BASE DE MÉTRICAS PARA A ESTAÇÃO TABA

Mario Henrique da Rocha Estolano

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

Aprovada por:

Profa. Ana Regina Cavalcanti da Rocha, D. Sc.

Prof. Geraldo Bonorino Xexeo, D. Sc.

Profa. Ana Cristina Rouiller, D. Sc.

RIO DE JANEIRO, RJ - BRASIL

AGOSTO DE 2005

ESTOLANO, MARIO HENRIQUE DA ROCHA

Base de Métricas para a Estação TABA [Rio de Janeiro] 2005

VIII, 125 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2005)

Dissertação – Universidade Federal do Rio de Janeiro, COPPE

1. Métricas

2. Medição de Software

3. Ambientes de Desenvolvimento de Software Orientados à Organização

I. COPPE/UFRJ II. Título (série)

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

BASE DE MÉTRICAS PARA A ESTAÇÃO TABA

Mario Henrique da Rocha Estolano

Agosto/2005

Orientadora: Ana Regina Cavalcanti da Rocha

Programa: Engenharia de Sistemas e Computação

Medir software é essencial para se desenvolver uma disciplina madura de Engenharia de Software, sendo utilizado para coletar dados quantitativos e qualitativos necessários para compreender, gerenciar e melhorar os processos e produtos de software. Também figura como disciplina chave na avaliação da maturidade e capacidade dos processos de software, tornando-se cada vez mais importante em acordos comerciais, onde fornece uma base para gerenciamento de projetos e aceitação de produtos.

A Estação TABA é um Ambiente de Desenvolvimento de Software Orientado à Organização (ADSOrg) e como tal deve possuir um processo e infra-estrutura voltados para medição. A abordagem para Medição e Análise da Estação TABA foi elaborada a partir do método *Goal-Question-Metrics* (GQM), dos requisitos da área de processo Medição e Análise do CMMI (*Capability Maturity Model Integration*) e do processo de medição do MR MPS.BR (Modelo de Referência de Melhoria de Processo de Software Brasileiro).

Este trabalho define a Base de Métricas para a Estação TABA, elemento fundamental ao processo de medição e análise, e que permite o armazenamento de métricas bem especificadas, juntamente com questões e objetivos, para utilização na definição de planos de medição para as organizações e seus projetos.

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

A METRICS REPOSITORY FOR THE TABA WORKSTATION

Mario Henrique da Rocha Estolano

August/2005

Advisor: Ana Regina Cavalcanti da Rocha

Department: Systems and Computing Engineering

Software measurement is fundamental for the development of a mature Software Engineering discipline. It can be used for collecting quantitative and qualitative data used for understanding, managing and improving software processes and products, featuring also as a key discipline for the evaluation of software processes capability and maturity. Besides, it has increasingly importance in commercial arrangements, supporting project management and acceptance criteria for products.

TABA Workstation is an Enterprise-Oriented Software Development Environment, having therefore a process and foundation for measurement. The measurement and analysis approach for TABA Workstation was developed based on the Goal-Question-Metrics (GQM) method, CMMI (Capability Maturity Model Integration) Measurement and Analysis process area requirements and on the MR MPS.BR (Reference Model for Brazilian Software Process Improvement) measurement process.

This work defines the Metrics Repository for the TABA Workstation, a fundamental building block for its measurement and analysis process, which allows storing well-defined metrics together with the questions and goals to be used for the definition of measurement plans for organizations and its projects.

A minha família e meus amigos.

Agradecimentos

À minha orientadora, Profa. Ana Regina, por acreditar e confiar em mim, e por fazê-lo com tanto carinho e incentivo.

À minha esposa, Cirineia, por ter paciência e carinho com minhas crises produtivas e improdutivas, oferecer o ombro sempre que preciso e por vibrar comigo a cada etapa concluída.

À minha mãe, por tudo que sou.

Ao amigo Edvaldo, por sua amizade e compreensão, pelo imenso incentivo e apoio incondicional.

À amiga Maria Luiza, pela sabedoria e apoio nos momentos críticos.

Ao meu irmão, Antonio Carlos, pelo apoio e pela guarida na reta final.

Aos professores Guilherme Travassos e Cláudia Werner, por todo apoio.

A Gleison, Mariano, Sávio e Sômulo, pelo auxílio com seus conhecimentos e por sua boa vontade em colaborar sempre.

Aos professores membros da banca, por prestigiarem meu trabalho com sua honrosa contribuição e presença.

A todos da LIPE Pesquisas, por sua garra, dedicação e superação, e pela torcida.

À minha família e aos meus amigos, um grande obrigado pelo incentivo e pela compreensão de minhas ausências.

A Deus, pelos desafios que me engrandecem.

Conteúdo

Capítulo 1

Introdução

Neste capítulo são apresentados as motivações que levaram à realização deste trabalho, seu principal objetivo, o contexto no qual foi desenvolvido, a metodologia de trabalho e a forma como esta dissertação está organizada.

1.1. Motivações

1.1.1. Medição de Software

As atividades de gerência de projetos e melhoria de processos requerem informações baseadas em dados quantitativos e qualitativos para a tomada de decisões. Nesse contexto, medição de software é uma infra-estrutura essencial para se desenvolver uma disciplina madura de Engenharia de Software (ROMBACH, 1991), sendo utilizada para coletar dados quantitativos e qualitativos necessários para compreender, gerenciar e melhorar os processos e produtos de software.

Dentre as principais motivações para se medir software, estão (BASILI, 1999) (BRIAND *et al.*, 1996):

- Compreender e caracterizar processos e produtos de software, para definir o estado ou desempenho atual do processo ou produto de software;
- Monitorar tendências e a evolução do desempenho / estado dos produtos e processos de software através dos diversos projetos de uma organização;
- Avaliar o alcance de objetivos de qualidade, a avaliação do impacto de tecnologias em produtos e processos e a compreensão dos aspectos onde as tecnologias precisam ser melhoradas ou adaptadas a um contexto específico;
- Predizer atributos externos relevantes a partir da identificação de relacionamentos entre características dos processos, produtos e recursos;
- Controlar a execução projetos de software, minimizando seus riscos, a partir da identificação de relações causais entre características de processos, produtos e recursos;
- Melhorar a qualidade e a produtividade dos projetos futuros, com base na identificação de relacionamentos causais que influenciem o estado ou desempenho dos processos e/ou produtos.

Para alcançar esses objetivos, aspectos relevantes dos processos, produtos e recursos devem ser observados e medidos, tornando evidente que a medição é uma infra-estrutura essencial para o sucesso no gerenciamento dos projetos de software e na melhoria de processos.

A medição de software também figura como disciplina chave na avaliação da capacidade dos processos de software, tornando-se cada vez mais importante em acordos comerciais, onde fornece uma base para especificação, gerenciamento e critérios de aceitação (ISO/IEC 15939, 2002).

De um modo geral, medir tem como objetivo capturar características de entidades e de seus relacionamentos, e então manipulá-los de maneira formal (KITCHENHAM, PFLEEGER e FENTON, 1995). Medição é o processo através do qual valores (símbolos ou números) são atribuídos a atributos de entidades do mundo real, de modo a descrevê-los de acordo com regras claramente definidas (ROBERTS, 1979). Assim, a medição permite quantificar atributos de entidades de modo a manipulá-los e aprender mais a seu respeito.

1.1.2. A Estação TABA

A Estação TABA (ROCHA *et al.*, 1990) é um Ambiente de Desenvolvimento de Software Orientado à Organização (ADSOrg) (VILLELA, 2004) e como tal necessitava de um processo e de infra-estrutura voltados para medição.

A abordagem para medição e análise pretendida para a Estação TABA (SCHNAIDER *et al.*, 2004) deveria ser baseada no método *Goal-Question-Metric* (GQM) (BASILI *et al.*, 1994) (BASILI e ROMBACH, 1988) (BASILI e WEISS, 1984) e nos requisitos da área de processo Medição e Análise do CMMI (*Capability Maturity Model Integration*) (CMU/SEI, 2002). A proposta dessa abordagem seria apoiar a medição e análise disponibilizando o conhecimento sobre medições que pudesse ser útil às organizações e aos gerentes de projeto durante a execução de atividades relacionadas a medição (SCHNAIDER *et al.*, 2004).

1.1.3. Bases de Métricas

Dada a heterogeneidade das fontes de informação relativas a métricas em uma organização, é importante prover mecanismos para estruturação, classificação, recuperação e uso das informações. Repositórios de métricas e ambientes de catalogação podem fornecer para os diferentes envolvidos (*stakeholders*) meios para

consulta, busca e recuperação e mecanismos de reuso, tendo como base a especificação certa do tipo de entidade a ser medida (uma atividade ou um artefato de software, por exemplo), a definição e motivação dos atributos, definição da métrica, do tipo de medição, fórmula para cálculo da métrica, unidade, tipo de escala, instrumentos para coleta de dados, regras de contagem (definição operacional), entre outros metadados necessários.

Dessa forma, os ambientes TABA precisavam de uma base de métricas que viabilizasse o processo de medição e, conseqüentemente, as ferramentas dele decorrentes, e que possibilitasse o armazenamento de métricas bem definidas, juntamente com questões e objetivos, para utilização na definição de planos de medição para as organizações e seus projetos.

1.2. Objetivo da Tese

Para tornar possível a implementação dos processos de melhoria, e em particular do processo de medição e análise de resultados na Estação TABA, este trabalho teve como principal objetivo definir sua Base de Métricas – uma estrutura capaz de comportar métricas, questões e objetivos previamente definidos para serem utilizadas durante a construção do plano de medição da organização ou de um projeto – infra-estrutura até então inexistente na Estação TABA.

Para atingir esse objetivo, foi definido e implementado um repositório para um catálogo de métricas, que pudesse ser utilizado para definição de planos de medição nos ambientes TABA, possibilitando o registro de métricas de forma detalhada e de modo que as métricas definidas possam ser aplicadas sem ambigüidade.

Os conceitos fundamentais inerentes à definição e à classificação das métricas também foram utilizados na estrutura definida, contemplando o registro de métricas diretas ou indiretas, atômicas ou calculadas a partir de outras.

1.3. Contexto de Desenvolvimento

A partir de 2003 a nova geração de ambientes TABA passou a ser utilizada na pratica mostrando o potencial da Estação TABA nas empresas (MONTONI, 2005a). Também neste período surgiu a demanda de organizações com o objetivo de serem avaliadas conforme o CMMI e/ou o Modelo de Referência para Melhoria do Processo de Software Brasileiro (MR mpsBR).

No ambiente de constante evolução da Estação TABA e das demandas por suporte mais avançado para os modelos de maturidade, foi percebida a carência de recursos na Estação para o tema "melhoria de processos", que motivou a formação de um grupo de trabalho sobre o tema: dois trabalhos de doutorado, o de BESSA (2005) e o de CAMPOS (2005), e três de mestrado, o de ANDRADE (2005), o presente trabalho, e posteriormente o de CABRAL FILHO (2005), em andamento. Este grupo definiu a estratégia de melhoria em níveis, que apresenta uma visão macro sobre melhoria de processos na Estação TABA, definindo os grandes processos e a interação entre os mesmos.

Foi nesse contexto que teve início o desenvolvimento deste trabalho, dada a necessidade imperativa do uso de métricas nos processos de melhoria e de estruturas para seu arquivamento, catalogação e uso, até então inexistentes na Estação TABA.

1.4. Metodologia de Trabalho

Uma vez definido o escopo da Tese, ao longo do desenvolvimento das cadeiras do curso de mestrado foi feita uma revisão da literatura incluindo artigos, livros, normas e padrões relacionados a métricas e a processos de medição.

Em reuniões com o grupo de melhoria criado, foram estabelecidos os requisitos básicos para a definição e registro de uma métrica, utilizando a literatura revisada, e em particular os trabalhos de KITCHENHAM et al. (1995) e KITCHENHAM et al. (2000). A partir destes requisitos de definição de uma métrica, foi criado um *template* detalhado para registro de métricos, o qual foi submetido a avaliações através de sua aplicação para registro de métricas que iam sendo encontradas na bibliografia em estudo pelo grupo. Na medida em que o *template* se mostrava incompleto para registrar determinada métrica, o mesmo era revisado, e assim sucessivas versões foram criadas.

Uma versão estável e revisada desse *template* foi utilizada na criação do modelo conceitual da Base de Métricas. Essa primeira versão do modelo não contemplava classes e atributos especificamente voltados a mecanismos de recuperação de informação e também ainda não tinha sido desenvolvida a parte do modelo referente a GQM e a planos de medição.

Quando da definição do processo de medição e análise, trabalho desenvolvido paralelamente a este, foram incorporados ao modelo da Base de Métricas as estruturas para suporte a GQM e planos de medição, e também para recuperação de informação, e levados em consideração também detalhes de estrutura propostos pelo ISO 15939.

A implementação da base de métricas na Estação TABA foi executada pelo grupo de desenvolvimento do projeto TABA a partir de um modelo de classes de projeto definida por este trabalho. Ao longo da implementação foram necessários ajustes no modelo, principalmente em função das classes de domínio da Estação TABA. Foi também do escopo deste trabalho o acompanhamento da implementação da base de métricas nele definida.

A conclusão do trabalho deu-se com a redação desta dissertação.

1.5. Organização da Dissertação

Este trabalho contém, além deste capítulo introdutório, mais cinco capítulos, resumidamente descritos a seguir.

O segundo capítulo apresenta os conceitos-chave e abordagens para medição de software, com foco nos conceitos e enfoques mais relevantes a este trabalho.

O terceiro capítulo discorre sobre como a medição de software é tratada nas principais normais internacionais e nos modelos de maturidade nacional (MPS.BR) e internacional (CMMI).

O quarto capítulo descreve as principais características dos Ambientes de Desenvolvimento de Software Orientados à Organização (ADSOrg) e apresenta a Estação TABA e seus ambientes. O capítulo também descreve a abordagem de medição e análise nos ambientes TABA.

No quinto capítulo, é descrita a Base de Métricas – objeto deste trabalho, sua especificação e implementação nos ambientes TABA.

No sexto e último capítulo, são apresentadas as considerações finais deste trabalho, ressaltando suas contribuições e as perspectivas para a continuidade do mesmo.

Capítulo 2

Medição de Software e Métricas

Neste capítulo são apresentados os conceitos-chave e abordagens para medição de software, com foco nos conceitos e enfoques mais relevantes a este trabalho.

2.1. Por que medir?

Para atingir seus objetivos (referentes, por exemplo, a custo e a qualidade), um projeto de software deve ter práticas de gerenciamento e de melhoria baseadas em informação como parte integrante do projeto.

Planejar projetos de software inclui o estabelecimento dos objetivos do projeto em termos quantitativos, a partir dos modelos da organização. A boa execução dos projetos de software baseia-se em capturar dados de medições definidos a priori, de modo a prover *feedback* sobre a aderência do projeto em relação aos objetivos planejados, permitindo, se necessário, o replanejamento do projeto em execução. O aprendizado organizacional e a criação de competências fundamentais no domínio da Engenharia de Software são obtidos "empacotando-se" [1] a experiência obtida em um projeto para uso em projetos futuros (ROMBACH, 1991). As atividades de gerência de projetos e melhoria dos processos requerem informações baseadas em dados para a tomada de decisões. Nesse contexto, medição de software é uma infra-estrutura essencial para se desenvolver uma disciplina madura de Engenharia de Software (ROMBACH, 1991), sendo utilizada para coletar dados quantitativos e qualitativos necessários para compreender, gerenciar e melhorar os processos e produtos de software.

Dentre as principais motivações para se medir software, estão (BASILI, 1999) (BRIAND *et al.*, 1996):

- Compreender e caracterizar processos e produtos de software, com o objetivo de definir o estado ou desempenho atual do processo ou produto de software. A medição permite diferenciar ambientes de projeto, compreender

[1] Adaptar o conteúdo do conhecimento obtido (análise, edição, interpretação, tradução e síntese do conhecimento), e transformar seu formato físico em um que seja adequado para sua transferência (reestruturação do formato de representação do conhecimento) (MONTONI, 2003).

os processos e produtos em utilização nesses ambientes e fornecer *baselines* e modelos descritivos para uso em avaliações posteriores.

- Monitorar, com o objetivo de observar tendências e a evolução do desempenho ou do estado dos produtos e processos de software através dos diversos projetos de uma organização.
- Avaliar, visando à verificação do alcance de objetivos de qualidade, a avaliação do impacto de tecnologias em produtos e processos e a compreensão dos aspectos onde as tecnologias precisam ser melhoradas ou adaptadas a um contexto específico.
- Predizer, visando identificar relacionamentos entre características dos processos, produtos e recursos e utilizar esses relacionamentos para predizer atributos externos relevantes.
- Controlar, com o objetivo de identificar relações causais entre características de processos, produtos e recursos e utilizar esses relacionamentos para influenciar na execução de um projeto de software de modo a minimizar riscos.
- Melhorar, visando à identificação de relacionamentos causais que influenciem o estado ou desempenho dos processos e/ou produtos, indicando modificações no processo de modo a melhorar a qualidade e a produtividade nos projetos futuros.

Para alcançar esses objetivos, aspectos relevantes dos processos, produtos e recursos devem ser observados e medidos. Utilizando os dados coletados, modelos de *baseline* podem ser construídos (indicando, por exemplo, a distribuição típica de esforço durante um processo de desenvolvimento), relacionamentos entre processos, produtos e recursos de software podem ser identificados (por exemplo, entre o tipo de inspeção empregada e sua eficiência na redução de falhas no produto final) e teorias podem ser confirmadas ou refutadas. Essas questões evidenciam que a medição é uma infra-estrutura essencial para o sucesso no gerenciamento de projetos de software e na melhoria de processos.

Nas seções a seguir, são descritos conceitos fundamentais relacionados a medição de software e à abordagem GQM (*Goal Question Metrics)* para o planejamento e execução de programas de medição baseados em objetivos.

2.2. Fundamentos da medição de software

De um modo geral, medir tem como objetivo capturar características de entidades e seus relacionamentos, e então manipulá-los de maneira formal (KITCHENHAM, PFLEEGER e FENTON, 1995). MEDIÇÃO é o processo através do qual valores (símbolos ou números) são atribuídos a atributos de entidades do mundo real, de modo a descrevê-los de acordo com regras claramente definidas (ROBERTS, 1979). Assim, a medição permite quantificar atributos de entidades de modo a manipulá-los e aprender mais a seu respeito (figura 2.1).

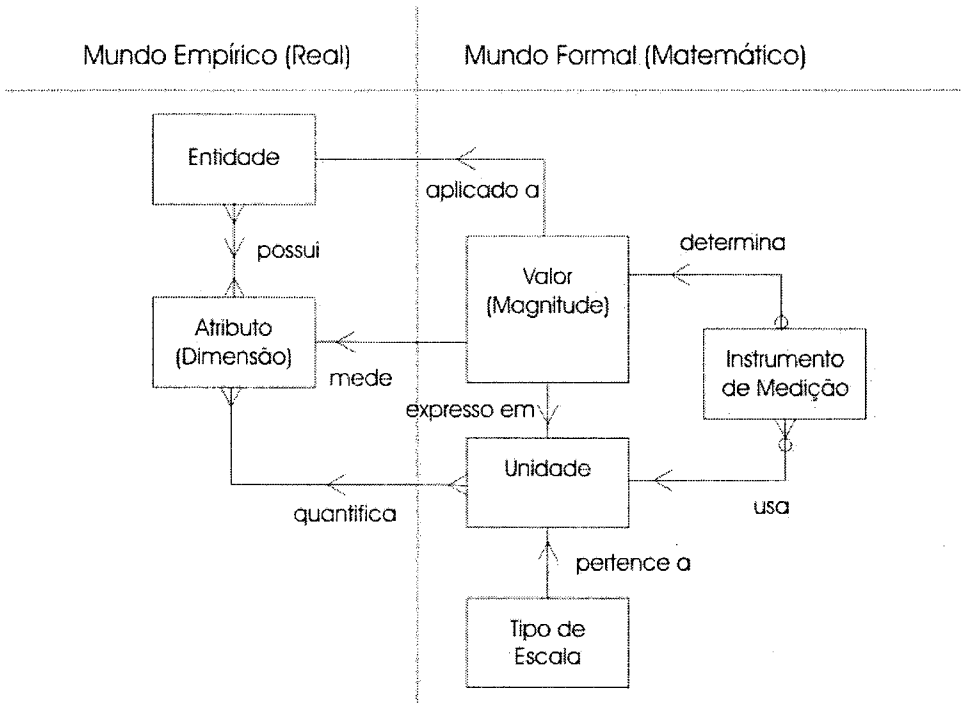

Figura 2.1 – Objetivo da medição (KITCHENHAM., B., PFLEEGER, S. L., FENTON, N., 1995)

Uma ENTIDADE pode ser um objeto ou evento no mundo real. Entidades relevantes no domínio de software incluem:

- Processos: qualquer atividade relativa a desenvolvimento e/ou manutenção, em diferentes níveis de granularidade (ex.: codificação).
- Produtos: qualquer artefato produzido ou modificado durante o desenvolvimento ou a manutenção de software (ex.: código-fonte, documentação de projeto do software), também em diferentes níveis de granularidade.
- Recursos: pessoas, equipamentos, softwares que são utilizados ou consumidos nos processos.

Entidades são descritas através de um conjunto de atributos, os quais são importantes para se distinguir uma entidade de outra. Um ATRIBUTO é uma característica ou propriedade de uma entidade. Atributos típicos de entidades de engenharia de software incluem: tamanho, complexidade, esforço, duração e experiência.

Uma entidade pode possuir diversos atributos (ex.: uma atividade de um processo pode possuir duração e custo). Em contrapartida, um mesmo atributo pode qualificar diferentes entidades (o atributo produtividade, por exemplo, pode-se aplicar a pessoas, equipes ou projetos de software).

Para descrever entidades, seus atribudos são normalmente definidos através de valores numéricos ou simbólicos. Diz-se, por exemplo, que a duração de um projeto de software é de 12 "meses corridos", ou que é "alta" a experiência de um desenvolvedor em uma determinada técnica. Esses números e símbolos são abstrações utilizadas para refletir a percepção do mundo real. Utilizando esses valores numéricos ou simbólicos, pode-se fazer julgamentos acerca de entidades do mundo real através do conhecimento e da análise de seus atributos. Nesse contexto, medição é o processo através do qual valores são associados a atributos de entidades (ROBERTS, 1979).

Para se atingir interpretações válidas dos valores, a medição deve preservar qualquer observação intuitiva ou empírica sobre os atributos e entidades do mundo real. Assim, quando, por exemplo, se mede o tamanho de um módulo de software, números maiores devem ser atribuídos a módulos maiores. Isso significa que valores associados a módulos de software com respeito a seu tamanho (ex.: tamanho do módulo x = 20 KLOC, é maior que o tamanho do módulo y = 10 KLOC) devem corresponder a relações observadas no mundo real (o módulo x é realmente maior que o módulo y). Ou seja, a medição de um dado atributo de uma entidade deve ser precedido de uma compreensão intuitiva desse atributo.

Como a medição é feita a partir da comparação de entidades, essa compreensão intuitiva leva à identificação de relacionamentos empíricos entre as entidades. Por exemplo, considerando o atributo "tamanho" de módulos de software, o número de linhas de código de um módulo não expressa, por si só, se esse módulo é pequeno, médio ou grande. Somente através da comparação com outros módulos o número pode ser interpretado para se chegar a relações empíricas, tais como "maior que", "igual a".

Uma MÉTRICA é um mapeamento de atributos de entidades do mundo real para entidades formais (valores da medição), de modo a descrevê-la de acordo com regras

claramente definidas (BRIAND *et al.*, 1996). Em outras palavras, uma medição é um mapeamento entre o mundo real e o mundo formal, matemático.

O objetivo de uma medição é fornecer uma avaliação do grau ou da forma que um determinado atributo está presente em uma entidade (produto, processo ou recurso, por exemplo). As possíveis manipulações dos números na representação formal do atributo nas entidades podem permitir que se descubra mais acerca da entidade do que através de sua observação direta no mundo real. Por exemplo, quando se examina o tamanho de um módulo de software utilizando linhas de código (LOC), a métrica é o mapeamento que indica como contar as linhas de código de qualquer módulo de software, considerando ou não comentários, contando ou não declarações de variáveis. Como resultado, para cada módulo de software, uma representação formal pode ser criada através de seu número de linhas de código. O número de linhas de código de um módulo pode ser medido para um conjunto de módulos, de maneira que os módulos possam ser comparados. A partir dos dados das medições, pode-se determinar se o módulo é maior que outro ou se o tamanho de um módulo está aumentando, por exemplo.

Para uso nas tomadas de decisão, as mudanças formais são interpretadas com respeito a seu significado no mundo empírico. Se, por exemplo, as medidas indicam que um módulo que sofre constantes modificações está ficando muito grande, o módulo pode ser reprojetado. A figura 2.2 demonstra esse processo de medição e interpretação.

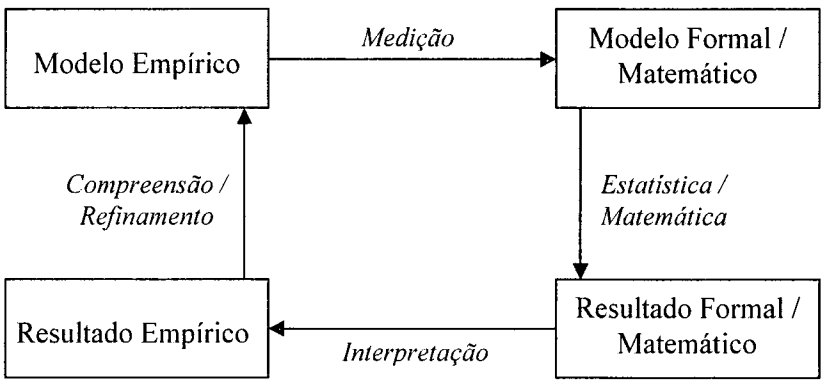

Figura 2.2 – Visão geral do processo de medição e interpretação (PFLEEGER e FENTON, 1997)

Um modelo formal explicita a compreensão intuitiva das entidades, devendo preservar as relações básicas que existem entre as entidades e seus atributos do mundo real. Isso significa que os dados obtidos devem representar as entidades observadas, e

que a manipulação dos dados deve preservar os relacionamentos observados entre as entidades. Supondo-se, por exemplo, a relação empírica "maior que" e a relação formal ">", μ será uma medida de tamanho se e somente se para quaisquer módulos de software m1 e m2 com m1 maior que m2, seja verdade que μ(m1)>μ(m2).

A condição de representação requer que qualquer medida seja associada com um modelo de como a medida mapeia as entidades e seus atributos no mundo real para os elementos de um sistema numérico. Tais modelos são essenciais para se compreender como as medidas são derivadas, para julgar seu uso apropriado e para interpretar os valores no contexto do mundo real.

A medição atribui uma representação ou mapeamento de um sistema de relações empíricas para algum sistema formal de relações, com o propósito de permitir a manipulação dos dados no sistema formal e utilizar os resultados para se chegar a conclusões sobre o atributo no sistema empírico. Mas nem todos os mapeamentos são iguais. Desta forma, as diferenças entre os mapeamentos podem restringir o tipo de análise que se faz com o resultado das medições. Essas diferenças tornam-se explícitas através do conceito de escala.

ESCALA, em uma medição, refere-se à relação entre os valores que são associados aos atributos. O mapeamento da medição, junto com o sistema empírico e o sistema formal, é chamado de escala de medição (BRIAND *et al.*, 1996). O tipo de escala determina as transformações admissíveis que podem ser aplicadas para se o obter outras medições aceitáveis. Os tipos mais comuns de escala são: nominal, ordinal, de intervalo, racional e absoluta (ZUSE, 1998) (BRIAND *et al.*, 1996) (FENTON e PFLEEGER, 1997) (FENTON, 1994) (SCHNEIDEWIND, 2002).

Uma escala nominal é uma simples classificação das entidades de um sistema relacional empírico, sem qualquer noção de ordem entre as classes. Qualquer representação numérica ou simbólica das classes é aceitável, mas não há qualquer noção de magnitude associada com os números ou símbolos. Na escala nominal, o valor do atributo é representado por um nome ou rótulo e, normalmente, é utilizado para a classificação de entidades. Por exemplo, uma possível classificação de linguagens de programação pode ser {C++, Delphi, Java}. A única transformação possível dos valores é aquela que preserva o fato que os objetos são diferentes, ou seja, transformação 1-para-1.

Uma medida com escala ordinal é uma classificação em ordem das entidades do sistema relacional empírico, de acordo com algum critério de ordenação com respeito ao

atributo. Porém, as distâncias entre os itens não têm qualquer significado. Qualquer mapeamento que preserve a ordem (ou seja, qualquer função monotônica) é aceitável. Por exemplo, o nível de experiência pode ser classificado de forma crescente em {nenhum, baixo, médio, alto, perito}.

Em uma medida com escala de intervalo, as diferenças entre os valores têm significado, mas os valores em si não têm, como é o caso, por exemplo, da medição de temperatura nas escalas Celsius e Fahrenheit. Essa escala captura informação sobre o tamanho dos intervalos que separam as classes, preservando ordem e diferença.

Em uma medida com escala racional, existe um valor zero com significado (ele representa a ausência total do atributo medido, funcionando como ponto inicial da escala) e as razões entre os valores também têm significado (por exemplo, esforço em homem-hora). Nesse tipo de mapeamento, são preservadas a ordem, o tamanho dos intervalos, e razão entre as entidades e todas operações aritméticas podem ser significativamente aplicadas.

Em uma medida com escala absoluta, existe apenas um mapeamento possível – a contagem – e qualquer análise aritmética dessa contagem é significativa. Por exemplo, a quantidade de defeitos observados em uma etapa do desenvolvimento de software pode ser medida contando os elementos do conjunto, ou seja, o número de defeitos encontrados.

A manipulação e a análise dos dados estão intimamente ligadas à escala utilizada. Somente com o entendimento das características e das restrições dos tipos de escala é possível determinar se alguma conclusão obtida a partir de uma análise ou manipulação de dados numéricos tem significado válido no mundo real.

O objetivo do mapeamento para o sistema relacional numérico é permitir a manipulação dos dados no sistema relacional formal e utilizar os resultados para obter conclusões sobre o atributo no sistema relacional empírico. Isso significa que os valores da medição não podem ser utilizados indiscriminadamente, pois a aplicação de operações matemáticas em qualquer tipo de valores obtidos através da medição pode levar a resultados que não fazem sentido.

A abordagem transformacional tem sido utilizada para classificar medições de acordo com suas escala de medição, de modo a entender que tipo de operações matemáticas podem ser aplicadas nos resultados de medição obtidos.

Transformações admissíveis

O tipo de escala afeta os tipos de operações e de análises estatísticas que podem ser aplicadas aos dados resultantes da medição. Para calcular, por exemplo, o valor de tendência central em medições feitas com uma escala nominal, é necessário utilizar moda; outras formas de se computar um valor central, como a média ou mediana não são aplicáveis e não fazem sentido nessa escala.

Escalas também são definidas através das transformações admissíveis. Para escalas reais, existe uma classificação de acordo com suas transformações admissíveis (ZUSE, 1998) (BRIAND *et al.*, 1996), conforme a tabela 2.1 .

Tabela 2.1 – escalas de medição e transformações (BRIAND *et al.*, 1996)

Tipo de escala	Descrição	Exemplos	Transformações admissíveis
Nominal	Classificação de objetos	Rotular, classificar	Mapeamentos do tipo 1-para-1
Ordinal	Classificação ordenada de objetos, de acordo com algum critério de ordenação	Preferência, ordenação	Transformações monotônicas crescentes
Intervalo	Diferenças entre os valores têm significado mas os valores em si, não	Horário de calendário, temperatura em graus Celsius ou Fahrenheit	M' = a*M + b (a>0)
Racional	Existe um valor zero significativo, e as razões entre os valores também têm significado	Tamanho, peso, intervalo de tempo, temperatura absoluta	M' = a*M (a>0)
Absoluta	Nenhuma transformação faz sentido.	Contar entidades	M' = M (identidade)

Significância

Determinar qual representação é a mais adequada para medir um atributo levando-se em consideração as transformações admissíveis conduz ao problema de "significância". Significância tem a ver com determinar que tipos de transformações fazem sentido sem que haja alteração do significado (interpretação) empírico.

Uma sentença é significativa se e somente se seu valor verdadeiro for invariante para todas as transformações admissíveis. Assim, para se obter interpretações significativas no mundo real, baseadas na análise realizada no mundo formal, é necessário considerar que tipos de transformação são significativas no mundo formal.

Valores de atributos normalmente são numéricos – o tamanho de um módulo de software, por exemplo, pode ser 10 KLOC – ou simbólicos (o nível de experiência pode ser classificado, por exemplo, em {nenhum, baixo, médio, alto, perito}. E ainda, por conveniência, os símbolos podem ser mapeados em números. O conjunto {nenhum, baixo, médio, alto, perito} poderia ser mapeado no conjunto {0,1,2,3,4}. Nesse caso, porém, os valores numéricos são simplesmente rótulos arbitrários e operações matemáticas como adição, por exemplo, não têm nenhum significado (KITCHENHAM, PFLEEGER e FENTON, 1995).

Valor de medição e unidade de medida

Um atributo é medido aplicando-se uma determinada unidade de medição para um dado atributo de uma dada entidade, com o objetivo de se obter um valor. Um valor de medição somente pode ser interpretado dentro do contexto da entidade, atributo e unidade de medida aos quais se aplica.

Uma unidade de medição determina como o atributo é medido (KITCHENHAM, PFLEEGER e FENTON, 1995). Um atributo pode ser medido com uma ou mais unidades de medição, e uma mesma unidade de medição pode ser utilizada para medir mais de um atributo. A duração de um projeto pode, por exemplo, ser medida em dias, semanas, meses ou anos, e o tamanho de um código pode ser medido contando-se as linhas de código ou o número de rotinas.

Uma unidade pode ser aplicável a diferentes atributos e a diferentes unidades. Porém, uma unidade de medição específica não pode ser definida independentemente do atributo e da entidade aos quais se aplica. Por exemplo, pode-se referir ao grau de experiência numa escala ordinal: {baixa, média, alta, perito} ou em uma escala nominal, como {"tem experiência", "não tem experiência"}.

KITCHENHAM, PFLEEGER e FENTON (1995) estendem o uso das unidades para permitir a definição dos pontos de escala de métricas com escala ordinal e para as categorias utilizadas para métricas de escala nominal. Por exemplo, na definição de categorias de erros como {grande, pequeno e negligenciável }, é necessário definir esses termos mais detalhadamente para possibilitar uma coleta consistente de dados. Nesse

caso, as "unidades" seriam a descrição de "grande" (por exemplo, um erro que resulte numa falha do software), "pequeno" (um erro que resulta numa saída incorreta para o usuário, por exemplo) e "negligenciável" (por exemplo, uma estrutura de código que vai de encontro com práticas-padrão de codificação). Portanto, no contexto de métricas de escala nominal e ordinal onde as métricas são mapeamentos para rótulos ordinários, uma unidade é necessária para garantir que essas métricas sejam utilizadas consistentemente.

Faixas de medição

Métricas válidas devem ser definidas a partir de um conjunto de valores permissíveis. Por exemplo, tamanho de módulos de software em linhas de código (LOC) é definida somente nos inteiros não-negativos.

O conjunto de valores permitidos para uma métrica é definido através de uma faixa. Este conjunto de valores pode ser finito ou infinito, limitado ou ilimitado, discreto ou contínuo. Faixas de métricas com escala nominal e ordinal normalmente são discretas (por exemplo, o mapeamento para inteiros), enquanto que para métricas de intervalo ou racionais podem ser contínuas ou discretas.

Protocolos de medição

Protocolos de medição permitem que sejam feitas medidas de um atributo de uma dada entidade de maneira comparável e repetível (KITCHENHAM, PFLEEGER e FENTON, 1995), definindo como os mesmos devem ser medidos. Repetível significa que duas pessoas que coletam independentemente os dados devem obter o mesmo valor quando medirem o mesmo item. Comparável significa que valores obtidos de diferentes itens tenham sido obtidos utilizando os mesmos procedimentos de medição. Dessa forma, protocolos de medição ajudam a tornar as métricas independentes de quem as utiliza e do ambiente. Além disso, são importantes pelo fato que, apesar de o entendimento ser baseado em abstrações das entidades, o que se mede são entidades reais. Por exemplo, um protocolo para medir LOC pode considerar somente as linhas de código-fonte que não estejam em branco e que não sejam de comentários, iniciando da primeira e seguindo até a última linha do programa, incluindo todas as sub-rotinas.

Uma vez identificadas as métricas que deverão ser utilizadas num programa de medição, estas ainda precisam ser claramente definidas de modo a facilitar a sua repetição e comunicação. A definição operacional das métricas é um passo importante

para um programa de medição, pois quando os usuários dos dados não sabem como estes foram coletados e quais as suas definições, visões distorcidas da realidade podem ser geradas. Este aspecto poderá levar a decisões equivocadas.

Atributos Internos e Externos

Medição é uma quantificação direta de um atributo de uma entidade. Muitas das propriedades de interesse empregam mapeamentos diretos de atributos para números. Porém, ao observar atributos compostos, atributos externos ou relações entre atributos, pode ser impossível medir diretamente o atributo ou a relação.

Atributos internos são atributos que podem ser medidos com base apenas nas entidades aos quais se referem, como por exemplo: tamanho de um documento, esforço demandado na codificação ou experiência da equipe de desenvolvimento.

Atributos externos são aqueles que podem ser medidos somente através da observação do relacionamento que suas entidades têm com o ambiente. São exemplos de atributos externos: a compreensibilidade de uma especificação de requisitos e produtividade de uma equipe.

Nesse contexto, é possível distinguir os conceitos de medição direta e indireta. A medição de atributos internos utiliza mapeamentos diretos de atributos para números. Porém, quando existem relacionamentos complexos entre os atributos ou quando um atributo precisa ser medido através da combinação de diversos aspectos (como é o caso dos atributos externos), então é necessário um modelo que indique como as medidas relacionadas devem ser combinadas.

Medição direta e Indireta

A medição direta de um atributo de uma entidade é aquela na qual não existe dependência da medição de outro atributo (FENTON, 1994). Uma medição direta é resultado de um processo direto de quantificação, como por exemplo medir o tamanho de um módulo de software em LOC ou a duração em horas das atividades relacionadas a teste do produto. Porém, em muitos casos as métricas são obtidas a partir de equações envolvendo outras métricas. Essas métricas são chamadas de indiretas.

A medição indireta de um atributo de uma entidade envolve a medição de um ou mais atributos (FENTON, 1994): as medições individuais são feitas e seus resultados combinados em um único valor que reflete o atributo que se deseja compreender. Exemplos de métricas indiretas são: produtividade calculada em LOC produzidas por

homens-mês; densidade de defeitos de um módulo calculada através do número de defeitos do módulo pelo tamanho do mesmo.

Para realizar medição indireta, é necessário um modelo que reflita um ponto de vista de uma entidade em um determinado ambiente, baseado em uma associação observada empiricamente entre atributos. Esse modelo pode ser formalizado através de uma equação matemática, e os atributos (ou o atributo) utilizados na equação podem ser referentes a uma ou mais entidades diferentes daquela cujo atributo está sendo medido. Por exemplo, para realizar a estimativa de esforço (que é um atributo de um processo ou de um recurso), pode-se utilizar o tamanho (que é atributo de um produto).

Classificação de métricas

FENTON e PFLEEGER (1997) classificam as métricas de software em três categorias:

- Métricas de processos: procuram obter informações a respeito das atividades realizadas durante o desenvolvimento de *software*. Normalmente, essas métricas possuem alguma relação com a noção de tempo, devido à seqüência existente entre as atividades. As métricas de processo são mais difíceis de serem definidas devido, em grande parte, à falta de entendimento das atividades que compõem o processo.
- Métricas de produto: procuram obter informações a respeito de qualquer artefato que resulte da execução de uma atividade.
- Métricas de recursos: procuram obter informações a respeito de qualquer entidade necessária para a execução de uma atividade.

Além dessas classificações, as métricas ainda podem ser classificadas como subjetivas ou objetivas. Métricas subjetivas são aquelas que dependem do julgamento humano, e seus resultados podem variar de pessoa para pessoa refletindo o julgamento de quem realizou a medição. Utilizando uma métrica subjetiva, se uma mesma pessoa medir um atributo em momentos diferentes poderá encontrar respostas distintas, mesmo que o atributo não tenha se modificado. Métricas objetivas, ao contrário, pressupõem que o processo de quantificação seja baseado em regras numéricas bem definidas, permitindo que um mesmo resultado seja obtido independentemente da pessoa que realizou a medição, do momento ou do ambiente em que isto foi feito. Para isso, basta que a condição em que se encontrava o atributo não tenha se modificado.

Ao selecionar-se uma métrica é interessante optar pelas métricas objetivas sempre que for possível. No entanto, não se deve desprezar o valor de uma métrica subjetiva quando esta for capaz de ajudar na caracterização, avaliação, previsão ou melhoria de algum processo ou produto (CHRISTENSEN, THAYER, 2001).

Pode-se dizer que não existe nenhuma métrica absolutamente objetiva, pois sempre haverá algum grau de subjetividade na caracterização das entidades e seus atributos (FENTON e PFLEEGER, 1997). Além disso, a comunicação sem ambigüidades dos resultados da medição é inerentemente difícil (PARK *et al.*, 1996).

Modelos de medição

Esses modelos podem ser utilizados para caracterizar, avaliar ou predizer atributos desejados. Dependendo do uso que se pretenda desses modelos, há três tipos básicos de modelos de relacionamento de atributos (BRIAND *et al.*, 1996):

- modelos descritivos
- modelos avaliativos
- modelos preditivos

Um modelo descritivo descreve um atributo de uma determinada entidade de uma maneira operacional que seja adequada aos objetivos da medição e que faça suposições plausíveis sobre o ambiente. Sua definição é feita em termos de atributos da entidade que sejam diretamente mensuráveis (métricas diretas) ou em termos de outros atributos da própria entidade ou mesmo outras entidades (métricas indiretas). Tais modelos expressam um relacionamento entre atributos diretamente baseado no conhecimento e compreensão que se tem acerca do atributo em questão.

Definindo matematicamente, um modelo descritivo é uma função $m = f(x_1, ..., x_n)$, onde m é uma métrica baseada num modelo que integra n outras métricas (BRIAND *et al.*, 1996). Se por exemplo o modelo definisse um métrica para a densidade de defeitos de um módulo de software, definido como a razão entre o número de defeitos e o tamanho do módulo, então X1 seria o número de defeitos, X2 seria a medida do tamanho do módulo e a densidade seria descrita por $m = X1 / X2$.

Modelos avaliativos capturam situações nas quais um determinado atributo precisa ser avaliado com base em uma ou mais medições do mesmo. A faixa de valores das medições é mapeada em alternativas de decisão que acarretam ações corretivas ou preventivas. Por exemplo, com base no número de defeitos detectados durante uma

inspeção de código, um gerente de projeto pode decidir pela reengenharia de um determinado módulo ou componente do software.

Um modelo avaliativo é uma função d = f(x1, ..., xn), onde d $\in$ {d1, ..., dm} (o conjunto de todas as possíveis alternativas de decisão, mutuamente exclusivas, baseadas no modelo de avaliação, e f é a função de decisão com critérios de entrada {x1, ..., xn} (BRIAND *et al.*, 1996). Por exemplo, a decisão de realizar ou não uma inspeção de código pode ser feita em função da complexidade ciclomática de um módulo de software; se a complexidade ciclomática for maior que 15, então o módulo deve ser inspecionado.

O objetivo de um modelo preditivo é predizer um atributo de uma entidade que ainda não existe, como por exemplo a confiabilidade esperada de um futuro sistema ou a estimativa de custos de um sistema. Nos modelos preditivos, considera-se que os valores de dois ou mais atributos sejam relacionados, de modo que o valor de um ou mais atributos possam ser utilizados para predizer o valor de um outro atributo. Os modelos de custos, por exemplo, normalmente utilizam o tamanho para predizer esforço (esforço = $a*LOC^b$, com a,b > 0).

Os modelos de predição são constituídos de funções de um dos seguintes tipos (BRIAND *et al.*, 1996):

1. $\hat{e} = f(x1, ..., xn)$, onde "ê" é uma pontuação do modelo para estimar a variável "e" que se pretende predizer e {x1,..., xn} são as variáveis de entrada do modelo.

2. p(e) = f (x1, ..., xn), onde p(e) é a probabilidade de ocorrência do evento "e".

O modelo COCOMO (BOEHM, 1981), por exemplo, é do tipo 1, tendo o tamanho em termos de KLOC como entrada, para predizer esforço em termos de meses (homens-mês) com uma fórmula do tipo esforço = $a*tamanho^b$. Pode-se, por exemplo, predizer a probabilidade de detectar uma falha em um módulo de software utilizando uma função do tipo 2 baseada na complexidade e no acoplamento do módulo: p(detecção de falha) = f (complexidade, acoplamento).

Com base nesses modelos, pode-se definir os atributos que se deseja medir. Para esses atributos, métricas podem ser definidas de acordo com suposições claras sobre o processo de software em questão e com a definição explícita dos objetivos endereçados.

A partir desses objetivos e suposições, é possível identificar propriedades desejáveis nas métricas, e utilizar isso para direcionar e restringir a busca por métricas (BRIAND *et al.*, 1996). Podem existir diferentes métricas para um determinado

atributo. Porém, qualquer métrica que satisfaça a condição de representação é uma métrica válida (FENTON e PFLEEGER, 1997).

Validação de métricas

As métricas devem ser validadas para garantir que estejam realmente medindo o que se propõe a medir, e também indicar o quanto as mesmas são úteis. Uma métrica é válida se satisfaz a condição de representação, ou seja, se ela captura no mundo matemático o comportamento que é percebido no mundo empírico. Um pré-requisito para a validade é a confiabilidade – o quanto a repetida aplicação de um processo de medição pode levar exatamente ao mesmo resultado. Essas duas características podem ser verificadas teórica ou empiricamente (BRIAND *et al.*, 1996). A validação teórica é feita para demonstrar que uma métrica esteja realmente medindo o atributo que se propõe a medir, enquanto que a validação empírica é aplicada para demonstrar que uma métrica é útil, no sentido de ser relacionada da forma esperada com outras variáveis. Portanto, uma métrica é válida se satisfaz a ambos os tipos de validação. Nesse contexto, a teoria de medição é uma ferramenta formal útil para construir e validar métricas.

2.3. Enfoques para medição de software

2.3.1. Planos de medição

Para se fazer afirmações precisas sobre a qualidade dos produtos ou processos de software, é necessário descrever quantitativamente o conceito de qualidade (o que pode não ser fácil para muitos aspectos de qualidade, como, por exemplo, para usabilidade). A questão principal é: como identificar características relevantes dos processos e dos produtos que possam descrever o aspecto relativo à qualidade que se deseja avaliar num dado contexto, e como fazer a interpretação nesse contexto?

Existem diversas métricas que podem ser aplicadas a uma grande variedade de atributos de projetos, processos e produtos (BASILI, 1999) (FENTON e PFLEEGER, 1997) (KAN, 2003), o que inclui, por exemplo, dados sobre recursos (ex.: esforço por atividade, características da equipe, tempo decorrido), dados sobre modificações e defeitos no software (ex.: modificações e defeitos por fase do desenvolvimento), dados sobre o processo (definição do processo, aderência ao processo) e dados sobre os produtos. As medições podem ser feitas de várias maneiras: o tamanho de um módulo

de software pode ser medido através da contagem das linhas de código (LOC), através de análise de pontos de função (GARMUS e HERRON, 2001), ou contando-se o número de objetos e métodos. Dessa forma, o problema restringe-se a decidir que métricas devem ser escolhidas para um dado projeto ou para uma organização, quem e quando os dados devem ser coletados e como as métricas devem ser utilizadas nas decisões gerenciais e para identificar oportunidades de melhoria.

Para se definir as métricas relevantes para se medir o alcance de um objetivo, e também para definir como deverão ser a análise e a interpretação dos dados de medição em um determinado ambiente, é necessário se realizar um planejamento da medição – uma metodologia para decidir o que e como deve ser medido, e como interpretar os dados coletados, explicitada através de um Plano de Medição.

Basicamente, existem duas abordagens a se seguir (PFLEEGER e ROMBACH, 1994): *top-down* ou *bottom-up*. Na abordagem *bottom-up*, inicia-se verificando o que pode ser medido e a partir daí chega-se aos objetivos (HETZEL, 1993). É especificado um conjunto básico de métricas que devem ser coletadas para cada produto desenvolvido e utilizado, o que inclui métricas referentes aos recursos aplicados, às atividades do processo, aos produtos criados e aos resultados verificados (uso e eficácia dos produtos na satisfação dos requisitos, por exemplo).

A teoria por trás da abordagem *bottom-up* é que existe um conjunto de métricas fundamentais que são necessárias para responder qualquer questão, e que independem dos objetivos específicos da organização. Nela, o objetivo da medição é estimular questões e ajudar a fornecer informação e especulações sobre os processos de software. Na prática, porém, a abordagem se mostrou falha a partir do momento em que se começou a medir somente o que era conveniente ou fácil de medir, em vez de medir o que era realmente necessário. Como conseqüência, os dados resultantes terminam não sendo úteis para se atingir o objetivo da análise (BASILI, 1992) (FENTON e PFLEEGER, 1997).

Por outro lado, a abordagem *top-down* ajuda a derivar métricas a partir de objetivos e necessidades de informação, e a interpretar os dados no contexto dos objetivos (BASILI *et al.*, 1994). A identificação de métricas úteis é baseada nos assuntos tratados nos objetivos desejados. Por exemplo, para derivar métricas para o objetivo de analisar um módulo de software com o propósito de prever sua manutenibilidade, da perspectiva daquele que fará a manutenção, é necessário compreender o modelo de manutenção que se pretende adotar (ex.: quantos módulos

sofreram alteração) e também o código-fonte do módulo (aspectos estruturais que influenciem a manutenibilidade).

O termo "Medição orientada a objetivos" é a definição *top-down* para um programa de medição baseado em objetivos definidos de maneira precisa e explícita, que estabeleçam como as métricas deverão ser utilizadas, provendo suporte à adequação, consistência e completude do plano de medição, e conseqüentemente à coleta de dados. A medição orientada a objetivos ajuda a gerenciar a complexidade dos programas de medição a permite discussões estruturadas acerca da medição.

Diferentes abordagens *top-down* foram desenvolvidas (ROCHE *et al.*, 1994), como por exemplo o QFD (*Quality Function Development)* (KOGURE *et al.*, 1983), o SQM (*Software Quality Metrics)* (MURINE, 1980) (MCCALL *et al.*, 1977) (BOEHM *et al.*, 1976) e o GQM (*Goal-Question-Metric)* (BASILI *et al.*, 1994) (BASILI e ROMBACH, 1988) (BASILI e WEISS, 1984).

Essas abordagens *top-down* focam na relação das métricas com os objetivos de medição, representando um grande avanço em comparação à abordagem *bottom-up* (ROMBACH, 1991). Elas diferem significativamente em termos de escopo dos objetivos de medição que tratam e da forma como guiam para identificação de métricas relevantes. Nesse sentido, a abordagem GQM tem mostrado ser a mais flexível e mais aplicável delas (ROCHE e JACKSON, 1994) (BASILI, 1992) (ROMBACH, 1991), e por ser, também, a adotada neste trabalho é descrita com detalhes a seguir.

2.3.2. O Paradigma GQM (*Goal, Question, Metric*)

O Paradigma GQM (*Goal, Question, Metric*) (BASILI *et al.*, 1994) (BASILI e ROMBACH, 1988) (BASILI e WEISS, 1984) é uma abordagem orientada a objetivos para medição de produtos e processos de Engenharia de Software. Baseia-se na hipótese de que para uma organização medir de forma objetiva, ela deve identificar, explicitar e especificar precisamente os objetivos de medição da organização e também aqueles relativos a cada projeto; deve relacionar esses objetivos aos dados necessários, para defini-los operacionalmente; e também deve fornecer um *framework* para análise e interpretação dos dados com respeito aos objetivos definidos.

O GQM dá suporte a objetivos de medição relativos a qualquer tipo de produto ou processo de software, dirigidos a todos os tipos de propósito, desde a caracterização até o controle e a melhoria, com foco em qualquer aspecto de qualidade, definido a partir de qualquer ponto de vista (perspectiva) e em qualquer ambiente. Ele é um

mecanismo para definição e avaliação de um conjunto de objetivos operacionais, e representa uma abordagem sistemática para customização e integração dos objetivos com modelos de produtos, processos de software e perspectives de qualidade, com base nas necessidades especificas do projeto e da organização (BASILI, 1999).

O Paradigma GQM foi desenvolvido inicialmente em 1984 na Universidade de Maryland (BASILI e WEISS, 1984), em cooperação com o "NASA Goddard Space Flight Center" (BASILI *et al.*, 1992) e foi continuado como parte do Projeto TAME (BASILI e ROMBACH, 1988). Diversas pesquisas foram realizadas envolvendo sua aplicação e melhoria (SOLINGEN e BERGHOUT, 1999), tendo sido também utilizado com sucesso, levando a melhorias substanciais, em organizações como NASA-SEL (BASILI *et al.*, 1992) e Motorola (DASKALANTONAKIS, 1992).

Em um programa de medição baseado em GQM, a atividade de análise é definida de maneira precisa e explícita através de um objetivo de medição. Os objetivos são definidos de modo que sejam operacionalmente tratáveis, por meio do refinamento para um conjunto de questões quantificáveis que são utilizadas para extrair a informação apropriada dos modelos, representando as dimensões dos objetivos.

As métricas são derivadas utilizando a técnica *top-down*, com base nas questões, definidas a partir dos objetivos, formalizando o processo e levando à escolha e/ou definição de métricas relevantes (BRIAND *et al.*, 1996). Esse refinamento é documentado minuciosamente em um plano GQM, registrando todo o racional utilizado na escolha das métricas. Os dados coletados são interpretados de maneira *bottom-up*, no contexto dos objetivos e questões definidos, considerando as limitações e suposições relativas a cada métrica. A estrutura hierárquica de 3 níveis, denotada como plano GQM, é mostrada na figura 2.3 (BASILI *et al.*, 1994):

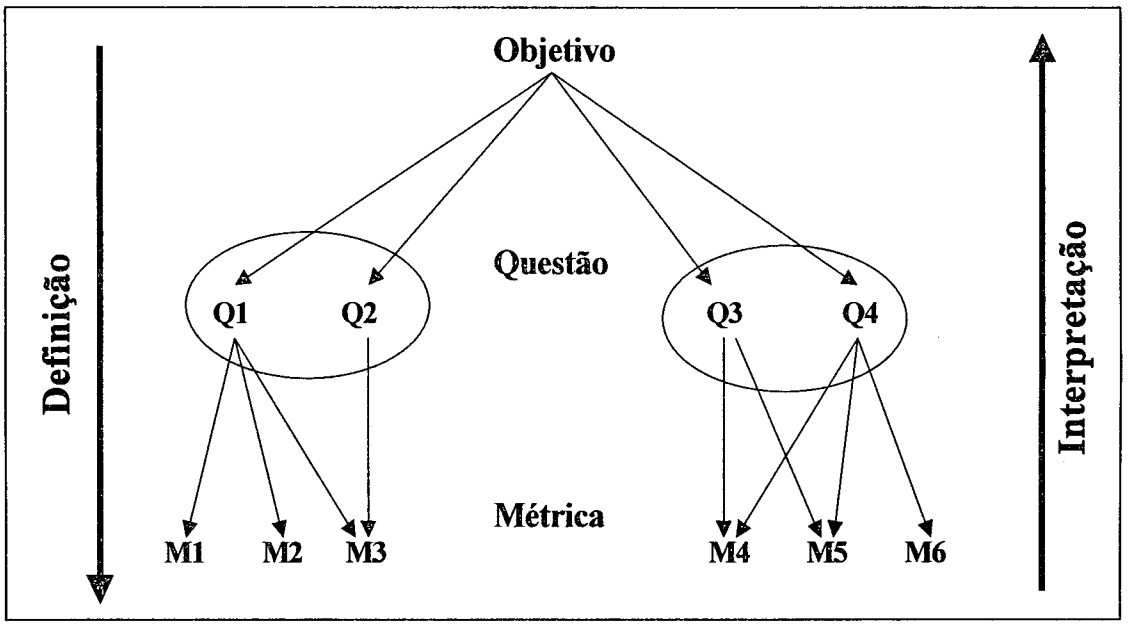

Figura 2.3 – Paradigma GQM (BASILI et al., 1994)

1. Nível conceitual (*Goal* / Objetivo): Um objetivo é definido para um objeto, com um propósito específico, com respeito a um certo modelo de qualidade, a partir de um dado ponto de vista relativo ao ambiente.
2. Nível operacional (*Question* / Questão): Um conjunto de questões é utilizado para definir como será feita a avaliação e como será atingido um objetivo específico. O objeto de medição é caracterizado através de questões que levam em consideração o modelo de qualidade e o ponto de vista definido no objetivo.
3. Nível quantitativo (*Measure* /Métrica): Um conjunto de métricas é definido para descrever o mapeamento do sistema relacional empírico para o modelo formal, com respeito às questões definidas.

Dessa forma, o GQM ajuda na realização do refinamento apropriado de questões abstratas de qualidade do domínio de Engenharia de Software, e na derivação de mapeamentos de medição adequados entre o mundo empírico e um modelo formal.

Como exemplo, suponha-se que o objetivo a ser atingido pelo programa de medição seja analisar a confiabilidade de um sistema, do ponto de vista do desenvolvedor, com o propósito de melhoria (figura 2.4). Em outras palavras, isso significa que se deseja analisar como melhorar a confiabilidade do sistema desenvolvido.

Para decidir quais são os fatores que devem ser modificados para se atingir a melhoria, é necessário identificar fatores de influência relevantes, o que exige que se defina como se enxerga a característica de qualidade "confiabilidade", por exemplo, em termos de quantos erros foram detectados por módulo (questão 1). Além disso, é necessário analisar a relação entre fatores potenciais de influência – por exemplo, através da avaliação da relação entre o tamanho dos módulos do sistema (questão 2) e o tipo de inspeção realizada (questão 3).

Uma vez definidas as questões, é necessário, para cada questão, definir o que precisa ser medido para respondê-las. Por exemplo, para analisar o tamanho de um módulo de software, é necessária uma métrica de tamanho – por exemplo LOC, pontos de função ou qualquer outra métrica de tamanho que seja aplicável no ambiente em questão.

Um objetivo é definido tão bem quanto as questões que ele gera e os modelos nos quais essas questões são baseadas. Uma vez que esses modelos são de difícil definição, na maioria das vezes eles ficam implícitos nas questões. Porém, o quanto mais formal, explícitos e completos forem os modelos, mais eficazes serão as questões e a definição dos objetivos. Cada questão gera um conjunto de métricas, e novamente as questões somente poderão ser respondidas com relação às métricas utilizadas, com respostas tão completas quanto as métricas permitirem.

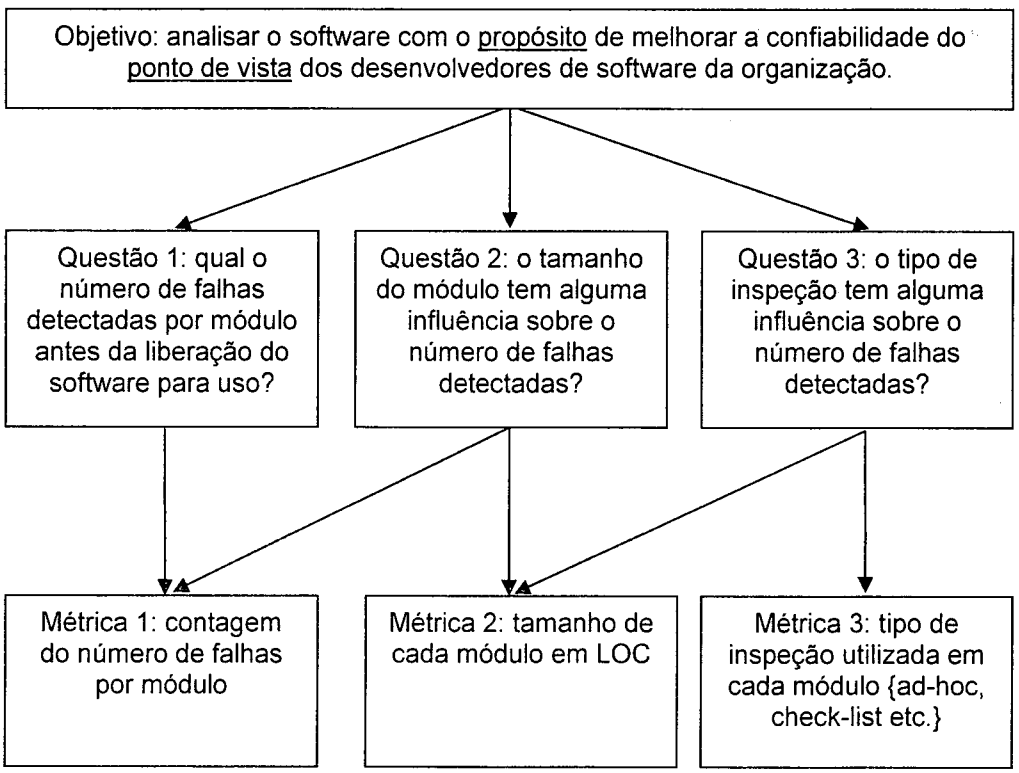

Figura 2.4 – Exemplo de Objetivo, Questões e Métricas

Assim como outras atividades, o processo de medição demanda eficiência no uso de recursos. Através do GQM, pode-se chegar a um conjunto ótimo de métricas: o menor número possível de métricas, com maior poder de resposta e que estejam efetivamente relacionadas aos objetivos. Uma mesma questão pode ser utilizada para definir vários objetivos, e as métricas podem ser utilizadas para responder mais de uma questão. As questões e métricas podem ser reutilizadas dentro um plano GQM ou mesmo entre diferentes programas de medição.

O Paradigma GQM baseia-se nos seguintes princípios (BASILI *et al.*, 1994), (ROMBACH, 1991):

- A atividade de análise deve ser definida precisa e explicitamente através de um objetivo de medição.
- As métricas devem ser derivadas de forma *top-down*, com base nos objetivos e questões. A estrutura de objetivos e questões não deve ser adaptada a partir de um conjunto pré-existente de métricas.
- Cada métrica deve ter uma fundamentação lógica que seja explicitamente documentada, que será utilizada para justificar a coleta de dados e guiar a análise e interpretação dos mesmos.
- Os dados coletados deve ser interpretados de maneira bottom-up, com atenção ao contexto dos objetivos de medição e às questões.
- Aqueles que utilizarão os resultados do programa de medição – e a partir dos quais o "ponto de vista" dos objetivos do plano GQM foi definido – precisam estar ativamente envolvidas na definição e interpretação do programa de medição.

A aplicação do GQM consiste em quatro fases (SOLINGEN, 1999), conforme ilustradas na figura 2.5:

(i) Fase de Planejamento, que envolve a seleção do que será mensurado e o planejamento do projeto de medição;

(ii) Fase de Definição, onde os objetivos, questões e métricas são definidos e documentados;

(iii) Fase de Coleta de Dados, onde é realizada a coleta de dados;

(iv) Fase de Interpretação, na qual os dos coletados são analisados para responder às questões e as respostas são usadas para verificar se os objetivos estabelecidos foram alcançados.

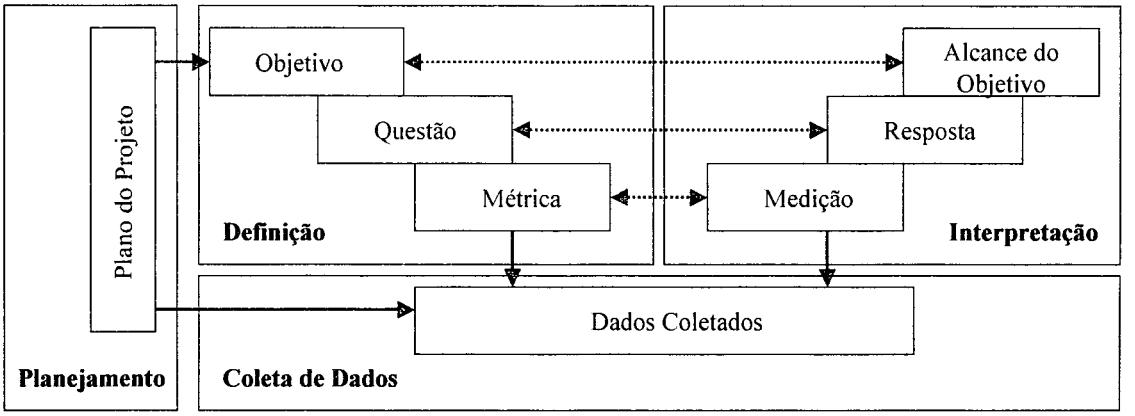

Figura 2.5 – Fases do GQM (SOLINGEN, 1999)

Durante a definição dos objetivos de medição, algumas vezes os objetivos são óbvios por estarem relacionados a necessidades urgentes da organização. Em outras, porém, são difíceis de serem selecionados e priorizados. Nestas situações a resposta às seguintes questões pode apoiar a seleção dos objetivos (SOLINGEN, 1999):

- Quais são os objetivos estratégicos da organização?
- Que forças têm impacto nestes objetivos estratégicos?
- Como pode ser melhorado o seu desempenho?
- Quais são os principais problemas?
- Quais são os objetivos de melhoria?
- Como atingir seus objetivos de melhoria?
- Quais são os possíveis objetivos de melhoria e quais são as prioridades?

Dentre as vantagens da abordagem GQM, podem ser citadas (ROMBACH, 1991): o suporte à definição operacional dos objetivos, a identificação de métricas úteis e relevantes, o suporte à análise e à interpretação dos dados coletados. A abordagem também permite uma avaliação da validade dos modelos obtidos e conclusões, a partir da documentação explícita do refinamento que é feito até se chegar a cada métrica.

O GQM ajuda, ainda, a garantir a adequação, a consistência e a completude do plano de medição. A adminstração da complexidade do programa de medição também é apoiada pelo GQM, permitindo uma discussão estruturada sobre medição e diminuindo a resistência da equipe de desenvolvimento, através de sua contínua participação no processo de medição (BRIAND *et al.*, 1996).

2.4. Considerações finais

Este capítulo apresentou os conceitos chaves, abordagens e enfoques para medição de software, com foco nos conceitos de métricas e de GQM que são os temas mais relevantes para este trabalho. Em (ZUSE, 1998) (BRIAND *et al.*, 1996) (KITCHENHAM, PFLEEGER e FENTON, 1995) (FENTON, 1994) pode-se consultar discussões mais aprofundadas da aplicação da teoria de medição em Engenharia de Software.

No próximo capítulo serão apresentadas as principais abordagens para medição definidas pelas normas e modelos nacionais e internacionais.

Capítulo 3

Medição de Software nas Normas e Modelos de Maturidade

Por sua importância, a medição de software é tratada nas principais normas internacionais, assim como nos modelos de maturidade nacional (MPS.BR) e internacional (CMMI). Neste capítulo são descritas as principais abordagens definidas por essas normas e modelos.

3.1. Introdução

Medições apóiam a gestão dos processos e a identificação de oportunidades de melhorias (FLORAC e CARLETON, 2000), e são essenciais para se conhecer a real situação dos processos.

Segundo a norma de Processo de Medição de Software, ISO/IEC 15939:2001 (2002), o objetivo da medição de software é coletar, analisar e reportar dados relacionados com os produtos desenvolvidos e processos implementados na organização e em seus projetos, para suportar a gestão efetiva dos processos e demonstrar objetivamente a qualidade dos produtos.

As abordagens para medição de software podem ser classificadas quanto à orientação do processo de medição. Nos processos de medição orientados por objetivos, há uma sistemática que apóia a definição do que deve ser medido, iniciando pelos objetivos (ex.: reduzir retrabalho) e definindo a partir destes as medições (métricas) a serem utilizadas para que seja possível verificar se os objetivos estão sendo atingidos. A abordagem GQM (SOLINGEN, 1999), descrita no capítulo 2, se enquadra nesta categoria.

Nos processos de medição orientados por métodos estatísticos, o processo de medição está mais voltado para medir o desempenho dos processos (efetividade, eficiência, produtividade, qualidade do produto entre outros) pelo uso de técnicas estatísticas, buscando-se controlar os processos. A abordagem para processos, proposta por FLORAC e CARLETON (2000) se enquadra neste grupo.

Diferentes técnicas de análise podem ser utilizadas nas diversas situações envolvendo medições.

As técnicas de análise quantitativa ou estatística (KAN, 2003) (FLORAC e CARLETON, 2000) visam detectar variações no comportamento esperado dos

processos, através de monitoração dos mesmos por meio de permanentes medições (aplicadas, por exemplo, no CMMI níveis 4 e 5).

As análises orientadas a objetivos (SOLINGEN, 1999) são voltadas para os processos de medição baseados em objetivos, como o GQM. Com os dados obtidos nas medições, são calculadas métricas e indicadores, e são elaboradas respostas para as questões derivadas dos objetivos estabelecidos no início do processo de medição.

As próximas seções descrevem as principais normas e modelos que definem abordagens e processos para medição de software. Todas elas apresentam processos de medição orientados por objetivos e prevêem procedimentos de análise orientados a objetivos.

3.2. CMMI/SW

O CMMI (*Capability Maturity Model Integration*) (CMU/SEI, 2002) tem como objetivo prover um *framework* único e integrado para melhoria de processos em organizações que envolvem diversas disciplinas. Através de um esforço para compilar as diversas visões e tendências surgidas para melhoria de processo baseada em modelos, o SEI (*Software Engineering Institute* da Universidade de Carnegie Mellon) em conjunto com órgãos militares americanos e a Indústria, busca a integração de ferramentas e técnicas utilizadas para melhoria individual das disciplinas de engenharia (dentre elas o CMM-SW), visando à qualidade e à eficiência da melhoria dos processos organizacionais. O CMMI surgiu para resolver o problema de se usar vários modelos e é o resultado da evolução do SW-CMM, SECM® (*System Engineering Capability Model*) e IPD-CMM® (*Integrated Product Development Capability Maturity Model*). É, portanto, o sucessor destes modelos. Além disso, o *framework* CMMI foi desenvolvido para ser consistente e compatível com a ISO/IEC 15504 (Chrissis, 2003).

Os modelos do CMMI permitem sua utilização em uma única disciplina (Engenharia de Software, por exemplo) ou de maneira integrada, através de uma representação por estágios ou em uma representação contínua. Tais modelos incluem vasto conhecimento sobre engenharia e melhoria de processos, apresentados através de objetivos claros e guias extensivos sobre as melhores práticas para atingi-los e, sobretudo, um *framework* bem definido que propõe como novas disciplinas podem ser acrescentadas ao conjunto já tratado pelo CMMI, de modo a minimizar o desenvolvimento de modelos incompatíveis no futuro.

Para se avaliar a aderência ao CMMI o método criado pelo SEI é o SCAMPI, o qual está em conformidade com os requisitos para avaliação da ISO/IEC 15504.

3.2.1. Áreas de processo

Um conceito fundamental para todos os modelos CMMI é o de "área de processo" (*process area)*. Nem tudo que é relacionado a processos e melhoria de processo é incluído num modelo de melhoria de processo. O CMMI seleciona somente os tópicos mais importantes para melhoria de processo e agrupa-os em "áreas". Essa classificação resulta em um pequeno conjunto de áreas de processo: vinte e quatro, no caso do CMMI-SE/SW IPPD V.1.0 . Cada área de processo é descrita em termos de práticas para atingir seus objetivos, indicando a infra-estrutura e as atividades que mais contribuem para a efetiva implementação e institucionalização das áreas de processo.

As vinte e quatro áreas de processo do CMMI-SE/SW são dividas em quatro níveis de maturidade na representação por estágios, distribuídas de acordo com a tabela 3.1.

Tabela 3.1 – Níveis de maturidade do CMMI-SE/SW

Nível de maturidade	Áreas de processo
Nível 2	Gerência de Requisitos Planejamento do Projeto Monitoração e Controle do Projeto Gerência de Acordo com Fornecedores Medição e Análise Garantia da Qualidade do Processo e Produto Gerência de Configuração
Nível 3	Desenvolvimento de Requisitos Solução Técnica Integração do Produto Verificação Validação Foco no Processo Organizacional Definição do Processo Organizacional Treinamento Organizacional Gerência Integrada de Projeto Gerência de Riscos Gerência Integrada de Fornecedores Análise de Decisão e Resolução Ambiente Organizacional para Integração

Nível 4	Desempenho do Processo Organizacional Gerência Quantitativa do Projeto
Nível 5	Inovação e Implantação na Organização Análise e Resolução de Causas

3.2.2. Medição e análise no CMMI

No CMMI (2002) existe uma área de processo dedicada especificamente às questões das medições: "Medição e Análise". A área de processo de Medição e Análise tem como objetivo "desenvolver e manter uma capacidade de medição utilizada para dar suporte às necessidades de informações para o gerenciamento" (CHRISSIS, 2003). A área tem dois objetivos específicos: um para alinhar as atividades de medição com as necessidades de medição e outro para se obter resultados de medição que satisfaçam essas necessidades.

A medição não é um fim por si só, e assim as práticas de Medição e Análise são sempre executadas dentro do contexto de execução de outros processos. Normalmente focada no nível dos projetos, a capacidade de medição também pode estar orientada às necessidades de informação da organização como um todo (MUTAFELIJA e STROMBERG, 2003).

A Medição e Análise está relacionada a quase todas as demais áreas de processo do CMMI, funcionando como um guia sempre que houver necessidade de se medir. O planejamento de processos inclui a definição de objetivos de medição, e o monitoramento e controle dos processos inclui medir e comparar os resultados contra o que foi planejado.

Esta área de processo reconhece que uma capacidade de medição útil não acontece por acaso: responsabilidades devem ser atribuídas e procedimentos de coleta, armazenamento e análise de dados devem ser estabelecidos. A necessidade de se comunicar os resultados das análises dos dados aos *stakeholders*[2] relevantes é enfatizada, assim como a necessidade da tomada de ações corretivas quando apropriado (AHERN *et al.*, 2001).

A área de processo Medição e Análise envolve:

- a especificação dos objetivos de medição e análise de forma que estejam alinhados às necessidades de informação do projeto e/ou da organização;

[2] *Stakeholder*: no contexto do CMMI, grupo ou indivíduo envolvido ou afetado pelo resultado de uma tarefa. Pode incluir, entre outros, membros do projeto, fornecedores, clientes, usuários.

- a especificação de medidas, mecanismos de coleta e armazenamento de dados, técnicas de análise de dados e mecanismos para relato e *feedback* de resultados;
- a implementação efetiva da coleta, armazenamento, análise e relato dos dados;
- o fornecimento dos resultados das análises dos dados, que serão utilizados no processo de tomada de decisão na organização e suas apropriadas ações corretivas.

A integração das atividades de medição e análise aos demais processos de um projeto apóia os seguintes aspectos:

- realização de planejamento e estimativas de forma mais objetiva;
- acompanhamento do desempenho real do projeto em relação aos planos e objetivos pré-estabelecidos;
- identificação e solução de problemas relacionados ao projeto.

Através de uma abordagem que fornece os resultados dos objetivos de medição, esta área de processo propicia a execução das ações corretivas apropriadas, uma vez que a tomada de decisões é feita com base em informações reais e históricas da própria organização.

Quando utilizada em uma organização no nível 2, esta área fica focada em atender as necessidades de informação da gerência, com medições mais concentradas em projetos individuais, não havendo a necessidade de um padrão organizacional. Quando utilizada no nível 3, é requerido um padrão organizacional para o processo de medição e deve ser feita uma *baseline* organizacional das medidas. No nível 4, já se tem um histórico de medidas e a gerência quantitativa faz uso estatístico das medições.

A área de medição e análise não define e nem propõe métricas. Abordagens de medição como o GQM (SOLINGEN, 1999) e o PSM (PSMC, 2004) são aderentes às praticas da área de Medição e Análise do CMMI. A Tabela 3.2 mostra os objetivos e práticas específicos da área de Processo de Medição e Análise.

Tabela 3.2 – Medição e Análise no CMMI: Objetivos e Práticas Específicos

Objetivo Específico 1 – Definir Atividades de Medição e Análise
PE 1.1 – Estabelecer Objetivos de Medição
PE 1.2 – Especificar Medidas
PE 1.3 – Especificar Procedimentos de Coleta e Armazenamento de Dados
PE 1.4 – Especificar Procedimentos de Análise
Objetivo Específico 2 – Fornecer Resultados das Medições
PE 2.1 – Coletar Dados das Medições
PE 2.2 – Analisar Dados das Medições
PE 2.3 – Armazenar Dados e Resultados
PE 2.4 – Comunicar Resultados

A figura 3.1 mostra uma visão geral da área de processo Medição e Análise, apresentando seus objetivos e práticas específicos.

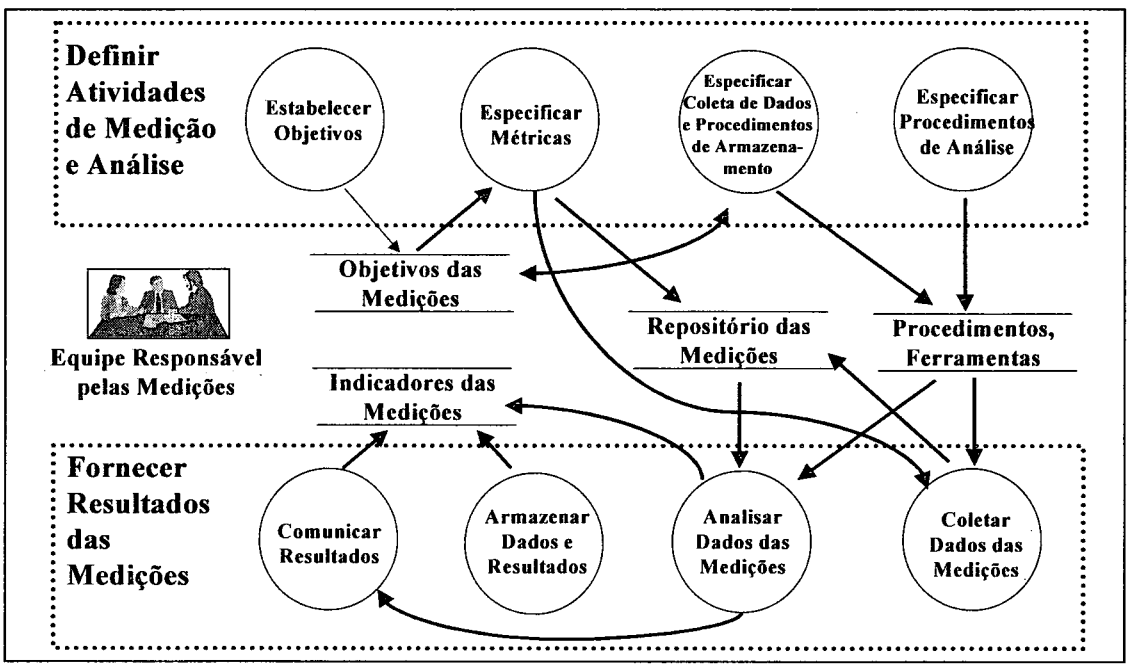

Figura 3.1 – Área de Processo de Medição e Análise do CMMI (Ahern, 2001)

As práticas específicas associadas ao primeiro objetivo "Definir Atividades de Medição e Análise" estabelecem um plano coerente para a medição e análise (Plano de Medição), tratando questões tais como: "o que se está medindo?", "porque se está medindo?", "como vai ser medido?" e "o que vai ser feito com os dados e resultados, uma vez obtidos?".

As práticas específicas associadas ao objetivo "Fornecer Resultados das Medições" apenas direcionam o usuário para fazer o que tem que ser feito. O objetivo

maior é então colocar os resultados das medições e análises nas mãos daqueles que vão tomar decisões e ações com base nestas informações.

3.3. O projeto SPICE e a norma ISO/IEC 15504

O SPICE *(Software Process Improvement and Capability dEtermination)* foi um projeto internacional, com participação de representantes de 25 países, que desenvolveu a norma internacional para avaliação de processos ISO-15504 *Information Tecnology-Process Assessment.* (EMAN *et al*, 1998). A ISO 15504 prove um *framework* para a avaliação de processos de software, que pode ser usado por organizações envolvidas com o planejamento, monitoração, controle, aquisição, fornecimento, desenvolvimento, operação, evolução, e suporte a software.

O projeto SPICE nasceu da necessidade do governo Britânico em avaliar melhor os fornecedores de grandes sistemas, com o objetivo de reduzir os riscos envolvidos, aumentando a chance de sucesso. Em 1992 foi encomendado um estudo intitulado *improve-IT* (DORLING, 1992) que pesquisasse os métodos até então disponíveis para a avaliação de fornecedores de software (determinação de capacidade) e que tratassem também da questão da melhoria dos processos, e que deveria considerar os métodos e normas já existentes (como por exemplo, o SW-CMM e a ISO 9001), abranger todos os processos de software e ser construído pelos especialistas que já desenvolviam e trabalhavam com os métodos e normas existentes à época. Este estudo chegou às seguintes conclusões:

- A maioria dos grandes compradores (governo e privados) têm dificuldades semelhantes em avaliar seus fornecedores;
- Grande parte dos fornecedores tem interesse em melhorar seus processos;
- Existem várias abordagens em uso;
- Existe o interesse em uma abordagem comum para a avaliação de processos;
- Existe muito interesse na auto-avaliação como preparação para melhorias.

O cenário descrito pelo estudo mostrava que a indústria estava usando diferentes abordagens, de diferentes formas, com adaptações e extensões que cada setor julgava necessário. Os resultados deste estudo foram apresentados ao Sub-comitê 7 (SC7) da ISO (*International Organization for Standardization*), que trata da normalização na área de engenharia de software, como argumento para a necessidade de uma norma que

padronizasse a avaliação de processos de software. A proposta foi aprovada e em 1993 iniciaram-se os trabalhos (EMAN et al, 1998).

O resultado final do projeto SPICE é a norma ISO/IEC-15504 IS, com cinco partes publicadas em 2003 e 2004. O *framework* resultante não provê um método único de avaliação de processos, mas todo um conjunto de requisitos que uma avaliação deve seguir para ter conformidade com a norma. Isso permite muita flexibilidade, pois podem existir diferentes modelos/métodos de avaliação, usando diferentes modelos de referência de processos, podendo ser todos conformes à norma. Portanto ela atua como um meta-modelo a partir do qual podem ser derivados diferentes modelos de referência e métodos de avaliação aderentes à norma. Uma avaliação que atenda aos requisitos da norma é dita uma "avaliação conforme a ISO 15504".

Uma das características principais da ISO 15504 é a sua grande flexibilidade, gerando várias possibilidades de adaptação para uso em diferentes contextos. Como não existe um método pré-definido de avaliação, mas sim um conjunto de requisitos a serem seguidos em uma "avaliação conforme", diferentes modelos/métodos podem existir em conformidade com a norma. Como o Modelo de Referência de Processos (PRM) também é escolhido em função da necessidade, diferentes domínios também têm feito uso do *framework* da ISO 15504 para elaboração de modelos para setores específicos.

A ISO/IEC 15504 presta-se à realização de avaliações de processos de software com dois objetivos: a melhoria de processos e a determinação da capacidade de processos de uma unidade organizacional. Se o objetivo for a melhoria de processos, a unidade organizacional pode realizar uma avaliação com o objetivo de gerar um perfil dos processos que será usado para a elaboração de um plano de melhorias. A análise dos resultados identifica os pontos fortes, os pontos fracos e os riscos inerentes aos processos. No segundo caso, a organização tem o objetivo de avaliar um fornecedor em potencial, obtendo o seu perfil de capacidade. O perfil de capacidade permite ao contratante estimar o risco associado à contratação daquele fornecedor em potencial para auxiliar na tomada de decisão de contratá-lo ou não (ISO/IEC, 2003).

O modelo de referência da ISO/IEC 15504 (2003) define a dimensão de processo – que corresponde à especificação de um conjunto de processos considerados universais e fundamentais para a boa prática da engenharia de software – e a dimensão de capacidade, que corresponde à definição de um modelo de medição com base na identificação de um conjunto de atributos que permite determinar a capacidade de um

processo para atingir seus propósitos, gerando os produtos de trabalho e os resultados estabelecidos (MOREAU, 2003).

Na dimensão de capacidade, as tarefas, atividades e práticas, bem como as características dos produtos de trabalho, são definidas como indicadores que demonstram se determinado processo é adequadamente praticado em um determinado nível de capacidade. Há seis níveis de capacidade em que um processo pode ser avaliado. São eles:

- Nível 0 - Processo incompleto: O processo não é implementado ou falha na consecução de seu propósito. Não existe evidência de que os produtos de trabalho sejam adequadamente produzidos ou que os resultados sejam alcançados.
- Nível 1 - Processo executado: O processo implementado alcança seu propósito mas sua execução talvez não seja rigorosamente planejada e acompanhada.
- Nível 2 - Processo gerenciado: O processo executado anteriormente é agora implementado de forma gerenciada (planejado, monitorado e ajustado) e seus produtos de trabalho são apropriadamente estabelecidos, controlados e mantidos.
- Nível 3 - Processo estabelecido: O processo gerenciado anteriormente é agora implementado utilizando um processo definido e é capaz de alcançar seus resultados de processo.
- Nível 4 - Processo previsível: O processo estabelecido anteriormente opera agora dentro de limites para alcançar os resultados de processo.
- Nível 5 - Processo otimizado: O processo previsível anteriormente é melhorado continuamente para satisfazer os objetivos de negócio atuais e projetados mais relevantes.

3.4. NBR ISO/IEC 12207:1995 (Tecnologia da Informação – Processos de Ciclo de Vida de Software) e suas emendas 1 e 2

A norma ISO/IEC 12207 (1995) e suas emendas 1 e 2 (ISO/IEC 12207:1995/Amd 1:2002) (ISO/IEC 12207:1995/Amd 2:2004) estabelecem uma arquitetura comum para o ciclo de vida de processos de software com uma terminologia bem definida. Contém processos, atividades e tarefas a serem aplicadas durante o

fornecimento, desenvolvimento, operação e manutenção de produtos de software e serviços correlatos. As emendas 1 e 2 da ISO/IEC 12207 acrescentaram o propósito e os resultados para cada processo, além de inserir novos ou expandir o escopo de processos.

Os processos definidos na NBR ISO/IEC 12207 podem ser utilizados como referência na implantação e avaliação do MR MPS.BR. É possível realizar exclusões, inclusões ou alterações de processos que não sejam pertinentes ao negócio seguindo o processo de adaptação da NBR ISO/IEC 12207.

A Norma foi publicada em 1995 como norma internacional, ganhando em 1998 uma versão brasileira com o mesmo nome que a internacional, somente acrescida das iniciais NBR.

Em outubro de 2002 e 2004, foram feitas atualizações na norma ISO/IEC 12207, chamadas de emendas 1 e 2 respectivamente, com o objetivo de representar a evolução da Engenharia de Software, as necessidades vivenciadas pelos usuários da norma e a harmonização com a série ISO/IEC 15504 (Avaliação de Processos de Software).

A norma tem por objetivo auxiliar os envolvidos na produção de software a definir seus papéis, por meio de processos bem definidos, proporcionando às organizações que a utilizarem um melhor entendimento das atividades a serem executadas nas operações que envolvem, de alguma forma, software.

A arquitetura da norma utiliza uma terminologia bem definida e é composta de processos, atividades e tarefas para aquisição, fornecimento, desenvolvimento, operação e manutenção de software, estabelecendo uma arquitetura de alto nível para o ciclo de vida do software, que abrange desde a concepção até a descontinuidade do mesmo. Essa arquitetura é baseada em processos-chave e no inter-relacionamento entre eles, seguindo dois princípios básicos (ROCHA *et al.*, 2001):

- Modularidade: os processos têm alta coesão e baixo acoplamento, ou seja, todas as partes de um processo são fortemente relacionadas e o número de interfaces entre os processos é mínimo.
- Responsabilidades: cada processo é de responsabilidade de uma "parte envolvida", que pode ser uma organização ou parte dela. As partes envolvidas podem ser da mesma organização ou de organizações diferentes.

Os processos da ISO / IEC 12207 são agrupados em três classes que representam sua natureza, conforme a tabela 3.3. Cada processo é definido em termos de suas próprias atividades, e cada atividade é adicionalmente definida em termos de suas tarefas.

Tabela 3.3 – Classes de processo da ISO/IEC 12207

Classe de processo	Objetivo
Processos fundamentais	Atendem ao início, à contratação entre adquirente e fornecedor e à execução do desenvolvimento, da operação ou da manutenção de produtos de software durante seu ciclo de vida.
Processos de apoio	Auxiliam e contribuem para o sucesso e a qualidade do projeto de software.
Processos organizacionais	São empregados por uma organização para estabelecer e implementar uma estrutura constituída pelos processos de ciclo de vida e pelo pessoal envolvido no desenvolvimento do software. São geralmente empregados fora do domínio de projetos e contratos específicos, contribuindo para a melhoria da organização.

Considerando o escopo deste trabalho, dentre os processos destacam-se dois:

Processo de garantia de qualidade – Define as atividades para garantir a conformidade dos processos e produtos de software, no ciclo de vida projeto, com seus requisitos especificados e sua aderências aos planos estabelecidos. A Abrangência do processo inclui questões como garantia da qualidade do produto (ISO 13596:1996 – Tecnologia da Informação, avaliação de produto de software, características de qualidade e diretrizes para seu uso) e do processo e do sistema de qualidade (ISO 9000 e 9001).

Processo de melhoria – Define as atividades básicas que uma organização executa para estabelecer, avaliar, medir, controlar e melhorar um processo de ciclo de vida de software. Além desses processos a norma traz um processo de adaptação, essencial para definir as atividades necessárias para adaptar a norma para sua aplicação a qualquer organização, projeto, modelo de ciclo de vida, cultura e técnica de desenvolvimento.

A norma é flexível para as abordagens de engenharia de software envolvidas, sendo utilizável com qualquer modelo de ciclo de vida (em cascata, incremental, evolutivo, RAD etc.), com qualquer métodos ou técnica de engenharia de software (projeto orientado a objetos, técnicas estruturadas, prototipação etc.) e com quaisquer linguagens de programação. Essas escolhas dependerão do projeto e das tecnologias disponíveis e acessíveis, sendo deixadas a critério dos usuários da norma.

3.4.1. Modificações da Emenda 1

A Emenda 1 define um Modelo de Referência de Processos, em um nível de abstração mais alto do que os requisitos detalhados contidos no corpo da norma. Esse

modelo é aplicável a organizações que estejam realizando avaliação de seus processos com a finalidade de determinar a capacidade dos mesmos. O modelo traz objetivos e resultados esperados para cada processo, permitindo a avaliação da efetividade dos processos. Foram definidos novos processos, como processo de gerência de ativos, processo de reuso e de engenharia de domínio. A arquitetura de processos da norma, já incluindo as alterações realizadas nos FDAM-1 e FDAM-2, é apresentada na figura 3.2.

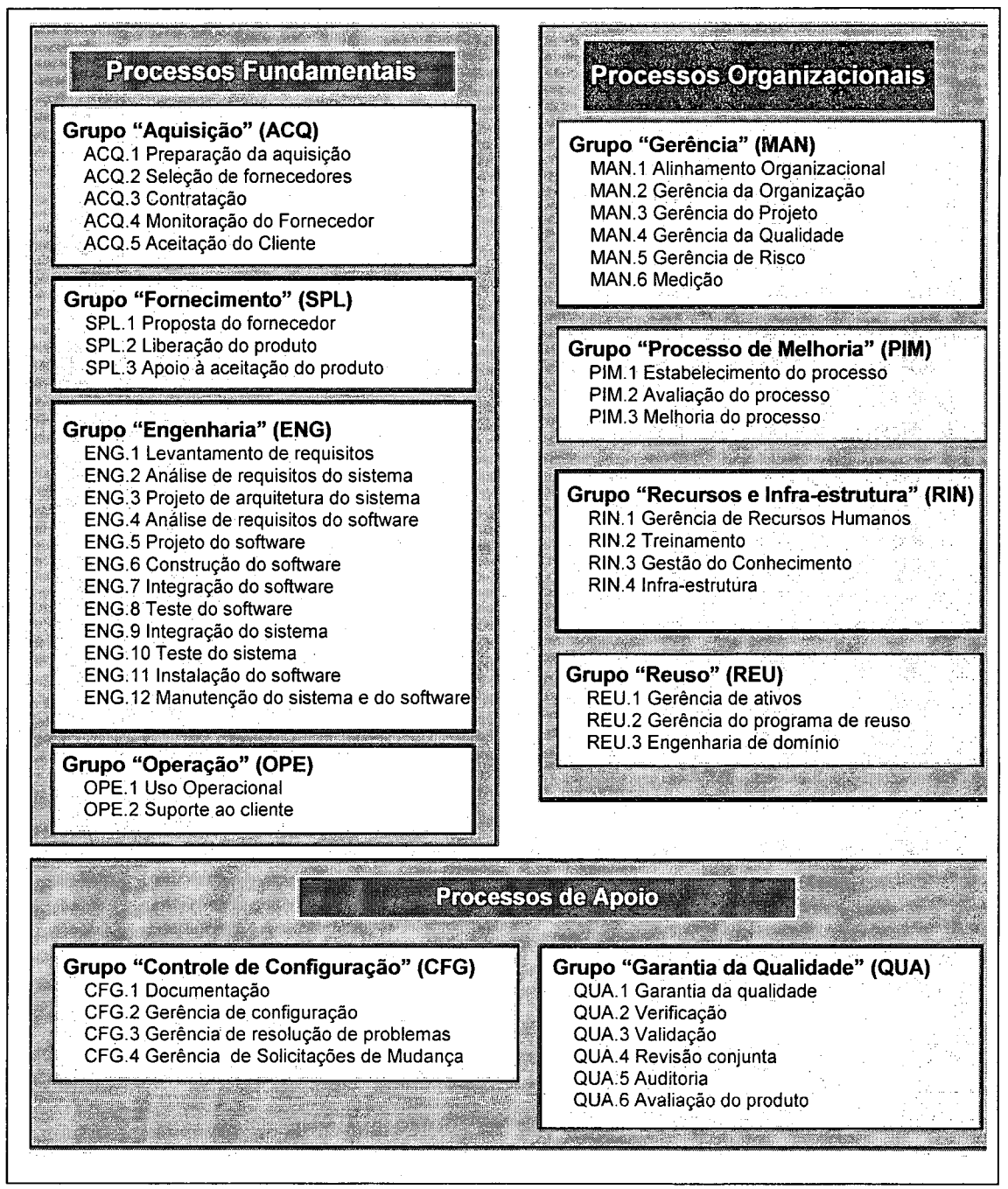

Figura 3.2 – Arquitetura da ISO 12207 com FDAM-1 e FDAM-2

3.4.2. Medição e análise na ISO/IEC 12207

Na ISO 12207, a medição é descrita como uma atividade pertencente ao processo de gerenciamento. A FDAM-1 da 12207 também trouxe a definição das atividades e tarefas do processo de gerenciamento, em particular da atividade de medição.

Pela definição da presente na ISO 12207 FDAM-1, o propósito de medição é "coletar e analisar dados relativos aos produtos desenvolvidos e processos implementados dentro da organização e de seus projetos, dar apoio à uma gerência efetiva dos processos e demonstrar de maneira objetiva a qualidade dos produtos".

São definidos os seguintes resultados esperados, como resultado da implementação com sucesso da medição:

- Compromisso Organizacional é estabelecido e mantido para implementar o processo de medição;
- As necessidades de informação sobre medição, demandadas pela organização e pelos processos gerenciais, são identificadas;
- Um conjunto adequado de métricas, orientado pelas necessidades de informação e objetivos de medição, é identificado e/ou desenvolvido;
- As atividades de medição são identificadas e executadas;
- Os dados requeridos são coletados, armazenados e analisados, e os resultados são interpretados;
- As informações produzidas são usadas para apoiar decisões e para fornecer uma base objetiva para comunicação aos interessados (*stakeholders*);
- O processo de medição e as métricas são avaliados e o resultados comunicados ao responsável pelo processo.

A atividade de medição consiste das tarefas apresentadas na tabela 3.4 .

Tabela 3.4 – Tarefas da atividade de medição da ISO/IEC 12207 FDAM-1

Tarefa	Resultados esperados
1. O gerente deverá estabelecer e manter o compromisso de medição. Garantir que todos os recursos, pessoal e pré-requisitos para o processo de medição sejam satisfeitos.	Compromisso da gerência em apoiar o processo de medição, e que as pessoas competentes na área nessa área sejam identificadas e tenham responsabilidades atribuídos no processo de medição, e que os recursos sejam colocados à disposição para o planejamento da medição e execução do processo.

2. O gerente deverá planejar o processo de medição. Desenvolver um plano detalhados para iniciar, guiar, monitorar e avaliar a coleta, análise, interpretação e armazenamento dos dados.	Definição do planejamento das necessidades de informação específicas da unidade; aquisição e liberação para uso de quaisquer tecnologias de apoio necessárias.
3. O gerente deverá executar a medição de acordo com o planejado. Produzir informações e realizar as medições de acordo com o que foi estabelecido no planejamento.	Dados são coletados, dados são armazenados de forma adequada à posterior recuperação e análise, a informação demanda é produzida e comunicada à unidade organizacional; medidas sobre o desempenho do processo de medição são coletadas.
4. O gerente deverá avaliar a medição. Avaliar as métricas e as atividades de medição, armazenar lições aprendidas dessa avaliação na "base de experiências de medição".	Avaliação das métricas e das atividades de medição de acordo com critérios específicos; armazenamento das lições aprendidas na "base de experiências de medição".

3.5. ISO/IEC 15939:2001 – Information Technology – Software Engineering – Software Measurement Process

A Norma Internacional ISO 15939:2001 define um processo de medição de software, descrito através de um modelo que especifica as atividades e tarefas necessárias para se identificar, definir, selecionar e melhorar a medição de software dentro da estrutura de um projeto ou organização, auxiliando a especificar, de forma adequada, quais são as necessidades de medição, como as métricas e a análise devem ser aplicadas e como determinar se os resultados da análise são válidos.

A norma ISO 15939:2001 não traz um catálogo de métricas de software, nem fornece qualquer conjunto de métricas recomendado para aplicação em projetos de software. No entanto, ela identifica um processo que apóia a definição de um conjunto de métricas adequado às necessidades de informação específicas. Ela também fornece definições para termos de medição que são comumente utilizados na Indústria de Software.

A ISO 15939:2001 pode ser utilizada por fornecedores e adquirentes de software, incluindo pessoal envolvido na realização de tarefas gerenciais, técnicas e de gestão de qualidade, no desenvolvimento, manutenção, integração e suporte de produtos de software, assim como compradores e usuários. Exemplos de sua utilização incluem:

- Fornecedor: Implementação de um processo de medição de software visando às necessidades de informação específicas de sua organização e de seus projetos.

- Adquirente: Avaliar a conformidade do processo de medição de software de seu fornecedor com essa norma.
- Fornecedor e adquirente: Estabelecer um contrato como método para definição das informações de medição de processo e produto que devem ser trocadas entre eles.

A norma contém um conjunto de atividades e tarefas que consistem num processo de medição de software que visa atender de um modo mais amplo as necessidades das organizações e projetos de software, podendo ser customizada. O processo de customização consiste em modificar as descrições não-normativas de tarefas, de modo a atingir os objetivos e resultados do processo de medição de software específicos das organizações e de seus projetos em particular. Todas as cláusulas normativas devem ser satisfeitas, e novas atividades e tarefas não definidas na norma podem ser adicionadas.

Não são prescritas pela norma estruturas organizacionais e nem nome, formato ou conteúdo explícito da documentação que deve ser produzida ao longo do processo de medição, cabendo essas decisões a seus usuários, de acordo com a cultura e restrições da organização. É desejável que o processo de medição seja devidamente integrado ao sistema de qualidade da organização.

A norma traz termos e definições, com base no Vocabulário de Termos básicos e Gerais de Metrologia da ISO/IEC, necessários para um entendimento comum da norma, dentre os quais vale destacar os da tabela abaixo, mais diretamente relacionados a este trabalho.

Tabela 3.5 – Alguns termos e definições da ISO 15939:2001

Termo	Definição
Atributo	Propriedade ou característica de uma entidade que pode ser distinguida quantitativa ou qualitativamente por meios humanos ou automáticos.
Métricas básicas	Métrica definida em termos de um atributo e dos métodos para quantificá-lo. Uma métrica básica é funcionalmente independente de outras métricas.
Dados	Coleção de valores atribuídos a métricas básicas, métricas derivadas e/ou indicadores.
Métrica derivada	Métrica que é definida como uma função de dois ou mais valores de métricas básicas.

Entidade	Objeto que se pretende caracterizar através da medição de seus atributos. Pode ser um processo, produto, projeto ou recurso.
Indicador	Métrica que fornece uma estimativa ou avaliação de atributos específicos derivados de um modelo relativo às necessidades de informação definidas.
Valor do indicador	Resultado numérico ou categórico atribuído a um indicador.
Necessidade de informação	Informação necessária para gerenciar objetivos, metas, riscos e problemas.
Produto de informação	Um ou mais indicadores e suas respectivas interpretações, atendendo uma determinada necessidade de informação. Exemplo: a comparação das taxas de defeitos medida planejada, junto com uma avaliação que indique se a diferença constitui ou não um problema.
Medida	Variável à qual um valor é atribuído como resultado de uma medição.
Métricas	Utilizado para referenciar de um modo genérico métricas básicas, métricas derivadas e indicadores.
Medir	Fazer uma medição
Conceito mensurável	Relação abstrata entre atributos de entidades e necessidades de informação.
Métrica derivada	Métrica que é definida como uma função de dois ou mais valores de métricas básicas.
Medição	Conjunto de operações que têm por objetivo determinar o valor de uma medida
Função de medição	Algoritmo ou cálculo realizado para combinar duas ou mais métricas básicas.
Método de medição	Seqüência lógica de operações, descritas genericamente, utilizada para quantificar um atributo com relação a uma dada escala. O tipo de medição depende da natureza das operações utilizadas para quantificar um atributo, podendo ser: Subjetiva – quantificação envolvendo julgamento humano; Objetivo – Quantificação baseada em regras numéricas.
Procedimento de medição	Conjunto de operações, descritas detalhadamente, utilizadas na realização de uma determinada medição, de acordo um dado método.
Processo de medição	O processo de estabelecer, planejar, executar e avaliar medição de software de um projeto ou

	organização.
Modelo	Algoritmo ou cálculo combinando uma ou mais métricas básicas e/ou derivadas, envolvendo critérios de decisão.
Observação	Instância resultante da aplicação de um procedimento de medição, visando produzir um valor para uma métrica básica.
Escala	Conjunto ordenado de valores, contínuo ou discreto, ou conjunto de categorias às quais os atributos são mapeados. O tipo da escala depende da natureza da relação entre os valores da escala. A norma cita quatro tipos de escala: Nominal: os valores de medição são categóricos, sem definição de ordem entre as categorias. Ordinal: os valores de medição são classificados em ordem. Intervalo: os valores de medição mantêm distâncias iguais, correspondentes a quantidades iguais do atributo medido. Racional: os valores de medição mantém distâncias iguais correspondentes a iguais quantidades do atributo medido, possuindo um valor zero que corresponde à ausência do atributo.
Unidade de medição	Quantidade definida e adotada por convenção, com a qual outras quantidades do mesmo tipo são comparadas de modo a expressar sua magnitude com relação a essa quantidade definida.
Valor	Resultado numérico ou categórico atribuído a uma métrica básica, derivada ou indicador.

O objetivo do processo de medição de software definido pela norma é "coletar, analisar e reportar dados relativos aos produtos desenvolvidos e processos implementados dentro uma unidade organizacional; dar apoio a gerência efetiva dos processos; e demonstrar objetivamente a qualidade dos produtos, em alinhamento com a norma ISO 15504-2:1998". Como resultados da implementação bem sucedida do processo de medição espera-se que:

- O compromisso organizacional para medição seja estabelecido e mantido;
- As necessidades de informação dos processos técnicos e gerenciais sejam identificadas;
- Um conjunto apropriado de métricas, guiado pelas necessidades de informação, seja identificado ou desenvolvido;

- As atividades de medição sejam identificadas;
- As atividades de medição identificadas sejam planejadas;
- Os dados necessários sejam coletados, armazenados, analisados e que os resultados sejam interpretados;
- Os produtos de informação sejam utilizados para dar suporte à decisão e para fornecer base objetiva para comunicação;
- O processo de medição e as métricas sejam avaliadas; e
- Melhorias sejam comunicadas ao responsável pelo processo de medição.

A norma define as atividades e tarefas necessárias para implementar um processo de medição de software. Uma atividade é um conjunto de tarefas relacionadas que contribuem para se atingir objetivo e os resultados esperados do processo de medição. Uma tarefa é um segmento de trabalho bem definido. Cada atividade é composta de uma ou mais tarefas. A norma descreve um conjunto detalhado de tarefas para cada atividade porém não especifica os detalhes de como devem ser executadas as tarefas. Mantendo a especificação de atividades definida pela ISO 12207:1995, os critérios de entrada e saída para cada atividade não são definidos pela norma.

O processo de medição de software consiste de quatro atividades, conforme mostrado no modelo de processo da figura 3.2. As atividades são seqüenciadas em um ciclo iterativo, que propicia *feedback* e melhoria contínuos do processo de medição. Esse modelo de processo de medição é uma adaptação do ciclo PDCA (*Plan-Do-Check-Act*), normalmente utilizado como base para processos de melhoria de qualidade. Dentro de cada atividade, as tarefas também são iterativas.

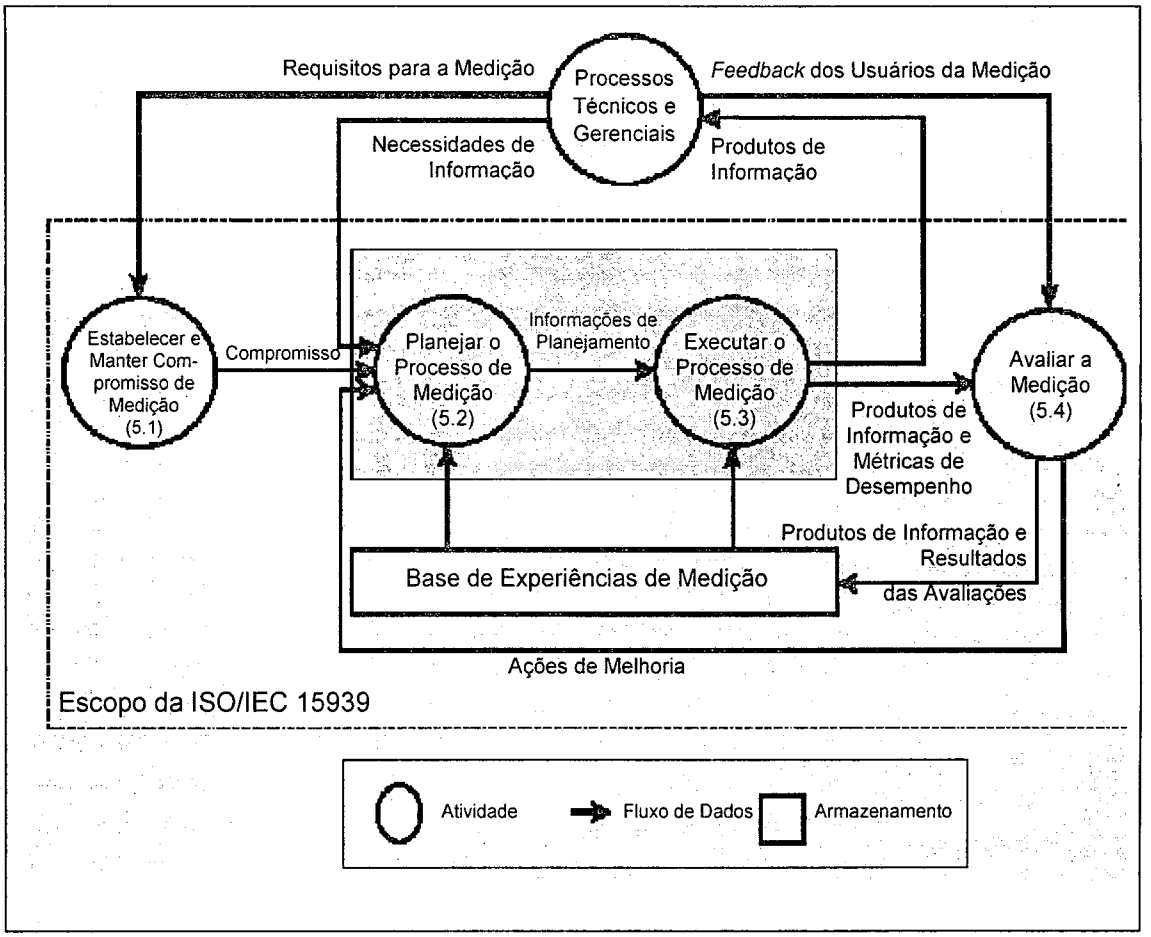

Figura 3.2 – Modelo de processo de medição de software da ISO 15939:2001

A norma não inclui descrição de processos técnicos e gerenciais, que podem ser encontrados na ISO/IEC 12207 ou no CMMI, por exemplo. No entanto, quando da utilização da norma, deve-se levar em conta as interfaces desses processos com as atividades do processo de medição.

No modelo definido, duas atividades são consideradas como núcleo do processo de medição: o Planejamento do Processo de Medição e a Execução do Processo de Medição, focando nos objetivos principais dos usuários da medição. As outras duas atividades, Estabelecer e Manter Compromisso de Medição, e Avaliar a Medição, fornecem o alicerce para esse núcleo, e fornecem o feedback necessário a ele; são atividades que endereçam os objetivos do responsável pelo processo de medição. A figura 3.2 mostra que esse núcleo do processo de medição é guiado pelas necessidades de informação da organização. Para cada necessidade de informação, esse núcleo do processo produz um produto de informação que satisfaz à necessidade de informação. O produto de informação é comunicado para a organização, servindo como base para tomada de decisão.

A atividade de avaliação incluída no modelo de processo enfatiza a importância da avaliação e do *feedback* como componentes fundamentais do processo, e que devem levar a melhoria do processo de medição e das métricas. Devem ser avaliados – se possível quantitativamente – a qualidade do processo de medição e de seus resultados, assim como o valor que as métricas agregam à organização. Esse ciclo inclui uma Base de Experiências de Medição, destinada a capturar produtos de informação dos ciclos de iteração anteriores, avaliações desses produtos e avaliações prévias do processo de medição. Os artefatos (produtos de informação, dados históricos, lições aprendidas etc.) armazenados nessa base podem e devem ser reutilizados em interações futuras do processo de medição.

Os artefatos de trabalho produzidos durante a execução do processo de medição são descritos num anexo da norma, e mapeados às tarefas nas quais são produzidos. A estrutura de atividades e tarefas é mostrada na figura 3.3. Para cada tarefa, a norma descreve "normativas" (ações necessárias para satisfazer a tarefa).

O processo definido não prescreve o uso de uma base ou conjunto de métricas previamente definidas, mas a norma apresenta um anexo com critérios que podem ser utilizados para seleção das métricas.

1. Estabelecer e manter o compromisso de medição
    1.1. Aceitar os requisitos para medição
    1.2. Atribuir recursos
2. Planejar o processo de medição
    2.1. Caracterizar a unidade organizacional
    2.2. Identificar as necessidades de informação
    2.3. Selecionar as métricas
    2.4. Definir os procedimentos para coleta, análise e divulgação dos dados
    2.5. Definir os critérios para avaliação dos produtos de informação e do processo de medição
    2.6. Revisar, aprovar e prover recursos para as tarefas de medição
    2.7. Obter e implementar tecnologias de apoio.
3. Executar o processo de medição
    3.1. Integrar os procedimentos
    3.2. Coletar os dados
    3.3. Analisar os dados e desenvolver os produtos de informação
    3.4. Comunicar os resultados
4. Avaliar a medição
    4.1. Avaliação os produtos de informação e o processo de medição
    4.2. Identificar melhorias em potencial

Figura 3.3 – Estrutura de Atividades e Tarefas do Modelo de Processo da ISO 15939

As atividades são descritas na ordem em que normalmente são executadas. No entanto, a interação de uma atividade para a anterior freqüentemente ocorre. A ordem na qual as tarefas para cada atividade são apresentadas não implica necessariamente na ordem de implementação das mesmas. Para cada tarefa, um ou mais requisitos normativos para a implementação da tarefa são definidos. Para muitas tarefas há também um excelente guia informativo, muitas vezes com exemplos, que ajuda na interpretação dos requisitos normativos e na implementação das tarefas na prática.

Durante a implementação de um processo de medição que atenda a norma, a unidade organizacional deve realizar as atividades relacionadas. O processo é iniciado a partir das demandas de "Necessidades de Medição" presentes nos processos técnicos e gerenciais definidos e utilizados pela organização.

3.5.1. Modelo Informação-Medição

A norma traz um modelo (figura 3.4) que mostra como as necessidades de informação são ligadas às entidades relevantes e aos atributos de interesse que devem ser medidos. As entidades incluem processos, produtos, projetos e recursos. O modelo descreve como os atributos relevantes são quantificados e transformados nos indicadores que fornecem a base para a tomada de decisão.

A seleção ou definição de métricas apropriadas para satisfazer as necessidades de medição começa com um "conceito mensurável": uma idéia de quais atributos mensuráveis têm relação com uma dada necessidade de informação, e como são relacionados. Aquele que planeja a medição deve definir os conceitos de medição que ligam esses atributos a uma determinada necessidade de informação.

O modelo identifica os termos e conceitos básicos, e ajuda a definir o que se precisa especificar durante o planejamento, execução e avaliação da medição.

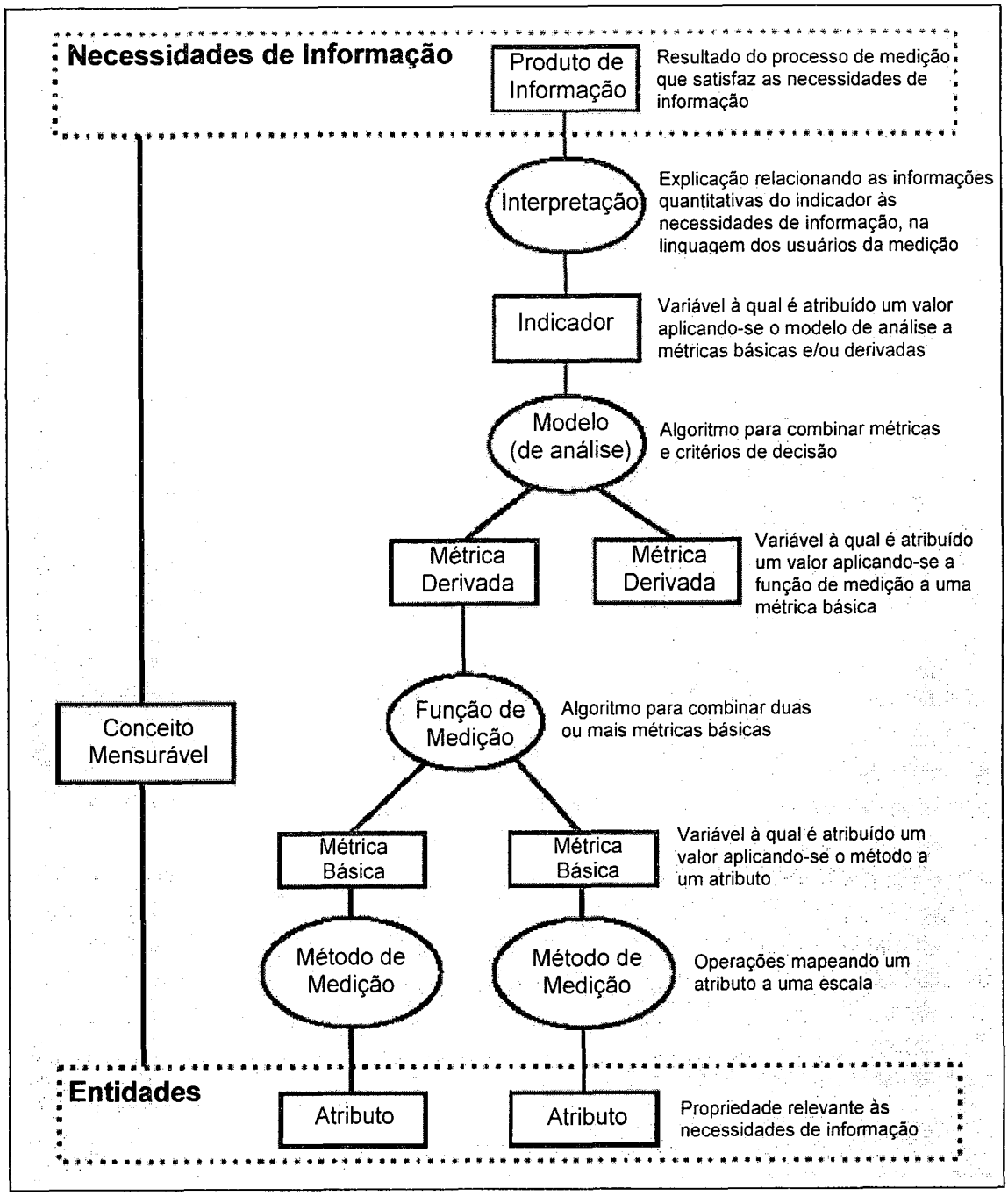

Figura 3.4 – Modelo Informação-Medição da ISO 15939

O modelo define três tipos de métricas: métricas básicas, métricas derivadas e indicadores. O conteúdo de informação das métricas aumenta na medida em que elas se aproximam, no modelo, da necessidade de informação. A tabela 3.6 descreve os principais elementos desse modelo.

Tabela 3.6 – Componentes do modelo informação-medição da ISO 15939:2001

Componente do Modelo	Descrição
1. Entidade	Objeto que se pretende caracterizar através da medição de seus atributos. Pode ser um processo, produto, projeto ou recurso. Uma entidade pode possuir uma ou mais propriedades que sejam importantes para se satisfazer às necessidades de medição.
2. Atributo	Propriedade ou característica de uma entidade que pode ser distinguida quantitativa ou qualitativamente, por meios humanos ou automáticos. Uma entidade pode ter vários atributos, e apenas alguns serem de interesse para a medição. O primeiro passo na instanciação do modelo é selecionar os atributos mais relevantes para as necessidades de informação dos usuários da medição. Um determinado atributo pode ser incorporado em vários conceitos de medição, dando apoio a diferentes necessidades de informação.
3. Métrica básica	Métrica definida em termos de um atributo e dos métodos para quantificá-lo. Uma métrica básica é funcionalmente independente de outras métricas, e captura informação acerca de um único atributo. O processo de coleta de dados consiste em atribuir valores às métricas básicas.
3.1. Método de medição	Seqüência lógica de operações, descritas genericamente, utilizada para quantificar um atributo com relação a uma dada escala. As operações podem envolver atividades como contagem de ocorrências ou observação do tempo decorrido. O mesmo método de medição pode ser aplicado a múltiplos atributos. Porém, cada diferente combinação de um atributo com um método de medição produz uma métrica básica diferente. Alguns métodos de medição podem ser implementados de diferentes maneiras. O conceito de "procedimento de medição" descreve uma implementação específica de um dado método de medição em um determinado contexto organizacional.
3.1.1. Tipo de método de medição	O tipo de medição depende da natureza das operações utilizadas para quantificar um atributo, podendo ser: Subjetiva – quantificação envolvendo julgamento humano; Objetivo – Quantificação baseada em regras numéricas, que podem ser implementadas por meios humanos ou automatizados.
3.1.2. Escala	Conjunto ordenado de valores, contínuo ou discreto, ou conjunto de categorias às quais os atributos são mapeados. O método de medição faz o mapeamento da magnitude do atributo medido para um valor em uma escala. Uma unidade de medição é normalmente associada a uma escala.
3.1.2.1. Tipo de Escala	O tipo da escala depende da natureza da relação entre os valores da escala. A norma cita quatro tipos de escala: Nominal: os valores de medição são categóricos, sem definição de ordem entre as categorias. Ordinal: os valores de medição são ordenados. Intervalo: os valores de medição mantém distâncias iguais, correspondentes a quantidades iguais do atributo medido.

	Racional: os valores de medição mantém distâncias iguais correspondentes a iguais quantidades do atributo medido, possuindo um valor zero que corresponde à ausência do atributo. O método de medição em geral afeta o tipo de escala que pode ser utilizada de maneira confiável com um determinado atributo. Métodos subjetivos de medição, por exemplo, normalmente só suportam escalas ordinais ou nominais.
3.1.2.2. Unidade de medição	Quantidade definida e adotada por convenção, com a qual outras quantidades do mesmo tipo são comparadas de modo a expressar sua magnitude com relação a essa quantidade definida. Somente quantidades expressas na mesma unidade podem ser diretamente comparadas.
4. Métrica Derivada	Métrica que é definida como uma função de dois ou mais valores de métricas básicas. As métricas derivadas capturam informação sobre mais de um atributo ou de um mesmo atributo de várias entidades. O modelo considera que transformações simples de métricas básicas (como por exemplo a aplicação de raiz quadrada) não agregam informação, não produzindo, portanto, métricas derivadas. A normalização de dados envolve freqüentemente a conversão de métricas básicas em métricas derivadas que podem ser utilizadas para comparação de diferentes entidades.
4.1. Função de Medição	Algoritmo ou cálculo realizado para combinar duas ou mais métricas básicas. A escala e a unidade da métrica derivada depende das escalas e das unidades das métricas básicas a partir das quais é composta, e também da forma como são combinadas pela função.
5. Indicador	Métrica que fornece uma estimativa ou avaliação de atributos específicos derivados de um modelo relativo às necessidades de informação definidas. Indicadores são a base para a análise e tomada de decisão, e são o que deveria ser apresentado aos usuários da medição, contendo um nível de informação maior do que uma métrica. A medição pode ser baseada em informações imperfeitas; assim, quantificar o grau de incerteza, acurácia e importância dos indicadores é essencial para a apresentação de seus valores.
5.1. Modelo	Algoritmo ou cálculo combinando uma ou mais métricas básicas e/ou derivadas, envolvendo critérios de decisão. É baseado na compreensão ou no que foi estabelecido sobre as relações esperadas entre as métricas que o compõe e/ou sobre seu comportamento ao longo do tempo. Os modelos produzem estimativas ou avaliações relevantes para as necessidades de informação definidas. A escala e o método de medição afetam a escolha das técnicas de análise ou dos modelos utilizados para produzir indicadores.
5.1.1. Critérios de decisão	São limites ou metas numéricas utilizados para determinar a necessidade de ação ou futura investigação, ou para descrever o nível de confiança em um dado resultado. Os critérios de decisão ajudam a interpretar os resultados da medição, devendo ser calculados ou baseados na compreensão conceitual do comportamento esperado. Eles podem ser derivados a partir de dados históricos, planos e heurísticas, ou computados como limites estatísticos de controle ou limites estatísticos de confiança.

6. Conceito Mensurável	Relação abstrata entre atributos de entidades e necessidades de informação. Por exemplo, para a necessidade de informação "comparar a produtividade no desenvolvimento de software de um projeto contra um valor esperado", um conceito mensurável poderia ser "produtividade de desenvolvimento de software", e para ser avaliado seria necessário medir o tamanho dos produtos de software e a quantidade de recursos necessários para criar esses produtos. Exemplos de outros conceitos mensuráveis: qualidade, risco, desempenho, capacidade, maturidade.

3.6. O Modelo de Referência MPS.BR

O projeto MPS.BR – Melhoria de Processo do Software Brasileiro – vem sendo desenvolvido, desde dezembro de 2003, pelas seguintes instituições: SOFTEX (Sociedade para Promoção da Excelência do Software Brasileiro - coordenadora do projeto); Núcleo SOFTEX Campinas - Sociedade Núcleo SOFTEX 2000; RIOSOFT - Sociedade Núcleo de Apoio à Produção e Exportação de Software do Rio de Janeiro; COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação da Universidade Federal do Rio de Janeiro, com participação; CESAR - Centro de Estudos e Sistemas Avançados de Recife; CenPRA - Centro de Pesquisas Renato Archer e CELEPAR - Companhia de Informática do Paraná.

O ponto de partida para definição do MPS.BR foi a análise da realidade das empresas brasileiras, a norma ISO/IEC 12207, a série de normas ISO/IEC 15504 (2003) e o modelo CMMI (*Capability Maturity Model Integration*). No MRMPS.BR (Modelo de Referência para a Melhoria de Processo do Software Brasileiro), a norma internacional ISO/IEC FDAM 12207 (2002) é o *framework* para a definição de processos.

O MPS.BR baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de produtos de software e serviços correlatos. Dentro desse contexto, o MPS.BR possui três componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS).

O MPS.BR está descrito através de documentos em formato de guias:

- Guia Geral: contém a descrição geral do MPS.BR e detalha o Modelo de Referência (MR-MPS), seus componentes e as definições comuns necessárias para seu entendimento e aplicação.

- Guia de Aquisição: contém recomendações para a condução de compras de software e serviços correlatos. Foi descrito como forma de apoiar as instituições que queiram adquirir produtos de software e serviços correlatos apoiando-se no MR-MPS.
- Guia de Avaliação: contém a descrição do processo de avaliação, os requisitos para o avaliador, os requisitos para a avaliação, o método e os formulários para apoiar a avaliação (ainda não publicado).

O MPS.BR tem como objetivo definir um modelo de melhoria e avaliação de processo de software, preferencialmente para as micro, pequenas e médias empresas, de forma a atender as suas necessidades de negócio e ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software. Este é o motivo pelo qual ele está aderente a modelos e normas internacionais. O MPS.BR também define regras para sua implementação e avaliação, dando sustentação e garantia de que o MPS.BR está sendo empregado de forma coerente com as suas definições.

A base técnica utilizada para a construção do MPS.BR é composta pelas normas NBR ISO/IEC 12207 e a ISO/IEC 15504 e seu Modelo de Avaliação de Processo de Software ISO/IEC 15504-5, portanto o modelo está totalmente aderente a essas normas. O MPS.BR também cobre o conteúdo do CMMI-SE/SW, através da inclusão de processos e resultados de processos em relação aos processos da Norma NBR ISO/IEC 12207, conforme a figura 3.5.

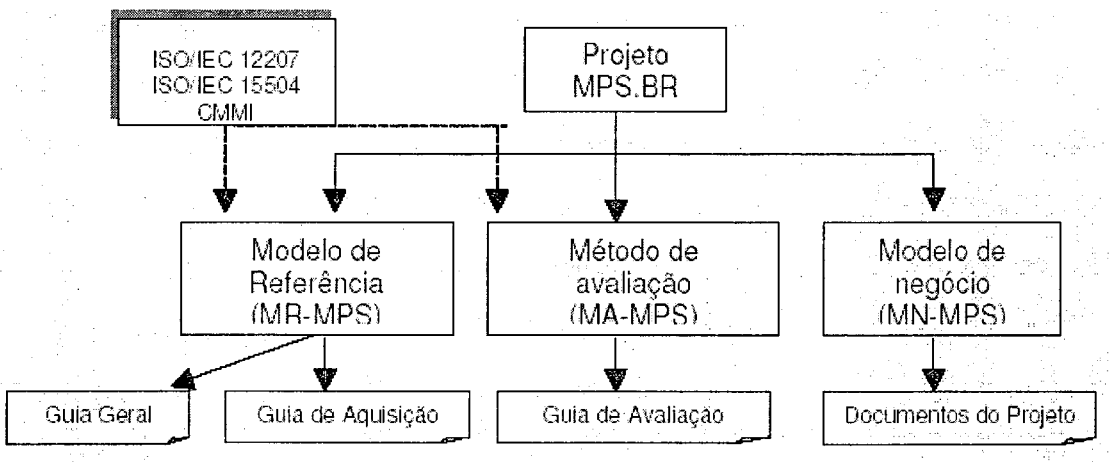

Figura 3.5 – MPS-BR

O Modelo de Referência de Melhoria de Processo de Software (MR-MPS) contém os requisitos que as organizações deverão atender para estar em conformidade com o MR-MPS. Ele contém as definições dos níveis de maturidade, da capacidade de

processos e dos processos em si, tendo sido baseado nas normas NBR ISO/IEC 12207 e suas emendas 1 e 2, ISO/IEC 15504, e adequado ao CMMI/SW.

3.6.1. Descrição do MR-MPS.BR

O Modelo de Referência MR-MPS.BR define níveis de maturidade que são uma combinação entre processos e capacidade de processos, conforme a estrutura apresentada na figura 3.6.

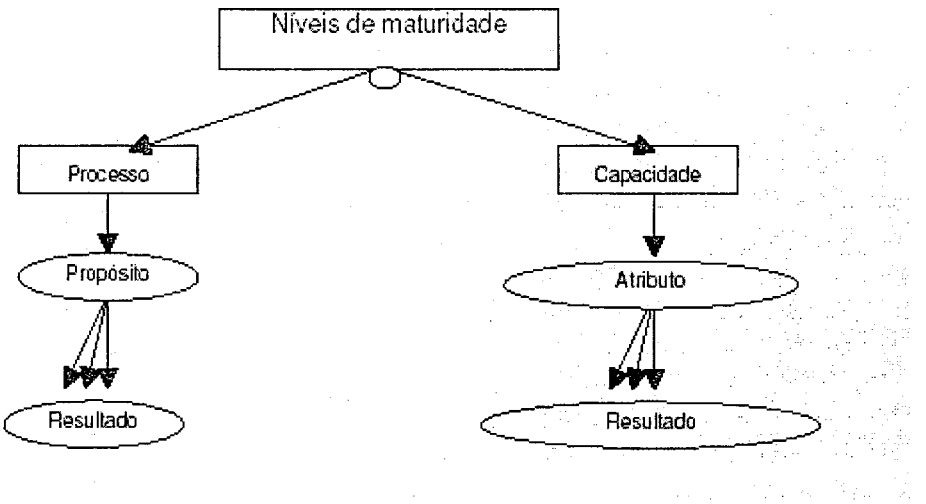

Figura 3.6 – Estrutura do MR-MPS.BR

A definição dos processos (figura 3.7) segue a forma apresentada na Emenda 1 da ISO/IEC 12207, declarando o propósito e os resultados de sua execução. Isso permite avaliar e atribuir graus de efetividade na execução dos processos. As atividades e tarefas necessárias para atender ao propósito e aos resultados não são definidas nesse guia devendo ficar a cargo dos usuários do MR-MPS.BR. A capacidade de um processo é sua habilidade para alcançar os objetivos de negócio, atuais e futuros. Ela está relacionada com o atendimento dos atributos do processo associados aos processos propriamente dito de cada nível de maturidade.

3.6.2 Níveis de maturidade

Os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria de implementação de processos na organização. O nível de maturidade em que se encontra uma organização permite prever seu desempenho futuro em uma ou mais disciplinas. O MR-MPS.BR define sete níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado).

A escala de maturidade se inicia no nível G e progride até o nível A. Para cada um destes sete níveis de maturidade foi atribuído um perfil de processos e de capacidade de processos que indicam onde a organização tem que colocar esforço para melhoria de forma a atender os objetivos de negócio, conforme a tabela 3.7.

O progresso e o atendimento do nível de maturidade se obtêm quando são atendidos todos os resultados e propósito do processo, e os atributos de processo relacionados àquele nível e aos anteriores. A divisão em estágios, embora baseada nos níveis de maturidade do CMMI SW, tem graduação diferente, com o objetivo de possibilitar uma implementação e avaliação mais gradual e adequada às pequenas e médias empresas. A possibilidade de se realizar avaliações considerando mais níveis permite visibilidade dos resultados de melhoria de processos com prazos mais curtos.

Figura 3.7 – Processos do MR-MPS

Na tabela 3.7 a seguir estão apresentados os mapeamentos das áreas de processo do CMMI para o MRMPS.BR, indicando, ainda, em que nível de maturidade do MPS.BR se encontram.

Tabela 3.7 – Mapeamento do CMMI para MR mps.BR (ROCHA *et al.*, 2005)

MPS.BR		CMMI-SE/SW
Nível	**Processo**	**Áreas de Processo**
A (mais alto)	Inovação e Implantação na Organização	Inovação e Implantação na Organização (CMMI-5)
	Análise de Causas e Resolução Análise de Causas e Resolução	Análise de Causas e Resolução (CMMI-5)
B	Desempenho do Processo Organizacional	Desempenho do Processo Organizacional (CMMI-4)
	Gerência Quantitativa do Processo	Gerência Quantitativa do Processo (CMMI-4)
C	Análise de Decisão e Resolução	Análise de Decisão e Resolução (CMMI-3)
		Gerência Integrada de Fornecedores (CMMI-3)
	Gerência de Riscos	Gerência de Riscos (CMMI-3)
D	Desenvolvimento de Requisitos	Desenvolvimento de Requisitos (CMMI-3)
	Solução Técnica	Solução Técnica (CMMI-3)
	Integração do Produto	Integração do Produto (CMMI-3)
	Instalação do Produto	
	Liberação do Produto	
	Verificação	Verificação (CMMI-3)
	Validação	Validação (CMMI-3)
E	Treinamento	Treinamento Organizacional (CMMI-3)
	Avaliação e Melhoria do Processo Organizacional	Foco no Processo Organizacional (CMMI-3)
	Definição do Processo Organizacional	Definição do Processo Organizacional (CMMI-3)
	Adaptação do Projeto para Gerência de Projetos	Gerência de Produto Integrado (CMMI-3)
F	Medição	Medição e Análise (CMMI-2)
	Gerência de Configuração	Gerência de Configuração (CMMI-2)
	Aquisição	Gerência de Acordo com Fornecedores (CMMI-2)
	Garantia da Qualidade	Garantia da Qualidade do Processo e do Produto (CMMI-2)
G	Gerência de Requisitos	Gerência de Requisitos (CMMI-2)
	Gerência de Projeto	Planejamento de Projeto (CMMI-2)
		Monitoração e Controle de Projeto (CMMI-2)

3.6.3. Medição no MR MPS.BR

No MR MPS.BR, o processo de Medição é definido no nível F (gerenciado). O processo é referenciado como "MED" e tem como propósito "coletar e analisar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos, de forma a apoiar os objetivos organizacionais".

Os resultados esperados a partir da implementação e do uso do processo são descritos na tabela 3.8.

Tabela 3.8 – resultados esperados do processo de Medição do MR-MPS

Resultado	Descrição
MED 1	Objetivos e atividades de medição são estabelecidos a partir das necessidades de informação e objetivos da organização
MED 2	Um conjunto adequado de medidas, orientado pelas necessidades de informação e objetivos de medição, é identificado e/ou desenvolvido, priorizado, documentado, revisado e atualizado
MED 3	As atividades de medição (coleta e armazenamento) são especificadas, incluindo-se métodos e ferramentas
MED 4	As atividades de análise são especificadas, incluindo-se métodos e Ferramentas
MED 5	Os dados requeridos são coletados e analisados
MED 6	Os dados e os resultados são armazenados
MED 7	As informações produzidas são usadas para apoiar decisões e para fornecer uma base objetiva para comunicação aos interessados (*stakeholders*)

3.7. ISO/IEC 9126

A especificação detalhada e a avaliação da qualidade de produtos de software é um fator chave para garantir a qualidade adequada. Isso pode ser obtido através da definição apropriada das características de qualidade levando em conta o objetivo do uso dos produtos de software. É importante que cada característica de qualidade relevante nos produtos de software seja especificada e avaliada, utilizando, sempre que possível, métricas validadas.

A ISO/IEC 9126 (2000) define características de qualidade e métricas associadas que podem ser utilizadas na especificação de requisitos funcionais e não funcionais. É definido um modelo de qualidade dividido em duas partes: a) qualidade interna e qualidade externa; b) qualidade em uso.

A primeira parte do modelo especifica seis características para qualidade interna e externa que são posteriormente subdivididas em subcaracterísticas, manifestadas externamente quando o software é utilizado como parte de um sistema, e que são resultado dos atributos internos do software. A segunda parte do modelo especifica quatro características de qualidade do software em uso, mas não desenvolve um modelo de subcaracterísticas. As características de qualidade definidas são aplicáveis a qualquer tipo de software e estabelecem um *framework* para especificação de requisitos de qualidade, fornecendo, também, uma terminologia consistente para qualidade de produtos de software.

A norma traz um anexo com recomendações e requisitos para métricas de produtos de software e para métricas do software em uso. As partes 2, 3, 4 da ISO/IEC 9126 trazem diversos exemplos de métricas definidas, que podem ser aplicadas quando da especificação e da avaliação de requisitos de qualidade incluindo produtos intermediários.

A ISO/IEC 9126-1 permite que a qualidade dos produtos de software seja especificada e avaliada a partir de diferentes perspectivas, associadas a aquisição, especificação de requisitos, desenvolvimento, uso, avaliação, suporte, manutenção garantia de qualidade e auditoria do software. Exemplos de uso do modelo de qualidade definido incluem:

- Validação da completude da definição de requisitos;
- Identificação dos requisitos do software;
- Identificação dos objetivos para os testes do software;
- Identificação dos critérios para garantia de qualidade;
- Identificação dos critérios de aceitação dos produtos.

A avaliação dos produtos de software para satisfazer as necessidades de qualidade é um dos processos no ciclo de desenvolvimento do software. A qualidade dos produtos pode ser avaliada através da medição de atributos internos ou da medição de atributos externos (normalmente medindo-se o comportamento do código quando executado) ou medindo-se a qualidade de atributos referentes ao uso do software. O objetivo é que o produto atenda às necessidades específicas de um determinado contexto de uso (figura 3.8). Atributos internos apropriados são um pré-requisito para se atingir o comportamento externo desejado, assim como o comportamento externo apropriado é pré-requisito para atingir a qualidade do software em uso.

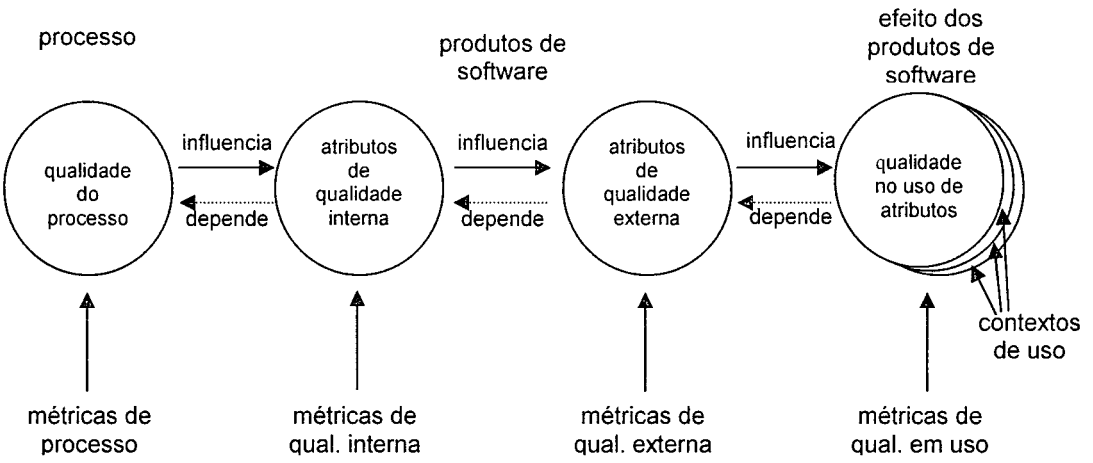

Figura 3.8 – Qualidade no ciclo de vida do software (ISO/IEC 9126)

Os itens de qualidade podem ser avaliados através de medição direta ou, indiretamente, através da medição de suas conseqüências. O processo pode ser avaliado indiretamente, medindo-se e avaliando seus produtos, ou um produto pode ser avaliado indiretamente, medindo o desempenho de uma tarefa no mesmo por um usuário (utilizando métricas da qualidade do software em uso).

O produto de software completo pode ser avaliado através de métricas externas escolhidas que descrevem sua interação com o ambiente e que são avaliadas observando-se o software em operação. A qualidade do software em uso pode ser medida confrontando-se o que ele realiza com as necessidades de seus usuários para atingir objetivos específicos de eficácia, produtividade, segurança e satisfação.

Nos estágios iniciais do desenvolvimento somente recursos e processos podem ser medidos. Quando produtos intermediários (especificações, código fonte etc.) tornam-se disponíveis eles podem ser avaliados através de métricas internas – que podem ser utilizadas para predizer valores de métricas externas. O modelo de qualidade para atributos internos e externos é mostrado na figura 3.9.

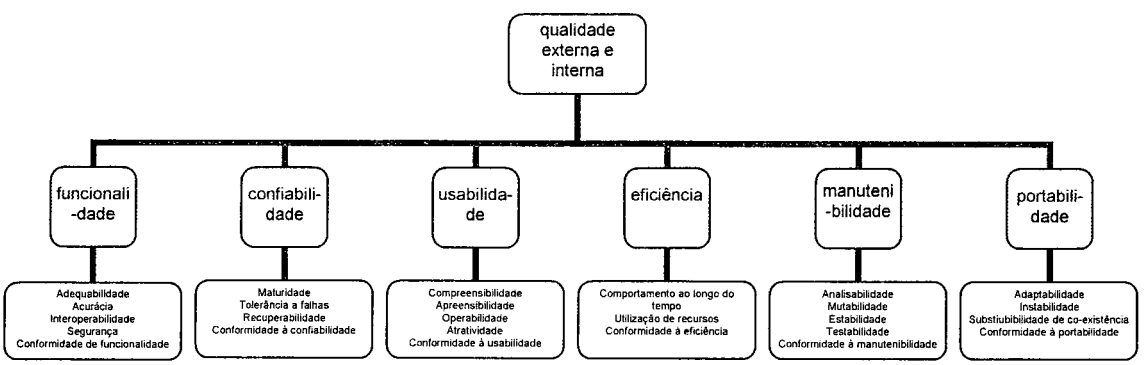

Figura 3.9 – Modelo de Qualidade Interna e Externa na ISO/IEC 9126

3.8. Outras normas e padrões relacionados à medição de software

3.8.1. Norma IEEE 982.1-1988 (*Standard Dictionary of Measures to Produce Reliable Software)*

A norma IEEE 982.1-1988 (1988) provê um conjunto de métricas para avaliação da confiabilidade[3] de software, que podem ser aplicadas tanto aos produtos de software quanto aos processos de desenvolvimento e suporte. Seu objetivo é fornecer uma definição comum e consistente de um conjunto de métricas que possam ser utilizadas desde o início do processo de desenvolvimento, e que sirvam como indicadores da confiabilidade do produto, de modo a apoiar gerentes e desenvolvedores. As métricas são apresentadas com indicação das condições próprias para uso e os métodos para computação dos resultados, através de um *framework* visando estabelecer uma linguagem comum entre os usuários.

A norma é aqui detalhada por ser um das poucas a efetivamente propor um conjunto de métricas, classificadas e definidas através de *templates*. Algumas de suas métricas foram utilizadas na definição do conjunto inicial de métricas proposto para a base de métricas deste trabalho.

Enfatizando a necessidade da avaliação da confiabilidade desde o início, a norma sugere métodos utilizando a medição para melhoria da confiabilidade do produto, e se propõe a fornecer meios para avaliação contínua do processo de desenvolvimento e dos produtos, em cada fase do ciclo de vida. Com relação a processos, a norma fornece métricas que podem ser aplicadas ao longo do ciclo de vida, visando à auto-avaliação e à melhoria da confiabilidade. Com relação a produtos, ela se propõe a oferecer meios de aumentar a confiabilidade do software em seu real ambiente de uso, durante as fases de operação e suporte.

A maioria das métricas utilizam os conceitos de erro, defeito e falha (*error, fault e failure*, respectivamente, no original) como unidades de dados primitivas, cujas definições estão descritas a seguir.

Erro (*error*)**:** ação humana que resulta em software contendo um defeito, incluindo, por exemplo, omissão ou má interpretação dos requisitos especificados pelos usuários, tradução incorreta e omissão de requisitos na especificação do projeto.

---

[3] Confiabilidade: probabilidade que o software, por um dado intervalo de tempo, em dadas condições, não causará falha de um sistema.

Defeito (*fault*): (1) Condição acidental que leva uma unidade funcional a falhar na execução de sua função (ISO; ANSI/IEEE Std 729-1983); (2) manifestação de um erro em um software. Um defeito, quando encontrado, pode causar uma falha. Sinônimo de *bug* (ANSI/IEEE Std 729-1983).

Falha (*failure*): (1) O término da habilidade de uma unidade funcional de executar sua função (ISO; ANSI/IEEE Std 729-1983); (2) Evento no qual um sistema ou componente do sistema não realiza uma função desejada dentro dos limites especificados. Uma falha pode ser produzida quando ocorre um defeito (ANSI/IEEE Std 729-1983).

Para melhor compreensão da norma, são descritos a seguir os conceitos que ela define para métrica, primitiva e gerenciamento da confiabilidade de software.

Métrica: avaliação quantitativa do grau em que um produto ou processo de software possui um dado atributo (característica).

Primitiva: Dados relacionados ao desenvolvimento ou uso do software e que são utilizados para gerar métricas ou descrições quantitativas do software. Primitivas são diretamente mensuráveis ou contáveis, ou são dadas através de valores constantes ou condições para uma determinada métrica. Exemplos incluem: erros, defeitos, falhas, tempo decorrido, intervalo de tempo, data, número de linhas de código com comandos.

Gerenciamento de confiabilidade de software: processo de otimização da confiabilidade do software através de um programa que enfatiza a prevenção de erros, detecção e remoção de defeitos, e o uso de medição para maximizar a confiabilidade perante as restrições do projeto, como recursos, cronograma e desempenho.

As 39 métricas definidas pela norma são divididas em duas categorias funcionais: produto e processo. As métricas de produto são aplicáveis a artefatos de software produzidos e são divididas em 6 subcategorias, enquanto que as métricas de processo são aplicáveis a atividades de desenvolvimento, teste e manutenção, e são divididas em três subcategorias (tabela 3.9).

Tabela 3.9 – categorias de métricas IEEE 982.1-1988

Categorias de métricas de produto	
Erros, defeitos e falhas	Contagem dos defeitos com relação a causas humanas, *bugs* nos programas, anomalias de funcionamento no sistema.
Tempo médio para falha (*mean-time-to-failure*); Taxa de Defeitos	Métricas derivadas em função da contagem de defeitos e do intervalo de tempo em que são encontrados
Crescimento e projeção de confiabilidade	A avaliação de alterações no grau de ausência de defeitos no produto sob teste ou operação.
Defeitos restantes no produto	Para acompanhamento e projeção do número de defeitos, da confiabilidade e do esforço a se empregar para removê-los.
Completude e consistência	A avaliação da presença e da concordância da necessidade de todos os componentes do software.
Complexidade	Avaliação de fatores que podem trazer complicações em um sistema.
Categorias de métricas de processo (diretamente relacionadas a gerência de processos)	
Controle do gerenciamento	Avaliação da capacidade da gerência de guiar os processos de desenvolvimento e manutenção.
Cobertura	Avaliação da presença de todas as atividades necessárias para desenvolver e manter produtos de software.
Risco; Benefício; Avaliação de Custo	Avaliação de custo / benefício das ações.

Exemplos de métricas da norma incluem: densidade de defeitos, complexidade ciclomática, cobertura dos testes funcionais e tempo médio para falha (*mean-time-to-failure)*. Todas são definidas através de um *template* simples com os campos a seguir: nome da métrica; aplicação (indica para que e onde a métrica pode ser utilizada); primitivas (define os itens de dados necessários para cálculo da métrica); implementação (descrição da definição operacional, passos necessários para coleta das métricas e fórmula para cálculo). A tabela 3.10 contém um exemplo de métrica definida pela norma.

Tabela 3.10 – Exemplo de definição de métrica pela IEEE 982.1-1988

Nome	Densidade de defeitos
Aplicação	Pode ser utilizada para: 1) Prever os defeitos restantes através de comparação com a densidade de defeitos esperada. 2) Determinar se foram realizados testes suficientes, comparando com metas pré-determinadas de densidade de defeitos. 3) Estabelecer padrões de valores para densidade de defeitos, para comparação e predição.
Primitivas	• Estabelecer os níveis de severidade para falhas • *F:* número total de defeitos exclusivos encontrados em um dado intervalo de tempo, que resultam em falhas de um nível específico de severidade. • *K*SLOC: número de linhas de código-fonte, incluindo linhas executáveis e não executáveis com declaração de dados, contadas aos milhares.
Implementação	1) Estabelecer níveis de severidade, tipos de falha, tipos de defeitos. 2) Falhas podem incluir falhas de I/O e falhas do usuário. Falhas podem ser resultantes do projeto, codificação ou documentação, por exemplo. 3) Observar e registrar toda e qualquer falha. 4) Determinar o(s) defeito(s) do programa que provocaram a falha. Classificar os defeitos por tipo. Defeitos adicionais podem ser encontrados sem que ocorra falha, assim como um defeito pode levar a mais de uma falha. Como conseqüência, o número total defeitos pode ser maior que o número total de falhas; é necessário calcular a densidade de falhas e a densidade de defeitos separadamente. 5) Determinar o número total de milhares de linhas de código [executáveis e mais não-executáveis com declaração de dados (*K*SLOC)]. 6) Calcular a densidade de defeitos para cada nível de severidade como: *Fd* = *F*/*K*SLOC

3.8.2. Norma IEEE 1045-1992 (*Standard for Software Productivity Metrics*)

A norma IEEE 1045-1992 (1992) estabelece um framework para medição e registro de produtividade no desenvolvimento de software, com foco na definição de como medir a produtividade de software e como reportar os resultados. Sua definição de produtividade é a razão entre a saída produzida (*output)* e o quanto de entrada (*input*) é necessário para produzi-la.

De acordo com a norma, para que um indicador de produtividade seja realmente útil, os dados utilizados devem ser acurados e completos. Uma produtividade descrita, por exemplo, como "5000 linhas de código por ano" deixa muitas questões em aberto (o

que é uma linha de código? Quanto dura um ano em horas de trabalho? Que atividades foram incluídas? Que esforço foi computado?).

A interpretação da produtividade com base em um único número deixa muita dúvida sobre o processo que está sendo medido. Sem conhecimento do escopo e das características do processo medido, ou a precisão dos dados utilizados nos cálculos, os valores resultantes são não-conclusivos. Assim, a norma se propõe a definir de maneira precisa unidades de medida para uso em métricas, para que se possa medir a produtividade acuradamente. Nas situações onde não é possível uma definição precisa da medição, o padrão demanda que as descrições utilizadas para os processos e as medições sejam feitas em um formato (*template*) específico.

A norma descreve o processo para coleta de dados e cálculos para medição de produtividade de software e formaliza o formato de apresentação dos dados (resultados) sobre produtividade, de modo que sejam úteis para qualquer um que deseje melhorar o processo de software no escopo do projeto ou da organização. A norma traz anexos com *templates* para coleta de dados das métricas quantitativas nele definidas e também prescreve métricas para caracterização do processo de software, com a intenção de fornecer meios para melhorá-lo.

De acordo com a norma, medir produtividade de software é semelhante a medir outras formas de produtividade: mede-se a razão entre as unidades de saída pelas unidades de entrada. As entradas consistem no esforço demandado para produzir um produto de saída. Para software, as saídas tangíveis são o código-fonte e a documentação.

A norma padroniza a terminologia de métricas de produtividade de software para garantir compreensão dos dados de medição, tanto para código quanto para a documentação produzida, definindo um conjunto de unidades de medida para os produtos de saída e os esforços (entradas). O nível mais atômico de medição definido pelo padrão é chamado de "primitiva" (equivalente à métrica atômica definida neste trabalho). As primitivas de saída medidas são: linhas de código, páginas documentadas e, opcionalmente, pontos de função. As primitivas de entrada medem o esforço daqueles que desenvolvem os produtos de software, em homem-hora. Os ganhos com o uso de ferramentas de apoio automático não são medidos diretamente pela norma, mas são indiretamente medidos pelas melhorias na produtividade das pessoas que as utilizam.

As primitivas são categorizadas por atributos (definidos como uma característica mensurável de uma primitiva"), e o padrão demanda que todos os atributos sejam

medidos para cada primitiva. Por exemplo, para a primitiva "linhas de código", há atributos como "novo / reutilizado". Utilizando o esquema de primitivas e atributos, a norma garante que os elementos de produtividade sejam consistentemente categorizados.

Visto que as medições de produtividade são tão válidas quanto as primitivas utilizadas para calculá-las, acurácia e consistência dos dados utilizados são fundamentais para que os resultados sejam significativos. O "grau de granularidade" (nível de detalhe com que os dados são coletados ou apresentados), é importante na determinação da precisão e na consistência dos dados. Dados de granularidade mais fina são obtidos coletando-se dados em termos de homem-hora para cada hora trabalhada. Rastreando o esforço demandado nesse nível reduz o potencial de se encontrar erros maiores. Estimativas, tais como aquelas feitas no início ou no final de um projeto, por uma aproximação do esforço, são um exemplo da granularidade maior. Nesse caso, erros maiores podem ser introduzidos nos resultados.

3.8.3. ISO/IEC 14598-5 (*Information Technology – Software Product Evaluation – Process for Evaluators)*

De maneira análoga à ISO/IEC 12207, que define os processos desejáveis e necessários para o desenvolvimento de software – sem que para isso descreva os detalhes dos processos, ou seja, servindo como um framework – a ISO/IEC 14598 define as atividades necessárias para analisar requisitos, especificar, projetar e realizar ações de avaliação, e tirar conclusões referentes à avaliação de qualquer tipo de produto de software.

A norma fornece requisitos e faz recomendações para a implementação prática de avaliações de produtos de software quando há diversas partes envolvidas, e que precisam entender, aceitar e confiar nos resultados da avaliação, podendo ser utilizada para aplicar os conceitos descritos na ISO/IEC 9126. O processo de avaliação pode ser utilizado para avaliar produtos de software já existentes bem como produtos em desenvolvimento.

A norma destina-se a:

- avaliadores em laboratórios que prestam serviços de teste e avaliação de software
- fornecedores de software, na avaliação de seus produtos

- compradores de software, quando da requisição de informação de avaliação
- usuários de software, na avaliação de produtos ou quando recebendo relatórios de avaliação
- organismos de certificação, na definição de novos mecanismos de certificação para produtos de software

Dentre os conceitos definidos pela norma, destacam-se:

- método de avaliação: procedimento descrevendo a ação que deve ser realizada pelo avaliador para se obter o resultado de uma medição ou verificação específica, plicada em componentes específicos do produto ou no produto como um todo.
- avaliação de produto de software: operação técnica que consiste em produzir uma avaliação de uma ou mais características de um produto de software, de acordo com um procedimento específico.

A qualidade de produtos de software pode ser descrita em termos de características de qualidade, conforme definição da ISO/IEC 9126. No entanto a medição de tais características não é prática. Normalmente, o possível é avaliar essas características com base na medição de atributos do produto de menor grau de abstração. Nesse contexto, o avaliador pode utilizar sua experiência em engenharia de software para fazer a avaliação – o que pode reduzir a objetividade da avaliação, ou utilizar métodos não-determinísticos de avaliação em que o avaliador precise fazer escolhas que não estejam pré-definidas. Nos procedimentos da norma é presente a preocupação de manter a objetividade da avaliação o mais alto o possível.

O principal objetivo do processo de avaliação definido na ISO 14598 é garantir as seguintes características desejáveis:

- repetibilidade: avaliações repetidas de um mesmo produto, seguindo a mesma especificação de avaliação, quando aplicadas pelo mesmo avaliador, devem produzir resultados considerados como idênticos.
- reprodutibilidade: avaliações do mesmo produto, seguindo a mesma especificação de avaliação, quando aplicadas por diferentes avaliadores devem produzir resultados considerados idênticos.
- imparcialidade: a avaliação não deve sofrer e nem demonstrar qualquer tipo de viés.

- objetividade: os resultados da avaliação devem ser factuais, ou seja, não ser "contaminados" pelas impressões ou opiniões do avaliador.

O processo de avaliação é composto por cinco atividades:

1) definição dos requisitos da avaliação;

2) especificação da avaliação, com base nos requisitos da avaliação e na descrição dos produtos;

3) projeto da avaliação, para produzir um plano de avaliação com base na especificação da avaliação;

4) execução do plano de avaliação: inspecionar, modelar, medir e testar os produtos e seus componentes de acordo com o plano de avaliação, sendo os resultados registrados em um *draft*;

5) conclusão da avaliação: confecção de um relatório da avaliação.

Como insumos para o processo de avaliação, o solicitante deve fornecer um documento básico de requisitos da avaliação, assim como a descrição do produto e seus componentes (o que pode incluir documentação referente ao planejamento e desenvolvimento dos mesmos); o avaliador deve fornecer as especificações pré-definidas, os métodos de avaliação e eventuais ferramentas utilizadas no processo. Os resultados esperados do processo de avaliação são:

- registros da avaliação, incluindo plano de avaliação e registro das ações de avaliação;
- o *draft* do relatório de avaliação, incluindo requisitos da avaliação, especificação da avaliação e resultados resumidos da avaliação;
- o relatório final de avaliação.

Nos requisitos da avaliação devem ser descritos os objetivos da avaliação, uma descrição geral do domínio da aplicação do produto, assim como um descrição geral de seus objetivos. Uma lista dos requisitos de qualidade do produto também deve estar presente, podendo, por exemplo, ser descrito em termos de características de qualidade da ISO 9126. Nesse caso, quando for demandada uma característica que não seja definida pela ISO 9126, devem ser incluídas referências definindo-a, e tanto o solicitante da avaliação quanto o avaliador devem explicitar sua compreensão mútua sobre a característica. A importância relativa de cada característica de qualidade deve ser indicada, podendo seguir uma gradação inclusa num anexo da norma. Para cada

requisito, a especificação da informação que deve estar contida no produto e em seus componentes devem ser fornecidos e, sempre que possível, fazendo referência padrões e normas de engenharia de software.

Na especificação da avaliação, é estabelecido o escopo da avaliação e são definidas todas as análises e medições que devem ser realizadas no produto e em seus componentes. O nível de detalhamento deve ser tal que garanta repetibilidade e reprodutibilidade. Deve conter também um mapeamento entre a especificação das medições e verificações e os requisitos da avaliação, juntamente com referência aos padrões ou justificativas para cada medição ou verificação listadas.

Para especificar as medições necessárias, o avaliador deve alocar os requisitos da avaliação no produto alvo da avaliação e em seus componentes – fazendo a decomposição dos requisitos da avaliação em sub-características – e especificar as medições que pretende utilizar para avaliar as características, sub-características e atributos do produto e de seus componentes. Essas especificações devem ser formuladas através de especificações formais de métricas, incluindo instruções de como as medidas devem ser apresentadas e analisadas no relatório de avaliação, assim como através de especificação de procedimentos que serão utilizados para verificar a presença dos requisitos do software em cada componente.

O plano de avaliação descreve os procedimentos operacionais necessários para implementar a especificação da avaliação, devendo conter todos os métodos e ferramentas que serão utilizados na avaliação. Os métodos de avaliação são relacionados a elementos da especificação da avaliação, que por sua vez são atrelados aos requisitos da avaliação. Como cada um dos métodos elementares de avaliação deve ser aplicado a diversos componentes, pode ocorrer de o planejamento considerar a aplicação de diversos métodos elementares de avaliação sobre um mesmo componente. Para otimizar as medições demandadas, de modo a reduzir riscos de erros e reduzir os esforços de avaliação. O plano de avaliação também deverá conter um cronograma das ações de avaliação.

O relatório final da avaliação contém os requisitos, a especificação da avaliação, resultados das medições e análises executadas e qualquer outra informação necessária para ser repetir ou reproduzir a avaliação. A norma traz ainda, em seus anexos:

- um *template* de relatório de avaliação
- a sugestão de uma escala de gradação para a definição de requisitos de qualidade
- uma descrição dos possíveis componentes de software a ser analisados (listando todos os artefatos descritos na ISO 12207)
- um guia dos pontos de interação entre o solicitante da avaliação e o avaliador
- um *template* de contrato de avaliação.

A avaliação de um produto de software pode ser realizada dentro do contexto de um processo de ciclo de vida, conforme definido na ISO 12207, podendo acontecer particularmente inserida nos processos de aquisição, fornecimento, desenvolvimento, operação ou manutenção. O quanto antes estiver inserido o processo de avaliação no processo de desenvolvimento do software, mais será possível definir no próprio processo de desenvolvimento medições e testes que devam ser realizados visando a avaliação, o que aumenta a possibilidade de o produto se aproximar mais das requisitos esperados no processo de avaliação, bem como reduzir o risco de custos extras.

Os conceitos e procedimentos definidos pela norma devem ser levados em consideração quando da definição de um plano de medição, assim como na especificação de procedimentos e definições operacionais de métricas e de sua análise, pois ela preza aspectos que vão além da definição técnica das métricas e procuram garantir que os resultados do processo de avaliação serão compreensíveis e confiáveis.

3.9. Considerações finais

Neste capítulo, foi descrito um conjunto de normas e modelos de maturidade relevantes para este trabalho, com destaque para o CMMI, o modelo de referência do MPS.BR e a ISO 15939. Os modelos de processo de medição propostos por essas normas e padrões – demonstram extrema convergência no uso de medição orientada a objetivos, mesma proposta da metodologia GQM.

No próximo capítulo, será descrita a Estação TABA, ambiente para o qual foi definida e implementada a base de métricas, proposta por esta Tese.

Capítulo 4

Ambientes de Desenvolvimento de Software e a Estação TABA

Este capítulo faz uma breve apresentação dos conceitos de Ambientes de Desenvolvimento de Software e, mais especificamente, de Ambientes de Desenvolvimento de Software Orientados à Organização (ADSOrg). O capítulo apresenta a Estação TABA e seus ambientes, comentando brevemente seus objetivos, sua estrutura e as ferramentas disponíveis, com uma visão geral da abordagem de medição e análise nos ambientes TABA.

4.1. Introdução

Ambiente de Desenvolvimento de Software (ADS) é um sistema computacional que provê suporte para o desenvolvimento e a manutenção de software e para o gerenciamento destas atividades, contendo uma base de dados central e um conjunto de ferramentas de apoio. ADSs visam apoiar os desenvolvedores em suas atividades, buscando aumento da produtividade, com impacto nos prazos, custos e qualidade dos projetos.

Os primeiros ambientes de desenvolvimento (PSEs – *Programming Support Environments*) apoiavam tipicamente as atividades de codificação e depuração, ficando as demais atividades do ciclo de vida sem apoio (HARRISON *et al.*, 2000).

A necessidade de suporte às demais atividades do ciclo de vida de um software deu início aos Ambientes de Engenharia de Software (SEEs – *Software Engineering Environments*) que suportam uma faixa mais ampla de atividades. Dentro dos ambientes de engenharia de software surgiu uma ramificação de trabalhos relacionados aos Ambientes de Engenharia de Software Centrados em Processos (PSEEs – *Process-centered Software Engineering Environments*), que integram o ferramental de suporte à elaboração de artefatos, com o suporte à definição e execução dos processos que produzem estes artefatos (HARRISON *et al*, 2000).

Segundo BALZER e GRUNHN (2001), Ambientes Centrados em Processos trazem diversas vantagens ao desenvolvimento:

- Guiam os desenvolvedores através dos processos suportados;
- Automatizam atividades dos processos;
- Auxiliam no acompanhamento do status das atividades dos processos;
- Forçam um alto nível de consistência entre os artefatos produzidos.

Segundo GARG e JAZAYERI (1996) os ambientes centrados em processos estendem os ambientes convencionais em duas direções:

- Suportam o uso de um processo de desenvolvimento explícito. O ambiente deve pelo menos apoiar a definição do processo de desenvolvimento e a monitoração da execução do processo, para detectar desvios;
- Tratam o processo de software como algo que vai além de uma simples descrição estática do ciclo de vida, considerando a engenharia de processos na definição e avaliação dos modelos de processo.

O estudo de ADS teve início na década de 50 e evoluiu rapidamente ao longo dos anos. As pesquisas iniciais em ADS visavam desenvolver ferramentas de automatização do processo de desenvolvimento de software.

Atualmente, os estudos em ADS exploram o desenvolvimento de ferramentas integradas para apoiar o desenvolvedor de software na execução das atividades do processo de desenvolvimento. Neste contexto, o grupo de engenharia de software da COPPE/UFRJ iniciou na década de 90 estudos em ambientes de desenvolvimento de software com o objetivo de definir e criar a Estação TABA, um meta-ambiente de desenvolvimento de software capaz de gerar, através de instanciação, outros ADS (ROCHA *et al.*, 1990). Ao longo desses anos de trabalho, o conceito de ADS evoluiu para a definição de ADS com suporte à utilização do conhecimento do domínio de aplicação durante o desenvolvimento e, mais recentemente, para a utilização de conhecimento organizacional. Para acompanhar essa evolução, a Estação TABA passou a considerar, Ambientes de Desenvolvimento de Software Orientados a Domínio (ADSOD) e Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg) (OLIVEIRA,1999) (VILLELA, 2004).

4.2. A Estação TABA

A Estação TABA, conforme sua primeira definição (ROCHA *et al.*, 1990), é um meta-ambiente capaz de gerar, através de instanciação, ambientes de desenvolvimento de software adequados às particularidades de processos de desenvolvimento e de projetos específicos. ROCHA *et al.* (1990) definem meta-ambiente como um ambiente que abriga um conjunto de programas que interagem com os usuários para definir interfaces, selecionar ferramentas e estabelecer os tipos de objetos que irão compor o ambiente de desenvolvimento específico.

A Estação TABA é um ADS centrado em processos, que agrega conhecimentos e funcionalidades desenvolvidos durante vários trabalhos de mestrado e doutorado e experiências na sua implantação em empresas. A Estação TABA cobre boa parte das características dos ambientes centrados em processos, com recursos adicionais:

- Suporte de conhecimento sobre o domínio para o qual se desenvolve (OLIVEIRA 1999);
- Suporte à aquisição e disseminação do conhecimento na execução dos processos (MONTONI, 2004);
- Suporte de conhecimento sobre a organização desenvolvedora (VILLELA, 2004) (SANTOS, 2003);
- Suporte a medições e avaliações de processos (ANDRADE, 2005);
- Opera em ambiente Windows, o que facilita a avaliação da sua eficácia na indústria;
- Em uso na indústria desde junho de 2003;
- Suporte explícito a áreas de processo do CMMI 2 e 3 e do MR MPS.BR, níveis G a C.

O projeto TABA foi criado a partir da constatação de que domínios de aplicação diferentes possuem características distintas e que estas devem incidir nos ambientes de desenvolvimento através dos quais os desenvolvedores de software desenvolvem aplicações. Desta forma, a Estação TABA tem por objetivo auxiliar na definição, implementação e execução de ADS adequados a contextos específicos.

Com o intuito de atender a este objetivo, quatro funções foram definidas originalmente para a Estação TABA (TRAVASSOS, 1994):

(i) Auxiliar o engenheiro de software na especificação e instanciação do ambiente mais adequado ao desenvolvimento de um produto específico a partir do processo de software e/ou de uma definição do domínio de aplicação;

(ii) Auxiliar o engenheiro de software na implementação das ferramentas necessárias ao ambiente definido;

(iii) Permitir aos desenvolvedores do produto de software a utilização da estação através do ambiente desenvolvido;

(iv) Permitir a execução do software na estação configurada para o seu desenvolvimento.

Em 1997, iniciou-se uma grande evolução da Estação TABA para torná-la um ambiente capaz de instanciar Ambientes de Desenvolvimento de Software Orientados a Domínio (ADSOD), considerando não apenas as características específicas dos projetos, mas também o domínio da aplicação. A motivação para esta evolução foi a identificação de que um dos principais problemas no desenvolvimento de software era o desconhecimento do domínio por parte dos desenvolvedores de software. Isso foi constatado pela autora da proposta em sua experiência em várias organizações (OLIVEIRA, 1999).

Durante o desenvolvimento e a manutenção de software desenvolvedores lidam de forma intensa com diferentes tipos de conhecimento ao longo dos processos de software e, por isso, é preciso dar atenção à produção, armazenamento, compartilhamento e uso dos conhecimentos relevantes neste contexto. Uma abordagem natural para tratar esta questão foi a introdução de gerência do conhecimento nos ambientes de desenvolvimento de software (VILLELA *et al.*, 2005). Desta forma, os Ambientes de Desenvolvimento de Software Orientados à Organização (ADSOrg) foram definidos como sendo ADS que apóiam a gerência do conhecimento ao longo dos processos de desenvolvimento e manutenção de software.

O ADSOrg tem como objetivo: (a) apoiar os desenvolvedores de software na execução de suas atividades, fornecendo todo o conhecimento que tenha sido capturado e acumulado pela organização por sua importância para o desenvolvimento e a manutenção de software, e (b) apoiar o aprendizado organizacional em Engenharia de Software a partir do aprendizado dos desenvolvedores da organização nos projetos de software específicos. A finalidade é evitar erros já cometidos e possibilitar a reutilização de soluções já aprovadas na execução de tarefas similares, buscando melhorar a produtividade e a qualidade, bem como diminuir custos.

A instanciação de ADS convencionais e orientados a domínio continua sendo possível, mas agora podem ser instanciados ADSOrg a partir de ambientes configurados pela Estação TABA para uma organização utilizando como base o processo padrão de desenvolvimento da organização, processos especializados e teorias do domínio dos softwares desenvolvidos pela organização.

4.3. Os ambientes da Estação TABA

As principais funções atuais dos ambientes da Estação TABA são (VILLELA, 2004):

(a) Auxiliar o engenheiro de software na configuração do ambiente mais adequado para apoiar o desenvolvimento e a manutenção de software em uma organização específica (Ambiente Configurado), considerando seu processo de software e o conhecimento organizacional relevante neste contexto;

(b) Auxiliar os gerentes de projeto na instanciação de ambientes de desenvolvimento de software para projetos específicos a partir do Ambiente Configurado;

(c) Apoiar, através dos ADSOrg, o desenvolvimento e a manutenção de software, bem como a gerência destas atividades.

Os ambientes que contemplam essas funções (figura 4.1) são os seguintes:

- Meta-Ambiente: ambiente que apóia a configuração de ambientes para organizações específicas;
- Ambiente Configurado: ambiente configurado a partir do Meta-ambiente que apóia a instanciação de ADSOrgs para projetos específicos;
- Ambiente Instanciado (ADSOrg): ambiente de desenvolvimento de software instanciado a partir do Ambiente Configurado, que é utilizado pelo gerente e desenvolvedores de cada projeto, sendo um (1) ADSOrg para cada projeto.

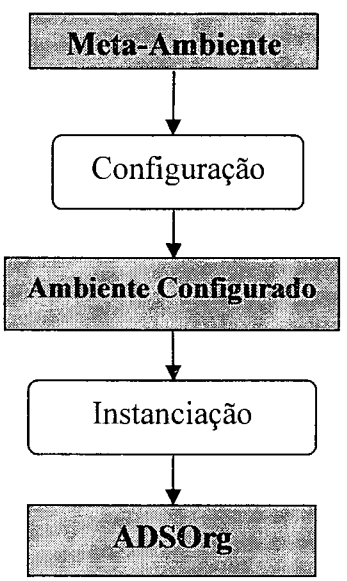

Figura 4.1 – Ambientes da Estação TABA (VILLELA 2004)

Os ambientes da Estação TABA são utilizados em diferentes contextos por profissionais de perfis distintos. O Meta-ambiente é utilizado em uma entidade externa à organização desenvolvedora, sendo utilizado por profissionais especializados em engenharia de software, que fazem a manutenção e evolução dos ativos de processos e demais recursos disponíveis neste ambiente. O principal processo que ocorre neste ambiente é o de Configuração, que é a criação dos ambientes que serão utilizados nas organizações, os chamados Ambientes Configurados. Portanto a principal função do Meta-ambiente é produzir ambientes para as organizações, conforme suas características individuais.

O Ambiente Configurado, gerado pelo Meta-ambiente, é utilizado por profissionais de nível gerencial, e pelo grupo de processos. O principal processo que ocorre neste ambiente é o de Instanciação, que é a geração dos ambientes que serão utilizados pelos profissionais dos projetos. Este processo leva em consideração as características individuais de cada projeto, portanto para cada projeto que é iniciado na organização é gerado um ADSOrg pelo ambiente configurado.

O Ambiente Instanciado (ADSOrg) para cada projeto é utilizado pelo gerente do projeto e demais membros da equipe do projeto. Este é o ambiente que dá suporte às várias atividades dos processos do ciclo de vida de um software, com ferramentas específicas, procedimentos e *templates*.

4.3.1. Definição de processos em níveis

ROCHA *et al.* (2001) descreve a estratégia de definição de processos em níveis. De acordo com esse modelo (figura 4.2), a definição de processos de software se realiza em três etapas: definição do processo padrão da organização, especialização do processo padrão e instanciação para projetos específicos. Como resultado, têm-se processos de software em diferentes níveis de abstração.

A definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software e constitui a base para a definição de todos os processos. Dessa forma, estabelece-se um processo básico que servirá como ponto de partida para a posterior definição dos processos de software adequados às diferentes características de cada projeto, permitindo economia de tempo e esforço. Para esta definição é considerada a norma ISO/IEC 12207 (1995), o CMMI (CMU/SEI, 2000), e/ou o MR MPS.BR (Sociedade SOFTEX, 2004a) (Sociedade SOFTEX, 2004b), de acordo com a necessidade de cada organização. O processo

padrão é definido com o apoio da ferramenta Config no processo de Configuração de Ambientes (VILLELA, 2004), que ocorre no meta-ambiente.

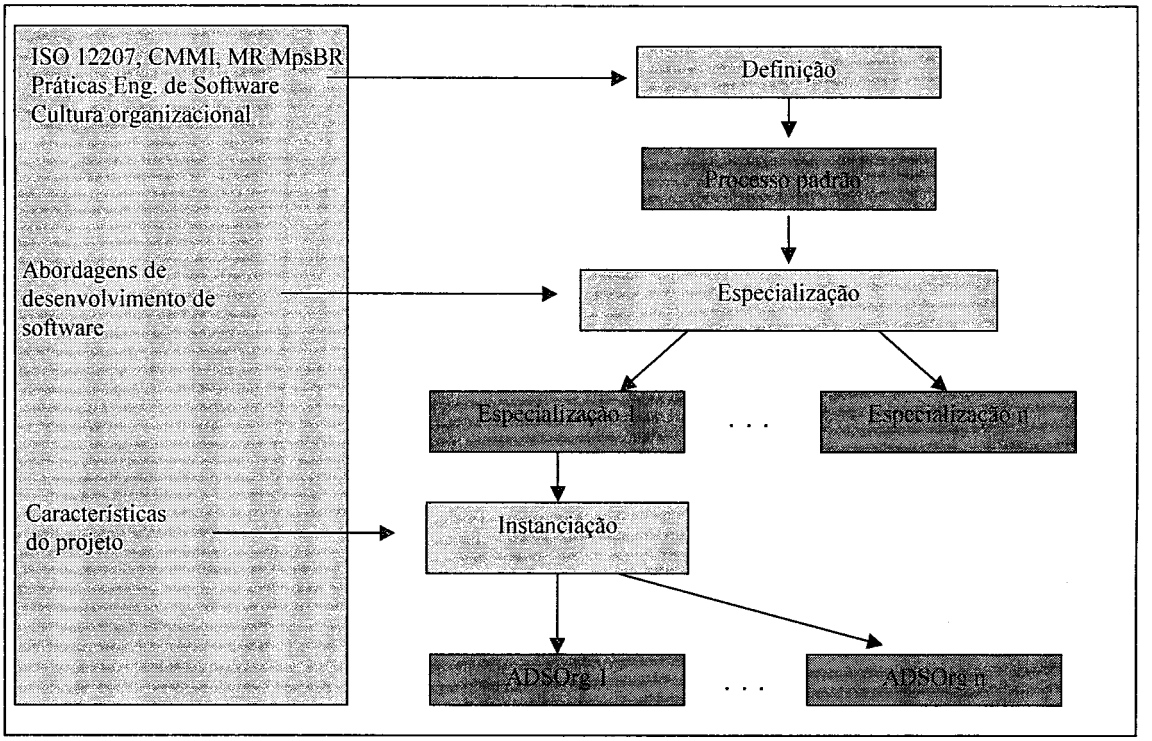

Figura 4.2: Modelo para a definição de processos de software

Tendo em vista que os diferentes paradigmas de desenvolvimento possuem características distintas e requerem diferentes abordagens de desenvolvimento, o processo de software padrão da organização deverá ser adaptado (especializado), considerando-se as características relacionadas ao paradigma de desenvolvimento utilizado (p.ex.: orientação a objetos). Assim, durante a etapa de especialização do processo padrão, atividades poderão ser adicionadas ou modificadas, de acordo com o contexto para o qual se está realizando a especialização. Os processos especializados também são definidos no processo de Configuração de Ambientes no meta-ambiente. Portanto, os processos padrão e especializados são definidos usando o meta-ambiente, e ficam disponíveis para uso no ambiente configurado.

A instanciação para projetos específicos consiste na adaptação de um processo especializado para um determinado projeto, considerando-se as suas peculiaridades. Nessa etapa, são definidos o modelo de ciclo de vida, os métodos e as ferramentas que serão utilizadas no projeto, os recursos humanos e suas responsabilidades ao longo do processo e os artefatos consumidos e produzidos. As atividades do processo especializado deverão ser mapeadas para o modelo de ciclo de vida escolhido para o projeto, e novas atividades poderão ser inseridas em um processo com um determinado

ciclo de vida. O processo de instanciação ocorre no Ambiente Configurado com o suporte da ferramenta AdaptPro (BERGER, 2003). O processo instanciado é definido no ambiente configurado, e fica disponível para uso em um projeto a partir dos ambientes instanciados.

Cabe notar que o único nível de processo que é de fato executado nos projetos é o último nível (processo instanciado). Desta forma, qualquer avaliação e análise dos processos da organização deve começar por este nível.

4.4. Estágio atual da implementação da Estação TABA

A implementação da Estação TABA foi iniciada em 1994 a partir do trabalho de TRAVASSOS (1994). No início, foi desenvolvida utilizando a linguagem Eiffel numa estação de trabalho da *Sun Microsystems*®. Com o passar do tempo, percebeu-se que o poder e a robustez da plataforma, apesar de adequadas ao ambiente de pesquisas, dificultava a experimentação das idéias em outros ambientes devido à falta de portabilidade do código para plataformas mais acessíveis e comumente utilizadas. Com o intuito de solucionar este problema, foi realizada uma re-implementação onde se optou pela plataforma de microcomputadores e pelo uso da linguagem C++ (OLIVEIRA, 1999) (SANTOS e ZLOT, 1999).

Com a criação dos ADSOrg, o modelo foi revisto para adequá-lo ao novo esquema de configuração/instanciação de ambientes e para permitir a definição de novas ferramentas.

Desde 2003, os ambientes TABA, antes restritos à área acadêmica, passaram a ser utilizado por empresas brasileiras desenvolvedoras de software (MONTONI *et al.*, 2005a) (MONTONI *et al.*, 2005b) (SANTOS, *et al.*, 2005).

Devido a estas novas exigências, a Estação TABA necessitou atender a novos requisitos. Estes requisitos, atualizados por VILLELA (2004), foram revisados recentemente dando origem ao conjunto de requisitos listados na tabela 4.1.

Tabela 4.1 - Lista de requisitos revisada da Estação TABA

#	Requisito	Descrição	Meta-	Ambient	ADSOrg
1	possuir interface consistente	os ambientes da Estação TABA devem possuir mecanismos de interface que permitam a utilização consistente de seus recursos e ferramentas;			

2	possuir um modelo comum de armazenamento de dados	a forma de representação das informações nos ambientes da Estação TABA deve possibilitar que as ferramentas compartilhem e utilizem estas informações de forma natural e consistente;			
3	Apoiar a reutilização de conhecimento	os ambientes da Estação TABA devem fornecer mecanismos que possibilitem a reutilização de qualquer espécie de conhecimento em contexto diferente do contexto para o qual foi criado;			
4	apoiar o controle de versões e a gerência de configuração	os ambientes da Estação TABA devem controlar as modificações feitas nos componentes de conhecimento, itens de software e ativos de processo, mantendo-os disponíveis em suas diferentes versões e gerenciando onde essas versões estão sendo utilizadas			
5	possuir conhecimento sobre processo de software e abordagens de desenvolvimento	o Meta-ambiente e os Ambientes Configurados devem possuir conhecimento sobre processo de software e as várias alternativas de modelos de ciclo de vida, paradigmas de desenvolvimento e métodos possíveis de serem utilizados, bem como sobre a adequabilidade da aplicação de cada uma dessas alternativas em diferentes contextos;			
6	possuir mecanismo de integração de ferramentas internas	o Meta-ambiente deve permitir e facilitar a integração de ferramentas internas (desenvolvidas como parte da Estação TABA);			
7	possuir mecanismo de integração de ferramentas externas	o Meta-ambiente, os Ambientes Configurados e os Ambientes Instanciados devem permitir e facilitar a integração de ferramentas externas;			
8	permitir a descrição de tarefas	o Meta-ambiente deve possuir mecanismos que facilitem a descrição de tarefas genéricas, que independem de um domínio de aplicação;			
9	apoiar a definição de Teorias de Domínio	o Meta-ambiente deve possuir mecanismos que facilitem a definição de Teorias de Domínio para diferentes domínios de aplicação e para o domínio de Engenharia de Software, identificando as tarefas genéricas que são executadas nesses domínios;			
10	apoiar a definição de processos	o Meta-ambiente deve apoiar a definição e a especialização do processo padrão de uma organização;			
11	gerar Ambientes Configurados para organizações específicas	o Meta-ambiente deve ser capaz de gerar um Ambiente Configurado para uma organização específica, considerando os processos padrão e especializados que foram definidos e os domínios de aplicação nos quais a organização atua;			
12	permitir a incorporação de novos conhecimentos e experiências	o Meta-ambiente deve possuir mecanismos que permitam incorporar conhecimentos e experiências registrados nos ambientes configurados e instanciados como, por exemplo, evoluções em lições aprendidas e novas melhores práticas;			
13	permitir novas configurações sem perda do conhecimento organizacional	o Meta-ambiente deve permitir a configuração de um novo ambiente para uma organização (nova versão do Ambiente Configurado), sem acarretar a perda do conhecimento acumulado na versão anterior.			
14	permitir a evolução das Teorias de Domínio que fazem parte do ambiente	um Ambiente Configurado deve permitir que novos conceitos e relações, bem como novas instâncias de conceitos e relações, sejam incluídos nas Teorias dos Domínios de			

		aplicação e de Engenharia de Software que fazem parte do ambiente;			
15	permitir a descrição da estrutura organizacional	um Ambiente Configurado deve permitir a descrição da estrutura organizacional de qualquer organização envolvida com os projetos das organizações desenvolvedoras e a definição das competências desejadas para cada posição definida pela estrutura;			
16	permitir a descrição dos profissionais da organização	um Ambiente Configurado deve permitir a descrição do perfil dos profissionais das organizações envolvidas e a alocação dos mesmos à estrutura organizacional;			
17	permitir a descrição dos processos organizacionais	um Ambiente Configurado deve permitir a descrição dos processos organizacionais das organizações envolvidas, o que envolve representação gráfica destes processos e descrição dos elementos representados;			
18	apoiar a definição de processos para projetos específicos a partir dos processos especializados	um Ambiente Configurado deve apoiar a definição de um processo para um projeto de software específico, a partir de um dos processos especializados da organização, considerando, para isto, as características do projeto;			
19	gerar ADSOrg para projetos específicos	um Ambiente Configurado deve ser capaz de gerar um ADSOrg para um projeto de software específico a partir do processo definido para o projeto;			
20	apoiar a produção de conhecimento organizacional	um Ambiente Configurado deve apoiar a filtragem e empacotamento de conhecimento adquirido nos ADSOrg e seu armazenamento no repositório da organização, o que inclui a associação do mesmo com as atividades dos processos às quais se refere e com os conceitos e instâncias de conceitos que o descrevem.			
21	apoiar a definição de arquiteturas de referência	o Meta-ambiente e os Ambientes Configurados devem possuir uma base de conhecimento sobre arquiteturas de software que sirvam de referência no desenvolvimento de diferentes produtos de um mesmo tipo e/ou domínio de aplicação;			
22	apoiar a execução do processo e a sua gerência	um ADSOrg deve apoiar a execução do processo, e de suas atividades, e a gerência do mesmo, através de orientação, automação e/ou monitoração do processo e, quando necessário, do suporte a sua modificação;			
23	apoiar o trabalho cooperativo	um ADSOrg deve definir protocolos de coordenação, colaboração e comunicação que facilitem o trabalho em equipe, o que é especialmente importante para projetos de desenvolvimento ou manutenção de software em larga escala;			
24	possuir suporte para a avaliação do produto e do processo	um ADSOrg deve apoiar a medição dos produtos gerados ao longo do processo, de forma a permitir a garantia da qualidade do produto final e a avaliação do processo.			
25	permitir a instanciação da Teoria do Domínio referente à aplicação	ADSOrg deve permitir que novas instâncias de conceitos e relações sejam incluídas na Teoria do Domínio referente à aplicação;			
26	fornecer acesso ao conhecimento sobre o domínio da aplicação	este requisito refere-se à necessidade ADSOrg oferecer mecanismos e ferramentas de acesso ao conhecimento sobre o domínio da aplicação.			

27	apoiar a localização de profissionais	um ADSOrg deve apoiar a localização dos profissionais das organizações envolvidas no projeto mais adequados para auxiliar na execução de uma atividade ou na solução de um problema;			
28	apoiar o entendimento dos processos organizacionais	um ADSOrg deve permitir a visualização dos processos organizacionais das organizações envolvidas no projeto e a navegação através dos seus diferentes níveis de abstração, fornecendo, sob solicitação, detalhes sobre os elementos representados e permitindo acesso às informações e conhecimentos disponíveis no ambiente;			
29	fornecer acesso ao conhecimento organizacional	este requisito refere-se à necessidade do ADSOrg oferecer, de acordo com a atividade do processo sendo executada, mecanismos e ferramentas de acesso ao conhecimento acumulado pela organização ao longo do tempo;			
30	apoiar a aquisição de conhecimento para a organização	um ADSOrg deve oferecer mecanismos que permitam a aquisição de conhecimento ao longo das atividades do processo e ao final do projeto.			
31	permitir a evolução do modelo das organizações clientes envolvidas	um ADSOrg deve permitir que os modelos das organizações envolvidas, já disponível no Ambiente Configurado e composto da estrutura organizacional, dos profissionais alocados a esta estrutura e dos processos organizacionais, sejam atualizados e expandidos de acordo com as informações obtidas no projeto corrente.			

4.5. Serviços e ferramentas de apoio presentes no ADSOrg

A seguir são apresentados os serviços e as ferramentas atualmente integradas aos ambientes TABA, que apóiam o cumprimento de seus requisitos e auxiliam os desenvolvedores de software na definição de processos de desenvolvimento e na execução das atividades de desenvolvimento de software. São eles:

- *Acknowledge*, ferramenta para apoiar o processo de captura de conhecimento ao longo dos processos de software, englobando registro, filtragem e empacotamento de conhecimento (MONTONI, 2003) (MONTONI *et al.*, 2004);
- *ActionPlanManager*, ferramenta para apoiar a elaboração de planos de ação ao longo do projeto;
- *AdaptPro*, ferramenta para apoiar a instanciação de processos de software para projetos específicos a partir dos processos especializados da organização (BERGER, 2003);
- *Config*, ferramenta para apoiar a configuração de ambientes para as organizações (VILLELA, 2004);
- *ControlManager*, ferramenta para apoiar o planejamento do acompanhamento e controle do projeto;

- *DocPlan* e *GeraDoc*, ferramentas para, respectivamente, apoiar o planejamento da documentação a ser produzida em um projeto de software e para gerar documentos a partir da agregação de outros documentos (MARTINS, 2004);
- *Editar*, ferramenta para apoiar a definição de teorias de domínio e tarefas (OLIVEIRA, 1999, ZLOT, 2002, ZLOT *et al.*, 2002);
- *Edited*, ferramenta para apoiar a definição de teorias de domínio e tarefas (OLIVEIRA *et al.*, 2000);
- *GConf*, ferramenta para apoio à gerência de configuração dos artefatos produzidos em um projeto de software (FIGUEIREDO *et al.*, 2004).
- *Genesis*, ferramenta para apoiar a atividade de investigação do domínio (GALOTTA, 2000);
- *Navegue*, ferramenta para apoiar a atividade de investigação do domínio (GALOTTA, 2000);
- *OrgPlan*, ferramenta para elaboração do plano de organização para um projeto;
- *Planilha de Atividades*, ferramenta para registro das atividades do desenvolvedor no projeto;
- *ProjectStatus*, ferramenta que informa a situação dos projetos sendo conduzidos pela organização;
- *QFuzzy*, ferramenta para apoiar a identificação dos requisitos de qualidade de produtos;
- *QualityPlan*, ferramenta para apoiar o planejamento do controle da qualidade que deve ser efetuado no projeto;
- *Regcon*, ferramenta para apoiar a atividade de investigação do domínio (GALOTTA, 2000);
- *ReqManager*, ferramenta para apoiar a gerência de requisitos com a construção da matriz de rastreabilidade.
- *RHPlan* e *RHManager*, ferramentas para apoiar o planejamento, monitoração e avaliação da alocação de profissionais aos projetos de software, que inclui a solicitação de contratação e capacitação, o acompanhamento das horas dedicadas a cada atividade, além da atualização, ao final do projeto, das competências por eles possuídas (SCHNAIDER, 2003);

- *RiscManager*, ferramenta para apoiar o planejamento e a monitoração de riscos em projetos de software que baseia-se na reutilização do conhecimento organizacional sobre riscos (FARIAS, 2002) (FARIAS *et al.*, 2003);
- *Sapiens*, ferramenta que permite a descrição e visualização de estruturas organizacionais, englobando os profissionais alocados e as competências requeridas e possuídas ao longo dessas estruturas (SANTOS, 2003) (SANTOS *et al.*, 2003) (SANTOS *et al.*, 2004);
- *TempPlan*, *TempManager*, *CustPlan* e *CustManager*, ferramentas para apoiar o planejamento e o controle de tempo e custo em projetos de software, baseadas na reutilização do conhecimento organizacional e nos modelos paramétricos COCOMO II e Análise de Pontos de Função (BARCELLOS, 2003) (BARCELLOS *et al.*, 2003);

4.6. Medição e análise nos ambientes TABA

A melhor abordagem para seleção das medidas adequadas é a que define, antes de tudo, o que a organização deseja ou precisa saber, e somente então escolhe as medidas apropriadas. O TABA apresenta uma abordagem para o processo de Medição e Análise em projetos de desenvolvimento de software (SCHNAIDER, L., SANTOS, G., MONTONI, M., ROCHA, A. R., 2004) baseada no método *Goal-Question-Metrics* (GQM), nos requisitos da área de processo Medição e Análise do CMMI e do processo de Medição do MR MPS.BR. A proposta é apoiar a medição e análise disponibilizando o conhecimento sobre medições que possa ser útil às organizações e aos gerentes de projeto durante a execução das atividades que compõem o processo de *Medição e Análise.* Para execução do processo, foram criadas duas ferramentas de apoio à abordagem proposta (SCHNAIDER, L., SANTOS, G., MONTONI, M., ROCHA, A. R., 2004).

O Processo de medição e análise e as ferramentas de apoio foram projetados e implementados tendo como infra-estrutura a Base de Métricas definida por este trabalho (figura 4.3).

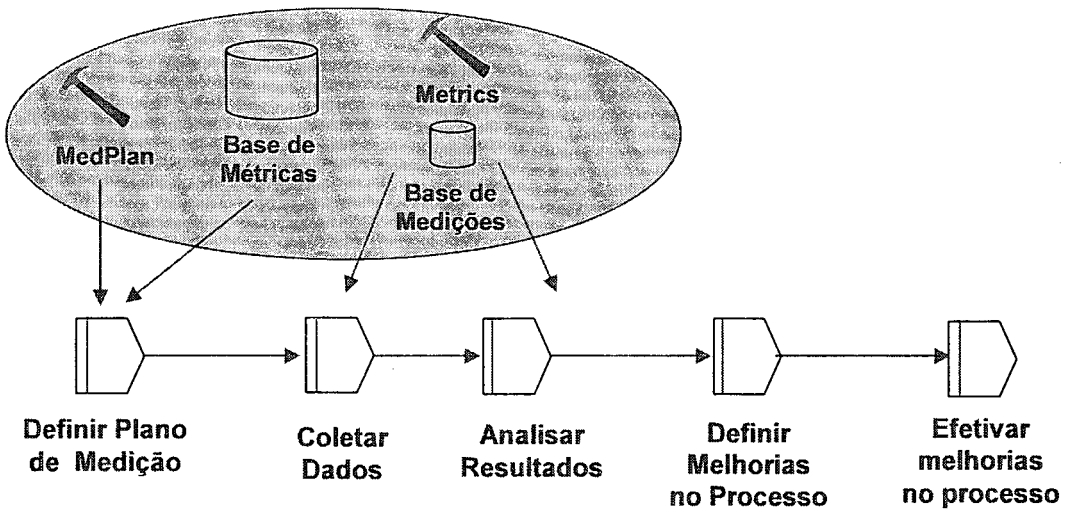

Figura 4.3 – Processo de Avaliação e Melhoria TABA

De acordo com FLORAC E CARLETON (1999), um processo de medição deve ser modelado de forma independente do processo de desenvolvimento, e conter atividades para coleta de métricas que contenham a descrição de cada métrica assim como uma descrição clara de seus objetivos.

O objetivo do processo de Medição e Análise é desenvolver e implantar a capacidade de avaliação das medidas (métricas) que servirão como importantes fontes de informação para a gerência da organização. A figura 4.4 descreve o processo de Medição e Análise definido para os ambientes TABA.

A proposta do processo de Medição e Análise definido é apoiar as necessidades de medição e análise da organização como um todo, possibilitando:

- a elaboração do Plano de Medição a partir dos objetivos de informação definidos pela organização, e
- o fornecimento dos resultados das medições estabelecidas.

A seguir são apresentadas, brevemente, as ferramentas *MedPlan* e *Metrics*, cujos objetivos, por sua vez, são apoiar o processo definido.

PROCESSO DE MEDIÇÃO E ANÁLISE

Atividade: **Especificar Atividades de Medição e Análise**

Descrição: Identificar objetivos e práticas para medições de forma que estes estejam alinhados às necessidades e objetivos de informação da Organização. A identificação de objetivos, questões e métricas é feita utilizando-se a abordagem GQM (*Goal-Question-Metrics*) de Basili.

Sub-Atividades:

Nome: **Identificar Objetivos de Medição**

Descrição: Estabelecer objetivos para medições derivados das necessidades e objetivos de informação da Organização.

Nome: **Identificar Questões de Medição**

Descrição: Estabelecer questões a serem respondidas para que os objetivos de medição da Organização possam ser mensurados.

Nome: **Identificar Medidas**

Descrição: Especificar medidas (métricas) para avaliar o atingimento dos objetivos definidos.

Nome: **Especificar Procedimentos de Coleta e Armazenamento de Dados**

Descrição: Especificar como os dados de medições serão obtidos e armazenados

Nome: **Especificar Procedimentos de Análise**

Descrição: Especificar como os dados de medições serão analisados e reportados

Atividade: **Fornecer Resultados das Medições**

Descrição: Obter e fornecer os resultados das medições realizadas de acordo com as necessidades e objetivos de informação identificados pela Organização

Sub-Atividades:

Nome: **Coletar Dados de Medições**

Descrição: Obter os dados das medições de acordo os procedimentos especificados para coleta, garantindo sua integridade

Nome: **Analisar Dados de Medições**

Descrição: Analisar e interpretar os dados obtidos nas medições

Nome: **Armazenar Dados e Resultados**

Descrição: Gerenciar e armazenar os dados obtidos nas medições assim como as especificação das medições e os resultados de análise.

Nome: **Comunicar Resultados**

Descrição: Comunicar os resultados das medições e das atividades de análise a todos os envolvidos.

Figura 4.4 – Processo de Medição e Análise Taba
(SCHNAIDER, L., SANTOS, G., MONTONI, M., ROCHA, A. R., 2004)

4.6.1 As ferramentas *MedPlan* e *Metrics*

A proposta da ferramenta *MedPlan* é apoiar a elaboração dos planos de medição da organização e do Projeto. Baseando-se no método GQM, esta ferramenta disponibiliza ao usuário o conhecimento sobre objetivos, questões, métricas e procedimentos de coleta, armazenamento e análise de dados a serem utilizados – armazenados na Base de Métricas do TABA, definida por este trabalho.

O Plano de Medição deve ser elaborado primeiramente para a organização como um todo, e neste deverão estar definidos os objetivos corporativos de medição, as questões a serem respondidas para que se possa avaliar o alcance destes objetivos e as medidas (métricas) propriamente ditas (assim como seus procedimentos de coleta, armazenamento e análise de dados) que serão utilizadas para responder às questões formuladas.

Uma vez elaborado o plano de medição da organização seus itens farão parte do plano de medição de cada projeto desta mesma organização. O gerente de projeto pode, se desejar, acrescentar (nunca retirar) novos objetivos, questões ou métricas que serão específicos do projeto em questão. Para tal, a ferramenta disponibiliza para o gerente de projeto o plano de medição da organização, assim como o conhecimento do ambiente a respeito de objetivos de medição e suas respectivas questões e métricas associadas.

A figura 4.5 apresenta a interface básica da ferramenta *MedPlan.* No lado esquerdo da interface pode-se identificar a atividade "Elaborar Plano de Medição da Organização" e suas respectivas sub-atividades, e no lado direito identifica-se a sub-atividade que está sendo realizada pelo usuário. Especificamente, a Figura 4.6 apresenta a sub-atividade "Definir Plano de Medição da Organização".

Os ícones localizados abaixo da barra de título permitem a busca e o registro de conhecimento no que diz respeito às atividades do processo. O gerente do projeto pode consultar idéias e lições aprendidas registradas por gerentes de projetos anteriores e também pode registrar suas próprias idéias e lições. A *MedPlan* disponibiliza também o conhecimento explícito em relação ao processo de Medição e Análise através de interface com a ferramenta de Aquisição do Conhecimento – *Acknowledge* (MONTONI *et al.,* 2004).

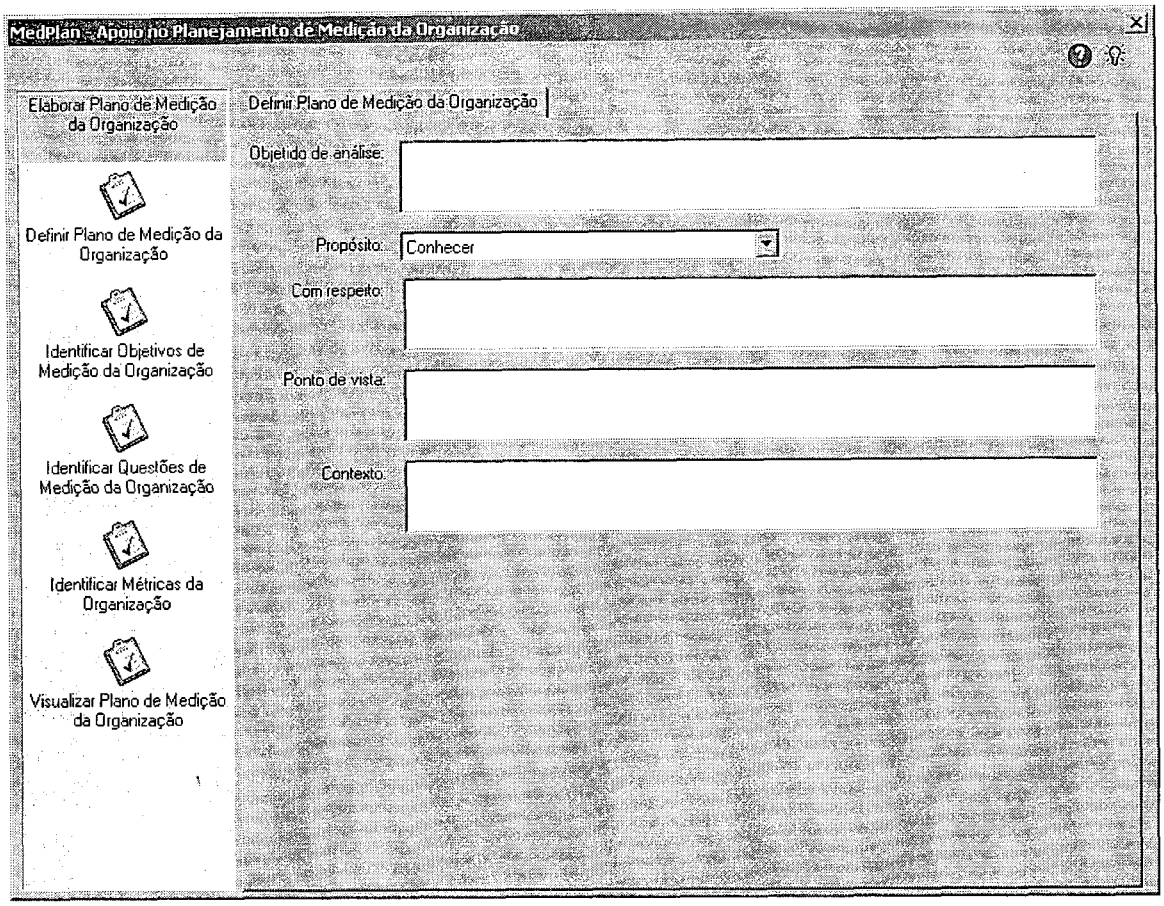

Figura 4.5 – Interface Básica da Ferramenta *MedPlan*

Na sub-atividade "Definir Plano de Medição da Organização" o usuário define pontos tais como o propósito das medições a serem realizadas (melhorar ou conhecer a Organização, por exemplo) e o contexto em que o Plano será elaborado. Em "Identificar Objetivos da Organização" o usuário seleciona os objetivos a serem alcançados pelas medições a serem efetuadas na Organização. Para cada objetivo identificado o usuário deve selecionar uma ou mais questões que devem ser respondidas para seu atingimento na sub-atividade "Identificar Questões da Organização". Na atividade "Identificar Métricas da Organização", por sua vez, o usuário deve selecionar uma ou mais métricas a serem utilizadas para responder a cada uma das questões previamente identificadas. Para cada métrica apresentada a ferramenta informa seus respectivos objetivos, descrições e procedimentos de coleta, armazenamento e análise. A figura 4.6 apresenta a tela da sub-atividade "Identificar Métricas da Organização".

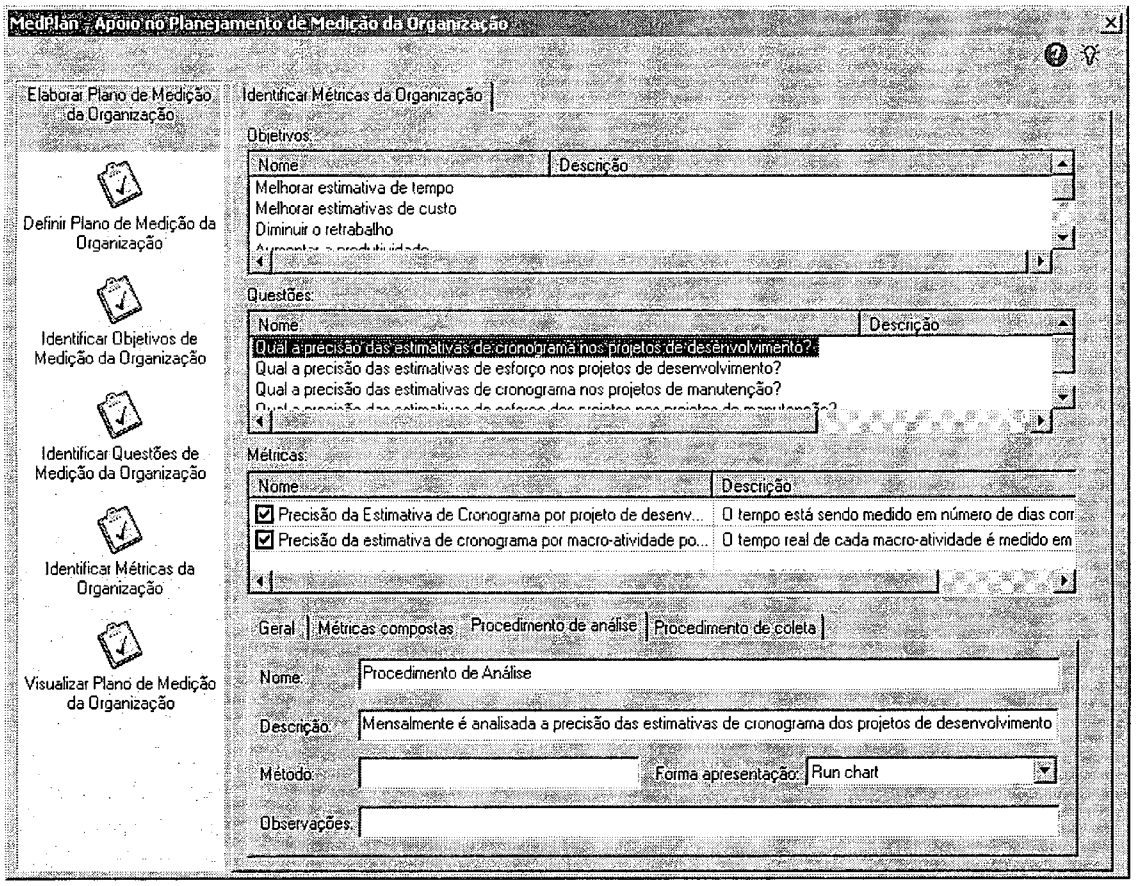

Figura 4.6 – Tela da sub-atividade "Identificar Métricas da Organização"

No Plano de Medição da Organização gerado com o auxílio da ferramenta na sub-atividade "Visualizar Plano de Medição da Organização" estarão registrados os objetivos, questões e métricas definidas para a Organização, assim como seus respectivos procedimentos de coleta e armazenamento e análise de dados.

O objetivo da ferramenta *Metrics* é apoiar a obtenção e o fornecimento dos resultados das medições realizadas de acordo com as necessidades e objetivos dos Planos de Medição da Organização e dos projetos.

Os cálculos e análises de resultados das medições realizadas são elaborados e apresentados tanto no âmbito da Organização como um todo quanto no âmbito de cada projeto. Os resultados obtidos pela Organização representam a "consolidação" dos resultados obtidos por cada um de seus projetos.

Para que os seus objetivos sejam alcançados, a ferramenta *Metrics*:

- calcula as métricas da Organização e de cada projeto separadamente;
- auxilia a execução do Processo de Medição e Análise da Organização através da disponibilização de gráficos para visualização dos valores e resultados das métricas calculadas e do apoio à análise dos resultados e geração de relatórios.

A figura 4.7 apresenta a interface básica da ferramenta.

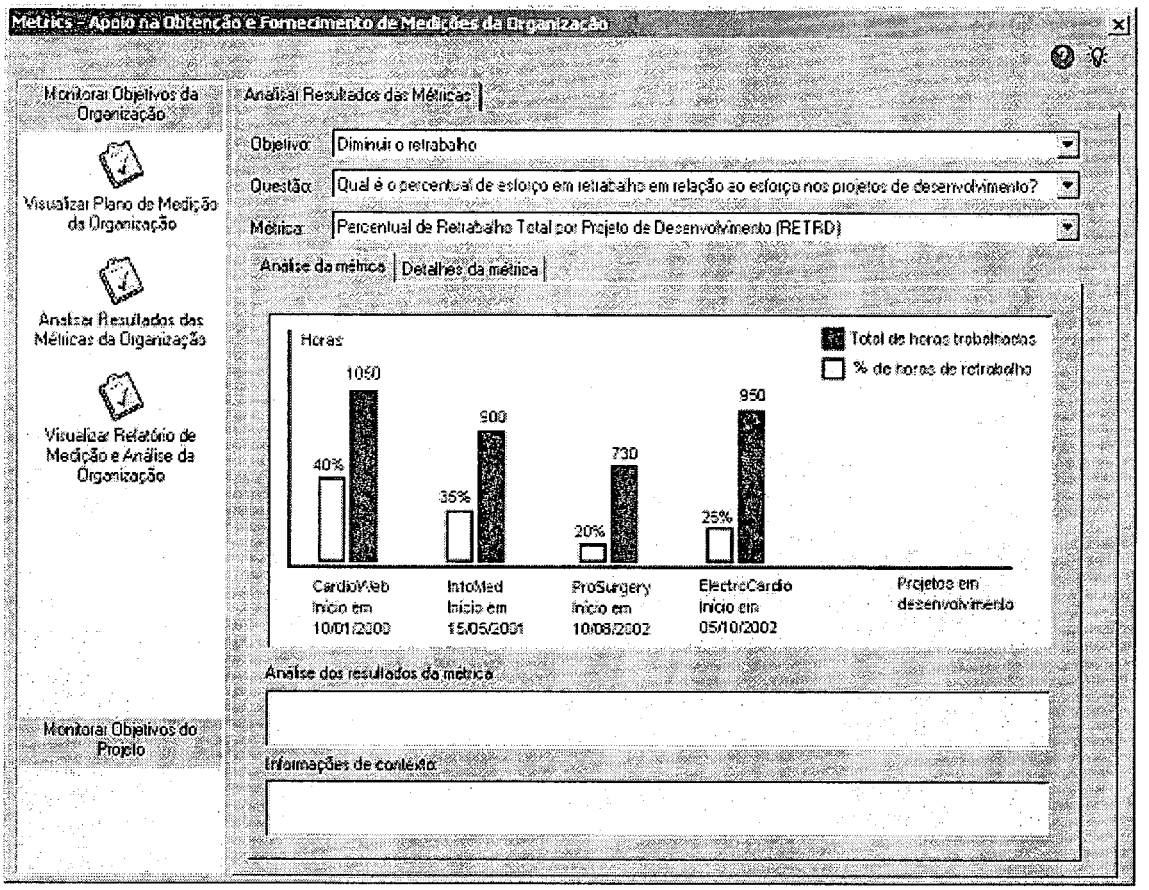

Figura 4.7 – Tela da sub-atividade "Analisar Resultados das Métricas da Organização" da Ferramenta Metrics

No lado esquerdo da interface pode-se identificar as atividades "Monitorar Objetivos da Organização" e "Monitorar Objetivos do Projeto", e suas respectivas sub-atividades. No lado direito identifica-se a sub-atividade que está sendo realizada pelo usuário. Assim como na ferramenta *MedPlan*, os ícones localizados abaixo da barra de título permitem a busca e o registro de conhecimento no que diz respeito às atividades do processo e ao conhecimento explícito em relação ao processo de Medição e Análise são disponibilizados através de interface com a ferramenta de Aquisição do Conhecimento *Acknowledge* (MONTONI *et al.*, 2004).

Na atividade selecionada ("Monitorar Objetivos da Organização"), o usuário pode visualizar o Plano de Medição da Organização na sub-atividade "Visualizar Plano de Medição da Organização. A tela da figura 4.7 apresenta detalhes da sub-atividade "Analisar Resultados das Métricas", onde o usuário tem acesso aos resultados das métricas que fazem parte do Plano de Medição da Organização.

Para visualizar o resultado de cada métrica o usuário deve selecionar um dos objetivos da Organização, uma das questões associadas a este objetivo e uma das

métricas associadas a esta questão. No quadro "Análise Resultado da Métrica" a ferramenta *Metrics* apresenta os resultados da métrica selecionada nos vários projetos da Organização. O usuário tem espaço então para registrar a análise dos resultados da métrica apresentada e, opcionalmente, informações de contexto. Na sub-atividade "Visualizar Relatório de Medição e Análise da Organização" a ferramenta apresenta o resultado de todas as métricas da Organização e seus respectivos gráficos acumulativos para que o usuário visualize, imprima e envie aos profissionais responsáveis.

4.7. Considerações finais

Neste capítulo foi apresentada a Estação TABA e os Ambientes de Desenvolvimento de Software Orientados a Organização. Foram também apresentadas as ferramentas já disponíveis na Estação e uma visão geral do apoio dado à área de Medição e Análise nos Ambientes TABA. Com isso, situou-se o foco deste trabalho – Base de Métricas – no contexto da Estação TABA.

Para fornecer infra-estrutura ao processo de medição e análise, foi definida por este trabalho uma Base de Métricas – uma estrutura capaz de comportar métricas, questões e objetivos para serem utilizados durante a construção do plano de medição da organização ou de um projeto. A Base é descrita com detalhes no próximo capítulo.

Capítulo 5

A Base de Métricas da Estação TABA

Este capítulo descreve a Base de Métricas, objeto deste trabalho, elemento fundamental definido e implementado para dar suporte a medição e análise dos Ambientes TABA.

5.1. Fundamentos e requisitos da Base de Métricas

Um importante elemento compartilhado por diversos métodos e técnicas de avaliação que apóiam à garantia de qualidade é a especificação de requisitos não-funcionais, baseada em uma definição e documentação apropriada de atributos e métricas para entidades (produtos, processos, recursos e suas sub-entidades). Dada a heterogeneidade das fontes de informação relativas a métricas em uma organização, é importante prover mecanismos para estruturação, classificação, recuperação e uso das informações, de modo que possam ser úteis para as diversas atividades relacionadas à garantia da qualidade. (OLSINA *et al.*, 2003).

Para essa finalidade, repositórios de métricas e ambientes de catalogação podem fornecer para os diferentes envolvidos (*stakeholders*) meios para consulta, busca e recuperação e mecanismos de reuso. Os repositórios são construídos a partir da definição da métrica, do tipo de medição, da fórmula para cálculo da métrica, da unidade e do tipo de escala, dos instrumentos para coleta de dados, das regras de contagem (definição operacional), da especificação do tipo de entidade e do atributo medido, entre outros metadados necessários.

Esforços iniciais foram feitos recentemente para se classificar métricas para algum tipo de entidade como, por exemplo, métricas para produtos de software: ISO/IEC 9126-1 (2000) e os *drafts* ISO/IEC 9126-2 (2002) e ISO/ISC 9126-3 (2002). Porém, o *template* (roteiro no estilo de formulário) em papel para se descrever tais métricas não é suficientemente completo, e não é o ideal para ser automatizado por ferramentas de busca, recuperação e medição.

Também é de menção importante o esforço realizado por KITCHENHAM *et al.* (2000) na definição de um *framework* e uma infra-estrutura conceitual (baseada no modelo entidade-relacionamento) para especificar entidades, atributos e

relacionamentos para medição e instanciação de projetos de software, com o objetivo de analisar métricas e conjuntos de dados de um modo consistente.

A Base de Métricas tem como objetivo ser um repositório para um catálogo de métricas, que possam ser utilizadas para definição de planos de medição nos ambientes configurados TABA. A base de métricas possibilita o registro de métricas de forma estruturada e detalhada.

Além das métricas, a Base também contempla uma estrutura para aplicação do GQM na construção de planos de medição, de modo a poder receber objetivos, questões e suas relações com as métricas. A estrutura inicial definida por este trabalho destinava-se a conter apenas a definição das métricas (e também das questões e objetivos), e não o resultado de sua aplicação (medições). No entanto, durante o desenvolvimento das ferramentas, a estrutura foi estendida para permitir a coleta e análise de dados.

5.1.1 Fundamentos e Requisitos relativos a Métricas

Como definido em (BRIAND *et al.*, 1996), uma métrica é um mapeamento entre o mundo real (dos produtos e processos) no mundo formal (matemático). Mais especificamente, através de uma métrica é feita a representação, através de um símbolo (pertencente a um conjunto pré-estabelecido) de um determinado atributo de uma determinada entidade.

Os conceitos fundamentais inerentes à definição e à classificação das métricas também devem ser contemplados na estrutura definida. A Base possibilita o registro de métricas diretas ou indiretas, atômicas ou calculadas a partir de outras. Outros conceitos como unidade de medida também são suportados.

Métricas normalmente são utilizadas em comparações e podem comparar o mesmo atributo para atividades ou produtos diferentes (ex.: comparar a precisão de uma estimativa de tempo para análise com a precisão da estimativa de tempo para implementação) ou comparar o mesmo atributo de uma mesma entidade em momentos distintos (ex.: número de requisitos do sistema logo após a análise e número de requisitos no momento em que se está entregando o sistema ao cliente), ou ainda comparar o mesmo atributo para uma mesma entidade entre projetos distintos (ex: comparação da densidade de defeitos dos projetos).

Para que se possa realizar comparações, a definição de uma métrica precisa ser robusta o suficiente para que medições feitas em momentos distintos, em contextos

distintos, ou por pessoas diferentes estejam efetivamente medindo o mesmo atributo da mesma forma, permitindo comparabilidade. Um requisito da Base, portanto, é que se possa especificar claramente o que a métrica vai medir e como vai fazê-lo.

Portanto, a definição da métrica deve ser feita de tal forma que sua aplicação seja repetível e reprodutível, ou seja, que dadas as mesmas circunstâncias, mesma entidade e mesmo atributo, o resultado da aplicação da métrica seja o mesmo, ainda que o "operador" seja outro. Para atingir esse objetivo, é importante que na descrição da métrica estejam explícitos sua definição operacional e sua fórmula de cálculo (para métricas indiretas).

Métricas são mapeamentos entre o mundo real e o mundo matemático, é necessário dizer que atributo de que entidade será medido, qual será o procedimento operacional para medição (ou seja, se for uma contagem, o que deve ser incluído ou excluído nessa contagem, se for uma métrica subjetiva ou que utilize uma escala categórica é importante deixar claro qual o critério utilizado para realizar a classificação). A métrica pode estar ligada a uma categoria de entidades em vez de uma entidade em particular (ex: uma métrica de precisão de estimativa de custo de uma atividade pode estar ligada a qualquer entidade que seja da categoria atividade). Do lado matemático é necessário também rigor na definição do tipo de escala utilizado, nos elementos que compõem essa escala, nos valores máximo e mínimo e, quando for o caso, da unidade de medida utilizada.

É importante deixar claro também qual o objetivo da métrica, em que situações se aplica, em que casos não pode ser utilizado e, se for o caso de coleta automatizada, que ferramenta será responsável pela coleta ou que formulário será utilizado no caso de coleta manual.

Nem todas as métricas são diretas ou atômicas, isto é, obtidas através de um único dado coletado. Assim, é necessário também definir as regras de cálculo, composição ou consolidação dos dados, bem como os itens de dados, que serão utilizados no cálculo. Para cada um desses itens, também é necessário indicar exatamente o que deverá ser medido e qual será o procedimento de medição. É também necessário informar no caso de métricas que medem entidades que são compostas por outras entidades, se será coletado/informado um único valor já consolidado ou se a métrica deverá ser calculada a partir da aplicação da métrica em cada uma das sub-entidades (ex: o esforço efetivamente demandado por uma atividade pode ser informado como um único valor ou pode ser calculado a partir do esforço demandado por cada

uma das sub-atividades que compõem esta atividade). Esse critério de composição ou de informação de um único valor pode ser informado no momento da especificação do plano de medição e é particularmente importante, durante a análise das medições, pois irá indicar o grão mínimo para a leitura das medições e conseqüentemente para localização mais pontual de eventuais problemas ou pontos de melhoria.

No momento de registrar uma métrica, para efeito de catalogação, é importante indicar a que categoria a métrica pertence (ex: custo e cronograma; qualidade etc.), as referências (de que livros ou artigos a métrica foi retirada) e eventualmente um conjunto de palavras chaves para localização da métrica. Outro atributo necessário é uma referência para outras métricas ("ver também").

5.1.2 Fundamentos e Requisitos relativos a Planos de Medição

Para análise e interpretação das medições realizadas através de uma métrica, é importante que seja definido o procedimento para análise: como os resultados deverão ser interpretados, o que se espera como um valor bom ou ruim, que tipo de gráfico ou tabela é sugerido para auxiliar na análise. É possível também que junto à definição da métrica esteja um valor ou uma faixa de valores esperados, considerando o histórico ou expectativa da organização ou um valor obtido a partir de um consenso ou de um *benchmark* entre outras organizações.

Como o objetivo principal da Base de Métricas é servir como subsídio para construção de planos de medição através do GQM, também foi considerada uma estrutura para a definição dos planos de medição e para aplicação do GQM. Um objetivo (*goal*) pode ser da organização ou especifico de um projeto e está ligado a questões que por sua vez apontam para métricas.

No plano de medição, além das métricas que serão utilizadas, deverão ser indicados quais foram os relacionamentos de objetivos, questões e métricas utilizados e, para cada métrica, poderão ser especificados procedimentos complementares ou alternativos para coleta e para análise e também os valores esperados especificamente para um projeto. Durante a confecção do plano de medição podem também ser atribuídas responsabilidades e indicação de ferramentas alternativas.

5.2. Descrição dos conceitos necessários à Base

A tabela 5.1 resume os conceitos necessários e utilizados na definição da Base, em função dos fundamentos e requisitos levantados na sessão anterior.

Tabela 5.1 – Conceitos necessários à definição da Base de Métricas

Conceito	Descrição do conceito	Exemplos de conteúdo
Nome	Nome dado à métrica.	Horas trabalhadas, custo estimado, densidade de erros.
Categoria	Categoria (de acordo com algum *framework* ou taxonomia a ser definido) à qual pertence a métrica.	Processo, Produto, Cronograma
Objeto de medição	Indica o que a métrica visa medir. Em geral, é composto por atributo e entidade. Tem relação com uma outra estrutura de dados que define as entidades e seus atributos.	Atributos: duração, quantidade, valor, custo. Entidade: atividade, código-fonte, módulo.
Descrição	Descrição textual breve do objetivo da métrica.	(descrição textual)
Definição operacional	Descrição textual detalhada do procedimento para utilização da métrica, ou seja, como devem ser realizadas as medições com a métrica, explicando como mapear os atributos medidos à escala proposta pela métrica, de modo a atingir de forma repetível e reprodutível o objetivo da métrica.	(descrição textual)
Unidade	Unidade em que será expressa uma medição feita através da métrica.	N (contagem), dias, homem/hora, KLOC
Tipo de escala	Tipo de escala utilizada para a métrica.	Nominal, Ordinal Intervalo, Racional, Absoluta.
Itens de dados	Valores de entrada que são necessários para se calcular a métrica.	Dias, Erros encontrados.
Equação de cálculo	Descrição dos passos necessários para o cálculo das métricas, a partir dos itens de dados.	X = somatório(Xi) X = média(Xi)
Direta ou indireta	Indica se a métrica é direta (aquela na qual não existe dependência da medição de outro atributo; também chamada de atômica) ou indireta.	Direta ou Indireta
Composição (métricas indiretas)	Caso a métrica seja indireta, contém a relação das outras métricas necessárias a seu cálculo.	Métrica de densidade de erros: é composta por número de erros encontrados no artefato E pelo tamanho do artefato.
Equação de composição (métricas indiretas)	Caso a métrica seja indireta, contém a equação para seu cálculo a partir das métricas primordiais.	Met3 = Met1 / Met2

Procedimento de coleta	Descrição de como os dados deverão ser coletados para a métrica, incluindo freqüência, condições, instrumento de coleta, responsável etc.	(descrição textual)
Procedimento para análise	Informações sobre como proceder com a análise dos resultados da métrica, que tipo análise utilizar etc.	
Referências	Referências bibliográficas de onde a métrica foi obtida; referências para uso.	Artigo XX, Livro YY, Norma ZZ, Link de Internet KK etc.
Palavras-chaves	Palavras-chaves (ou expressões) pelas quais a métrica pode ser localizada.	Qualidade, custo, erros.
Ver também	Outras métricas semelhantes ou relacionadas.	Pode indicar métricas para o mesmo fim mas com outra granularidade, por exemplo.
Observações	Qualquer observação ou atributo sobre a métrica que por enquanto não se encaixe em nenhum dos campos já definidos.	(descrição textual)
Objetivo (GQM)	Objetivo de um plano de medição GQM	"Aumentar a produtividade no desenvolvimento"
Questão (GQM)	Questão de um plano de medição GQM	"Qual é a produtividade das equipes dos projetos de desenvolvimento?"
Métrica (GQM)	Métricas de um plano de medição GQM	Produtividade da Equipe nos projetos de desenvolvimento
Plano de medição	Plano de Medição, feita para a organização ou para um projeto, utilizando a abordagem GQM.	(plano GQM)

5.3. Estrutura da Base de Métricas TABA

A estrutura da Base de Métricas foi especificada a partir dos requisitos e dos atributos descritos nas seções anteriores, e definidos através de um diagrama de classes UML, numa modelagem conceitual (figuras 5.1 e 5.2).

O núcleo da estrutura consiste na classe métrica, e nas suas relações com as classes atributo e entidade. De acordo com a especificação (definição) de métrica, pode ser necessário coletar mais de um dado (mais de uma informação) para chegar no valor da métrica, ou pode ainda ser necessária alguma transformação matemática (transformação de escala, por exemplo). A métrica pode ainda ser resultado de uma consolidação estatística ou matemática de um conjunto de dados semelhantes, como, por exemplo, do número médio de horas necessário para codificação de uma classe definida no projeto, que é calculado a partir da média dos tempos (em horas ou minutos,

de acordo com a definição da métrica) demandados no desenvolvimento de cada classe de um determinado sistema. Em outro exemplo, o tempo total para execução de um processo seria a soma do tempo utilizado em cada em cada atividade do processo.

Além das classes Objetivo (*"Goal"*), Questão (*"Question"*), que se relacionam com a classe métrica, também foram sugeridos no modelo classes de relacionamento para conter as ligações criadas entre objetivos e questões (que questões são relacionadas a cada objetivo), e entre questões e métricas (que métricas responde cada questão) no momento da criação do plano de medição.

As questões, objetivos e métricas irão compor o plano de medição. Para cada métrica que compõe o plano, pode ser necessário também definir informações operacionais específicas do projeto e/ou da organização, tais como freqüência de coleta e formato de apresentação.

5.3.1. Modelagem conceitual

A análise do modelo conceitual é a base para projetar e implementar repositórios de métricas (e sistemas de catalogação) para diferentes entidades, atributos e domínios. O modelo conceitual especifica as principais classes e relacionamentos para o domínio de métricas, necessários para se definir e explorar todas as informações para fins de catalogação.

Esse modelo conceitual não tem como objetivo o armazenamento e o processamento de instâncias de valores de métricas (ou seja, as medições) para diferentes atributos de entidades de projetos.

Do ponto de vista do processo de medição, no domínio empírico (PFLEEGER e FENTON, 1997) as classes principais são ENTIDADE e ATRIBUTO. Confirme visto no capítulo 2, a classe ENTIDADE representa objetos tangíveis ou intangíveis que são observados no mundo real (projeto, produto, processo e recurso, por exemplo). Uma ENTIDADE pode ser hierarquicamente partida em sub-entidades e, nesse caso a entidade de escopo maior é referenciada como ENTIDADE_PAI.

A classe ATRIBUTO representa o que é observado e atribuído considerando a entidade à qual pertence, sendo uma propriedade que interesse se avaliar. Para um dado atributo, há, no mínimo, um relacionamento empírico de interesse que pode ser capturado e representado em um domínio formal (através de uma métrica), propiciando a exploração do relacionamento matematicamente (FENTON e PFLEEGER, 1997).

ATRIBUTO é um objeto no nível "mais baixo" de abstração que pode ser quantificado diretamente ou indiretamente por uma métrica.

A classe METRICA representa o mapeamento específico de um atributo empírico (que pertence a uma entidade) para o mundo formal, além do critério específico e procedimentos para medi-lo. É importante ressaltar a diferença dessa definição com a dada em (KITCHENHAM, PFLEEGER e FENTON, 1995). No contexto deste trabalho, uma medida (aqui chamada de "item de dados") é parte de uma métrica. Uma ou mais métricas podem quantificar o mesmo atributo, através de diferentes procedimentos operacionais e/ou diferentes escalas e/ou unidades.

O campo definicaoOperacional especifica as regras de contagem e procedimentos que devem ser aplicados a cada métrica, servindo como um guia para a coleta de dados. No caso de métricas com procedimento de coleta automatizado, o algoritmo a ser utilizado pode ser especificado nesse campo.

O TIPO DE ESCALA de uma métrica afeta o conjunto de operações estatísticas e matemáticas que podem ser aplicados a um valor medido. Um tipo de escala é definido por transformações admissíveis (FENTON, 1994), conforme visto em detalhes no capítulo 2 desta dissertação. No modelo proposto, os tipos de escala são: nominal, ordinal, intervalo, racional e absoluta.

As classes METRICA_DIRETA e METRICA_INDIRETA são tipos de métricas, expressos através de herança no modelo conceitual. Seus conceitos no domínio formal são estritamente relacionados com a quantificação de atributos diretos e indiretos (no domínio empírico). Por exemplo, se X é uma variável dependente, então, para uma métrica indireta (de um atributo indireto), o valor X pode ser obtido a partir de um modelo matemático, uma equação que modela as dependências entre os atributos. O atributo equacaoComposicao especifica formalmente como o valor da variável pode ser obtido/calculado. Para uma métrica direta, o valor é computado diretamente.

Toda informação presente no modelo conceitual é necessária para ter uma especificação correta e completa dos atributos e métricas para fins de catalogação, consulta e uso automatizado.

O modelo apresenta outras classes, referentes à catalogação e definição do plano de medição baseado na abordagem GQM. A tabela 5.2 descreve as classes e atributos do modelo conceitual.

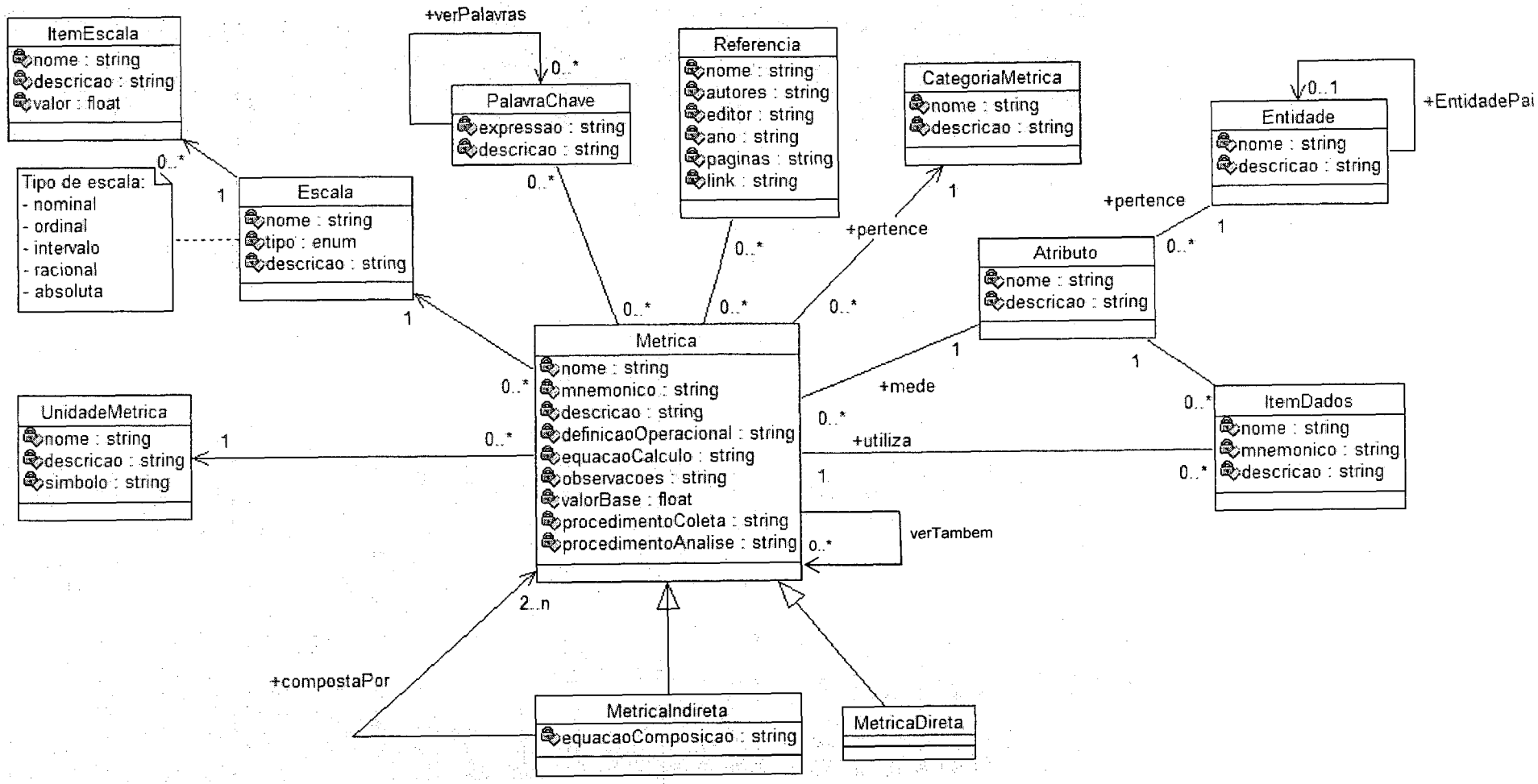

Figura 5.1 – Modelo Conceitual da Base de Métricas (parte 1)

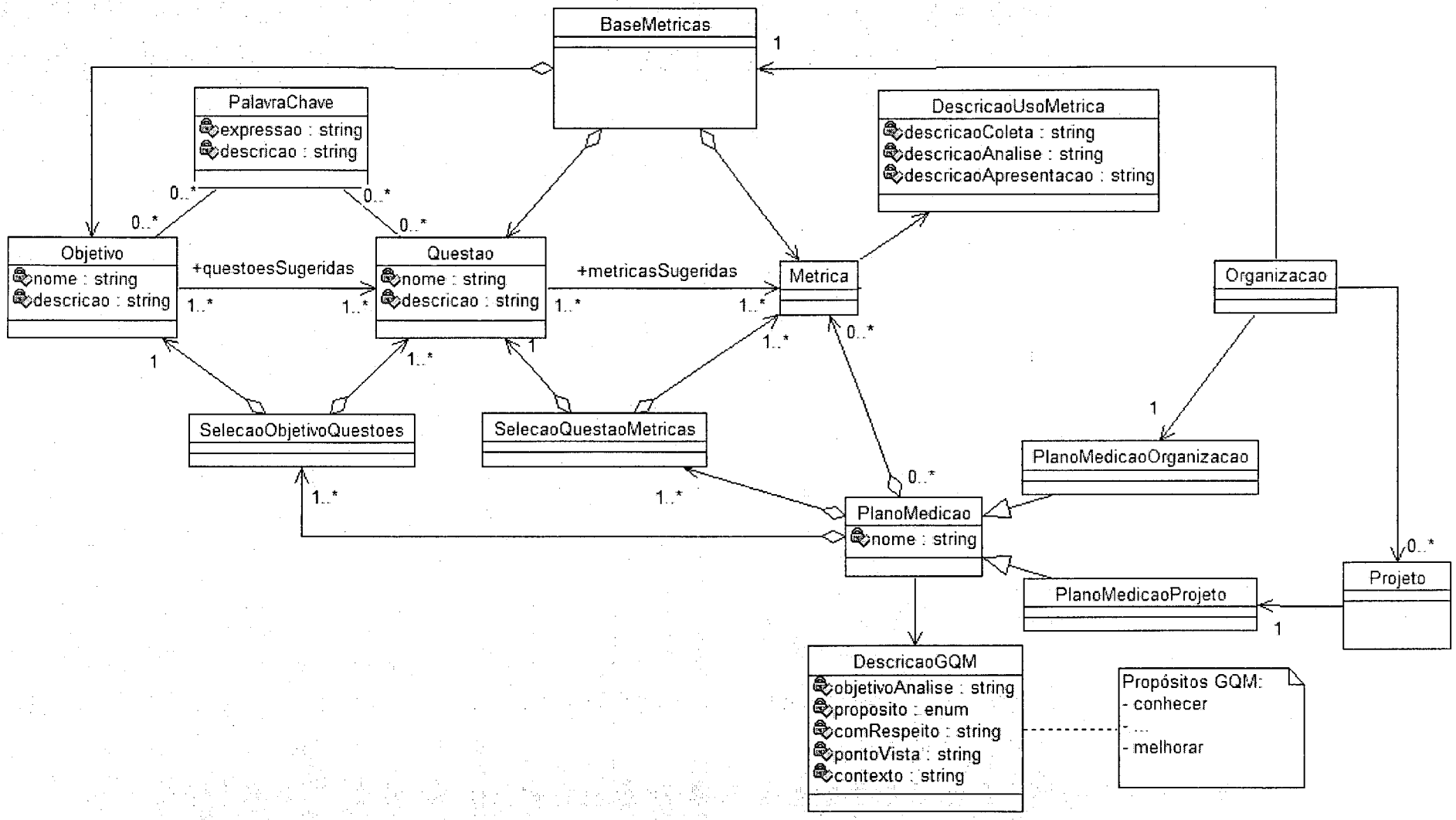

Figura 5.2 - Modelo Conceitual da Base de Métricas (parte 2)

Tabela 5.2 – Descrição das classes e principais atributos do modelo conceitual

CLASSE	*Descrição*
Atributo	Descrição
Metrica	*Classe fundamental do modelo*
nome	Expressão que define de forma concisa e única a métrica.
mnemonico	Identificador da métrica para ser usado nas fórmulas de cálculo de composição, equivalente ao nome de uma variável.
descricao	Descrição dos objetivos, finalidades e aplicação da métrica.
definicaoOperacional	Descrição detalhada de como a métrica deverá ser aplicada. No caso de escalas numéricas pode indicar quando começa a contagem, o que deve ser considerado e o que deve ser excluído da contagem e quando a mesma é interrompida. Quando é uma escala ordinal ou categórica indica que critérios devem ser considerados para seleção de uma ou outra categoria. Para métricas compostas ou indiretas, o procedimento também descreve como será realizado o cálculo de composição ou consolidação. A definição operacional é que garantirá que a métrica possa ser aplicada por outra pessoa ou em outras circunstâncias e que os resultados das medições sejam compatíveis e comparáveis.
equacaoCalculo	Define como a métrica será calculada a partir dos itens de dados. A equação é expressa através de uma fórmula matemática composta de mnemônicos (representando os itens de dados) e operadores aritméticos.
observacoes	Observações sobre o uso da métrica.
procedimentoColeta	Informações completas de como, quando, com que freqüência e por quem a métrica será aplicada.
procedimentoAnalise	Descreve como os resultados da métrica deverão ser analisados, incluindo o tipo de apresentação que será utilizado para análise (tabela, gráfico de pizza, gráfico de barras, linha de tendência etc.). Neste procedimento pode ser informado também se o valor obtido será confrontado com algum valor de referência.
verTambem	Indica outras métricas que possam ter finalidade semelhante.
ItemDados	*Indica o conjunto de dados a partir dos quais a métrica é composta. Todos os dados utilizados na equação de cálculo devem ser de itens de dados que estão presentes neste conjunto e vice-versa.*
MetricaDireta	*Classe herdada de Metrica para representar métricas diretas, também conhecidas como atômicas.*
MetricaIndireta	*Classe herdada de Metrica para representar métricas indiretas, também conhecidas como não-atômicas. São métricas compostas por outras métricas.*
equacaoComposicao	Define como a métrica indireta será calculada a partir das

	métricas que a compõem. A equação é expressa através de uma fórmula matemática composta de mnemônicos (representando as métricas subjacentes) e operadores aritméticos.
compostaPor (métrica)	Indica o conjunto de métricas a partir das quais a métrica indireta é composta. Todos os mnemônicos utilizados na equação de cálculo devem ser de métricas que estão presentes neste conjunto e vice-versa.
Escala	*Tipo de escala utilizado na métrica.*
ItemEscala	*Itens que compõem a escala, caso seja uma escala categórica.*
UnidadeMetrica	*Indica a unidade de medida da métrica.*
PalavraChave	*Indica o conjunto de palavras-chaves através da qual a métrica pode ser localizada por meio de buscas.*
Referencia	*Referências (livros, artigos, normas etc.) onde a métrica é definida e/ou descrito seu uso ou aplicação.*
CategoriaMetrica	*Indica a categoria à qual pertence a métrica, dentro de um conjunto pré-definido de categorias (ex: métricas satisfação do cliente, métricas de cronograma/custo etc.).*
Atributo	*Nome do atributo que será medido pela métrica. Esse atributo deve existir no tipo de entidade indicada em Entidade (ex: esforço).*
Entidade	*Indica que tipo ou categoria de entidade a métrica visa medir; deve ser o nome de um tipo de entidade do TABA (ex: atividade).*
PlanoMedicao	*Representa um plano de medição contendo métricas questões e objetivos, relacionadas através das classes de relacionamento SelecaoObjetivoQuestoes e SelecaoQuestaoMetricas.*
PlanoMedicaoOrganizacao	*Especialização da classe PlanoMedicao para o caso de planos de medição para a organização. Uma organização possui um e somente um plano de medição.*
DescricaoGQM	*Descrição formal do plano de medição de acordo com a abordagem GQM.*
PlanoMedicaoProjeto	*Especialização da classe PlanoMedicao para o caso de planos de medição para projetos.*
Organizacao	*Classe do modelo de dados TABA que representa a organização.*
Projeto	*Classe do modelo de dados TABA que representa o projeto. Uma organização possui um ou vários projetos,*

	que são acompanhados através de um plano de medição da organização e opcionalmente por planos de medição de cada projeto.
Objetivo	*Objetivo do plano de medição; através do relacionamento questoesSugeridas, são identificadas na Base as questões referentes ao objetivo.*
Questao	*Questão do plano de medição; através do relacionamento metricasSugeridas, são identificadas na Base as Métricas referentes à questão.*
SelecaoObjetivoQuestoes	*Indica quais foram as questões efetivamente selecionadas para atender um objetivo utilizado no plano de medição. É um subconjunto das questões sugeridas.*
SelecaoQuestaoMetricas	*Indica quais foram as métricas efetivamente selecionadas para atender uma questão utilizada no plano de medição. É um subconjunto das métricas sugeridas.*
DescricaoUsoMetrica	*Para cada métrica presente no plano de medição é possível fornecer instruções adicionais sobre coleta, análise e apresentação, além daquelas já cadastradas junto com a métrica.*
BaseMetricas	*Entidade que representa a Base como um todo contendo objetivo, questões e métricas.*

5.4. Implementação da Base de Métricas

A Base de Métricas foi implementada no *framework* já existente da Estação TABA. Cabe ressaltar que foi implementada uma versão simplificada da Base definida por este trabalho, o suficiente para atender aos processos de medição e análise definidos. A redução do formalismo no modelo implementado deveu-se basicamente a restrições de tempo para liberação e uso das ferramentas Metrics e MedPlan, e também buscando facilidade na definição de novas métricas pelos usuários finais da Estação. Por outro lado, o modelo previsto teve a adição de algumas classes e atributos referentes ao mecanismo de coleta e análise de dados, assim como para integração com o modelo de dados TABA já existente. O diagrama das classes implementado é apresentado na figura 5.3 .

As principais alterações no modelo estão relacionadas abaixo:

- O conceito de "item de dados" foi eliminado. Uma métrica atômica passa a registrar as medidas mais simples necessárias.

- As classes MetricaIndireta e MetricaDireta foram implementadas através de um atributo da classe Métrica chamado "atomica", que apresenta valor "verdadeiro" (*true*) quando a métrica é direta (atômica), e "falso" quando contrário.
- Os atributos "procedimentoColeta" e "procedimentoAnalise" foram transformados em classes, pela necessidade de informações adicionais e estruturadas para a coleta e a análise de dados, respectivamente.
- As classes Entidade e Atributo foram implementadas através de referências, no procedimento de coleta, a entidades e atributos do modelo TABA.
- Foi criada a classe RegraMetrica para apoiar na coleta de dados, restringindo a entidades e atributos que satisfaçam as regras definidas (ex.: coletar métricas somente de atividades concluídas).
- Não foram implementadas as classes e conceitos referentes à catalogação (PalavraChave, Referencia, CategoriaMetrica)
- O conceito de unidade foi implementado de forma simplificada, através de um atributo "string" (perdendo, dessa forma, carga semântica).
- Foi criada uma classe Medida, para registrar os valores coletados a partir da aplicação das métricas.
- Os atributos da classe DescricaoGQM foram incorporados à classe PlanoMedicao.
- As classes planoMedicaoOrganizacao e planoMedicaoProjeto foram implementadas através do relacionamento existente ou não da classe PlanoMedicao com as classes Projeto e Organização.
- As classes de relacionamento selecaoObjetivoQuestoes e selecaoQuestaoMetricas foram substituídas por outros relacionamentos entre as classes Objetivo / Questao e Questao / Métricas, respectivamente.

Quando se implementou a Base, foi cadastrado um conjunto inicial de objetivos, questões e métricas. Para que essas métricas pudessem ter sua coleta automatizada já na primeira versão das ferramentas, o algoritmo relativo à definição operacional dessas métricas foi implementado diretamente no código do TABA. Para as métricas que o usuário cadastra a partir da Ferramenta Metrics (seção 5.5), o procedimento é descrito textualmente.

Nessa implementação, foi adotado um mecanismo de persistência com infra-estrutura para representação de medidas em formato XML, permitindo integração com processos e ferramentas já existentes nas organizações para coleta de dados de métricas. Os sistemas existentes podem, então, exportar as medidas, utilizando XML, para o repositório de medições, e a ferramenta Metrics coleta os dados a partir dessa estrutura. Para tornar a Metrics mais genérica, as medidas geradas pelas ferramentas internas do TABA também são exportadas nesse mesmo repositório XML.

A figura 5.3 apresenta o modelo de classes da Base de Métricas implementada na Estação TABA com as reduções e extensões incorporadas ao modelo proposto neste trabalho. São apresentadas as classes acrescidas para atender a Base de Métricas e suas descrições, assim como seu uso durante as medições.

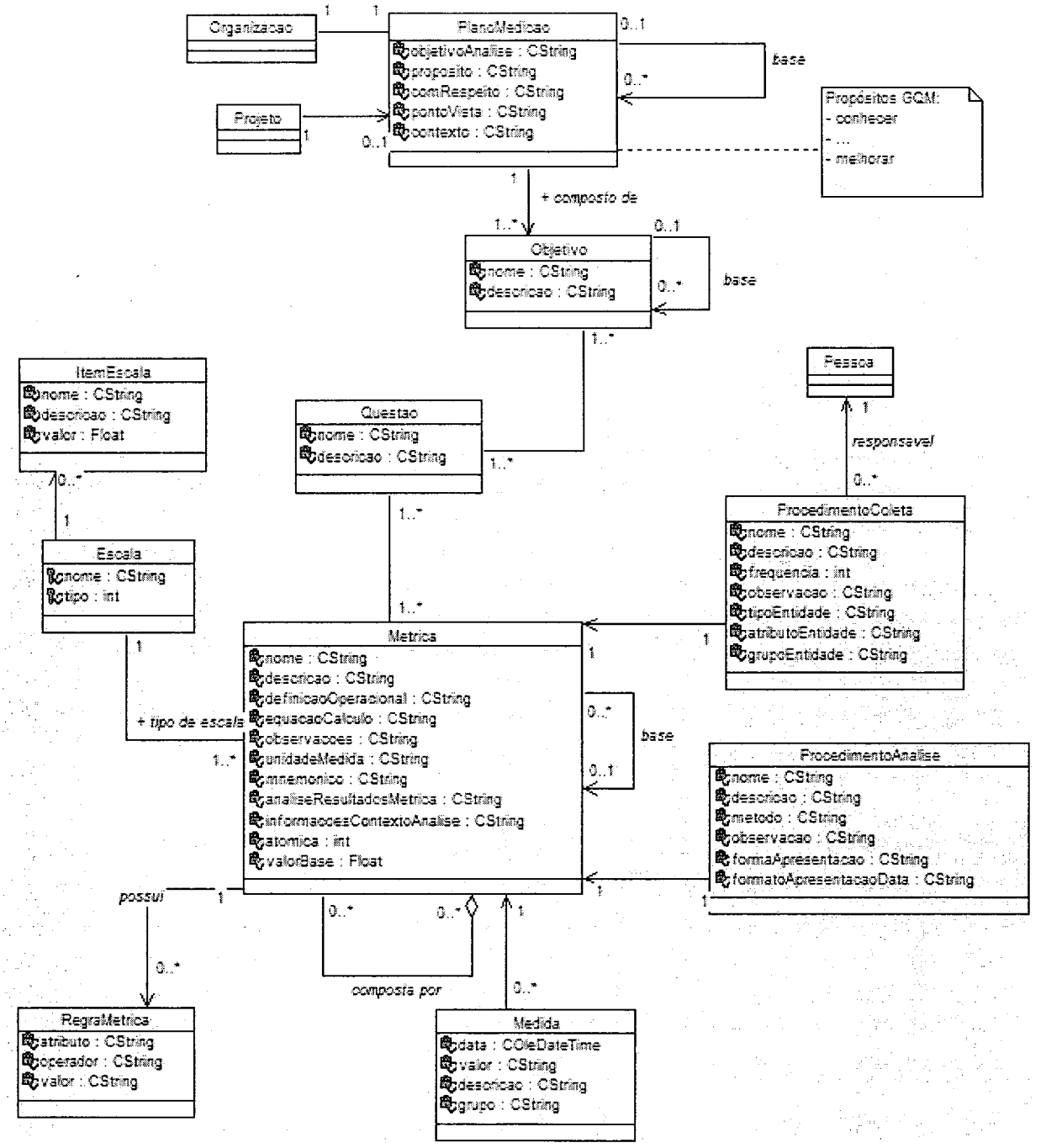

Figura 5.3 – Classes do modelo de Base de Métricas implementado no TABA

5.5. Utilização da Base de Métricas

Nos ambientes TABA, a interface do usuário com a Base de Métricas é feita através das ferramentas Metrics e Medplan. Através da MedPlan, é possível:

- Fazer a manutenção da Base, sendo possível incluir, alterar e excluir métricas, questões e objetivos (ambiente configurado).
- Definir o plano de medição da organização (ambiente configurado) e de seus projetos (ambiente configurado), a partir da seleção de objetivos, questões e métricas cadastrados na Base.

Na definição do plano de medição, após selecionar o(s) objetivos de medição, para cada objetivo são selecionadas dentre as questões cadastradas para aquele objetivo, as questões pertinentes à organização ou projeto. Uma vez selecionado o conjunto de questões, para cada um delas seleciona-se uma ou mais métricas dentre as métricas previamente cadastradas para cada questão, concluindo-se o plano de medição.

Para cada métrica selecionada, é possível definir ou alterar:

- um valor base (um valor de referencia)
- o procedimento de análise (indicando como será feita a leitura dos resultados, tipo de gráfico, granularidade de leitura)
- o procedimento de coleta (definindo, por exemplo, se as métricas serão coletadas no nível de atividade, fase ou projeto; ou, ainda, se o procedimento de coleta será manual ou automático, com coleta a partir do repositório do projeto).

A figura 5.4 mostra uma tela da ferramenta MedPlan. Durante a definição do plano de medição de uma organização é possível incluir novas métricas.

A ferramenta Metrics é responsável pela coleta (manual ou automática) e consolidação das métricas no repositório XML descrito na seção 5.4. É também na Metrics que ocorre a visualização dos resultados das métricas e a indicação dos resultados de análise pelo usuário. Na Metrics é ainda possível monitorar as áreas de processo da organização, através de métricas previamente definidas.

A figura 5.5 mostra uma tela da ferramenta, com a visualização de resultados de uma métrica.

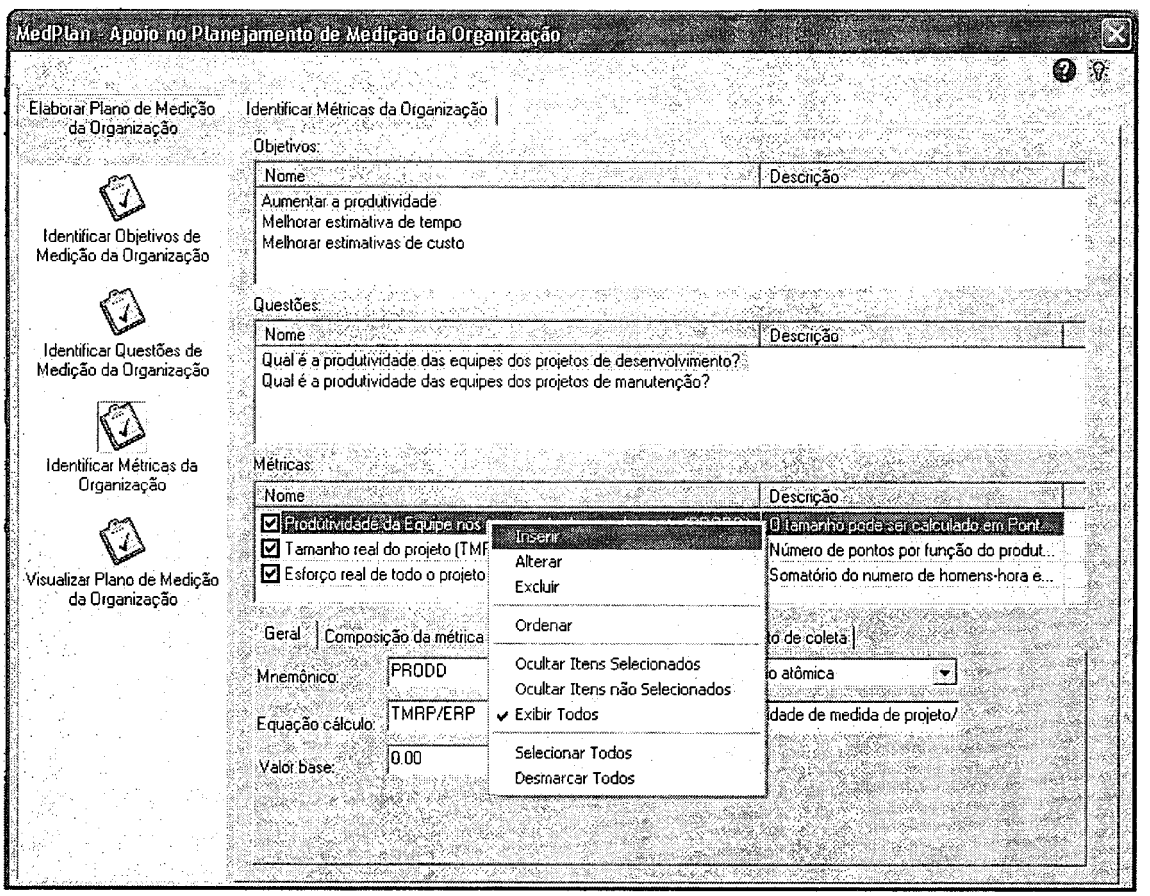

Figura 5.4 – Tela da ferramenta MedPlan: definição de plano de medição organizacional com manutenção do cadastro de métricas.

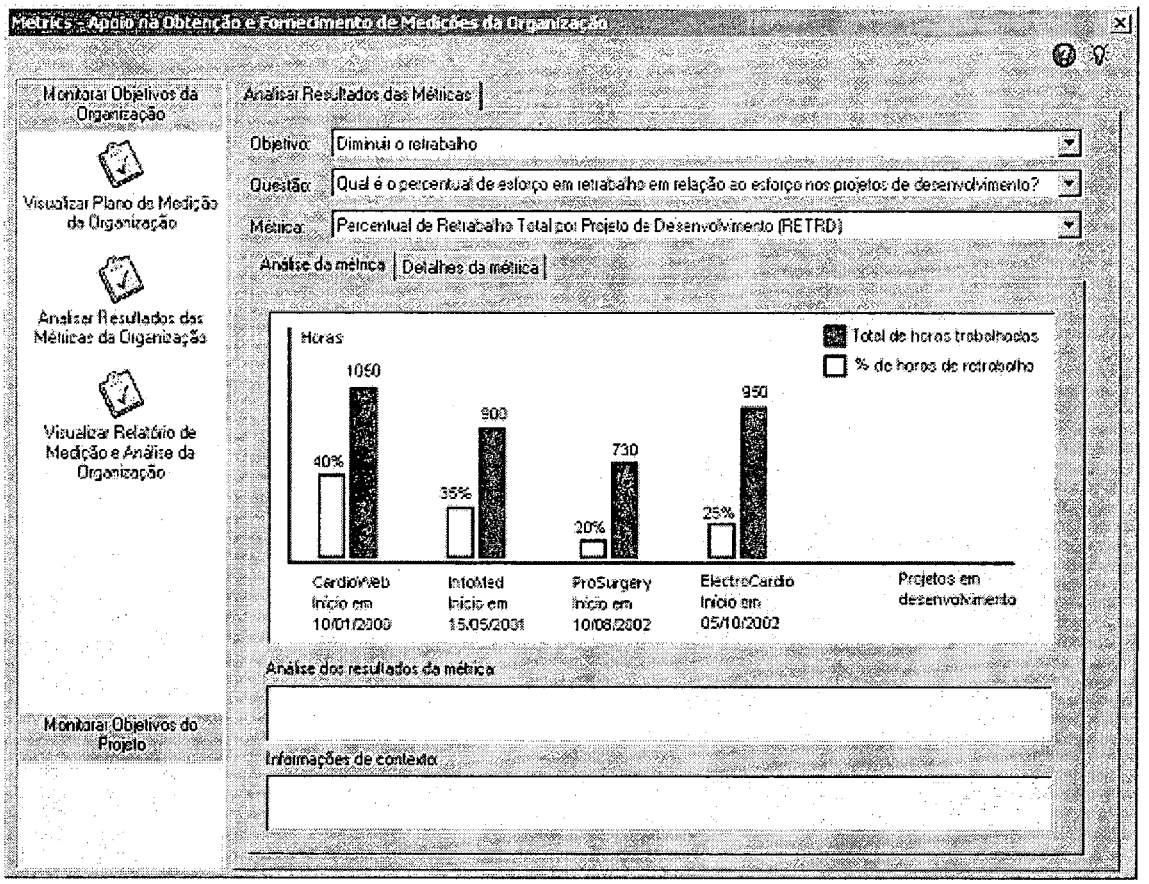

Figura 5.5 – Tela da ferramenta Metrics: visualização e análise de resultados de uma métrica

5.6. Exemplo de métricas utilizadas

A figura 5.6 mostra um relatório com exemplo de plano de medição para organização, criado através da ferramenta MedPlan, com descrição dos objetivos, questões e métricas e seus principais atributos. As métricas são apresentadas com seus principais atributos.

Plano de Medição da Organização

1. **Objetivo: Aumentar a produtividade**

    1. **Questão**: Qual é a produtividade das equipes dos projetos de desenvolvimento?

        1. **Métrica**: Produtividade da Equipe nos projetos de desenvolvimento (PRODD)

        Descrição: O tamanho pode ser calculado em Pontos por Função ou KLOC. A produtividade da equipe é o número de PF por HH ou o número de KLOC por HH.
        Atomicidade: Não atômica
        Mnemônico: PRODD
        Equação cálculo: TMRP/ERP
        Procedimento coleta: As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.
        Procedimento análise: Mensalmente é analisada a produtividade das equipes dos projetos de desenvolvimento encerrados no mês (projetos com aceitação do produto) e comparado com os resultados dos meses anteriores. Deve ser elaborado um histograma mostrando os projetos dos meses anteriores e os projetos do mês que está sendo analisado.
        Composição da métrica:

            1. Métrica: Tamanho real do projeto (TMRP)
            2. Métrica: Esforço real de todo o projeto (ERP)

        2. **Métrica**: Tamanho real do projeto (TMRP)

        Descrição: Número de pontos por função do produto ao final do projeto ou número de KLOC do produto ao final do projeto
        Atomicidade: Atômica
        Mnemônico: TMRP
        Procedimento coleta: A coleta é feita manualmente.

        3. **Métrica**: Esforço real de todo o projeto (ERP)

        Descrição: Somatório do numero de homens-hora empregados durante todo o projeto
        Atomicidade: Atômica
        Mnemônico: ERP
        Procedimento coleta: As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.

    2. **Questão**: Qual é a produtividade das equipes dos projetos de manutenção?

        1. **Métrica**: Tamanho real do projeto (TMRP)

        Descrição: Número de pontos por função do produto ao final do projeto ou número de KLOC do produto ao final do projeto
        Atomicidade: Atômica
        Mnemônico: TMRP
        Procedimento coleta: A coleta é feita manualmente.

        2. **Métrica**: Esforço real de todo o projeto (ERP)

        Descrição: Somatório do numero de homens-hora empregados durante todo o projeto
        Atomicidade: Atômica
        Mnemônico: ERP
        Procedimento coleta: As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.

        3. **Métrica**: Produtividade da Equipe nos projetos de manutenção (PRODM)

        Descrição: O tamanho pode ser calculado em Pontos por Função ou KLOC. A produtividade da equipe é o número de PF por HH ou o número de KLOC por HH.
        Atomicidade: Não atômica
        Mnemônico: PRODM
        Equação cálculo: TMRP/ERP
        Procedimento coleta: As métricas são derivadas automaticamente a partir de valores registrados no Ambiente TABA.
        Procedimento análise: Mensalmente é analisada a produtividade das equipes dos projetos de manutenção encerrados no mês e comparado com os resultados dos meses anteriores. Deve ser elaborado um histograma mostrando os projetos dos meses anteriores e os projetos do mês que está sendo analisado.
        Composição da métrica:

            1. Métrica: Tamanho real do projeto (TMRP)
            2. Métrica: Esforço real de todo o projeto (ERP)

Figura 5.6 – Exemplo de objetivos, questões e métricas cadastradas na Base de Métricas e utilizados em um plano de medição

5.7. Considerações finais

Este capítulo descreveu a Base de Métricas, objeto deste trabalho, elemento fundamental definido e implementado para dar suporte à abordagem para medição e análise dos Ambientes TABA. Foram apresentados os fundamentos e requisitos da especificação da Base, sua modelagem conceitual, a implementação nos Ambientes TABA e, brevemente, como é feita a utilização da Base nos Ambientes.

Capítulo 6

Considerações Finais e Perspectivas Futuras

Neste capítulo são apresentadas as considerações finais deste trabalho, suas contribuições e suas perspectivas futuras.

6.1. Considerações finais

Processos de medição se tornaram uma parte tão importante quanto necessária nas Organizações que desenvolvem software (McGARRY *et al.*, 2001), pois para competir em um ambiente caracterizado por rápidas e constantes mudanças, é fundamental trabalhar de maneira produtiva, eficiente e com alto nível de qualidade. É neste contexto que a medição se insere: a partir da existência de dados e análises históricas sobre a Organização é possível melhorar em muito o processo de tomada de decisão (HOLMES, 2002).

Segundo a norma de Processo de Medição de Software, ISO-15939:2001 (2002), o objetivo da medição de software é coletar, analisar e reportar dados relacionados com os produtos desenvolvidos e processos implementados na organização e em seus projetos, para suportar a gestão efetiva dos processos, e demonstrar objetivamente a qualidade dos produtos.

A medição de software apóia o gerenciamento de projetos e a melhoria dos processos e produtos de software, funcionando como ferramenta essencial para gerenciamento das atividades do ciclo de vida de software, avaliação da viabilidade dos planos de projetos e monitoração da aderência das atividades dos projetos a esses planos. A medição de software é também uma disciplina chave na avaliação da qualidade de produtos de software e da capacidade dos processos de software organizacionais, tornando-se cada vez mais importante em acordos comerciais, onde é utilizado para especificação de requisitos, gerenciamento das atividades e definição de critérios de aceitação.

A Estação TABA apresenta uma abordagem para o processo de Medição e Análise em projetos de desenvolvimento de software (SCHNAIDER, L., SANTOS, G., MONTONI, M., ROCHA, A. R., 2004) baseada no método *Goal-Question-Metrics* (GQM), nos requisitos da área de processo Medição e Análise do CMMI e do processo de Medição do MR MPS.BR. A proposta dessa abordagem é apoiar a medição e análise

disponibilizando o conhecimento sobre medições que possa ser útil às Organizações e aos gerentes de projeto durante a execução das atividades que compõem o processo de *Medição e Análise.* Para execução do processo, foram criadas duas ferramentas de apoio à abordagem proposta, definidas e implementadas em outro trabalho (SCHNAIDER, L., SANTOS, G., MONTONI, M., ROCHA, A. R., 2004).

Este trabalho apresentou a definição e a implementação da Base de Métricas para a Estação TABA, pedra fundamental para o processo de medição e análise. A Base é uma estrutura capaz de comportar métricas, questões e objetivos para serem utilizadas durante a construção do plano de medição de uma organização ou de um projeto.

A Base de Métricas apresentada neste trabalho tem como objetivo ser um repositório para um catálogo de métricas, que possam ser utilizadas para definição de planos de medição nos ambientes configurados TABA, possibilitando o registro de métricas de forma detalhada e de modo que as métricas definidas possam ser aplicadas sem ambigüidade.

Foi implementada uma versão simplificada da Base definida por este trabalho, o suficiente para atender aos processos de medição e análise definidos. A redução do formalismo no modelo implementado deveu-se basicamente a restrições de tempos para liberação e uso das ferramentas Metrics e MedPlan, e também buscando facilidade na definição de novas métricas pelos usuários finais da Estação.

A principal contribuição deste trabalho foi a definição da Base de Métricas da Estação TABA, infra-estrutura até então inexistente na Estação, e que vem sendo utilizada em diversas empresas que utilizam Ambientes TABA desde 2004.

Destaca-se também a contribuição no grupo de trabalho de "melhoria de processos", no qual foi definida a estratégia de melhoria em níveis, que apresenta uma visão macro sobre melhoria de processos na Estação TABA, definindo os grandes processos e a interação entre os mesmos.

6.2. Perspectivas futuras

A implementação do modelo completo definido neste trabalho, o que poderá propiciar maior automatização na coleta e maior poder de análise, principalmente para novas métricas definidas pelos usuários, figura como a principal melhoria em continuidade a este trabalho, onde poderão ser identificadas evoluções necessárias ao modelo aqui proposto.

A base foi criada para ser um mecanismo de catalogação e recuperação de métricas para uso principalmente na definição de planos de medição. Sua interação com ferramentas automáticas de coleta de dados e de análise é um ponto importante para sua evolução. As classes unidades e escalas, por exemplo, ainda não guardam relações semânticas entre suas instâncias, o que permitiria a comparação de valores e de métricas coletadas utilizando diferentes unidades de medição. A evolução da Base para dar suporte a um maior nível de automatização da coleta.

A estrutura da Base também deve evoluir no sentido de ter uma definição cada vez mais detalhada das métricas, de modo a permitir um número maior de operações automáticas de sumarização, comparação, geração de resultados e integração aos demais processos da Estação TABA.

Um mecanismo mais refinado para buscas de métricas seria também uma melhoria possível, permitindo localização de métricas a partir de palavras-chaves, métricas semelhantes ("veja também") e por categoria.

A incorporação do uso de indicadores de uma forma mais explícita, reunindo conjuntos de métricas e medições para explicar um evento ou um cenário, conforme definido pela ISO 15939:2002, constitui outra melhoria significativa. Indicadores propiciam uma visão dos resultados da aplicação das métricas bem mais próxima do nível de tomada de decisão, seja gerencial ou organizacional.

A especialização da Base de Métricas para melhor suporte a determinados focos de medição (por exemplo, para medições estritamente relacionadas a produtos de software) e principalmente para acompanhar outras abordagens de medição – como é o caso dos processos de medição orientados por métodos estatísticos – também merece atenção como estudo complementar e de aprofundamento deste trabalho.

Referências Bibliográficas

AHERN, D. M., CLOUSE, A., TURNER, R., 2001, *CMMI Distilled: a Practical Introduction to Integrated Process Improvement*. Addison-Wesley.

ANDRADE, J. M. S., 2005, *Avaliação de Processos de Software em ADSOrg*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.

BARCELLOS, M. P., 2003, *Planejamento de Custos em Ambientes de Desenvolvimento de Software Orientados à Organização*, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

BARCELLOS, M., FIGUEIREDO, S., ROCHA, A. R., *et al.*, 2003, "Utilização de Métodos Paramétricos, Analogias, Julgamento de Especialistas e Conhecimento Organizacional no Planejamento de Tempo e Custos de Projetos de Software", *II Simpósio Brasileiro de Qualidade de Software*, Fortaleza, Brasil, Setembro.

BASILI, V. R. *et al.*, 1992, "The Software Engineering Laboratory - An Operational Software Experience Factory". *Proc. of the 14th International Conference on Software Engineering.*

BASILI, V. R., 1989, "The Experience Factory: Packaging Software Experience". In: *Proceedings of the 14th Annual Software Engineering Workshop,* Software Engineering Laboratory, Goddard Space Flight Center - NASA, Greenbelt, USA, Nov.

BASILI, V. R., 1992, *Software Modelling and Measurement: The Goal/Question/Metric Paradigm*. Technical Report CS-TR-2956, Department of Computer Science, University of Maryland, College Park, MD 20742, September.

BASILI, V. R., 1999, *Lecture material of the course on A Quantitative Approach to Software Management and Engineering*, Department of Computer Science, University of Maryland.

BASILI, V. R., CALDIERA, G., ROMBACH, H. D., 1994, "Goal Question Metric Paradigm", *Encyclopedia of Software Engineering*, 2 Volume Set, John Wiley & Sons, Inc.

BASILI, V. R., *et al.*, 1994a, "Measurement", in: The *Encyclopedia of Software Engineering*, Nova York, NY, John Wiley & Sons, Inc.

BASILI, V. R., *et al.*, 2002, "Lessons learned from 25 years of process improvement: The Rise and Fall of the NASA Software Engineering Laboratory", in: *Proceedings of ICSE*, Orlando, USA, pp.69-79.

BASILI, V. R., LINDVALL, M., COSTA, C., 2001, "Implementing the Experience Factory Concepts as a set of Experiences Bases". In: *Proceedings of 13th International Conference on Software Engineering & Knowledge Engineering*, pp. 102-109, Buenos Aires, Argentina, Jun.

BASILI, V. R., ROMBACH, H. D., 1988, "The TAME Project: Towards Improvement-Oriented Software Environments". *IEEE Transactions on Software Engineering, SE-14(6).*

BASILI, V. R., WEISS, D., 1984, "A Methodology for Collecting Valid Software Engineering Data", *IEEE Transactions on Software Engineering*, Vol. 10, No. 3, Nov, pp. 728-738.

BERGER, P., 2003, *Instanciação de Processos de Software em Ambientes Configurados na Estação TABA*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.

BESSA, A. , 2005, *Melhoria de Processos nos Ambientes Configurados TABA*, Exame de Qualificação para D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.

BOEHM, B. W., 1981, *Software Engineering Economics,* Prentice Hall, Upper Saddle River.

BOEHM, B. W., BROWN, J. R., LIPOW, M., 1976, "Quantitative Evaluation of Software Quality". *Proceedings of the 2nd International Conference on Software Engineering*, 1976.

BRIAND, L. C., DIFFERDING, C. M., ROMBACH, H. D., 1996, "Practical Guidelines for Measurement-Based Process Improvement". *Software Process*, 2(4), December.

BRIAND, L., EMAN, K. E., MORASCA, S., 1996, "On the Application of Measurement Theory in Software Engineering". *Journal on Empirical Software Engineering*, 1 (1).

CABRAL FILHO, R. C. S. *et al.*, 2005, *Uma Abordagem Experimental para a Avaliação da Melhoria de Processos,* artigo submetido ao IV Simpósio Brasileiro de Qualidade de Software (SBQS 2005).

CARLETON, A.D., PARK, R.E., GOETHERT, W.B., FLORAC, W.A., 1996, *Goal-Driven Software Measurement – A Guidebook*, CMU/SEI-96-HB-002, Pittsburgh, Software Engineering Institute, Carnegie Mellon University.

CHRISSIS, M. B., KONRAD, M., SHRUM, S., 2003, *CMMI: Guidelines for Process Integration and Product Improvement*, Addison-Wesley.

CHRISTENSEN, M. J., THAYER, R. H., 2001, "Software Metrics", In: *The Project Manager's Guide to Software Engineering Best Practices*, cap.15, Best Practices Series, Wiley.

CLARK, B., 2002, "Eight Secrets of Software Measurement", *IEEE Software Transactoins*.

CMU/SEI, 2001, *Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version1.1: Method Definition Document*, CMU/SEI-2001-HB-001, Pittsburgh, Software Engineering Institute, Carnegie Mellon University. URL: http://www.sei.cmu.edu

CMU/SEI, 2002, *Capability Maturity Model Integration (CMMI), Version 1.1 CMMI for Software Engineering (CMMI-SW, V1.1)*, Pittsburgh, Software Engineering Institute, Carnegie Mellon University. URL: http://www.sei.cmu.edu

CMU/SEI, 2002, *CMMI for Systems Engineering and Software Engineering V1.*, CMU/SEI-2002-TR-002.

DASKALANTONAKIS, M. K., 1992, "A Practical View of Software Measurement and Implementation Experiences Within Motorola", In: *Applying Software Metrics, IEEE Computer Society Press*, pp. 168-180.

EMAN, K. E., DROUIN, J., MELO W., 1998, *SPICE – The Theory and Practice of Software Process Improvement and Capability Determination*, IEEE Computer Society Press.

FARIAS, L. L., 2002, *Planejamento de Riscos em Ambientes de Desenvolvimento de Software Orientados a Organização*, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

FARIAS, L., TRAVASSOS, G. H., ROCHA, A. R. , 2003, "Knowledge Management of Software Risks" In: *Journal of Universal Computer Science*, vol. 9 n 7, 670- 681.

FENTON, N. E., 1994, "Software Measurement: A Necessary Scientific Basis". *IEEE Transactions on Software Engineering*, vol. 20, No. 3, March. FENTON, N.E., PFLEEGER, S.L, 1997, *Software Metrics – A Rigorous & Practical Approach*, Second Edition, Boston, MA, PWS Publishing Company.

FENTON, N. E., KITCHENHAM, B., 1991, "Validating Software Measures", *Journal of Software Technology, Verification and Reliability*, vol. 1, No. 2, pp. 27-42.

FENTON, N. E., NEIL, M., 2000, "Software Metrics: Roadmap", In: *Future of Software Engineering*, Limerick, Ireland, pp. 359-370.

FIGUEIREDO, S., SANTOS, G., ROCHA, A. R., 2004, "Gerência de Configuração em Ambientes de Desenvolvimento de Software Orientados à Organização", *III Simpósio Brasileiro de Qualidade de Software*, maio, Brasília, DF, Brasil.

FLORAC, W. A., CARLETON, A. E., 2000, *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, Addison-Wesley.

FLORAC, W. A., PARK, R.E., CARLETON, A.D., 1997, *Practical Software Measurement: Measuring for Process Management and Improvement*, CMU/SEI-97-HB-003, Pittsburgh, Software Engineering Institute, Carnegie Mellon University.

GALOTTA, C., 2000, *Netuno: um Ambiente de Desenvolvimento de Software Orientado ao Domínio de Acústica Submarinha*, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

GARG, P., JAZAYERI, M. 1996, "Process-Centered Software Engineering Environments: A grand Tour" in *Software Process*. A. Fuggetta and A. Wolf (eds.), John Wiley & Sons.

GARMUS, D., HERRON, D., 2001, *Function Point Analysis: Measurement Practices for Successful Software Projects,* Addison Wesley.

HARRISON, W., OSSHER, H., TARR, P., 2000, "Software Engineering Tools and Environments: A Roadmap". In: *Proceedings of the Conference on the Future of Software engineering - International Conference on Software Engineering*, pp. 261-277, Limerick, Ireland.

HETZEL, W., 1993, *Making Software Measurement Work: Building an Effective Software Measurement Program*, QED Publishing.

HOLMES, L., 2002, *IT Measurements Practical Advice from the Experts*, Addison-Wesley.

IEEE 1045-1992, 1992, *Standard for Software Productivity Metrics*, IEEE.

IEEE 982.1-1988, 1988, *Standard Dictionary of Measures to Produce Reliable Software*, IEEE.

ISO/IEC 12207:1995, 1995, *Information Technology – Software Life-Cycle Processes*, ISO/IEC.

ISO/IEC 12207:1995/Amd 1:2002, 2002, *ISO/IEC 12207 Amendment: Information Technology - Amendment 1 to ISO/IEC 12207*, ISO/IEC.

ISO/IEC 12207:1995/Amd 2:2004, 2004, *ISO/IEC 12207 Amendment: Information Technology - Amendment 2 to ISO/IEC 12207*, ISO/IEC.

ISO/IEC 14598-5, 1998, *Information Technology – Software Product Evaluation – Process for Evaluators*, ISO/IEC.

ISO/IEC 15504-1, 2003, *ISO/IEC FDIS 15504-1: Information Technology - Process Assessment – Part 1 - Concepts and Vocabulary*, ISO/IEC.

ISO/IEC 15504-2, 2003, *ISO/IEC FDIS 15504-2: Information Technology - Process Assessment – Part 2 - Performing an Assessment*, ISO/IEC.

ISO/IEC 15504-3, 2003, *ISO/IEC FDIS 15504-3: Information Technology - Process Assessment - Part 3 - Guidance on Performing an Assessment*, ISO/IEC.

ISO/IEC 15504-4, 2003, *ISO/IEC FDIS 15504-4: Information Technology - Process Assessment – Part 4 - Guidance on use for Process Improvement and Process Capability Determination*, ISO/IEC.

ISO/IEC 15504-5, 2004, *ISO/IEC FCD 15504-5: Information Technology - Process Assessment - Part 5: An exemplar Process Assessment Model*, ISO/IEC JTC1 SC7.

ISO/IEC 15939:2001, 2002, *ISO/IEC 15939:2001: Information technology — Software engineering — Software measurement process*, ISO/IEC JTC1 SC7.

ISO/IEC 9000:2000, *2000, Quality Management Systems – Fundamental and Vocabulary*, ISO/IEC.

ISO/IEC 9000-3:2004, *Software Engineering – Guidelines for the application of ISO 9000-1:2000 to computer software*, ISO/IEC.

ISO/IEC 9126-1, 2000, *ISO 9126-1: Information technology – Software product quality – Part 1: Quality model*, ISO/IEC.

ISO/IEC 9126-2, 2002, *ISO 9126-2: Software engineering –Product quality – Part 2: External metrics*, ISO/IEC.

ISO/IEC 9126-3, 2002, *ISO 9126-3: Software engineering –Product quality – Part 3: Internal metrics*, ISO/IEC.

KAN, S. H., 2003, *Metrics and Models in Software Quality Engineering*, Second Edition, Addison-Wesley.

KITCHENHAM, B., PFLEEGER, S., HOAGLIN, D., EMAN, K., 2002, "Preliminary Guidelines for Empirical Research in Software Engineering", *IEEE transactions on software engineering*, vol. 28, N0. 8, August.

KITCHENHAM., B., HUGHES, R.T., LINKMAN, S. G., 2000, "Modeling Software Measurement Data" *IEEE Trans. Software Eng.*, vol. 27, no. 9, pp. 788–804.

KITCHENHAM., B., PFLEEGER, S. L., FENTON, N., 1995, "Towards a Framework for Software Measurement Validation". *IEEE Transactions on Software Engineering*, 21 (12), December.

KOGURE, M., AKAO, Y., 1983, *Quality Function Deployment and CWQC in Japan. Quality Progress.*

MARTINS, F. R. S., 2004, *Ambiente de Desenvolvimento de Software Orientado a Organização Aplicado à Instrumentação Virtual*, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

MCCALL, J. A., RICHARDS, P.K, WALTERS, G. F., 1977, *Factors in Software Quality*. Report No. NTIS AS/A- 049014015, US Rome Air Development Center.

MCGARRY, J., CARD, D., JONES, C., *et al.*, 2001, *Practical Software Measurement: Objective Information for Decision Makers,* Addison-Wesley.

MELTON, A., GUSTAFSON, D., BIEMAN J., BAKER, A., "A mathematical perspective for software measurement research", *Journal of Software Engineering*, Vol. 5, No. 5, pp. 246-254, 1990.

MONTONI M., SANTOS G., VILLELA K., *et al.*, 2005a, "Enterprise-Oriented Software Development Environments to Support Software Products and Processes Quality Improvement", *Proceedings of the PROFES 2005.*

MONTONI, M. *et al.*, 2003, "ACKNOWLEDGE: uma Ferramenta para Aquisição do Conhecimento no Desenvolvimento de Software", *I Workshop Tecnologias da Informação e Gerência do Conhecimento*, Fortaleza, CE.

MONTONI, M., 2003, *Aquisição de Conhecimento: Uma Aplicação no Processo de Desenvolvimento de Software*, Dissertação de MSc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

MONTONI, M., MIRANDA, R., ROCHA, A. R.. TRAVASSOS, G. H., 2004, "Knowledge Acquisition and Communities of Practice: An Approach to Convert

Individual Knowledge into Multi-Organizational Knowledge", In: *Workshop Learning Software Organization*, Banff, Canada.

MONTONI, M., SANTOS, G., SCHNAIDER, L., *et al.*, 2005b, "An Approach to Support CMMI Measurement and Analysis Process Area", In: *Workshop Learning Software Organization*, Kaiserslautern, Germany, april.

MOREAU, B. *et al.*, 2003, "Software Quality Improvement in France Telecom Research Center", *Software Process: Improvement and Practice*, v.8:3, pp. 135 – 144.

MURINE, G.. E., 1980, "Applying Software Quality Metrics in the Requirements Analysis Phase of a Distributive System", *Proceedings of the Minnowbrook Workshop*, New York.

MUTAFELIJA, B., STROMBERG, H., 2003, *Systematic Process Improvement using ISO 9001:2000 e CMMI*, Artech House.

NBR ISO/IEC 12207, 1998, Tecnologia de informação – Processos de ciclo de vida de software. Rio de Janeiro, Brasil, Associação Brasileira de Normas Técnicas.

NGUYEN, M. et al., 1997, *Total Software Process Model Evolution*, em http://www.idi.ntnu.no/~epos/Papers/icse97.ps

OLIVEIRA, K. M., 1999, *Modelo para Construção de Ambientes de Desenvolvimento de Software Orientados a Domínio*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

OLIVEIRA, K.M, XIMENES, A., MATWIN, *et al.*, 2000, "A Generic Architecture for Knowledge Acquisition Tools in Cardiology"; *5th Intelligent Data Analysis in Medicine and Pharmacology - Workshop at the 14th European Conference on Artificial Intelligence*, pp. 43-45, Berlin, Alemanha, Agosto.

OLSINA, L., MARTIN, M. A., FONS, J., *et al.*, 2003, *Towards the Design of a Metrics Cataloging System by Exploiting Conceptual, Navigational, and Semantic Web*, Engineering School at UNLPam, Argentina.

PFLEEGER, S. L., 1997, "Use Realistic, Effective Software Measurement", in: *Constructing Superior Software*, Software Quality Institute Series, The Software Quality Institute.

PFLEEGER, S. L., 2000, "Improving Predictions, Products, Processes and Resources", In: *Software Engineering*, cap.13, pp. 563-592.

PFLEEGER, S. L., 2001, *Software Engineering: theory and practice*, 2nd edition, Prentice-Hall, Inc., ISBN 0-13-029049-1.

PFLEEGER, S. L., ROMBACH, H. D., 1994, "Measurement based Process Improvement", *IEEE Software*, July.

PFLEEGER, S., 1998, *Software Engineering – Theory and Practice*, New Jersey, USA, Prentice-Hall.

ROBERTS, F. S., 1979, "Measurement Theory with Applications to Decision Making, Utility, and the Social Sciences", *Encyclopedia of Mathematics and its Applications*, Addison-Wesley.

ROCHA, A. R., *et al.*, 2005, "Reference Model for Software Process Improvement: a Brazilian Experience", *Proceedings of the 3rd World Congress for Software Quality (3WCSQ)*, Munich, Germany.

ROCHA, A. R., SOUZA, J. M., AGUIAR, T. C., 1990, "TABA: A Heuristic Workstation for Software Development". In: *Proceedings of COMPEURO 90*, pp. 126-129, Tel Aviv, Israel, May.

ROCHA, A.R., MALDONADO, J.C., WEBER, K.C., 2001, *Qualidade de Software – Teoria e Prática*, 1a ed., Prentice Hall, São Paulo.

ROCHE, J., JACKSON, M., 1994, "Software measurement methods: recipes for sucess?", *Journal on Information and Software Technology*, 36(3).

ROMBACH, H. D., 1991, *Practical Benefits of Goal-Oriented Measurement. Software Reliability and Metrics*, Elsevier Applied Science.

SANTOS, G. *et al.*, 2003, "SAPIENS: uma Ferramenta para Descrição e Recuperação de Competências em uma Organização", *I Workshop Tecnologias da Informação e Gerência do Conhecimento*, Fortaleza,CE.

SANTOS, G., 2003, *Representação da Distribuição do Conhecimento, Habilidades e Experiências através da Estrutura Organizacional*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.

SANTOS, G., MONTONI, M., ROCHA, A. R. *el at.*, 2005, "Using a Software Development Environment with Knowledge Management to Support Deploying Software Processes in Small and Medium Size Companies", In: *Workshop Learning Software Organization*, Kaiserslautern, Germany, april.

SANTOS, G., VILLELA, K., SCHNAIDER, L., *et al.*, 2004, "Building ontology based tools for a software development environment", In: *Workshop Learning Software Organization*, Banff, Canada, (Lecture Notes in Computer Science, vol 3096, pp 19-30)

SANTOS, G., ZLOT, F., 1999, *Definição e Instanciação de Ambientes na Estação TABA*, Projeto Final de Curso, UFRJ, Rio de Janeiro, RJ, Brasil.

SCHNAIDER, L. *et al.*, 2004, "MedPlan: uma Abordagem para Medição e Análise em Projetos de Desenvolvimento de Software", *II Simpósio Brasileiro de Software*, Brasília, DF.

SCHNAIDER, L., 2003, *Planejamento da Alocação de Recursos Humanos em Ambientes de Desenvolvimento de Software Orientados à Organização*, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

SCHNAIDER, L., SANTOS, G., MONTONI, M., ROCHA, A. R., 2004, "Uma Abordagem para Medição e Análise em Projetos de Desenvolvimento de Software", *III Simpósio Brasileiro de Qualidade de Software*, maio, Brasília, DF, Brasil.

SCHNEIDEWIND, N. F., 2002, "Body of Knowledge for Software Quality Measurement", *Computer*, February, pp. 77-83.

SEI, 2002, *Capability Maturity Model Integration (CMMI) Version 1.1 - Staged Representation*, Carnegie Mellon University, Software Engineering Institute, Pittsburgh.

SEI, 2002, *CMMI for Systems Engineering/Software Engineering (CMMI-SE/SW), Staged Representation, Version 1.1,* Technical report CMU/SEI-2002-TR-02. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

SOFTEX, 2004a, "Uma Estratégia para Melhoria de Processo de Software nas Empresas Brasileiras", http://www.softex.br/media/QuaTIC.zip. Acessado em 02/2005.

SOFTEX, 2004b, "Modelo de Referência para Melhoria de Processo de Software: uma abordagem brasileira.", http://www.softex.br/media/artigoCLEI _versao_final.pdf. Acessado em 02/2005.

SOLINGEN, R., BERGHOUT, E., 1999, *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*, McGrawHill.

TRAVASSOS, G. H., 1994, O *Modelo de Integração de Ferramentas da Estação TABA*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

VILLELA, K., 2004, *Definição e Construção de Ambientes de Desenvolvimento de Software Orientados à Organização*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.

VILLELA, K., ROCHA, A.R., TRAVASSOS, G. H., 2005, "On the Importance Attributed to Different Knowledge in Software Development Environments", *In: Workshop Learning Software Organization*, Kaiserslautern, Germany, april.

ZLOT, F., 2002, *Conhecimento de Tarefa em Ambientes de Desenvolvimento de Software Orientados a Domínio*, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

ZUSE, H., 1998, *A Framework for Software Measurement*, Walter de Gruyter, Berlin.