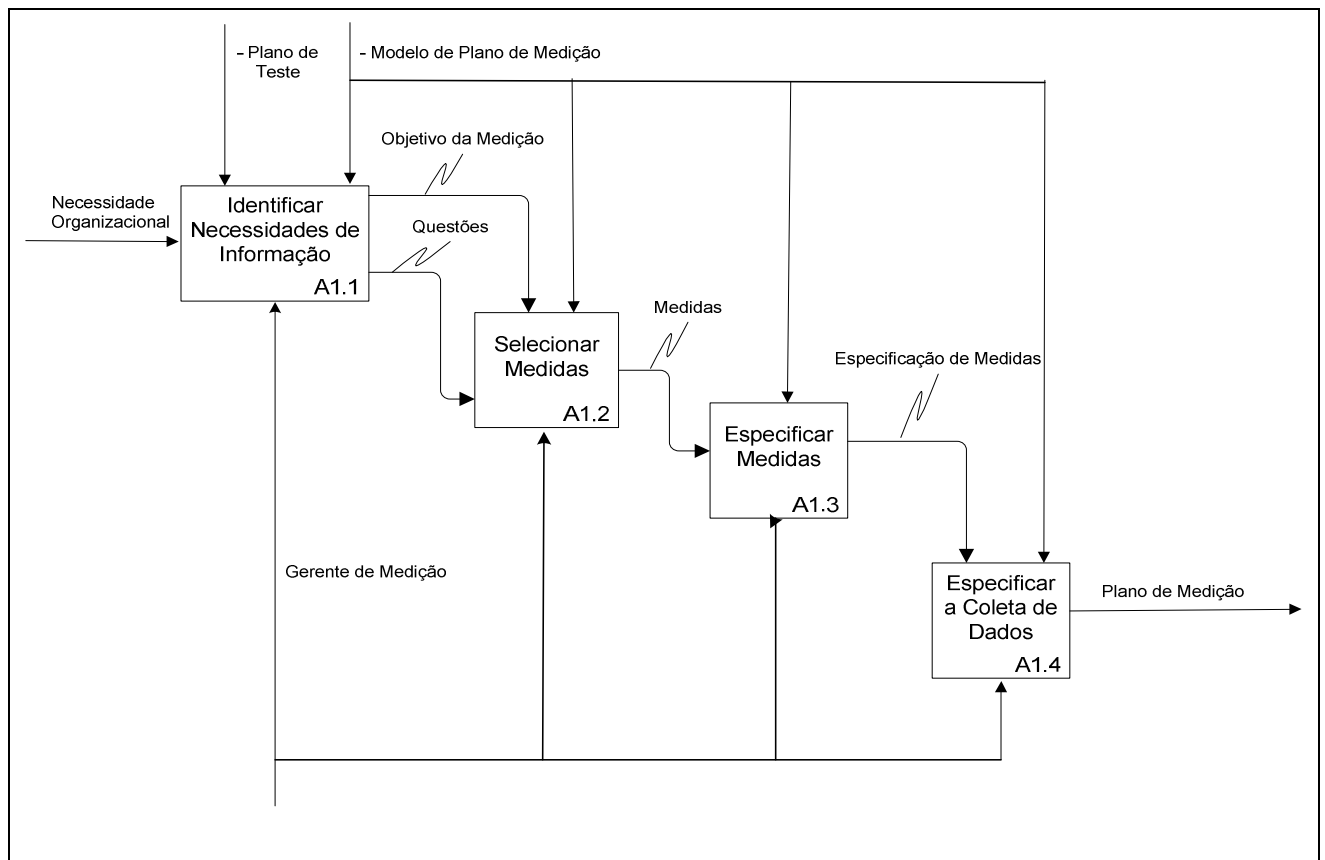


Figura 15 - Fase de Planejamento da Medição

A fase de Planejamento da Medição é efetuada com base no processo de teste, de acordo com cada tipo de estratégia estabelecida são definidos os objetivos da medição, o que se deseja conhecer, aprender ou avaliar com relação a essas estratégias. Essa fase é dividida em quatro atividades e em cada uma dessas

atividades são preenchidos alguns itens do modelo de plano de medição, mostrado na figura 16.

<b>Plano de Medição</b>					
<b>A1.1) Identificação das Necessidades de Informação</b>					
a) Descrição das Metas					
Cód. Meta	Propósito	Perspectiva	Ambiente		
	Analisar o (a) _____ (objeto: processo, produto).  Com a intenção de _____ (razão: conhecer, avaliar, prever, melhorar).	o (a) _____ (foco: custo, correção, confiabilidade).  do ponto de vista do (a) _____ (quem: usuário, cliente, desenvolvedor, gerente)	No seguinte contexto _____		
b) Papéis e Responsabilidades					
Papel		Responsabilidade			
c) Identificação das Necessidades de Informação					
Cód. Meta	Objetivo	Questões			
<b>A1.2) Seleção de Medidas</b>					
a) Entidades e Atributos					
Cód. Meta	Objetivo	Questões	Entidades Relevantes	Atributos	Medida Básica
<b>A1.3) Especificação de Medidas</b>					
a) Especificação de Medidas Básicas					
Cód. Meta	Medida Básica	Método de Medição	Tipo de Método	Tipo de Escala	Unidade de Medida
b) Especificação de Medidas Derivadas					
Cód. Meta	Medida Derivada			Função de Medição	
c) Especificação de Indicadores					
Cód. Meta	Medida Derivada	Descrição do Indicador	Modelo de Análise	Critérios de Decisão	
<b>A1.4) Especificação da Coleta de Dados</b>					
Periodicidade		Ferramentas	Descrição		

Figura 16 - Plano de Medição



Esse Modelo de Plano de Medição foi elaborado com base no plano de medição sugerido pelo PSM e pelas recomendações sugeridas pelo GQM, no entanto sofreu algumas alterações visando facilitar sua compreensão e preenchimento, por parte dos profissionais de software, e se adequar ao contexto do trabalho. As atividades estão sub-divididas da seguinte forma:

**A1.1) Identificar Necessidades de Informação:** de acordo com as estratégias definidas no plano de teste são identificadas as necessidades de informação. Para auxiliar nessa tarefa foram adotados alguns princípios do GQM. Esse método estabelece um conjunto de informações que a definição de uma meta deve conter, são elas: o propósito que especifica o objeto de estudo e a razão da medição. A perspectiva que define o foco da medição e o ponto de vista que será considerado. O ambiente que descreve o contexto que está sendo considerado para a medição.

Na figura 17, é mostrada a parte do modelo de plano de medição que deve ser preenchida para a identificação das metas.

**a) Descrição das Metas**

<b>Cód. Meta</b>	<b>Propósito</b>	<b>Perspectiva</b>	<b>Ambiente</b>
	Analisar o (a) _____ (objeto: processo, produto)  Com a intenção de _____ (razão: conhecer, avaliar, prever, melhorar)	o (a) _____ (foco: custo, correção, confiabilidade)  do ponto de vista do (a) _____ (quem: usuário, cliente, desenvolvedor, gerente.)	No seguinte contexto _____

**Figura 17 - Plano de Medição – Descrição da Meta**

Como exemplo, assume-se que uma organização tenha a necessidade de melhorar a produtividade do seu processo de teste de software. Para tanto será implantado o processo de medição proposto. O primeiro passo é seguir o plano de medição que possui como primeira atividade a identificação das metas. Como mostra a tabela 5, são preenchidos o propósito, a perspectiva e o ambiente. Nesse caso, a meta é “Analisar o processo de teste de software com a intenção de avaliar a produtividade do processo de teste e identificar oportunidades de melhoria”. Portanto, o foco da medição é fornecer embasamento quantitativo sobre a produtividade do teste e o ambiente considerado para a medição é o ambiente de desenvolvimento de *web sites*.

Tabela 5 - Exemplo do Plano de Medição - Descrição da Meta

Cód. Meta	Propósito	Perspectiva	Ambiente
1	<p>Analisar o processo de teste de software.</p> <p>Com a intenção de avaliar a produtividade do processo de teste e identificar oportunidades de melhoria.</p>	<p>Fornecer um embasamento quantitativo para a avaliação da produtividade das atividades de teste.</p> <p>Do ponto de vista do (a) do gerente de projeto.</p>	<p>No seguinte contexto: ambiente de desenvolvimento de <i>web sites</i>.</p>

Como nessa atividade é definido o contexto da medição cabe também definir o papel de cada envolvido no processo de medição, como recomendado no PSM, descrevendo suas responsabilidades. Na figura 18, é mostrada a parte do modelo de plano de medição onde devem ser inseridos os papéis e as responsabilidades de cada envolvido.

#### b) Papéis e Responsabilidades

Papel	Responsabilidade

Figura 18 - Plano de Medição – Papéis e Responsabilidades

Como mostrado na tabela 6, um exemplo de definição de papéis e responsabilidades seria definir o papel de gerente de medição que possui a responsabilidade de planejar a medição, analisar os indicadores e produzir recomendações.

Tabela 6 - Exemplo do Plano de Medição – Papéis e Responsabilidades

Papel	Responsabilidade
Gerente de Medição	<ul style="list-style-type: none"> <li>- Planejar a medição;</li> <li>- Analisar os indicadores resultantes da medição e produzir recomendações.</li> </ul>
Analista de Medição	<ul style="list-style-type: none"> <li>- Coletar e processar os dados de teste;</li> <li>- Analisar os dados de teste produzindo os indicadores de medição.</li> </ul>

Com as metas definidas e os papéis descritos, de acordo com o PSM, é necessário identificar quais os objetivos específicos a serem obtidos. Muitas vezes, meta e objetivos podem ser interpretados erroneamente, portanto é necessário ter bem definidos esses conceitos. A meta é mais ampla e corresponde ao que se deseja alcançar com a medição com relação ao negócio. Enquanto, o objetivo define o que é necessário conhecer do seu processo ou produto para atingir a meta

desejada. Portanto, a meta é ampla e o objetivo é mais específico e definido de acordo com a meta proposta.

Para cada meta podem ser gerados diversos objetivos e com base em cada um desses objetivos são formuladas questões que, ao serem respondidas quantitativamente, permitem atingir os objetivos. A figura 19, mostra a parte do modelo de plano de medição onde são descritos os objetivos e as questões relacionadas.

**c) Identificação das Necessidades de Informação**

Cód. Meta	Objetivo	Questões

**Figura 19 - Plano de Medição - Identificação das Necessidades de Informação**

De acordo com o exemplo, a meta considerada é “Analisar o processo de teste de software com a intenção de avaliar a produtividade na detecção de defeitos e identificar oportunidades de melhoria”. Portanto, um exemplo de objetivo para a meta definida, como mostra a tabela 7, seria “Avaliar o teste de integração quanto ao esforço gasto na detecção de defeitos”.

Com base nos objetivos são elaboradas questões que, ao serem respondidas, permitem atingir os objetivos. Portanto, para o objetivo “Avaliar o teste de unidade quanto à quantidade de defeitos detectados por hora” as questões a serem respondidas seriam “Qual a taxa de defeitos detectados no teste de unidade?” e “Qual o esforço gasto na execução do teste de unidade?”.

**Tabela 7 - Exemplo do Plano de Medição – Identificação das Necessidades de Informação**

Cód. Meta	Objetivo	Questões
1	Avaliar o teste de unidade quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de unidade? - Qual o esforço gasto na execução do teste de unidade?
	Avaliar o teste de integração quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de integração? - Qual o esforço gasto na execução do teste de integração?

**A1.2) Selecionar Medidas:** nessa atividade são utilizados os objetivos como base para direcionar a seleção de medidas. Com base nas questões formuladas na atividade anterior é identificado o que pode ser medido para se atingir os objetivos. No PSM essa idéia inicial do que pode ser medido é denominado conceito

mensurável e pode ser qualquer atividade ou produto de onde se obtêm elementos, possuindo atributos e podendo ser medidos. Para cada questão existente são identificadas as entidades que podem ser medidas e os atributos de cada entidade que serão considerados de forma a fornecer subsídios para responder a essas questões. Na figura 20, como recomendado no PSM, é apresentada a parte do modelo de plano de medição que, preenchida, relaciona os objetivos e questões às entidades e atributos permitindo identificar as medidas básicas correspondentes. A medida básica é a medida de um único atributo realizada por meio de um método especificado de medição.

**a) Entidades e Atributos**

Cód. Meta	Objetivo	Questões	Entidades Relevantes	Atributos	Medida Básica

**Figura 20 - Plano de Medição – Seleção da Medição**

Como mostra a tabela 8, considerando o objetivo “Avaliar o teste de unidade quanto à quantidade de defeitos detectados por hora” e a questão “Qual a taxa de defeitos detectados no teste de unidade?”, é necessário avaliar a entidade defeito e os atributos ou características dessa entidade como: a quantidade, a descrição e a data do defeito. Portanto, com base na avaliação da quantidade de defeitos detectados no teste de unidade obtêm-se a medida básica “Defeitos por teste de unidade”.

**Tabela 8 - Exemplo do Plano de Medição – Seleção da Medição**

Cód. Meta	Objetivo	Questões	Entidades Relevantes	Atributos	Medida Básica
1	Avaliar o teste de unidade quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de unidade?	Cada defeito detectado	- quantidade - descrição do defeito - data de detecção	Defeitos por teste de unidade.
		- Qual o esforço gasto na execução do teste de unidade?	Atividade de teste	- horas trabalhadas - data do teste	Esforço por teste de unidade.
	Avaliar o teste de integração quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de integração?	Cada defeito detectado	- quantidade - descrição do defeito - data de detecção	Defeitos por teste de integração
		- Qual o esforço gasto na execução do teste de integração?	Atividade de teste	- horas trabalhadas - data do teste	Esforço por teste de integração
	Avaliar o teste de sistema quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de sistema?	Cada defeito detectado	- quantidade - descrição do defeito - data de detecção	Defeitos por teste de sistema.
		- Qual o esforço gasto na execução do teste de sistema?	Atividade de teste	- horas trabalhadas - data do teste	Esforço por teste de sistema

**A1.3) Especificar Medidas:** após identificar as necessidades de informação e selecionar conceitos mensuráveis é necessário efetuar a construção da medição. Essa construção consiste em especificar ou descrever as medidas que serão utilizadas na medição. Nessa atividade são descritos: as medidas básicas, o método de medição e os indicadores a serem analisados. O primeiro passo na construção da medição, de acordo com o PSM, é especificar cada medida básica definida na atividade de seleção de medidas. Na figura 21, é apresentada a parte do modelo de plano de medição onde devem ser descritas as medidas básicas. Para cada medida básica descreve-se: o método de medição, o tipo de método, o tipo de escala e a unidade de medida utilizada.

O método de medição é uma seqüência de operações que quantificam atributos em uma determinada escala de medição. Um método de medição pode ser classificado como objetivo ou subjetivo, dependendo da natureza da operação utilizada para quantificar o atributo. Um exemplo de tipo de método objetivo é a operação de contagem de defeitos que quantifica o atributo de forma direta.

A escala define um mapeamento do mundo real para o mundo dos números. Os tipos de escala podem ser: escala absoluta, por exemplo, as medidas de contagem; escala de razões, por exemplo, medição de massa, intervalos de tempo; escala de intervalos, por exemplo, temperatura medida em graus Celsius e datas; escala ordinal, por exemplo, medidas de preferência e dureza de materiais e escala nominal, por exemplo, linguagem de programação (C, Pascal, Fortran), metodologia utilizada (nenhuma, orientado a objetos, análise estruturada).

**a) Especificação de Medidas Básicas**

Cód. Meta	Medida Básica	Método de Medição	Tipo de Método	Tipo de Escala	Unidade de Medida

**Figura 21 – Plano de Medição - Especificação da Medição – Medidas Básicas**

Considerando a medida básica "Defeitos por teste", como mostrado na tabela 9, são definidos: o método de medição, o tipo de método, o tipo de escala e a unidade de medida. Portanto, para a medida defeitos por teste o método de medição será contar cada defeito detectado no teste, como trata-se de uma contagem, o tipo de método é objetivo, o tipo de escala é absoluta e a unidade de medida utilizada é número de defeitos.

Tabela 9 - Exemplo do Plano de Medição - Especificação da Medição – Medidas Básicas

Cód. Meta	Medida Básica	Método de Medição	Tipo de Método	Tipo de Escala	Unidade de Medida
1	Defeitos por teste de unidade	Contar defeitos	Objetivo	Absoluta	Número de defeitos
	Esforço por teste de unidade	Contar horas trabalhadas	Objetivo	Absoluta	Horas trabalhadas
	Defeitos por teste de integração	Contar defeitos	Objetivo	Absoluta	Número de defeitos
	Esforço por teste de integração	Contar horas trabalhadas	Objetivo	Absoluta	Horas trabalhadas
	Defeitos por teste de sistema	Contar defeitos	Objetivo	Absoluta	Número de defeitos
	Esforço por teste de sistema	Contar horas trabalhadas	Objetivo	Absoluta	Horas trabalhadas

O passo seguinte, de acordo com o PSM, é a descrição das medidas derivadas a serem utilizadas com base no modelo de plano de medição, mostrado na figura 22. A medida derivada é uma medida definida em função de duas ou mais medidas básicas, ou ainda por outras medidas derivadas, obtendo informações de mais de um atributo e sendo definida com o uso de uma função de medição. A função de medição define uma medida derivada e corresponde a um algoritmo que combina dois ou mais valores de medidas básicas ou derivadas. A combinação das escalas e unidades das medidas básicas na função determina as escalas e unidades das medidas derivadas que a função define.

#### b) Especificação de Medidas Derivadas

Cód. Meta	Medida Derivada	Função de Medição

Figura 22 – Plano de Medição - Especificação da Medição – Medidas Derivadas

Como mostra a tabela 10, com a combinação da medida “Defeitos por teste de unidade” e a medida “Esforço por teste de unidade”, por meio da função de medição que divide defeitos por horas trabalhadas, temos a medida derivada “Quantidade média de defeitos detectados por hora no teste de unidade”.

Tabela 10 - Exemplo do Plano de Medição - Especificação da Medição – Medidas Derivadas

Cód. Meta	Medida Derivada	Função de Medição
1	1. Quantidade média de defeitos detectados por hora no teste de unidade.	Dividir defeitos detectados por horas trabalhadas no teste de unidade.
	2. Quantidade média de defeitos detectados por hora no teste de integração.	Dividir defeitos detectados por horas trabalhadas no teste de integração.
	3. Quantidade média de defeitos detectados por hora no teste de sistema.	Dividir defeitos detectados por horas trabalhadas no teste de sistema.

O último passo da especificação da medição é a descrição dos indicadores com base nas medidas selecionadas. Um indicador é uma medida da qual pode-se extrair uma estimativa ou avaliação dos atributos, determinados a partir de um modelo de análise associado a uma necessidade original de informação. Um indicador corresponde ao que é apresentado ao usuário para que este possa realizar a análise da medição e tomar decisões sobre o processo, projeto ou produto. Como mostra a figura 23, para especificar os indicadores deve-se indicar a medida correspondente e, descrever o indicador, o modelo de análise e os critérios de decisão. O modelo de análise é a combinação de dois ou mais valores de medidas, básicas ou derivadas, com algum critério de decisão, formando um algoritmo.

O critério de decisão possibilita a interpretação dos resultados gerados por um modelo de análise. Para a identificação da necessidade de alguma ação ou informar o nível de confiabilidade do resultado ao tomador de decisão, são considerados valores limites ou pontos referenciais.

#### c) Especificação de Indicadores

Cód. Meta	Medida Derivada	Descrição do Indicador	Modelo de Análise	Critérios de Decisão

Figura 23 – Plano de Medição - Especificação da Medição – Indicadores

Como mostra a tabela 11, calculando-se a média de defeitos detectados por hora em cada tipo de teste, utilizando o modelo de análise definido e as medidas derivadas definidas, obtêm-se o indicador “Estimativa de produtividade na detecção de defeitos no teste de software“, que será interpretado de acordo com o critério de análise definido. Nesse caso, o critério de decisão é “Quanto maior a quantidade de defeitos detectados por hora melhor a produtividade” considerando que a meta inicial era “Analisar o processo de teste de software com a intenção de avaliar a produtividade do processo de teste e identificar oportunidades de melhoria” o indicador, que é o produto da informação, deve responder à meta estabelecida inicialmente.

Tabela 11 – Exemplo do Plano de Medição - Especificação da Medição – Indicadores

Cód. Meta	Medida Derivada	Descrição do Indicador	Modelo de Análise	Critérios de Decisão
	1. Quantidade média de defeitos detectados por hora no teste de unidade; 2. Quantidade média de defeitos detectados por hora no teste de integração; 3. Quantidade média de defeitos detectados por hora no teste de sistema.	Estimativa de produtividade na detecção de defeitos no teste de software	Média de defeitos detectados por hora em cada tipo de teste.	Quanto maior a quantidade de defeitos detectados por hora melhor a produtividade.

Para auxiliar na construção da medição, juntamente com o plano de medição, pode ser utilizado o esquema hierárquico proposto pelo PSM, mostrado na figura 24. Com a utilização desse esquema é possível simular a medição antes do preenchimento detalhado do plano de medição, permitindo montar o fluxo da medição com base nas necessidades, definindo as entidades, os atributos, as medidas básicas até se obter os indicadores.

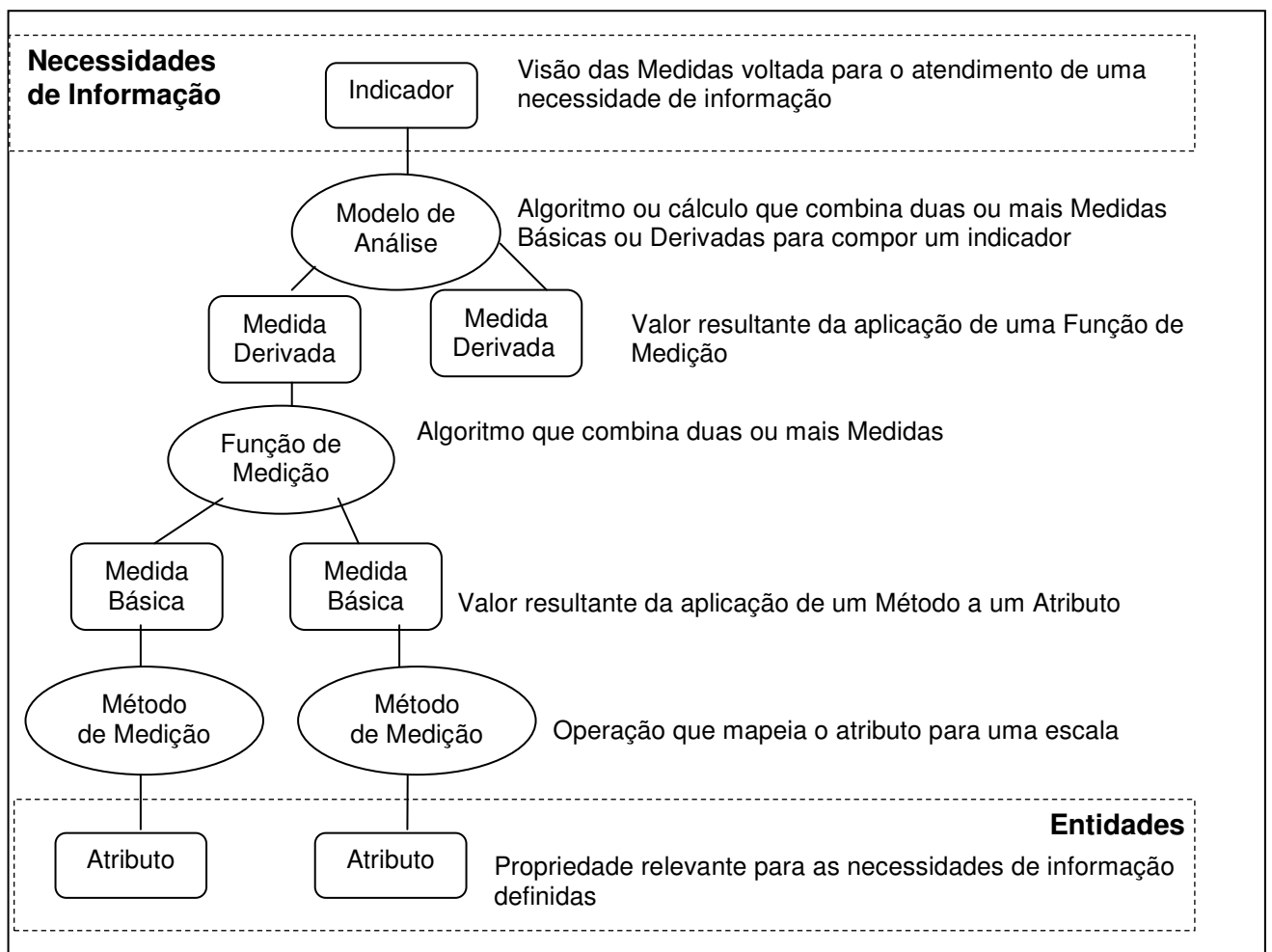


Figura 24 – Construção da medição – Esquema hierárquico



Na figura 25 é apresentado o exemplo da construção da medição utilizando o esquema hierárquico para a meta “Analisar o processo de teste de software com a intenção de avaliar a produtividade do processo de teste e identificar oportunidades de melhoria”.

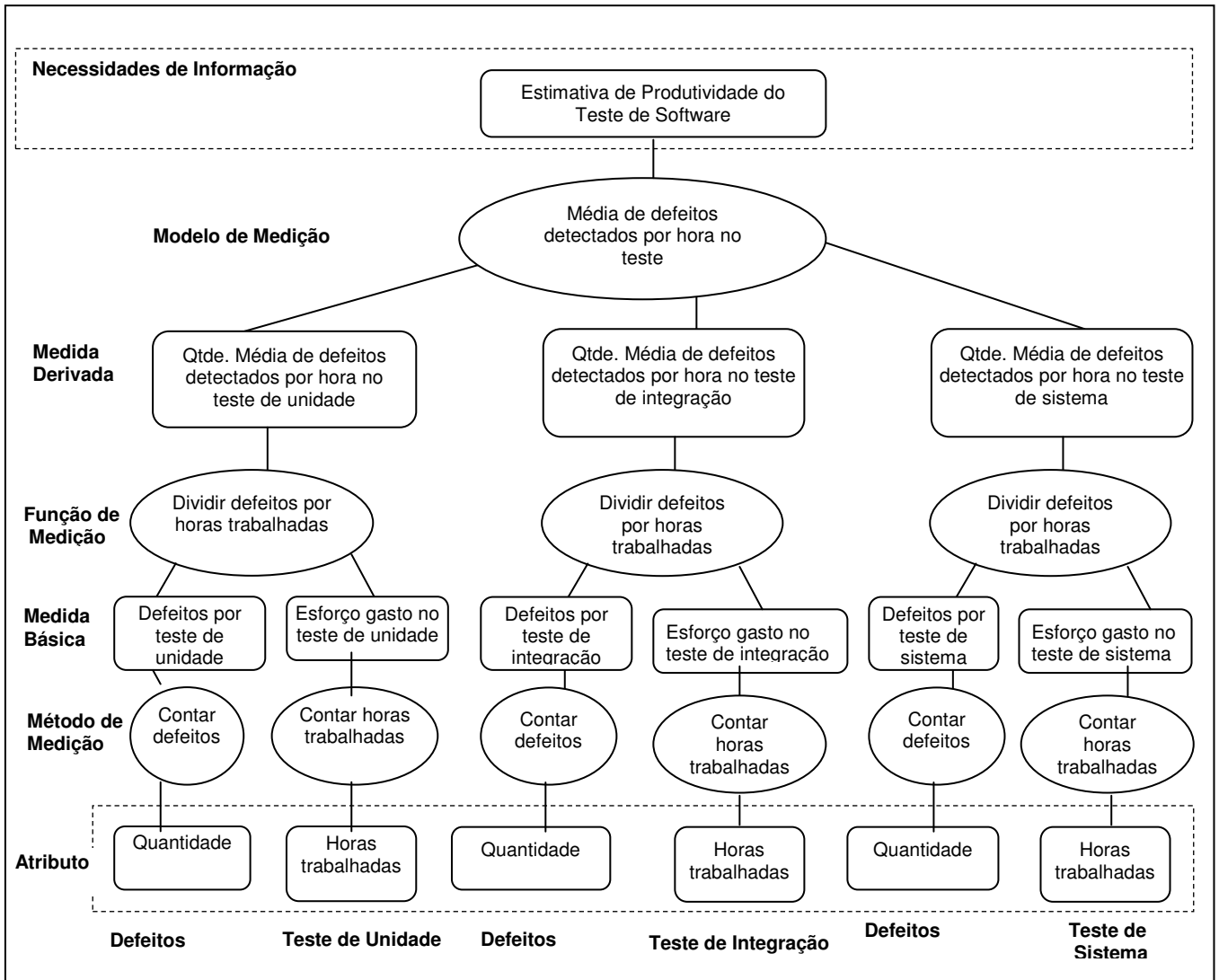


Figura 25 - Exemplo de Construção da medição – Esquema hierárquico

**A1.4) Especificar a Coleta de Dados:** nessa fase, de acordo com o PSM, são descritos a periodicidade da coleta dos dados, os recursos e ferramentas a serem utilizados. Como resultado da fase de planejamento é gerado um plano de medição que servirá como guia para a execução da medição. Nele consta, além dos objetivos, questões e medidas, também a descrição da forma de coleta e armazenamento dos dados obtidos na medição. Na figura 26, é apresentada a parte do modelo de plano de medição onde deve ser especificada a coleta de dados.

Periodicidade	Ferramentas	Descrição

Figura 26 – Plano de Medição - Especificação da Medição - Coleta de Dados

Como mostra a tabela 12, um exemplo de especificação da coleta de dados é definir a periodicidade da coleta dos dados como quinzenal, utilizando como ferramenta uma planilha do Word para o registro dos dados e na descrição indicar que serão considerados na medição somente os defeitos encontrados na fase de teste de software.

Tabela 12 - Exemplo do Plano de Medição - Especificação da Medição - Coleta de Dados

Periodicidade	Ferramentas	Descrição
Quinzenal	Planilhas Word	Aplicado na fase de teste de software do processo de desenvolvimento.

## A2) Execução da Medição

Como apresentado na figura 27, a fase de Execução da Medição é composta por três atividades: Coletar e Processar Dados, Analisar os Dados e Produzir Resultados da Medição. Essa fase, da mesma forma que a fase de planejamento, apresenta uma relação direta com o processo de teste de software.

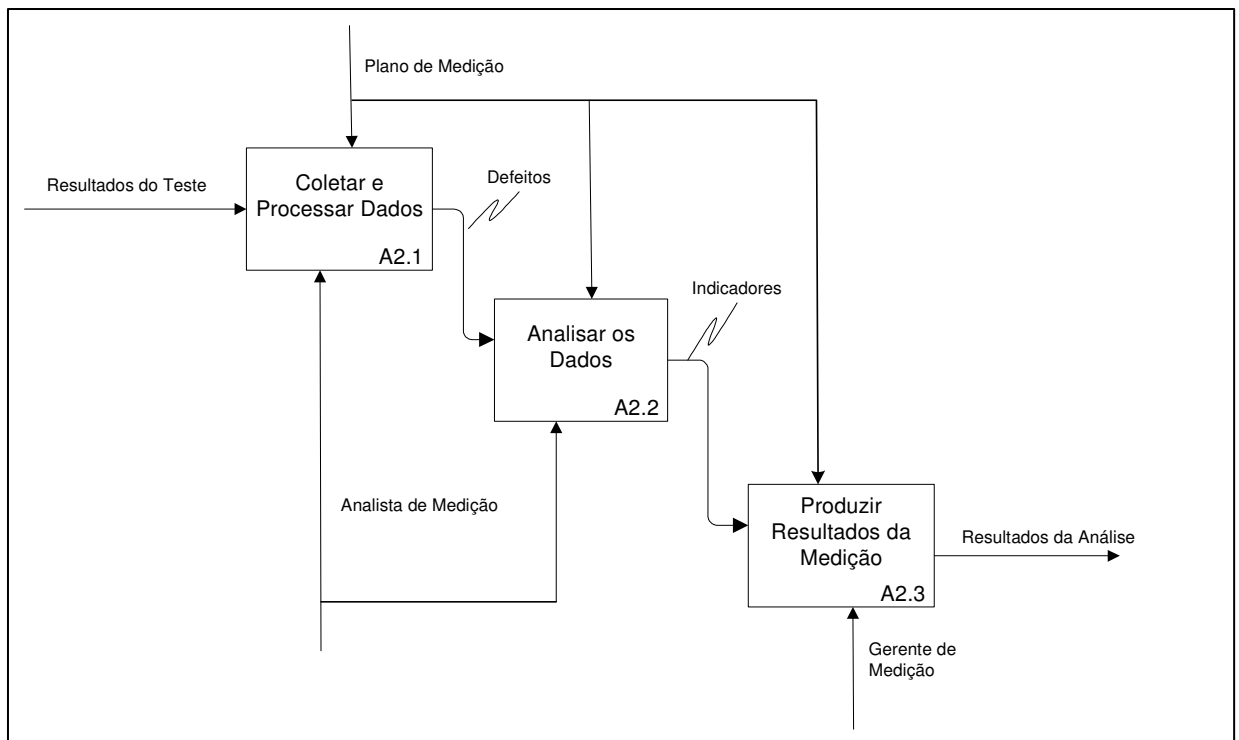


Figura 27 - Fase de Execução da Medição

Na fase de Coleta e Processamento dos Dados, o Analista de Medição coleta e processa os resultados das atividades de teste e os defeitos são apurados de acordo com as medidas especificadas no Plano de Medição. Ainda com base nesse plano, o Analista de Medição, efetua a análise das medidas considerando o modelo de análise dos indicadores, definido no Plano de Medição, e gera os indicadores.

De acordo com os critérios de decisão definidos no plano de medição, o Gerente de Medição, analisa os indicadores e produz os Resultados da Análise.

Esses resultados são informados aos responsáveis pelo processo de teste que o utilizarão como base para efetuar melhorias.

Nesse capítulo foi apresentado o contexto e a estrutura do processo de medição proposto, elaborados com base nos estudos realizados sobre o teste e a medição de software.

Foi apresentado o contexto da proposta descrevendo a relação existente entre o processo de teste e o processo de medição, ressaltando que a medição não interfere na definição da estrutura do processo de teste vigente, cabendo à organização selecionar e estruturar seu processo de teste.

O Processo de Teste fornece os subsídios para a aplicação do Processo de Medição, gerando o Plano de Teste e os Resultados do Teste. No Plano de Teste são descritas as estratégias de teste que servirão como base para o Planejamento da Medição. De acordo com as estratégias de teste adotadas são determinadas as metas e medidas a serem utilizadas. O resultado obtido nos testes fornece os dados sobre defeitos que serão a base para a fase de Execução da Medição.

Para o presente trabalho foi adotado um processo de teste que será utilizado como referência. Esse processo contempla a documentação do teste, que segue as recomendações da Norma IEEE 829 (1983), abrangendo os relatórios de plano de teste e de resultado do teste que são cruciais para estabelecer uma relação entre os processos de teste e de medição.

O Processo de Medição proposto foi estruturado com base no PSM, considerando alguns princípios do GQM, e abrange as atividades de planejamento e execução da medição. Portanto, dada uma necessidade organizacional são identificadas metas, são definidos os objetivos e questões. Com base nesses objetivos e questões são identificados as medidas e os indicadores que devem fornecer informações que atendam as necessidades identificadas. Para guiar o

planejamento foi elaborado um formulário, denominado Modelo de Plano de Medição, que auxilia na identificação das necessidades, na seleção e especificação de medidas e serve como guia para a execução da medição.

A seguir, é mostrada a aplicação do processo proposto por meio de dois estudos de caso, apresentando uma análise dos resultados obtidos.

## **CAPÍTULO 5 – ESTUDOS DE CASO**

Neste capítulo é apresentada uma aplicação prática do processo de medição proposto, por meio de dois estudos de caso, baseados em informações obtidas na área de desenvolvimento e manutenção de software de empresas distintas quanto ao processo de teste empregado e ao segmento de atuação. O objetivo destes estudos é analisar a aplicação de cada fase do processo proposto e avaliar como as informações obtidas contribuem para a melhoria do processo de teste.

O estudo de caso na empresa A foi realizado em outubro de 2007, coletando informações sobre os testes efetuados no período de janeiro a junho de 2007. Já o estudo na empresa B foi efetuado em outubro de 2007, coletando informações referentes aos testes realizados no *web site* no mês de setembro.

A seguir, são descritos as áreas, os projetos e o processo de teste de cada uma das empresas estudadas, bem como, é apresentado o detalhamento da aplicação do processo de medição.

### **5.1. Estudo de Caso – Empresa A**

#### **5.1.1. Caracterização dos Projetos da Área**

A empresa A, uma multinacional de tecnologia da informação há 25 anos no mercado internacional, atua há 7 anos no Brasil nos segmentos de manufatura, automotivo, financeiro e de serviços. Fornece a seus clientes soluções de gerenciamento de processos de negócio, desenvolvimento e integração de sistemas e infra-estrutura de TI.

As áreas da empresa relacionadas ao desenvolvimento e manutenção de software estão estruturadas de acordo com o conceito de fábrica de software, dividindo-se em fábrica de projetos e fábrica de programas. A fábrica de projetos é responsável pela especificação de novos produtos e de novas funcionalidades para os produtos já existentes, execução de testes e implantação do produto, enquanto, a fábrica de programas é uma linha de produção de programas padronizada para o desenvolvimento de software. Como numa linha de montagem, dada a especificação enviada pela fábrica de projetos, etapas seqüenciais de fabricação são seguidas até se obter, ao final, como produto o código a ser homologado pelo cliente. O cliente no

caso da fábrica de programas é a fábrica de projetos, de onde se originam as ordens de serviço.

Os programas desenvolvidos pela fábrica de programas são heterogêneos, originários de diversos projetos e sistemas e, em sua maioria, são desenvolvidos para o ambiente *mainframe*: *Cobol*, *Risc*, entre outros.

A fábrica de programas obteve a certificação CMMI nível 3 em 2006. Atualmente, mantêm uma política efetiva de qualidade visando garantir agilidade e qualidade a seus processos.

No estudo de caso da empresa A, a base para a aplicação do processo de medição são as informações, geradas pelo processo de teste, obtidas na fábrica de programas.

### **5.1.2. Processo de Teste da Área**

Como o processo de desenvolvimento empregado, pela empresa A, segue o conceito de fábrica de software são previstas, várias etapas a serem executadas de forma seqüencial, dentre as quais a etapa de teste de software.

No processo de teste adotado, pela fábrica de programas, estão definidos os testes do programador e da área de garantia da qualidade e a inspeção de código. Ao término dos testes os programas são enviados para a fábrica de projetos onde os analistas responsáveis efetuam o teste final de aceitação do produto.

O processo de desenvolvimento ocorre da seguinte forma: a fábrica de projetos envia um lote, formado por um ou mais programas, contendo uma ordem de serviço a ser executada e o plano de teste. O plano de teste somente descreve os passos que devem ser seguidos na execução dos testes.

Com base na documentação recebida é efetuado o planejamento das atividades de desenvolvimento e são distribuídos os programas entre os programadores. Os programadores efetuam a codificação e os testes de acordo com o plano de teste fornecido pela fábrica de projetos. Com a conclusão dos testes do programador, os programas são enviados para um segundo programador que efetua a inspeção de código, verificando a ocorrência de defeitos de padronização de código. Na ocorrência de defeitos, os programas são devolvidos, ao programador responsável, para as devidas correções. Caso contrário, são enviados para a área

de garantia da qualidade que efetua os testes de funcionalidade e de verificação de padrão de código.

Concluídos os testes da área de garantia da qualidade, sem ocorrência de defeitos, o lote é enviado para a fábrica de projetos onde os analistas efetuam os testes finais de aceitação do lote.

Todo o processo é controlado mantendo-se o registro das atividades executadas, horas gastas e os defeitos encontrados por meio da ferramenta *Rational ClearQuest*.

Com base nas atividades de teste executadas na fábrica de programas será aplicado o processo de medição e os resultados obtidos serão analisados como apresentado a seguir.

### **5.1.3. Detalhamento do Estudo de Caso**

Com as informações obtidas sobre a empresa A, principalmente quanto à estrutura do processo de teste empregado, pode-se ter uma visão geral do ambiente onde será aplicado o processo de medição. Como se trata de uma fábrica de programas, não é analisado um projeto em particular, são considerados todos os programas submetidos a testes na fábrica num determinado período, independente do projeto ao qual pertencem.

A partir destas informações inicia-se a aplicação do processo proposto que prevê duas fases: o Planejamento e a Execução da Medição. A seguir é apresentada a descrição do estudo de caso, detalhando cada fase do Processo de Medição.

#### **5.1.3.1. Planejamento da Medição**

A fase de Planejamento da Medição compreende as seguintes atividades: identificar necessidades de informação, selecionar medidas, especificar medidas e especificar a coleta de dados.

Para iniciar o Planejamento da Medição dois elementos são fundamentais: o Plano de Teste e as Necessidades Organizacionais. O Plano de Teste por fornecer as informações sobre as estratégias de teste utilizadas, que servirão de guia para a definição das medidas a serem consideradas no processo de medição. No caso da empresa A, o Plano de Teste não pôde ser utilizado por conter apenas a descrição

dos procedimentos a serem seguidos durante a execução do teste. No entanto, a empresa possui um segundo documento onde descreve o processo de teste, detalhando as estratégias de teste que, obrigatoriamente, devem ser aplicadas pela fábrica de programas. Portanto, na medição, esse documento é utilizado em substituição ao Plano de Teste.

Outro ponto importante no Planejamento da Medição é a identificação das Necessidades Organizacionais, que são informações que os tomadores de decisão precisam obter por meio da medição para direcionarem suas ações. Essas necessidades organizacionais serão a base para a identificação das metas, dos objetivos e das medidas a serem obtidas com a medição.

A empresa A, como numa linha de produção, necessita de agilidade e qualidade em seu processo produtivo de forma a obter produtos com qualidade a baixos custos. Portanto, um grande problema é a ocorrência de defeitos nos produtos, no ambiente do cliente, que geram prejuízos para a imagem da empresa e o retrabalho na correção dos defeitos, acarretando aumento nos custos do processo. Desta forma, uma necessidade da organização é avaliar o processo de teste identificando pontos problemáticos nas etapas do teste, visando propor melhorias no processo. De acordo com as estratégias de teste empregadas e as necessidades organizacionais identificadas, o primeiro passo do planejamento é, seguindo o Modelo de Plano de Medição, definir as metas da medição descrevendo o propósito, a perspectiva e o ambiente da medição como mostrado na tabela 13.

#### A1.1) Identificação das Necessidades de Informação

Tabela 13 - Estudo de Caso - Empresa A - Descrição das Metas

Cód. Meta	Propósito	Perspectiva	Ambiente
1	<p>Analisar o processo de teste de software da fábrica de programas.</p> <p>Com a intenção de avaliar os defeitos detectados no processo de teste.</p>	<p>Fornecer embasamento quantitativo para a avaliação da detecção de defeitos por estratégia de teste adotada.</p> <p>Do ponto de vista do gerente da fábrica de programas.</p>	Fábrica de desenvolvimento de programas <i>mainframe</i> .
2	<p>Analisar o processo de teste de software da fábrica de programas.</p> <p>Com a intenção de avaliar o teste de software quanto ao esforço despendido.</p>	<p>Fornecer embasamento quantitativo para a avaliação do custo do teste de software quanto ao esforço gasto em cada estratégia de teste adotada.</p> <p>Do ponto de vista do gerente da fábrica de programas.</p>	Fábrica de desenvolvimento de programas <i>mainframe</i> .

Como o foco é o processo de teste, foram identificadas duas metas para medição: avaliar o processo de teste quanto à distribuição de defeitos e avaliar o



esforço despendido no teste. Esta medição visa analisar cada estratégia de teste empregada de forma a reduzir custos e melhorar a qualidade dos processos.

O propósito descreve o que se deseja conhecer e com qual objetivo. A perspectiva define o foco da medição e o ponto de vista a ser considerado. Já o ambiente descreve o contexto de aplicação da medição. Neste caso, o ponto de vista considerado é do gerente da fábrica de programas e o ambiente é uma fábrica de desenvolvimento e manutenção de programas *mainframe*.

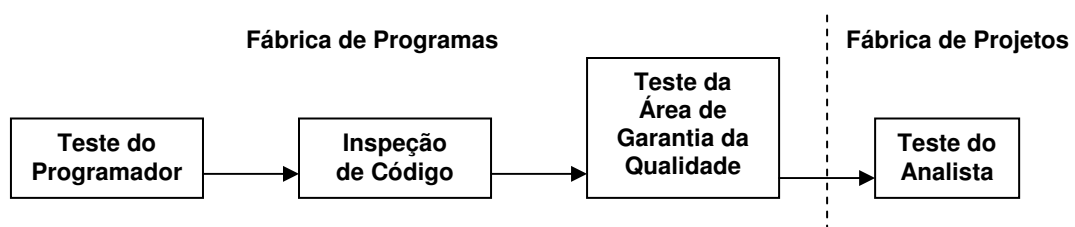
Com as metas definidas, foram descritos os papéis dos envolvidos na medição, como mostra a tabela 14.

**Tabela 14 - Estudo de Caso - Empresa A – Papéis e Responsáveis**

Papel	Responsabilidade
Gerente de Medição	- Planejar a medição; - Analisar os indicadores resultantes da medição e produzir recomendações.
Analista de Medição	- Coletar e processar os dados de teste; - Analisar os dados de teste produzindo os indicadores de medição.

O papel de Gerente de Medição foi assumido pela autora deste trabalho e o de Analista de Medição foi assumido, em parte, pela equipe da área de garantia da qualidade da empresa responsável pela coleta dos dados e pela autora, novamente, que efetuou o processamento e a análise dos dados.

Com as metas definidas e os papéis descritos, foram identificados os objetivos específicos a serem obtidos de acordo com cada meta definida. Como mostra a figura 28, o processo de teste contém as estratégias de teste do programador, da área de garantia da qualidade e do analista e a inspeção de código. Na medição para cada meta definida são identificados os objetivos acerca das estratégias de teste utilizadas pela empresa.



**Figura 28 - Estudo de Caso - Empresa A - Estratégias de Teste**

O objetivo define o que é necessário conhecer de cada estratégia de teste para se atingir as metas desejadas. No caso da meta 1 - “Analisar o processo de teste de software da fábrica de programas. Com a intenção de avaliar os defeitos

detectados no processo de teste”, são consideradas as estratégias de teste da área de garantia da qualidade e do analista e a inspeção de código. O teste do programador não foi considerado, por não fazer parte do processo de teste da empresa o registro dos eventuais defeitos. Já os defeitos encontrados no Teste do Analista foram considerados, mesmo o teste sendo efetuado na fábrica de projetos, porque os defeitos encontrados são reportados para a fábrica de programas que mantêm o registro desses defeitos para as correções necessárias.

No caso da meta 2 – “Analisar o processo de teste de software da fábrica de programas. Com a intenção de avaliar o teste de software quanto ao esforço despendido”, são consideradas as estratégias de teste do programador e da área de garantia da qualidade e a inspeção de código. Neste caso, o Teste do Analista não foi considerado, porque a fábrica de programas não tem acesso às horas gastas com os testes efetuados na fábrica de projetos.

Como mostra a tabela 15, para cada meta foram gerados diversos objetivos e com base em cada objetivo foram formuladas questões que, ao serem respondidas quantitativamente, permitirão atingir os objetivos.

**Tabela 15 - Estudo de Caso - Empresa A - Identificação das Necessidades de Informação**

<b>Cód. Meta</b>	<b>Objetivo</b>	<b>Questões</b>
<b>1</b>	Avaliar a quantidade de defeitos detectados na inspeção de código.	- Qual a quantidade de defeitos detectados na inspeção de código? - Qual a quantidade total de defeitos encontrados nos testes?
	Avaliar a quantidade de defeitos detectados no teste da área de qualidade.	- Qual a quantidade de defeitos detectados no teste da área de qualidade? - Qual a quantidade total de defeitos encontrados nos testes?
	Avaliar a quantidade de defeitos detectados no teste do analista.	- Qual a quantidade de defeitos detectados no teste do analista? - Qual a quantidade total de defeitos encontrados nos testes?
<b>2</b>	Avaliar o esforço gasto no teste do programador.	Qual o esforço gasto no teste do programador? Qual o valor médio hora do teste do programador?
	Avaliar o esforço gasto na inspeção de código.	Qual o esforço gasto na inspeção de código? Qual o valor médio hora da inspeção de código?
	Avaliar o esforço gasto no teste da área de qualidade.	Qual o esforço gasto no teste da área de qualidade? Qual o valor médio hora do teste da área de qualidade?

Concluída a identificação das necessidades de informação, o próximo passo é a seleção de medidas.

#### **A1.2) Seleção de Medidas**

De acordo com as metas e objetivos, para cada questão existente, foram identificadas as entidades que podem ser medidas e os atributos de cada entidade

que serão considerados de forma a fornecer subsídios para responder a essas questões.

As entidades determinam o que será avaliado e qual o objetivo da análise. Os atributos descrevem quais as características desse objeto que serão medidas. Com base nas entidades e atributos foram definidas as medidas básicas a serem obtidas com a medição, como mostra a tabela 16.

**Tabela 16 - Estudo de Caso - Empresa A - Seleção de Medidas**

<b>Cód. Meta</b>	<b>Objetivo</b>	<b>Questões</b>	<b>Entidades Relevantes</b>	<b>Atributos</b>	<b>Medidas Básicas</b>
<b>1</b>	Avaliar a quantidade de defeitos detectados na inspeção de código.	- Qual a quantidade de defeitos detectados na inspeção de código.	Defeitos detectados na inspeção de código.	Quantidade	Quantidade de defeitos detectados na inspeção de código.
		- Qual a quantidade total de defeitos encontrados nos testes?	Cada defeito encontrado no teste.	Quantidade	Total de defeitos encontrados nos testes.
	Avaliar a quantidade de defeitos detectados no teste da área de qualidade.	- Qual a quantidade de defeitos detectados no teste da área de qualidade.	Defeitos detectados no teste da área de qualidade.	Quantidade	Quantidade de defeitos detectados no teste da área de qualidade
		- Qual a quantidade total de defeitos encontrados nos testes?	Cada defeito encontrado no teste.	Quantidade	Total de defeitos encontrados nos testes.
	Avaliar a quantidade de defeitos detectados no teste do analista.	- Qual a quantidade de defeitos detectados no teste do analista.	Defeitos detectados no teste do analista.	Quantidade	Quantidade de defeitos detectados no teste do analista.
		- Qual a quantidade total de defeitos encontrados nos testes?	Cada defeito encontrado no teste.	Quantidade	Total de defeitos encontrados nos testes.
<b>2</b>	Avaliar o esforço gasto no teste do programador.	Qual o esforço gasto no teste do programador?	Teste do programador	Horas trabalhadas	Esforço do teste do programador
		Qual o valor médio hora do teste do programador?	Recursos alocados no teste do programador	Valor hora	Valor hora médio teste de programador
	Avaliar o esforço gasto na inspeção de código.	Qual o esforço gasto na inspeção de código?	Inspeção de código	Horas trabalhadas	Esforço da inspeção de código.
		Qual o valor médio hora da inspeção de código?	Recursos alocados na inspeção de código	Valor hora	Valor hora médio da inspeção de código
	Avaliar o esforço gasto no teste da área de qualidade.	Qual o esforço gasto no teste da área de qualidade?	Teste da área de qualidade	Horas trabalhadas	Esforço do teste da área de área de qualidade
		Qual o valor médio hora do teste da área de qualidade?	Recursos alocados no teste da área de qualidade	Valor hora	Valor hora médio teste da área de qualidade

Com a identificação das necessidades de informação e a seleção das medidas, o próximo passo é a construção da medição. Essa construção consiste em

especificar ou descrever as medidas que serão utilizadas na medição. Nessa atividade são descritos: as medidas básicas, o método de medição e os indicadores a serem analisados.

O primeiro passo na construção da medição é especificar cada medida básica definida na atividade de seleção de medidas, como mostrado na tabela 17. Para cada medida básica descreve-se: o método de medição, o tipo de método, o tipo de escala e a unidade de medida utilizada.

### A1.3) Especificação de Medidas

**Tabela 17 - Estudo de Caso - Empresa A - Especificação de Medidas Básicas**

<b>Cód.Meta</b>	<b>Medida Básica</b>	<b>Método de Medição</b>	<b>Tipo de Método</b>	<b>Tipo de Escala</b>	<b>Unidade de Medida</b>
<b>1</b>	Quantidade de defeitos detectados na inspeção de código.	Contar defeitos	Objetivo	Absoluta	Número de defeitos
	Quantidade de defeitos detectados no teste da área de qualidade	Contar defeitos	Objetivo	Absoluta	Número de defeitos
	Quantidade de defeitos detectados no teste do analista.	Contar defeitos	Objetivo	Absoluta	Número de defeitos
	Total de defeitos encontrados nos testes.	Contar defeitos	Objetivo	Absoluta	Número de defeitos
<b>2</b>	Esforço do teste do programador	Contar horas trabalhadas	Objetivo	Absoluta	Horas trabalhadas
	Esforço da inspeção de código	Contar horas trabalhadas	Objetivo	Absoluta	Horas trabalhadas
	Esforço do teste da área de qualidade	Contar horas trabalhadas	Objetivo	Absoluta	Horas trabalhadas
	Valor hora médio do teste de programador	Valor hora	Objetivo	Absoluta	Valor hora
	Valor hora médio da inspeção de código	Valor hora	Objetivo	Absoluta	Valor hora
	Valor hora médio do teste da área de qualidade	Valor hora	Objetivo	Absoluta	Valor hora

De acordo com a combinação das medidas básicas são identificadas as medidas derivadas, como mostra a tabela 18.

Tabela 18 - Estudo de Caso - Empresa A - Especificação de Medidas Derivadas

Meta	Medida Derivada	Função de Medição
1	1. Percentual de defeitos detectados na inspeção de código.	Dividir o total de defeitos detectados na inspeção de código pelo total de defeitos encontrados em todos os testes e multiplicar por 100.
	2. Percentual de defeitos detectados no teste da área de qualidade	Dividir o total de defeitos detectados no teste da área de qualidade pelo total de defeitos encontrados em todos os testes e multiplicar por 100.
	3. Percentual de defeitos detectados no teste do analista.	Dividir o total de defeitos detectados no teste do analista pelo total de defeitos encontrados em todos os testes e multiplicar por 100.
2	1. Esforço do teste do programador.	Calcular esforço do teste do programador / Total de esforço do teste
	2. Esforço da inspeção de código.	Calcular esforço da inspeção de código / Total de esforço do teste
	3. Esforço do teste da área de qualidade.	Calcular esforço do teste da área de qualidade / Total de esforço do teste
	4. Custo médio valor hora do teste	Calcular a média do valor hora do teste

Com a combinação das medidas derivadas obtêm-se os indicadores. Esses indicadores são especificados, como mostra a tabela 19, definindo-se o critério de decisão a ser considerado na análise do indicador. O indicador deve servir como resposta às necessidades de informação, definidas no plano de medição, e os critérios de decisão devem ser utilizados como referência para análise do indicador.

Tabela 19 - Estudo de Caso - Empresa A - Especificação de Indicadores

Cód Meta	Medida Derivada	Descrição do Indicador	Modelo de Análise	Crítérios de Decisão
1	1. Percentual de defeitos detectados na inspeção de código. 2. Percentual de defeitos detectados no teste da área de qualidade 3. Percentual de defeitos detectados no teste do analista.	Distribuição de defeitos detectados no teste de software.	Não se Aplica <sup>3</sup> . (Neste caso)	Quanto maior o percentual de defeitos detectados, por estratégia de teste, melhor.
2	1. Esforço total do teste de software 2. Custo Médio Valor Hora do Teste	Estimativa de custo do teste de software.	Calcular: Esforço Total do Teste x Custo Médio Valor Hora do Teste	Quanto menor o custo do teste de software melhor.

A figura 29, mostra a construção da medição para a meta 1 - “Analisar o processo de teste de software da fábrica de programas. Com a intenção de avaliar os defeitos detectados no processo de teste”, de forma hierárquica. Com base nas

<sup>3</sup> Neste caso não é necessária a definição de um modelo de análise para o indicador, considerando que somente a análise dos percentuais gerados e das medidas derivadas já permitem avaliar a distribuição dos defeitos detectados no teste de software.

necessidades identificadas são definidas: as entidades, os atributos, as medidas básicas até se obter os indicadores.

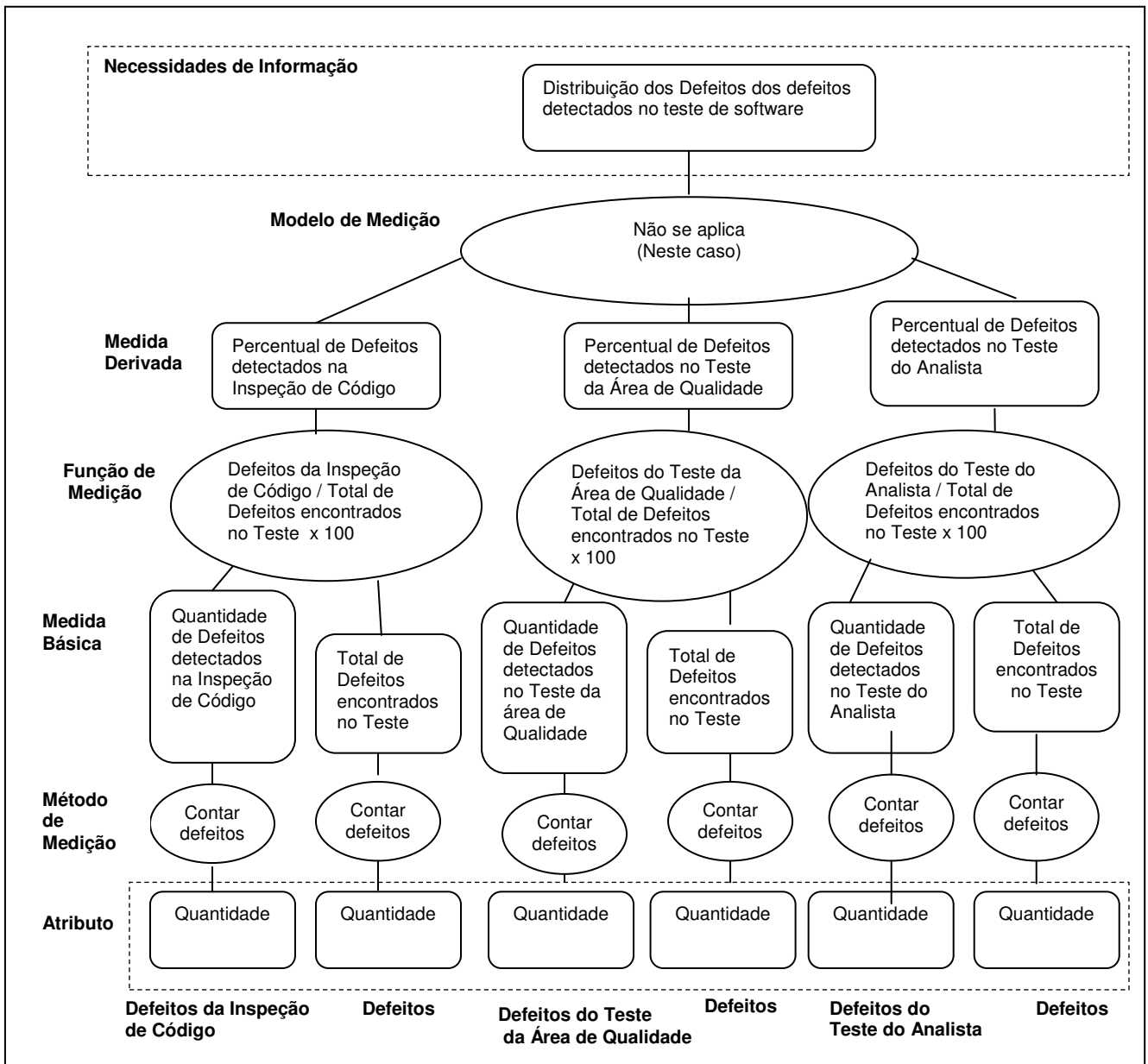


Figura 29 - Estudo de Caso - Empresa A - Construção da Mediç o (Meta 1)

A figura 30, apresenta a constru o da medi o para a meta 2 – “Analisar o processo de teste de software da f brica de programas. Com a inten o de avaliar o teste de software quanto ao esfor o despendido e identificar melhorias”.

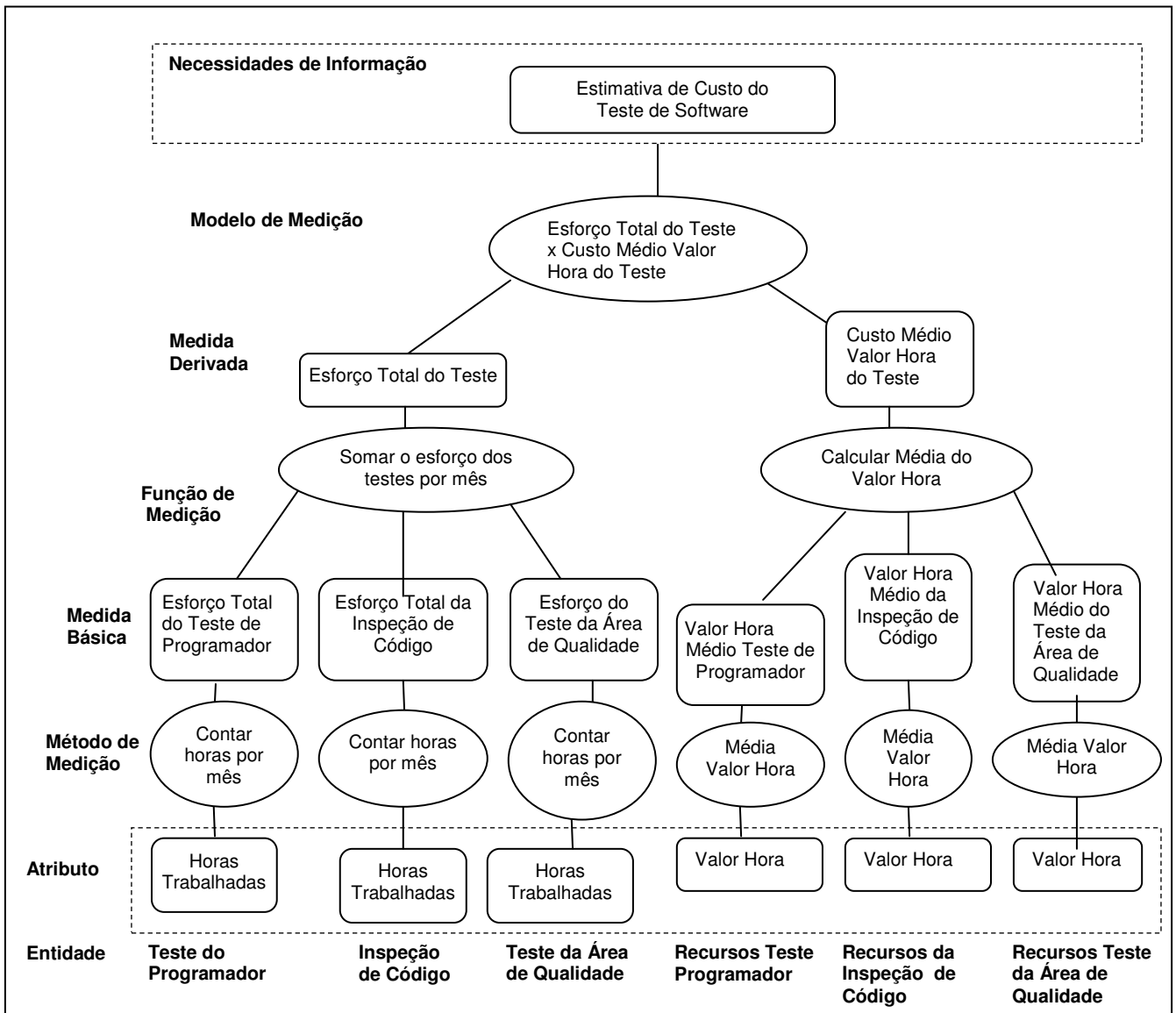


Figura 30 - Estudo de Caso - Empresa A - Construção da Medição (Meta 2)

A coleta de dados foi especificada, junto à área de Garantia da Qualidade, levantando-se a forma de armazenamento dos resultados dos testes. Foi apurado que todas as informações são armazenadas na ferramenta *Rational ClearQuest* da empresa IBM *Rational*, portanto consultas ao banco de dados desta ferramenta foram elaboradas de forma a recuperar as medidas definidas no plano de medição. A especificação da coleta de dados é mostrada na tabela 20.

Tabela 20 - Estudo de Caso - Empresa A - Especificação da Coleta de Dados

Periodicidade	Ferramentas	Descrição
Mensal Período de análise de janeiro a junho de 2007.	- Ferramenta <i>Rational ClearQuest</i> - Planilha Excel	Serão elaboradas consultas ao banco de dados SQL Server para coleta das informações necessárias.

Com a fase de Planejamento da Medição concluída, a próxima fase do processo é a Execução da Medição.

### 5.1.3.2. Execução da Medição

A fase de Execução da Medição compreende as seguintes atividades: coletar e processar dados, analisar os dados e produzir recomendações.

Nessa fase os dados resultantes da aplicação do teste de software foram coletados e processados de acordo com o Plano de Medição. Para a meta 1 – “Avaliar os defeitos detectados no processo de teste” foram consideradas as medidas básicas:

- quantidade de defeitos detectados na Inspeção de Código;
- quantidade de defeitos detectados no Teste da Área de Qualidade;
- quantidade de defeitos detectados no Teste do Analista;
- total de defeitos encontrados nos testes.

Os dados foram processados totalizando os defeitos para cada estratégia de teste e mês de detecção do defeito, obtendo-se as medidas básicas, como mostrado na tabela 21.

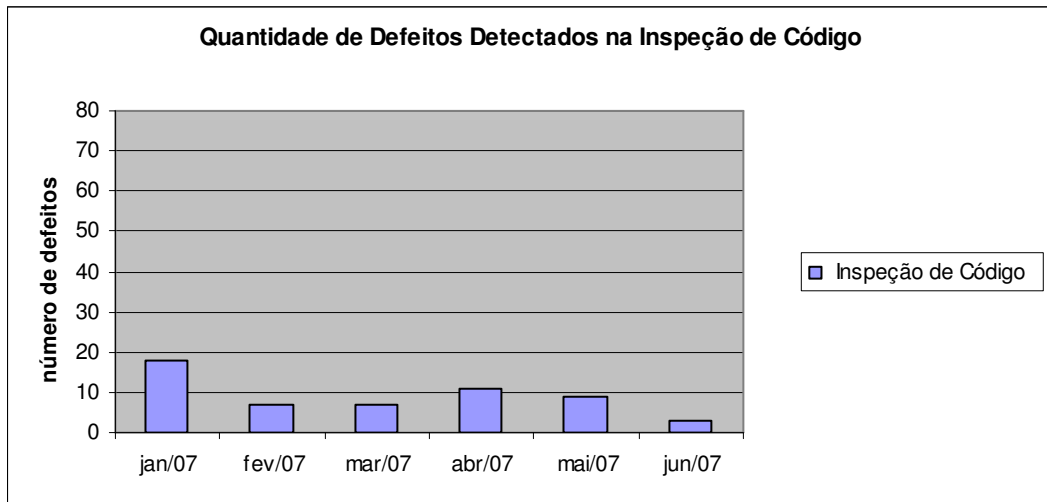
Tabela 21 - Estudo de Caso – Empresa A – Medidas Básicas (Meta 1)

Mês/Ano	Quantidade de Defeitos Detectados na Inspeção de Código	Quantidade de Defeitos Detectados no Teste da Área de Qualidade	Quantidade de Defeitos Detectados no Teste do Analista	Total de Defeitos encontrados nos Testes
Jan/07	18	52	15	85
Fev/07	7	18	12	37
Mar/07	7	20	20	47
Abr/07	11	38	15	64
Mai/07	9	36	4	49
Jun/07	3	8	1	12
<b>Total</b>	<b>55</b>	<b>172</b>	<b>67</b>	<b>294</b>

Com base nos defeitos contabilizados, na tabela 21, foram gerados os gráficos de distribuição dos defeitos detectados nos testes da área de garantia da qualidade e do analista e na inspeção de código no período de janeiro a junho de 2007, que são mostrados, respectivamente, nas figuras 31, 32 e 33.

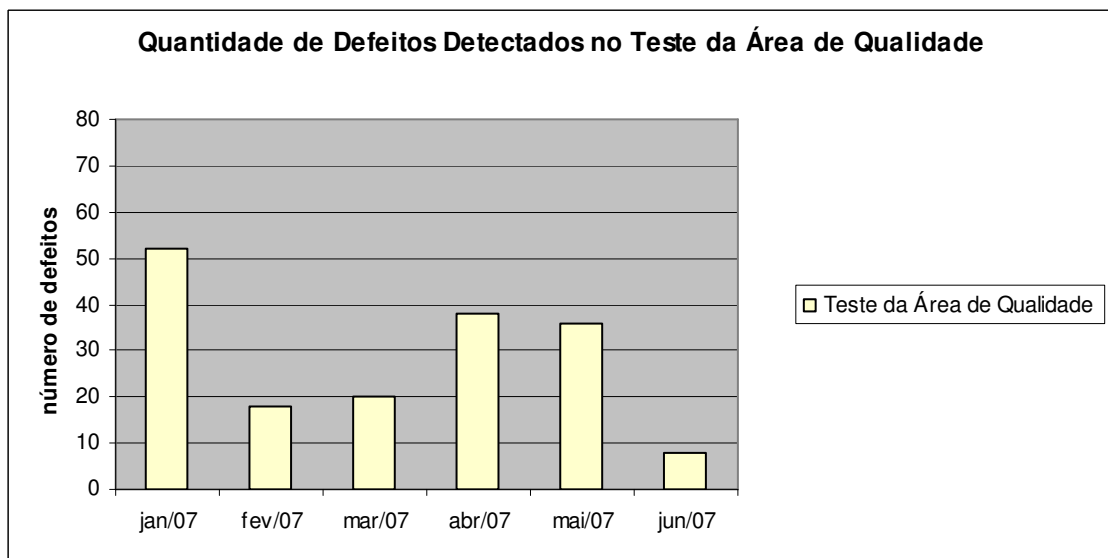
A figura 31 refere-se à medida básica “Quantidade de defeitos detectados na Inspeção de Código”, mostrando o número de defeitos encontrados por mês na inspeção de código no período de janeiro a junho de 2007.





**Figura 31 - Estudo de Caso - Empresa A - Gráfico de Defeitos de Inspeção de Código**

A figura 32 refere-se à medida básica “Quantidade de defeitos detectados no Teste da Área de Qualidade”, mostrando o número de defeitos encontrados por mês nos testes efetuados pela área de garantia de qualidade no período de janeiro a junho de 2007.



**Figura 32 - Estudo de Caso - Empresa A - Gráfico de Defeitos do Teste da Área de Qualidade**

Já a figura 33, refere-se à medida básica “Quantidade de defeitos detectados no Teste do Analista”, mostrando o número de defeitos encontrados por mês nos testes efetuados pelos analistas da fábrica de projetos e reportados à fábrica de programas no período de janeiro a junho de 2007.

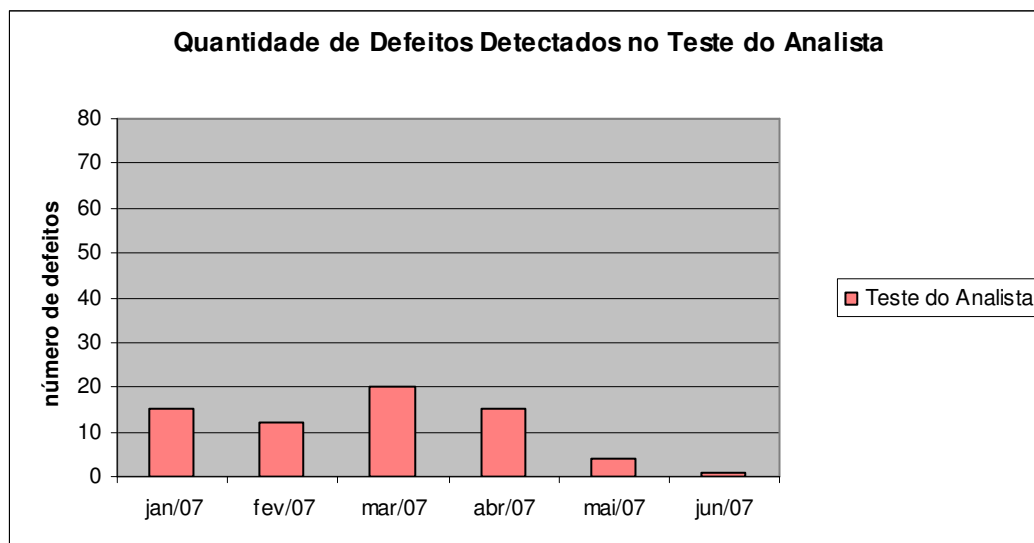


Figura 33 - Estudo de Caso - Empresa A - Gráfico de Defeitos do Teste do Analista

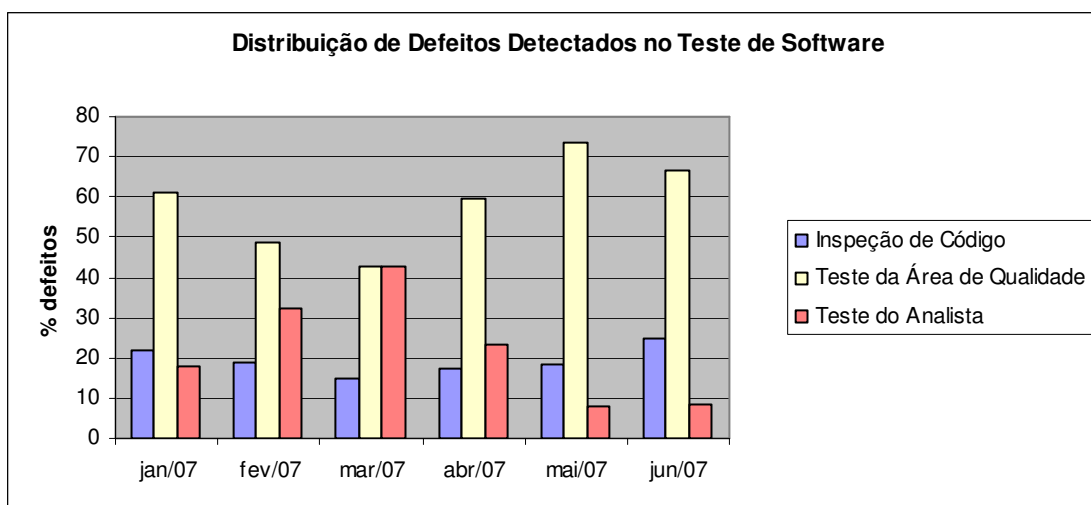
Utilizando as medidas básicas e a função de medição, que divide o total de defeitos obtidos para cada estratégia de teste pelo total de defeitos encontrados nos testes, foram geradas as medidas derivadas como mostra a tabela 22:

- percentual de defeitos detectados na Inspeção de Código;
- percentual de defeitos detectados no Teste da Área de Qualidade;
- percentual de defeitos detectados no Teste do Analista.

Tabela 22 - Estudo de Caso - Empresa A – Medidas Derivadas (Meta 1)

Mês/Ano	% Defeitos da Inspeção de Código	% Defeitos do Teste da Área de Qualidade	% Defeitos do Teste do Analista
Jan/07	21,75	61,18	17,65
Fev/07	18,92	48,65	32,44
Mar/07	14,9	42,55	42,55
Abr/07	17,19	59,38	23,44
Mai/07	18,37	73,47	8,16
Jun/07	25	66,67	8,34

Com base nas medidas derivadas foi obtido o indicador de “Distribuição de defeitos detectados no teste”, como mostrado na figura 34. Este indicador responde à necessidade de informação, definida no Plano de Medição, de avaliar o processo de teste quanto à detecção de defeitos. De acordo com o critério de decisão estabelecido para o indicador, o tomador de decisão, pode avaliar a quantidade de defeitos detectados de acordo com cada estratégia de teste aplicada e, desta forma, identificar os pontos problemáticos do teste e propor melhorias.



**Figura 34 - Estudo de Caso - Empresa A - Gráfico de Distribuição de Defeitos do Teste de Software**

A segunda meta definida no plano de medição é a meta 2 - “Avaliar o esforço despendido no teste”, com o foco na avaliação do custo do processo de teste. Neste caso, como mostrado na tabela 23, foram consideradas as medidas básicas de esforço de teste, definidas no plano de medição, que são:

- esforço gasto no Teste do Programador;
- esforço gasto na Inspeção de Código;
- esforço gasto no Teste da Área de Qualidade.

**Tabela 23 - Estudo de Caso – Empresa A – Medidas Básicas – Esforço (Meta 2)**

Mês/Ano	Esforço gasto no Teste do Programador	Esforço gasto na Inspeção de Código	Esforço gasto no Teste da Área de Qualidade	Esforço Total do Teste
Jan/07	480:14:00	69:37:00	216:26:00	766:17:00
Fev/07	341:01:00	45:19:00	94:15:00	480:35:00
Mar/07	522:54:00	30:32:00	148:45:00	702:11:00
Abr/07	350:32:00	60:34:00	243:16:00	654:22:00
Mai/07	251:02:00	58:50:00	231:45:00	541:37:00
Jun/07	255:05:00	29:46:00	98:55:00	383:46:00
<b>Total</b>	<b>2200:48:00</b>	<b>294:38:00</b>	<b>1033:22:00</b>	<b>3528:48:00</b>

Na tabela 24 são mostradas as medidas básicas de valor hora dos recursos empregados nos testes, por convenção são utilizados valores fictícios para as horas:

- Valor hora médio do Teste de Programador;

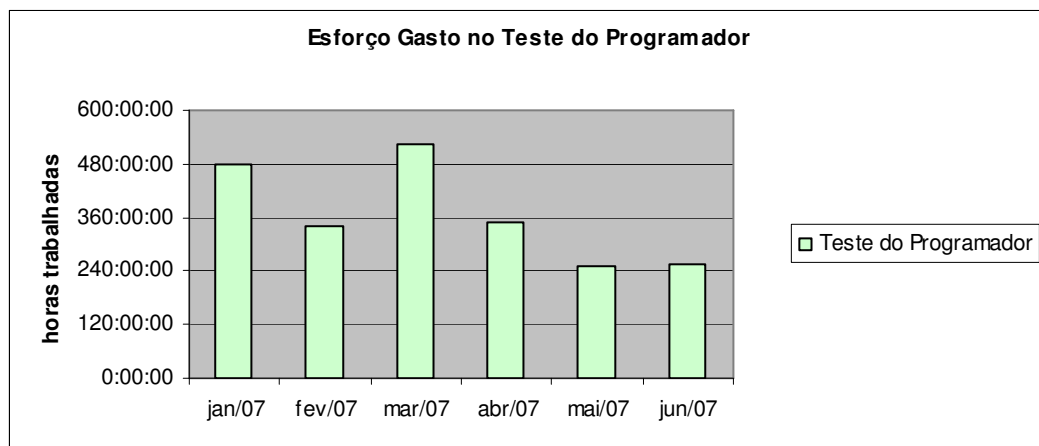
- Valor hora médio da Inspeção de Código;
- Valor hora médio do Teste da Área de Qualidade.

**Tabela 24 - Estudo de Caso – Empresa A – Medidas Básicas – Valor Hora (Meta 2)**

	Teste do Programador	Inspeção de Código	Teste da Área de Qualidade
Valor Hora	30	30	29

De acordo com os resultados contabilizados, mostrados na tabela 23, foram gerados os gráficos com a distribuição do esforço gasto nos testes do Programador e da área de Garantia de Qualidade e na Inspeção de Código, como apresentado, respectivamente, nas figuras 35, 36 e 37.

A figura 35 refere-se à medida básica “Esforço Gasto no Teste do Programador” que apresenta as horas gastas por mês nos testes do Programador, no período de janeiro a junho de 2007.



**Figura 35 - Estudo de Caso - Empresa A - Gráfico do Esforço do Teste do Programador**

A figura 36 refere-se à medida básica “Esforço Gasto na Inspeção de Código” que apresenta as horas gastas por mês na Inspeção de Código, no período de janeiro a junho de 2007.

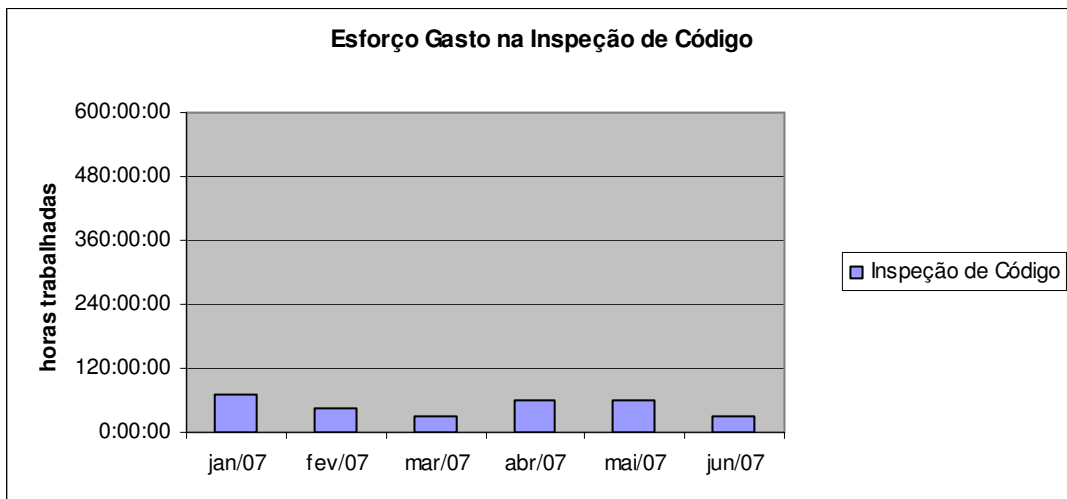


Figura 36 - Estudo de Caso - Empresa A - Gráfico do Esforço da Inspeção de Código

A figura 37 refere-se à medida básica “Esforço Gasto no Teste da Área de Qualidade” que apresenta as horas gastas por mês no teste efetuado pela Área de Garantia da Qualidade, no período de janeiro a junho de 2007.

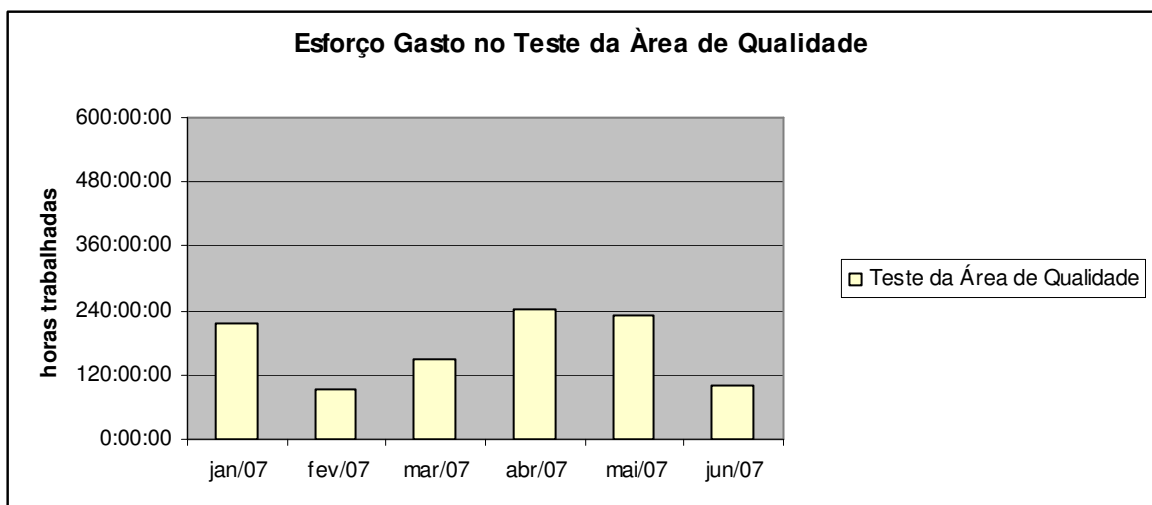
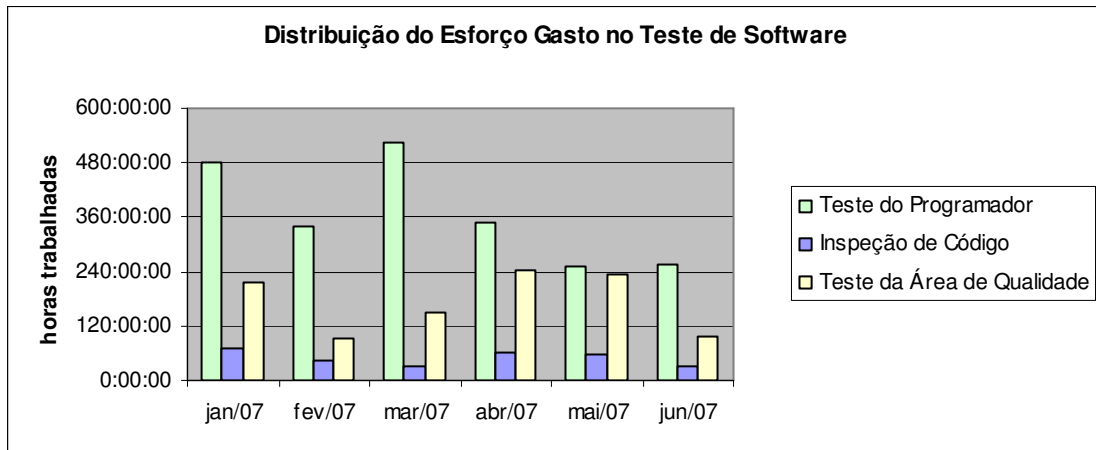


Figura 37 - Estudo de Caso - Empresa A – Gráfico do Esforço do Teste da Qualidade

Na figura 38 é apresentado o gráfico de distribuição do esforço do teste de software referente às medidas básicas obtidas, mostrando um comparativo entre as estratégias de teste do Programador e da Área de Garantia da Qualidade e da Inspeção de Código.



**Figura 38 - Estudo de Caso - Empresa A – Gráfico de Esforço do Teste de Software**

Com base nas medidas básicas foram geradas as medidas derivadas, como mostra a tabela 25:

- Esforço total do Teste de Software;
- Custo Médio Valor Hora do Teste

**Tabela 25 - Estudo de Caso - Empresa A - Medidas Derivadas (Meta 2)**

Mês/Ano	Esforço Total do Teste de Software	Custo Médio Valor Hora do Teste	Custo do Teste
Jan/07	766:17:00	30	22.988,50
Fev/07	480:35:00	30	14.417,50
Mar/07	702:11:00	30	21.065,50
Abr/07	654:22:00	30	19.631,00
Mai/07	541:37:00	30	16.248,50
Jun/07	383:46:00	30	11.501,50
<b>Total</b>	<b>3528:48:00</b>	<b>30</b>	<b>105.852,50</b>

Analisando as medidas derivadas tem-se o indicador “Estimativa de Custo do Teste de Software”, como mostrado na figura 39. O indicador obtido pode ser avaliado de acordo com o critério de decisão definido no plano de medição, que determina que quanto menor o custo do teste melhor. Esse indicador atende à necessidade de informação indicada pela empresa que é avaliar o esforço gasto no teste com o foco no custo.

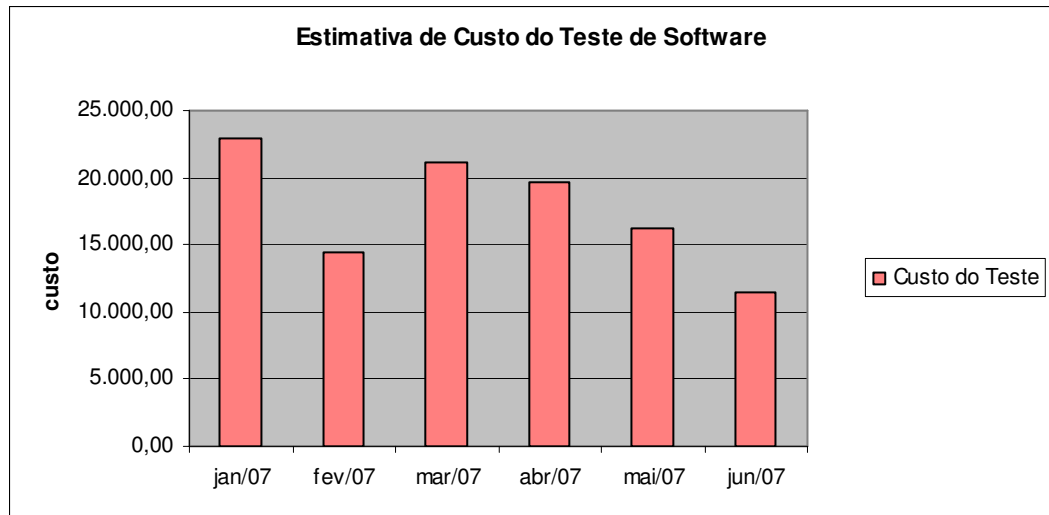


Figura 39 - Estudo de Caso - Empresa A - Gráfico de Estimativa de Custo do Teste de Software

#### 5.1.4. Análise da Aplicação do Processo de Medição

Com este estudo de caso pôde-se observar a aplicação do Processo de Medição, abordando as fases de Planejamento e Execução da Medição.

O Plano de Teste, previsto para ser a base para a extração de informações sobre o teste, não continha a descrição do Processo de Teste. As informações necessárias constavam em outro documento elaborado pela empresa, que foi utilizado sem acarretar prejuízos para a aplicação do Processo de Medição.

Com uma visão geral sobre a empresa e a estrutura do processo de teste adotado, pôde-se verificar a viabilidade da implantação do processo de medição observando se o processo de teste era efetivamente empregado e os resultados dos testes eram registrados de forma acessível. Como o processo de teste era executado e os dados eram registrados, o Processo de Medição pôde ser iniciado.

Na fase de Planejamento da Medição foram identificadas as necessidades de informação sobre o ponto de vista do gerente da fábrica de programas. Nesta fase é essencial o entendimento das necessidades de informação e a correta definição das metas de medição a serem alcançadas, portanto o gerente da fábrica de programas, como o principal interessado nos resultados da medição, teve participação efetiva em toda a fase de planejamento.

A fase de Planejamento da Medição foi efetuada seguindo o Plano de Medição, e de acordo com as necessidades de informação identificadas o plano foi preenchido, descrevendo as metas, os objetivos até se obter as medidas e

indicadores necessários para atender às necessidades de informação. Neste documento, também foi definida a forma de coleta, de processamento e de análise dos dados.

Na fase de Execução da Medição a coleta dos dados foi o ponto mais crítico do processo já que as medidas descritas no plano de medição precisavam ser extraídas do banco de dados da ferramenta *Rational ClearQuest*, exigindo um trabalho extra na elaboração de mecanismos para a recuperação dos dados, mostrando que as informações nem sempre eram registradas de uma forma que sua recuperação fosse simples e rápida. Neste ponto a necessidade de comprometimento da organização com a medição foi evidenciada, considerando que foi necessário a empresa dispor de membros da equipe para efetuarem a recuperação dos dados e auxiliarem no processamento e análise dos dados.

Durante as atividades de coleta e processamento dos dados foram identificadas oportunidades de melhoria no processo de teste, tais como: a inclusão de novas informações sobre o teste no banco de dados e, também, alterações na forma de registro dos dados visando facilitar a recuperação das informações sobre as estratégias de teste aplicadas. Com o processamento e a análise dos dados, foram gerados como resultados, gráficos e tabelas, utilizados pelos tomadores de decisão para direcionarem suas ações.

Neste estudo, foram utilizadas como base duas necessidades de informação definidas pelo gerente da fábrica de programas, mostrando que para ambas as necessidades o processo pôde ser aplicado, percorrendo todas as fases e atividades propostas. Ressalta-se que o envolvimento dos membros da organização em todas as fases do processo foi crucial para o êxito da medição.

Cabe observar que, durante a atividade de definição das metas, foi identificado que, dependendo das necessidades de informação, as metas definidas podem contemplar outras metas menores, que poderiam ser chamadas de submetas. No entanto, o Plano de Medição não possui uma área específica para as submetas o que representa uma oportunidade de desenvolvimento de um estudo que explore a análise de metas que contenham submetas propondo melhorias para a estrutura do Plano de Medição.



## **5.2. Estudo de Caso – Empresa B**

### **5.2.1. Caracterização dos Projetos da Área**

A empresa B, uma instituição pública de pesquisa tecnológica, possui uma área de desenvolvimento de software que atua há 14 anos no mercado nacional de tecnologia da informação. Essa área fornece serviços de desenvolvimento e verificação de sistemas para empresas privadas, mas tendo como seus principais clientes os órgãos públicos. Além, de atender a empresas públicas e privadas, também desenvolve sistemas para as áreas internas da instituição. Em 2005, obteve a certificação ISO 9001:2000 em desenvolvimento de sistemas computacionais.

Os projetos que compõem a área são bem heterogêneos, abrangendo diversas áreas de negócio. São desenvolvidos para ambiente Windows, rede local e WEB, utilizando banco de dados relacionais como SQL Server, Oracle e MySQL. Em geral, requerem agilidade no desenvolvimento, necessitando pelo menos uma versão preliminar em curto espaço de tempo. Esses projetos variam no tempo de execução dependendo do escopo definido, exigindo que os projetos sejam divididos em iterações que possibilitem a execução e a entrega do produto em prazos menores.

No estudo de caso da empresa B, a base para a aplicação do Processo de Medição são as informações geradas pelo processo de teste existente na área de desenvolvimento de software da instituição.

### **5.2.2. Processo de Teste da Área**

O processo de desenvolvimento empregado baseia-se no modelo espiral e prevê uma fase específica de teste de software que engloba as estratégias de teste: de Unidade, de Integração e de Homologação.

Os testes de unidade são efetuados pelo programador ao término do desenvolvimento dos programas. Com os testes de unidade concluídos são aplicados, pela equipe de teste, os testes de integração. Para guiar os testes de unidade e de integração há uma lista, contendo os defeitos mais comumente encontrados, que deve ser preenchida pelo responsável pelo teste. Todos os

defeitos encontrados nos testes são registrados numa ferramenta de acompanhamento de testes desenvolvida pela área em questão.

O teste de homologação é efetuado pelo cliente que, na ocorrência de defeitos, comunica a equipe de desenvolvimento.

Com base nas atividades de teste executadas por essa área será aplicado o processo de medição em um projeto selecionado para estudo e os resultados serão analisados como apresentado a seguir.

### **5.2.3. Detalhamento do Estudo de Caso**

Com as informações obtidas sobre a empresa B, principalmente quanto à estrutura do processo de teste empregado, têm-se uma visão geral do ambiente onde será aplicado o processo de medição.

Como a área de desenvolvimento de software trabalha com o foco em projetos, para a aplicação do processo de medição, foi selecionado para análise o projeto de um *web site*.

A partir dessas informações inicia-se a aplicação do processo proposto que prevê duas fases: o Planejamento e a Execução da Medição. A seguir o estudo de caso é descrito detalhando-se cada fase do Processo de Medição.

#### **5.2.3.1. Planejamento da Medição**

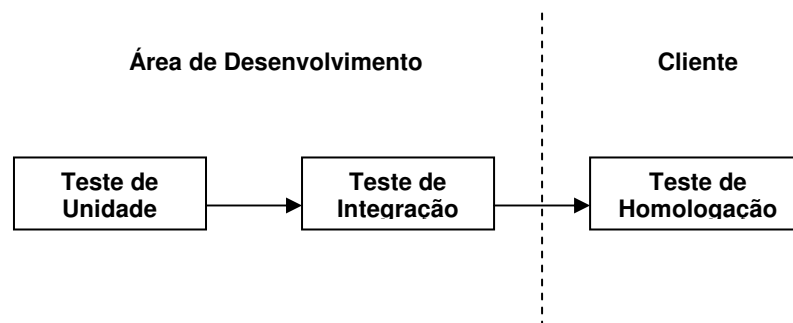
A fase de Planejamento da Medição compreende as seguintes atividades: identificar necessidades de informação, selecionar medidas, especificar medidas e especificar a coleta de dados.

Para iniciar o Planejamento da Medição dois elementos são fundamentais: o Plano de Teste e as Necessidades Organizacionais. O Plano de Teste por fornecer as informações sobre as estratégias de teste utilizadas e servir como base para a definição das medidas a serem consideradas no processo de medição. No caso da empresa B, o plano de teste não pôde ser utilizado por conter apenas a descrição dos procedimentos a serem seguidos durante a execução do teste. No entanto, a empresa possui um segundo documento onde descreve o processo de teste, detalhando as estratégias de teste que obrigatoriamente devem ser aplicadas em todos os projetos da área. Portanto, na medição, esse documento será utilizado em substituição ao Plano de Teste.

A Necessidade Organizacional é outro ponto importante no Planejamento da Medição, considerando que será a base para a identificação das necessidades de informação da área. As necessidades organizacionais são informações que os tomadores de decisão precisam obter por meio da medição para direcionarem suas ações.

A área de desenvolvimento de software não possui um Processo de Medição focado no processo de teste e os sistemas sob sua responsabilidade requerem agilidade no desenvolvimento e entregas em curtos espaços de tempo. Diante disso, foi identificada a necessidade organizacional de acompanhar o processo de teste de forma a conhecer o processo e identificar possíveis melhorias quanto à detecção do maior número de defeitos em menor tempo. Portanto, avaliar a produtividade do teste é a principal necessidade da organização.

O processo de teste utilizado, como mostra a figura 40, prevê as estratégias de teste de unidade, integração e homologação. No caso da medição somente são analisados os testes de unidade e de integração, considerando que o teste de homologação é efetuado pelo cliente não havendo registro de tempo de teste e de defeitos detectados.



**Figura 40 - Estudo de Caso - Empresa B - Estratégias de Teste**

De acordo, com o documento de descrição do processo de teste, as necessidades organizacionais e, seguindo o Modelo de Plano de Medição, o primeiro passo do planejamento é definir as metas da medição, descrevendo o propósito, a perspectiva e o ambiente de medição, como mostra a tabela 26.

Como a necessidade da empresa é a análise da produtividade do teste de software, é necessário avaliar o esforço despendido no teste e relacionar esse esforço com os defeitos encontrados.

### A1.1) Identificação das Necessidades de Informação

Tabela 26 - Estudo de Caso - Empresa B – Descrição das Metas

Cód. Meta	Propósito	Perspectiva	Ambiente
1	<p>Analisar o processo de teste de software.</p> <p>Com a intenção de avaliar a produtividade do processo de teste e identificar oportunidades de melhoria.</p>	<p>Fornecer um embasamento quantitativo para a avaliação da produtividade das atividades de teste.</p> <p>Do ponto de vista do (a) gerente de projeto.</p>	Área de desenvolvimento de <i>web sites</i> .

O propósito descreve o que se deseja conhecer e com qual objetivo. A perspectiva define o foco da medição e o ponto de vista a ser considerado. O ambiente descreve o contexto de aplicação da medição. Neste caso, o ponto de vista considerado é do gerente de projeto e o ambiente é uma área de desenvolvimento de *web sites*.

Com as metas definidas foram descritos os papéis dos envolvidos na medição, como mostra a tabela 27.

Tabela 27 - Estudo de Caso - Empresa B – Papéis e Responsabilidades

Papel	Responsabilidade
Gerente de Medição	<ul style="list-style-type: none"> <li>- Planejar a medição;</li> <li>- Analisar os indicadores resultantes da medição e produzir recomendações.</li> </ul>
Analista de Medição	<ul style="list-style-type: none"> <li>- Coletar e processar os dados de teste;</li> <li>- Analisar os dados de teste produzindo os indicadores de medição.</li> </ul>

Neste estudo, os papéis de gerente e analista de medição foram assumidos pela autora deste trabalho que efetuou o planejamento, a coleta, o processamento e a análise dos dados.

Com as metas definidas e os papéis descritos, foram identificados os objetivos específicos a serem obtidos de acordo com a meta definida. Como mostra a tabela 28, o objetivo define o que é necessário conhecer de cada estratégia de teste para se atingir as metas desejadas. No caso da meta de “Analisar o processo de teste de software. Com a intenção de avaliar a produtividade do processo de teste e identificar oportunidades de melhoria”, são considerados o esforço e os defeitos encontrados nas estratégias de teste de unidade e de integração.

Para cada meta podem ser gerados diversos objetivos e com base em cada um destes objetivos foram formuladas questões que, ao serem respondidas quantitativamente, permitirão o atingimento dos objetivos.

Tabela 28 - Estudo de Caso - Empresa B - Identificação das Necessidades de Informação

Cód. Meta	Objetivo	Questões
1	Avaliar o teste de unidade quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de unidade? - Qual o esforço gasto na execução do teste de unidade?
	Avaliar o teste de integração quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de integração? - Qual o esforço gasto na execução do teste de integração?

Concluída a identificação das necessidades de informação o próximo passo é a seleção de medidas.

### A1.2) Seleção de Medidas

Com base nas metas e objetivos, para cada questão existente, foram identificadas as entidades que podem ser medidas e os atributos de cada entidade que serão considerados de forma a fornecer subsídios para responder a essas questões.

As entidades determinam o que será avaliado e qual o objetivo da análise. Os atributos descrevem quais as características desse objeto que serão medidas. Com base nas entidades e atributos foram definidas as medidas básicas a serem obtidas com a medição, como mostra a tabela 29.

Tabela 29 - Estudo de Caso - Empresa B - Seleção de Medidas

Cód. Meta	Objetivo	Questões	Entidades Relevantes	Atributos	Medidas Básicas
1	Avaliar o teste de unidade quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de unidade?	Cada defeito detectado	- quantidade - data de detecção	Defeitos detectados no teste de unidade
		- Qual o esforço gasto na execução do teste de unidade?	Atividade de teste	- horas trabalhadas - data do teste	Esforço gasto no teste de unidade.
	Avaliar o teste de integração quanto à quantidade de defeitos detectados por hora.	- Qual a taxa de defeitos detectados no teste de integração?	Cada defeito detectado	- quantidade - data de detecção	Defeitos detectados no teste de integração.
		- Qual o esforço gasto na execução do teste de integração?	Atividade de teste	- horas trabalhadas - data do teste	Esforço gasto no teste de integração.

Com a identificação das necessidades de informação e a seleção das medidas, o próximo passo é a construção da medição. Esta construção consiste em especificar ou descrever as medidas que serão utilizadas na medição. Nessa atividade são descritos: as medidas básicas, o método de medição e os indicadores a serem analisados.

O primeiro passo na construção da medição é especificar cada medida básica definida na atividade de seleção de medidas, como mostrado na tabela 30. Para cada medida básica descreve-se: o método de medição, o tipo de método, o tipo de escala e a unidade de medida utilizada.

### A1.3) Especificação de Medidas

**Tabela 30 - Estudo de Caso - Empresa B - Especificação de Medidas Básicas**

Cód. Meta	Medida Básica	Método de Medição	Tipo de Método	Tipo de Escala	Unidade de Medida
1	Defeitos detectados no teste de unidade.	Contar defeitos	Objetivo	Absoluta	Número de defeitos
	Esforço gasto no teste de unidade.	Contar horas trabalhadas	Objetivo	Absoluta	Horas trabalhadas
	Defeitos detectados no teste de integração	Contar defeitos	Objetivo	Absoluta	Número de defeitos
	Esforço gasto no teste de integração.	Contar horas trabalhadas	Objetivo	Absoluta	Horas trabalhadas

De acordo com a combinação das medidas básicas são identificadas as medidas derivadas, como mostra a tabela 31.

**Tabela 31 - Estudo de Caso - Empresa B - Especificação de Medidas Derivadas**

Cód. Meta	Medida Derivada	Função de Medição
1	1. Quantidade média de defeitos detectados por hora no teste de unidade.	Dividir defeitos detectados por horas trabalhadas no teste de unidade.
	2. Quantidade média de defeitos detectados por hora no teste de integração.	Dividir defeitos detectados por horas trabalhadas no teste de integração.

Com base nas medidas derivadas são definidos os indicadores, como mostra a tabela 32, que devem responder as necessidades de informação identificadas para a medição. O critério de decisão determinado para o indicador serve como referência para a análise dos resultados obtidos.

**Tabela 32 - Estudo de Caso - Empresa B - Especificação de Indicadores**

Cód. Meta	Medida Derivada	Descrição do Indicador	Modelo de Análise	Critérios de Decisão
1	1. Quantidade média de defeitos detectados por hora no teste de unidade; 2. Quantidade média de defeitos detectados por hora no teste de integração.	Estimativa de produtividade na detecção de defeitos no teste de software	Média de defeitos detectados por hora em cada tipo de teste.	Quanto maior a quantidade de defeitos detectados por hora melhor a produtividade.

A figura 41, mostra a construção da medição para a meta de “Analisar o processo de teste de software. Com a intenção de avaliar a produtividade do

processo de teste e identificar oportunidades de melhoria”, de forma hierárquica. Com base nas necessidades identificadas são definidas: as entidades, os atributos, as medidas até se obter os indicadores.

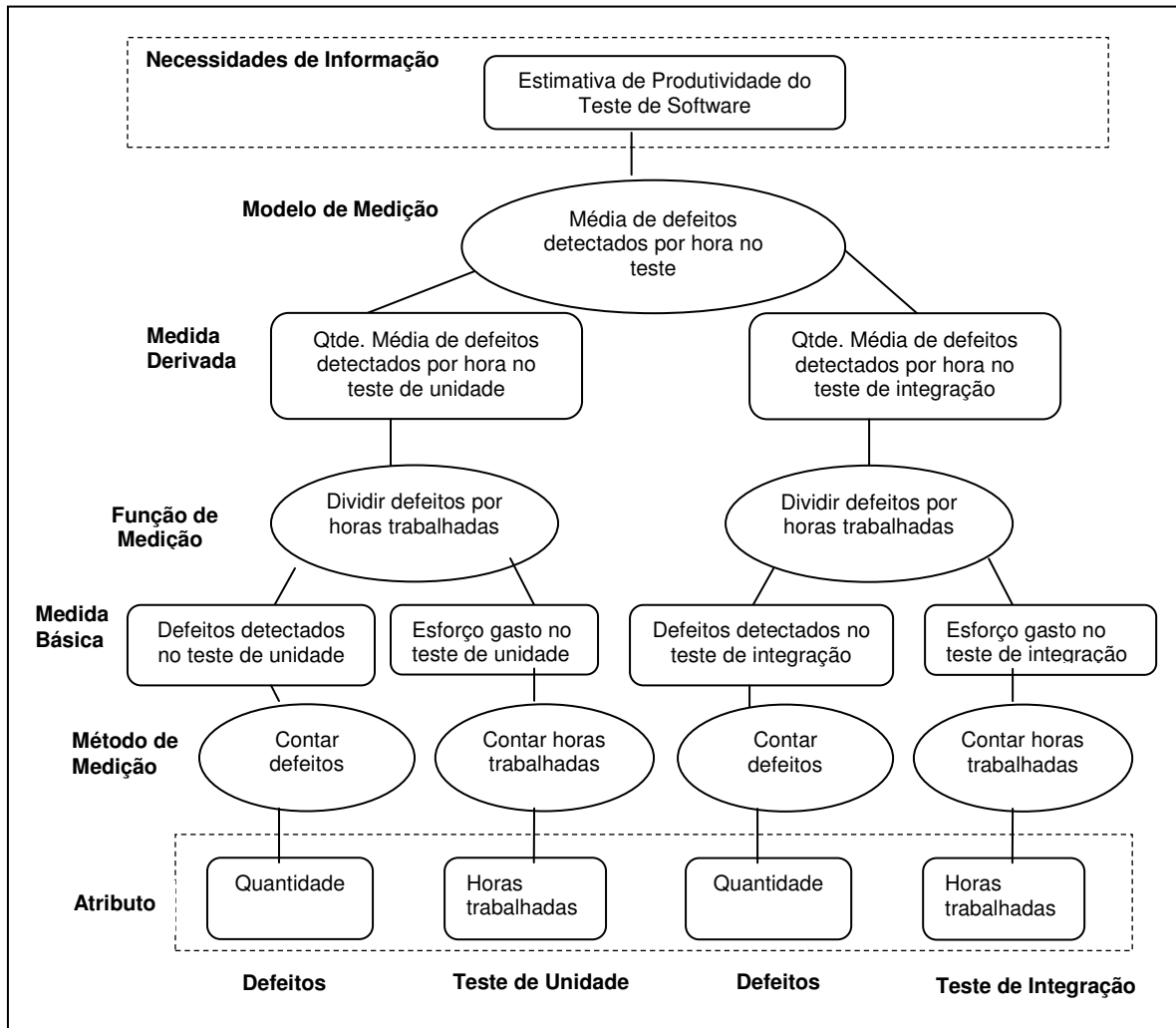


Figura 41 - Estudo de Caso - Empresa B - Construção da medição

Como mostra a tabela 33, foi especificado que a coleta de dados seria efetuada utilizando os dados registrados, em planilhas Excel, pelos programadores e testadores, sendo executada ao término da aplicação de cada estratégia de teste.

#### A1.4) Especificação da Coleta de Dados

Tabela 33 - Estudo de Caso - Empresa B - Especificação da Coleta de Dados

Periodicidade	Ferramentas	Descrição
Ao término de cada estratégia de teste aplicada.	Planilhas Excel	Aplicado na fase de teste de software do processo de desenvolvimento.

Com a fase de Planejamento da Medição concluída a próxima fase do processo é a Execução da Medição.

### 5.2.3.2. Execução da Medição

Na Execução da Medição os dados resultantes da aplicação do teste de software são coletados de acordo com a especificação de medidas contida no plano de medição.

Para facilitar o processamento dos dados foi gerada uma estrutura hierárquica do projeto descrevendo os programas, as unidades e os módulos a serem considerados na medição, como mostra a figura 42.

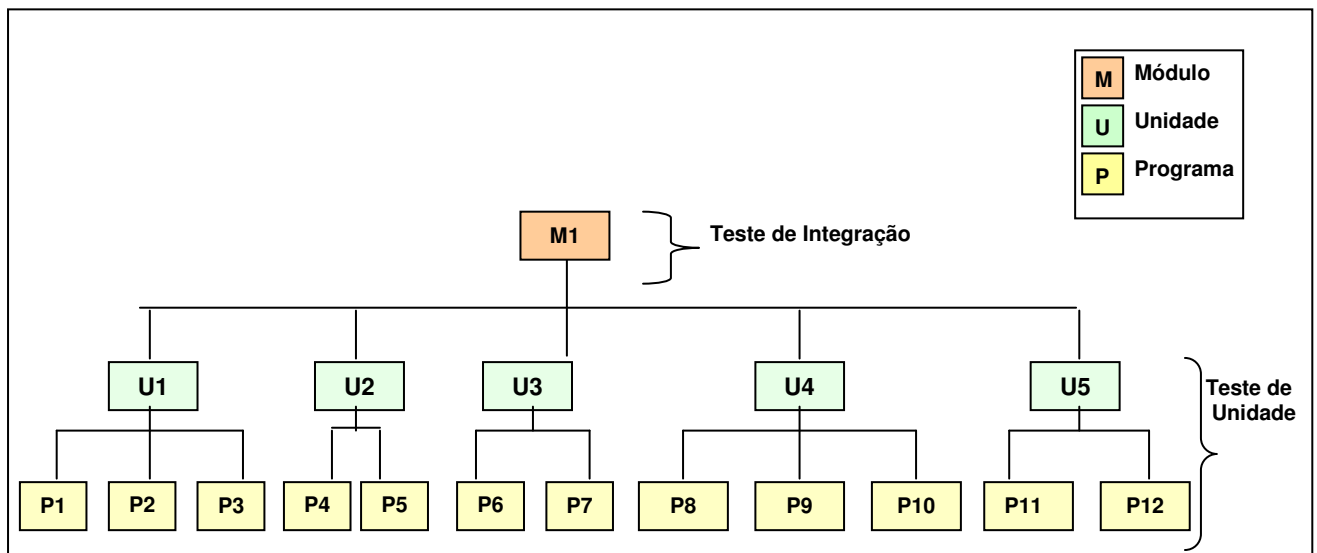


Figura 42 – Estudo de Caso – Empresa B - Estrutura do Projeto *Web Site*

As unidades são os itens avaliados no teste de unidade e os módulos são os itens avaliados no teste de integração. Cada unidade é formada por um ou mais programas, portanto ao contabilizar os defeitos e o esforço gastos nesses programas os valores serão totalizados nas unidades correspondentes.

Os dados resultantes de cada estratégia de teste foram processados separadamente com base nas medidas definidas no Plano de Medição. Foram contabilizados os defeitos e os esforços gastos no teste de unidade para cada programa testado, como mostrado na tabela 34.



Tabela 34 - Estudo de Caso - Empresa B - Teste de Unidade (programas)

Unidade	Programas	Defeito	Esforço
U1	P1	4	1:22
	P2	5	1:30
	P3	8	1:12
U2	P4	5	1:12
	P5	9	2:22
U3	P6	2	0:19
	P7	2	1:30
U4	P8	1	0:05
	P9	2	1:30
	P10	4	2:10
U5	P11	3	0:35
	P12	10	2:15

Como mostrado na tabela 35, os dados sobre defeitos e esforço gasto em cada programa foram totalizados e os resultados agrupados nas unidades. Com estes dados foram obtidas as medidas básicas de total de defeitos e esforço para cada unidade avaliada no teste de unidade. Com a combinação das medidas básicas obteve-se a medida derivada 'Quantidade média de defeitos detectados por hora no teste de unidade.

Tabela 35 - Estudo de Caso - Empresa B - Teste de Unidade – Medida Derivada

Unidade	Defeitos detectados no teste de unidade	Esforço gasto no teste de unidade	Defeitos/Hora
U1	17	4:04	4.20
U2	14	3:34	4.19
U3	4	1:49	2.68
U4	9	3:45	2.61
U5	13	2:50	5.20
<b>Média de Defeitos por hora</b>	<b>57</b>	<b>15:22</b>	<b>3.74</b>

A figura 43 apresenta o gráfico com a medida derivada “Quantidade média de defeitos detectados por hora no teste de unidade”. Para cada unidade testada foram contabilizados os defeitos e as horas gastas no teste de software, obtendo-se o percentual de defeitos por hora para as unidades U1, U2, U3, U4 e U5.

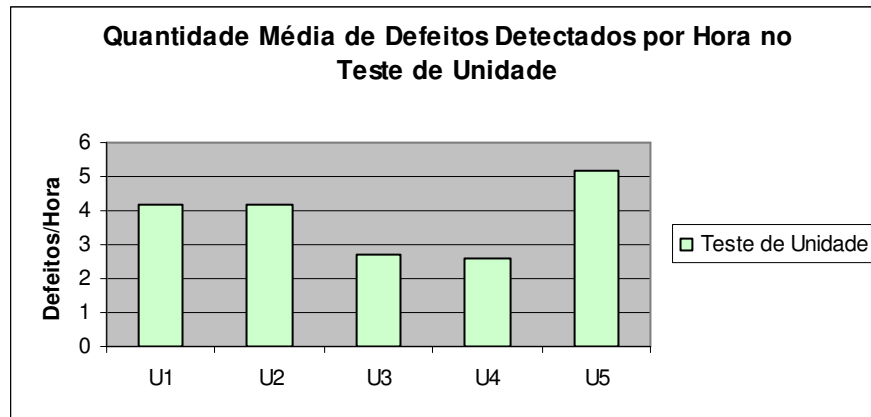


Figura 43 - Estudo de Caso - Empresa B – Gráfico de Defeitos por Teste de Unidade

Os dados obtidos no teste de integração também foram contabilizados de acordo com o plano de medição, obtendo-se as medidas básicas de defeitos e esforço do teste de integração e a medida derivada “Quantidade média de defeitos detectados por hora no teste de integração”, como mostra a tabela 36.

Tabela 36 - Estudo de Caso - Empresa B - Teste de Integração – Medida Derivada

Módulo	Defeitos detectados no teste de integração	Esforço gasto no teste de integração.	Defeitos/ Hora
M1	5	3:31	1.5
<b>Média de Defeitos por hora</b>	5	3:31	<b>1.5</b>

Na figura 44 é mostrado o gráfico referente à medida derivada “Quantidade média de defeitos detectados por hora no teste de integração” gerada com base nas medidas básicas de total de defeitos e esforço do teste de integração. Neste caso, o teste de integração foi efetuado no sistema como um todo considerando um único módulo de teste denominado M1.

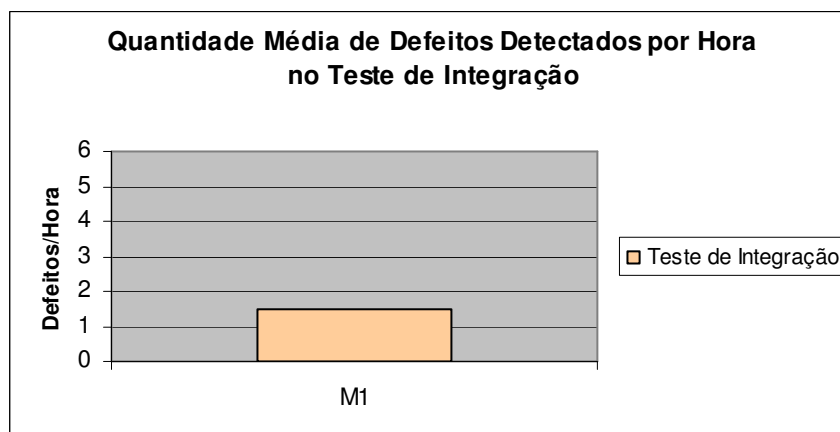


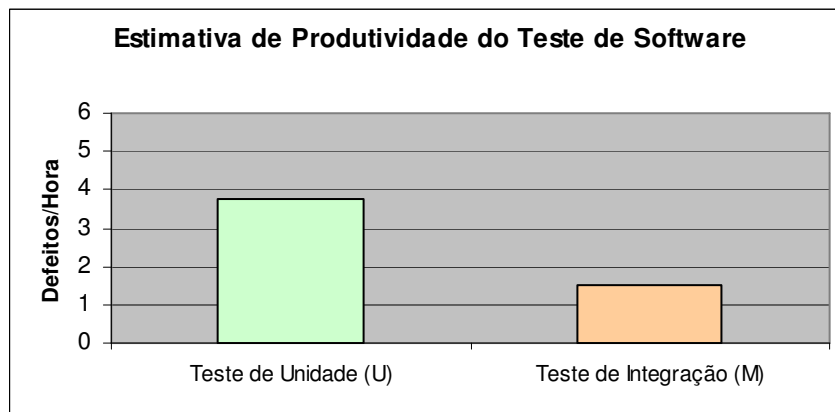
Figura 44 - Estudo de Caso - Empresa B - Gráfico de Defeitos por Teste de Integração

Com base nas medidas derivadas obtidas foi gerado o indicador “Estimativa de produtividade na detecção de defeitos no teste de software”, como mostra a tabela 37.

**Tabela 37 - Estudo de Caso - Empresa B – Estimativa de Produtividade do Teste**

Teste	Defeitos detectados no teste	Esforço gasto no teste	Defeitos/ Hora
Unidade (U)	57	15:22	3.74
Integração (M)	5	3:31	1.5
<b>Média de Defeitos por hora</b>	<b>62</b>	<b>19:33</b>	<b>3.21</b>

Na figura 45 é mostrado o gráfico de “Estimativa de Produtividade do Teste de Software” gerado com base nos percentuais de defeitos por hora dos testes de unidade e de integração, contidos na tabela 37. De acordo com o critério de decisão, definido para o indicador no Plano de Medição, quanto maior o percentual de defeitos por hora melhor é a produtividade do teste de software.



**Figura 45 - Estudo de Caso - Empresa B – Gráfico de Estimativa de Produtividade do Teste de Software**

#### 5.2.4. Análise da Aplicação do Processo de Medição

Com o estudo de caso pôde-se observar a aplicação do Processo de Medição, abordando as fases de Planejamento e Execução da Medição.

Na aplicação do processo, inicialmente, foi efetuado um estudo sobre a empresa fornecendo uma visão geral sobre seu método de trabalho e sua atuação no mercado.

Com uma visão geral sobre a empresa e sobre a estrutura do processo de teste foi verificado que o teste de software era, efetivamente, aplicado e os dados eram registrados de forma acessível. Foi escolhido um projeto para ser medido

dando início à fase de Planejamento da Medição. De acordo com as necessidades de informação identificadas foi elaborado o Plano de Medição descrevendo-se as metas, objetivos, entidades até obter as medidas e indicadores.

Como a medição era focada em projeto houve a necessidade de montar uma estrutura que separasse os programas testados agrupando em unidades, para os testes de unidade, e em módulos, para o teste de integração, facilitando o processamento dos dados na medição.

A fase de Execução da Medição foi efetuada com base no plano de medição e na estrutura de agrupamento do projeto, executando as atividades de coleta, processamento e análise dos dados. Como resultado foram gerados gráficos e tabelas que serão utilizados pelo tomador de decisão para direcionar suas ações.

Neste estudo todas as fases e atividades contidas no processo proposto foram executadas sem problemas. No entanto, observa-se que é fundamental que o processo de teste utilizado pela empresa seja bem estruturado e aplicado corretamente nos projetos de forma que os resultados sejam efetivamente registrados. Outra observação é que em todas as fases do Processo de Medição é fundamental a participação dos membros da empresa, auxiliando no planejamento da medição, relatando as necessidades existentes e os objetivos a serem alcançados e, na fase de Execução da Medição, auxiliando na coleta e processamento dos dados resultantes dos testes.

### **5.3. Conclusões Relativas aos Estudos de Caso**

Os estudos de caso apresentados tiveram como objetivo colocar em prática o Processo de Medição proposto utilizando duas empresas de desenvolvimento de software com segmentos de atuação, processos de teste e necessidades totalmente diferenciados.

Desta forma, foi observado que independente das características particulares de cada empresa o Processo de Medição pôde ser aplicado percorrendo as fases e atividades propostas. O Plano de Medição mostrou-se fundamental para guiar todo o planejamento da medição sendo preenchido, total ou parcialmente, de acordo com as necessidades de informação.

Foram identificados como pontos críticos para a implantação do Processo de Medição a existência de um processo de teste que efetivamente esteja sendo

executado e que os defeitos resultantes sejam armazenados de forma acessível. Outro fator crítico é o comprometimento da empresa com a medição considerando que no Planejamento da Medição o tomador de decisão é papel chave para a identificação das necessidades de informação ajudando na elaboração e na validação do plano de medição. Já na fase de Execução da Medição é fundamental que membros da equipe ligados ao processo de teste, que tenham conhecimento sobre o armazenamento das informações, auxiliem os analistas de medição na coleta e processamento dos dados.

Foi verificado que os Planos de Teste utilizados pelas empresas, analisados nos estudos, não seguem o padrão determinado pela Norma 829, não apresentando a descrição do processo de teste. Portanto, foram utilizados os documentos de descrição do processo de teste próprios de cada empresa sem representar prejuízos à execução do processo de medição.

Durante a aplicação dos estudos de caso oportunidades de melhoria do processo de teste foram percebidas já na fase de Coleta e Processamento dos Dados, como a inclusão de novas informações sobre o teste no banco de dados e alterações na forma de registro dos dados, visando facilitar a recuperação das informações sobre as estratégias de teste aplicadas. Outras melhorias para o processo de teste poderão ser identificadas considerando que os resultados obtidos com as medições serão a base para os tomadores de decisão conhecerem e avaliarem o processo de teste, identificando pontos problemáticos e direcionando suas ações para melhoria do processo.

Cabe observar que, no caso da empresa A, foi possível acompanhar a análise dos resultados da medição, efetuada pelo gerente da fábrica de programas, evidenciando como os resultados da medição contribuem para que o tomador de decisão identifique os problemas existentes e efetue melhorias no processo de teste. Diante dos resultados obtidos com a medição, o gerente da fábrica de programas pode identificar problemas no processo de teste, como o excesso de horas despendido no teste do programador e a inclusão incorreta dos resultados dos testes, como um meio de encobrir os erros dos programadores no desenvolvimento e manutenção dos programas. Os resultados da medição, também, servirão como uma base quantitativa para o gerente da fábrica de programas tomar ações de melhoria do processo de teste.

## **CAPÍTULO 6 – CONCLUSÕES**

Com a qualidade de software sendo considerada como fator determinante para a competitividade entre as organizações, o teste e a medição são considerados fatores fundamentais para assegurar a qualidade dos processos e produtos. O teste de software por ser um mecanismo que auxilia na detecção de defeitos no produto, a análise quantitativa dos defeitos encontrados nos testes, por meio da medição, são cruciais para o controle e a avaliação da qualidade dos processos, projetos e produtos da organização. Nesse ponto, o teste e a medição são complementares, o teste fornece os dados sobre o produto que analisados com base nas métricas definidas pelo processo de medição, permitem que a organização faça ajustes no seu processo de desenvolvimento de software para remover as causas de defeitos que prejudicam a qualidade do produto e do processo.

Como a definição do Processo de Medição não é uma tarefa fácil, há várias propostas de metodologias e modelos de medição. No entanto, quando o objetivo é tratar o teste de software e o processo de medição conjuntamente surgem diversos questionamentos no que se refere ao teste e à medição de forma relevante, estruturada e com boa qualidade. Diante disso, este trabalho propôs um processo de medição que une os conceitos de teste de software e processo de medição.

Este capítulo apresenta a conclusão do trabalho desenvolvido, suas contribuições e sugestões para trabalhos futuros.

### **6.1. Contribuições do Trabalho**

Neste trabalho foi proposto um Processo de Medição que une os conceitos de teste de software e de processo de medição, utilizando na medição as informações obtidas com base em um processo de teste. O foco do processo de medição é o processo de teste, desta forma, as estratégias de teste adotadas servem como referência para o planejamento da medição, e os defeitos resultantes dos testes, como fonte de informação sobre defeitos e insumo para a execução da medição.

O Processo de Medição abrange as fases de Planejamento e Execução da Medição, ressaltando a importância de, primeiramente, definir as metas e objetivos a serem alcançados com a medição, alinhando as necessidades de informação da organização com as medidas e indicadores a serem obtidos. A fase de execução da medição somente é aplicada após a conclusão da fase de planejamento da medição.

Além de mostrar a aplicação conjunta dos conceitos sobre teste de software e processo de medição é importante destacar as seguintes contribuições deste trabalho:

- Apresenta a utilização do PSM como base para a elaboração de um processo de medição. A estruturação do processo de medição proposto é baseada nas recomendações do PSM, no entanto, como esta técnica é abrangente foram efetuadas algumas adaptações para restringir o escopo do processo às fases de planejamento e execução da medição e agregar algumas características do GQM, no que tange a definição das metas de medição, com a finalidade de facilitar o preenchimento do plano de medição;

- Fornece o Modelo de Plano de Medição, como uma forma de apoiar os profissionais na medição de defeitos do software. O Modelo de Plano de Medição serve como roteiro para o planejamento da medição, contendo um item a ser preenchido correspondente a cada atividade da fase de Planejamento da Medição. Além de servir como um roteiro para o planejamento, o Plano de Medição permite o registro histórico da medição e, também, serve como referência para a fase de Execução da Medição, direcionando a coleta e a análise dos dados obtidos com os testes;

- Mostra a aplicação prática do processo de medição em ambientes de desenvolvimento de software. Para avaliar a aplicabilidade do processo de medição foram efetuados estudos de caso em duas empresas com segmentos de atuação, processos de teste e necessidades de informação diferentes. Nestes estudos pôde-se observar que independente das características particulares de cada empresa, dos processos de teste e das necessidades de informação diferenciados, o processo de medição pôde ser aplicado percorrendo todas as atividades definidas em cada fase do processo de medição.

O uso efetivo do processo de medição em cenários diferentes pode permitir a identificação de melhorias até mesmo para o próprio processo de medição. Desta forma, é importante que trabalhos futuros explorem mais amplamente o processo de medição proposto, como sugestões descritas a seguir.

## 6.2. Trabalhos Futuros

O processo proposto abrange somente as fases de Planejamento e Execução da Medição, focando no processo de teste de software. Portanto, trabalhos futuros podem ser desenvolvidos estudando a inclusão de novas fases para o processo de medição como as propostas pelo PSM, que são: Estabelecer e sustentar o comprometimento da empresa e Avaliar o processo de medição. Essas duas fases do PSM se tornam importantes por garantirem a manutenção e a avaliação do processo de medição, considerando que as fases de Planejamento e Execução da Medição já foram implantadas com êxito.

No presente trabalho o Processo de Medição proposto foi exercitado em duas empresas, com processos de teste e necessidades de informação diferentes. No entanto, seria importante a implantação deste processo em outras empresas, por pessoas diferentes, com necessidades e ambientes diferentes, de forma a avaliar a facilidade de utilização e compreensão do processo proposto. Outra oportunidade de trabalho futuro seria adequar este Processo de Medição para a avaliação de outros processos que não o teste de software.

Com base no Processo de Medição proposto, também, pode-se estudar a viabilidade do desenvolvimento de uma ferramenta para a geração do plano de medição e o registro dos resultados da medição de forma automatizada ou avaliar a customização do PSM *Insight* do Departamento de Defesa Norte-Americano, ferramenta que acompanha o PSM e auxilia na implantação do processo de medição, de forma a adequá-la ao processo de medição de defeitos.



## REFERÊNCIAS BIBLIOGRÁFICAS

- ADRION, W. R.; BRANSTED, M. A.; CHERNIAVSKY, J. C. Validation, Verification and Testing of Computer Software. **ACM Computing Surveys**, v. 14, n. 2, p. 159-192, jun. 1982.
- ALMENDRA, V. DA S. **Modelo para Avaliação de Métricas de Software**. 2003. 99f. Dissertação (Mestrado em Engenharia) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2003.
- BASILI, V.; CALDIERA, G.; ROMBACH, H. **Goal Question Metric paradigm**. In: Encyclopédia of Software Engineering. V. 2, 1994. p. 527 –532.
- BORGES, E. P. **Um Modelo de Medição para Processos de Desenvolvimento de Software**. 2003. 154f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, Belo Horizonte, 2003.
- CARVALHO, E. L. **Roteiro de Avaliação de Ferramenta de Automação de Teste**. 2004.152f. Dissertação (Mestrado em Engenharia da Computação) – Instituto de Pesquisas do Estado de São Paulo – IPT. São Paulo, 2004.
- DEPARTMENT OF DEFENSE AND US ARMY. **Practical Software and Systems Measurement A Foundation for Objective Project Management**. v. 4.0c, march 2003. Disponível em:< <http://www.psmc.com> > Acesso em: 08/05/2006.
- GELPERIN, D.; HETZEL, B. The Growth of Software Testing. **Communications of the ACM**, v.31, n. 6, p. 687-695, jun. 1988.
- GLADCHEFF, A.P.; SANCHES, R; SILVA, D.M. **Um instrumento de avaliação de qualidade de software educacional: como elaborá-lo**.Workshop de Qualidade de Software. Rio de Janeiro, 2001.
- HETZEL, W. **The Complete Guide to Software Testing**. Wellesley, Massachusetts, QED Information Sciences, 1984.280p.
- HUMPHREY, W. S. **A Discipline for Software Engineering**. Addison-Wesley, Reading, MA, 1995.790p.
- INSTITUTE OF ELETRICAL AND ELETRONIC ENGINEERS. **IEEE Standard For Software Engineering Terminology 610.12**, 1990.
- INSTITUTE OF ELETRICAL AND ELETRONIC ENGINEERS, **IEEE Standard For Software Test Documentation 829**, 1983.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO/IEC 15939; **Software Engineering – Software Measurement Process**, 2001.
- MCGARRY, J. **Practical Software Measurement A Foundation For Objective Project Measurement**. Addison Wesley, 2001 - 304p.

MYERS, G. J. **The Art of Software Testing**. New York, John Wiley & Sons, 1979. 256p.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **Integration Definition For Function Modeling (IDEF0)**. Draft Federal Information Processing Standards Publication 183. NIST,1993.

PARK , R. E.; GOETHERT, W.B.; FLORAC, W. A. **Goal-Driven Software Measurement — A Guidebook**. Software Engineering Institute, Carnegie Mellon University.1996. 171p.

PRESSMAN, R. S. **Engenharia de software**; tradução Rosângela Deloso Penteadó; revisão técnica Fernão Stella R. Germano, José Carlos Maldonato, Paulo Cesar Masiero;6. ed. - São Paulo,2006.720p.

PRESSMAN, R. S. **Engenharia de software**; tradução Mônica Maria G. Travieso; revisão técnica Paulo Cesar Masiero, José Carlos Maldonado, Fernão Stella R. Germano;5. ed.- Rio de Janeiro, 2002.843p.

ROTTA, S. R. **Um Método de Teste de Software Baseado em Casos de Teste**. 2001.161f. Dissertação (Mestrado em Engenharia) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2001.

SCHNEIDEWIND, N. F. Methodology for Validating Software Metrics. **IEEE Transactions on Software Engineering Computer**, vol. 18, maio 1992, p. 410-420.

SILVA, O. J. DA; CRESPO, A. N.; BORGES, C. A.; SALVIANO, C. F.; TEIVE, M. DE; JUNIOR, A.; JINO, MÁRIO. **Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo**. Simpósio Brasileiro de Qualidade de Software. São Paulo, 2004.16p.

WHITE, L. J.; COHEN, E. I. A Domain Strategy for Computer Program Testing. **IEEE Transactions on Software Engineering**, v.6, n.3, p. 247-257, maio 1980.

## GLOSSÁRIO

<b>Plano de Teste</b>	prescreve o escopo, abordagem, recursos e cronograma das atividades de teste e identifica os itens a serem testados, as características que serão testadas, as atividades de teste que serão executadas, os responsáveis por cada atividade e os riscos associados ao plano.
<b>Especificação do Produto</b>	descreve as características do produto como o escopo e as funcionalidades.
<b>Características do projeto</b>	consistem nas características específicas do projeto como recursos, tempo, custo, entre outros.
<b>Resultado do Teste</b>	corresponde aos resultados obtidos no processo de teste. São descritos em relatórios como gráficos, entre outros.
<b>Gerente de Teste</b>	é o responsável pelo planejamento do teste.
<b>Gerente de Medição</b>	responsável pelo planejamento da medição e pela produção dos resultados da execução da medição.
<b>Necessidade Organizacional</b>	corresponde à necessidade de informação que a organização possui para a tomada de decisão.
<b>Modelo de Plano de Medição</b>	é um formulário que contém itens correspondentes a cada atividade da fase de planejamento da medição. É um guia para o planejamento da medição.
<b>Casos de Teste</b>	é um conjunto de dados de entrada, para um software ou componente, e o seu respectivo resultado esperado com a finalidade de exercitar um componente ou um software para verificar a conformidade com os requisitos especificados.
<b>Procedimentos de Teste</b>	identifica todos os passos requeridos para operar o sistema e exercitar um caso de teste especificado.
<b>Analista de Teste</b>	responsável pelo projeto dos testes, execução dos testes definidos e registro dos resultados obtidos.
<b>Questões</b>	são perguntas formuladas com base nos objetivos de medição que ao serem respondidas devem permitir atingir os objetivos.
<b>Medidas</b>	uma atribuição objetiva empírica de um número (ou símbolo) a uma entidade que caracteriza um atributo particular da entidade.
<b>Especificação da Medição</b>	descrição das medidas que serão utilizadas na medição.
<b>Objetivo da Medição</b>	define o que é necessário conhecer do processo ou produto para atingir a meta desejada.
<b>Defeitos</b>	uma etapa, processo ou definição de dados incorreta, por exemplo uma instrução incorreta no código do software.
<b>Indicadores</b>	corresponde ao que é apresentado ao usuário para que este possa realizar a análise da medição e tomar decisões sobre o processo, projeto e produto.

**Plano de Medição**

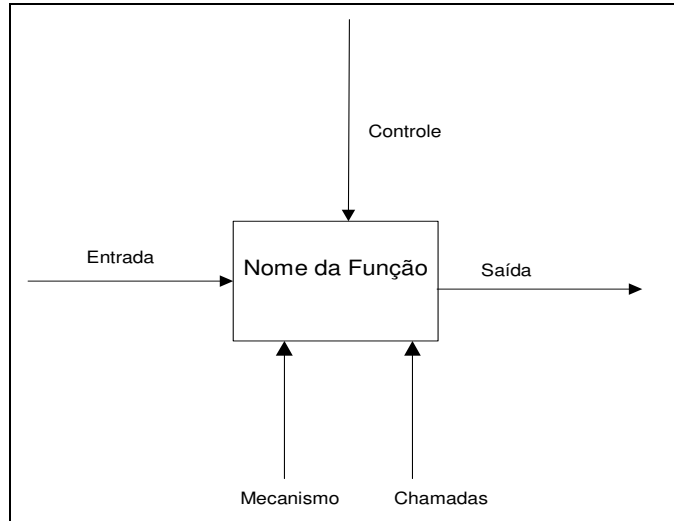
documento que contém o planejamento da medição, descrevendo as metas, objetivos e medidas utilizadas no processo de medição. Serve como base para a execução da medição.

## APÊNDICE A - IDEF0 (Integration Definition for Function Modeling)

Esta linguagem foi desenvolvida a partir de uma necessidade da força aérea americana, que trabalhava com diversas indústrias aeroespaciais. Como cada uma destas indústrias trabalhava a sua maneira, era difícil controlar e documentar todos esses processos. Estudos foram feitos para que se tivesse uma linguagem que compreendesse vários critérios e atendesse todas estas indústrias. Em 1972, foi desenvolvido o SADT (*Structured Analysis and Design Technique*) que foi escolhido para uso no projeto da Força Aérea Americana. Em seguida foram desenvolvidos o ICAM 1 e o ICAM 2. O ICAM 2 foi documentado e renomeado para IDEF0 (NIST, 1993). IDEF (*Integration Definition for Function Modeling*) é um conjunto de métodos para modelagem baseada em representações de diagramas, incluindo uma larga variedade de técnicas.

O IDEF0, que é o primeiro conjunto de padrões do IDEF, processa uma coleção de atividades e outras ações utilizando-se de ICOMs (*Input Control Output Mechanism*). O ICOM não inclui apenas dados e informações, mas também tudo que pode ser descrito como sendo um processo (esquema, estimativa, regulamentos, produtos, entre outros). Como mostrado na figura A.1, o ICOM é uma representação gráfica de uma tarefa ou um conjunto de tarefas, que possui “terminais” para que possa ser alimentada ou alimentar outras ICOMs. Esses “terminais” recebem o nome de entrada, controle, saída e mecanismo. A entrada recebe o dado a ser convertido pela atividade, o controle agrega responsabilidade de como e quando a entrada deve ser processada e executada, a saída apresenta o resultado de como a entrada foi processada e o mecanismo representa quem deve executar esta atividade (pode ser uma pessoa, equipamento, máquina ou outras organizações).

O modelo funcional IDEF0 é composto por um conjunto de ICOMs, setas e caixas. Cada atividade ou função é conceitualmente representada por uma caixa retangular. Cada atividade pode ser decomposta em vários níveis. Estes subníveis seguem as mesmas convenções. Portanto um modelo completo de IDEF0 é uma representação hierárquica do processo composta por: atividades ou funções em quantos níveis forem necessários.



**Figura 46 - Exemplo de ICOM**

**Fonte: NIST (1993)**