



UNIVERSIDADE
ESTADUAL DE LONDRINA

Departamento de Computação
Trabalho de Conclusão de Curso

**Métricas de Software com ênfase no processo de medição do
MPSBR**

Daniel Cesar Romano Luvizotto

Londrina

2008

Daniel Cesar Romano Luvizotto

**Métricas de Software com ênfase no processo de medição
do MPSBR**

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação em
Ciência da Computação da Universidade
Estadual de Londrina, como requisito
parcial à obtenção do grau de Bacharel.

Orientadora: Jandira Guenka Palma.

COMISSÃO EXAMINADORA

Prof. [Dr. ou Ms.] [nome]
[instituição]

Prof. [Dr. ou Ms.] [nome]
[instituição]

Prof. [Dr. ou Ms.] [nome]
[instituição]

Londrina, ____ de _____ de <ano>

Agradecimentos

Primeiramente a Deus pelo dom da vida e por proporcionar momentos tão especiais como este.

À orientadora Prof. Dra. Jandira Palma, braço amigo de todas as etapas deste trabalho.

À minha família, pelo apoio e pela confiança.

À minha querida namorada Jaqueline, sempre ao meu lado.

Aos professores e meus amigos, pela força e pelo companheirismo ao longo desta etapa importante de nossas vidas.

Aos profissionais entrevistados, pela concessão de informações valiosas para a realização deste estudo.

A todos que, com boa intenção, colaboraram para a realização e finalização deste trabalho.

A Deus, meus pais, meu irmão, minha
namorada e meus amigos...

Resumo

Gerentes de projeto encontram vários problemas no desenvolvimento de software, um dos maiores problemas, e que interferem em todo o desenvolvimento e gerenciamento do projeto, é conseguir dimensionar o tamanho do software. Uma ferramenta que pode apoiar os gerentes de projeto são as Métricas de software, pois ajudam dimensionar, planejar, e obter estimativas mais precisas de projetos, recursos e produtos de software. A medição associada a métricas é recomendada por vários modelos de controle de qualidade de software, como MPS.BR (Melhoria de Processo do Software Brasileiro), que será foco deste estudo, e ainda as métricas apóiam o gerenciamento do processo de desenvolvimento de software. Então, este trabalho efetivará estudo de métricas, avaliando ferramentas, modelos e como podem ajudar na implantação do processo de medição do nível F do MPS-BR numa empresa com o nível G.

Palavras-chave: Métricas de Software, Qualidade de Software, MPS.BR, Melhoria de Processo do Software Brasileiro, GQM.

Abstract

Project managers have many problems in software development, and one of the biggest problems that interfere throughout the development and management of the project is to scale the size of the software that is being managed. A tool that can support the project managers are the metrics of software, it helps assess, plan and obtain more accurate estimates of projects, resources and products. The measurement is associated with metrics recommended by various models of quality control software, as Brazilian Software Process Improvement the MPS.BR (*Melhoria de Processo do Software Brasileiro*), which will be the focus of study, and also helps improve the management of the software development process. Then carry out this work study of metrics, benchmarking tools, models and how they can assist in the deployment process of measuring the level of M MPS-BR in a company with the level G.

Keywords: Software Metrics, Software Quality, MPS.BR, GQM.

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
Lista de abreviaturas e siglas	x
Introdução	12
1. Métricas	14
1.1. Engenharia e Qualidade de Software	14
1.2. O Processo de Gerência de Projetos	14
1.3. Métricas	15
1.3.1. Métrica de Software Análise de Pontos de Função	16
1.3.2. Métricas de Software por Ponto de Caso de Uso	27
1.4. Processo de Medição	32
1.4.1 <i>Goal Question Metric</i> (GQM)	33
2. Melhoria de Processo do Software Brasileiro (MPS.BR)	38
2.1 Níveis do MPS.BR	42
2.2 O Processo de Medição do nível F do MPS.BR	47
3. Estudo de Casos de Métricas	50
4. Análise de Resultados	54
Conclusão	59
Referências Bibliográficas	61
Bibliografia Consultada	63

Lista de Figuras

Fig. 1.1: Modelo de Análise de Ponto por função [7].	17
Fig. 1.2: Hierarquia do processo GQM.	37
Fig. 2.1: Componentes do MPS.BR	39
Fig. 4.1: Estrutura de medição.....	57

Lista de Tabelas

Tab. 1.1: Complexidade de AIE e ALI [7].	20
Tab. 1.2: Contribuição de ALI e AIE [7].	21
Tab. 1.3: Complexidade funcional para entradas externas [7].	23
Tab. 1.4: Complexidade funcional para consultas externas e saídas externas [7].	23
Tab. 1.5: Contribuição de EE, SE e CE [7].	25
Tab. 1.6: Produtividade no desenvolvimento por setor de negócios em 1999 [7]	27
Tab. 1.7: Tabela de Pesos de Atores [8].	28
Tab. 1.8: Tabela de casos de uso por número de transações [8].	29
Tab. 1.9: Tabela de casos de uso por número de entidades [8].	29
Tab. 1.10: Fatores de complexidade técnica [8].	31
Tab. 1.11: Fatores de complexidade ambiental [10].	31
Tab. 2.1: Níveis de maturidade do MR-MPS [14].	40
Tab. 3.1: Análise por pontos de caso de uso do web site.	51
Tab. 3.2: Análise de pontos por função do sistema de gerenciamento.	52

Lista de abreviaturas e siglas

AIE	Arquivos de Interface Externa.
ALGA	Associação Londrinense de Ginástica Artística
ALI	Arquivo Lógico Interno.
AP	Atributo de processo
APF	Análise de Ponto de Função
AR	Arquivos Referenciados
BID	Banco Interamericano de Desenvolvimento
CE	Consulta Externa
EE	Entrada Externa.
FCA	Fator Complexidade Ambiental
FCT	Fator de complexidade técnica
FINEP	Financiadora de Estudos e Projetos
GQM	<i>Goal Question Metric</i>
ISO	<i>International Organization for Standardization</i>
MA-MPS	Método de Avaliação
MCT	Ministério da Ciência e Tecnologia
MED	Medição
MN-MPS	Modelo de Negócio
MPS.BR	Melhoria de Processo do Software Brasileiro.
MR-MPS	Modelo de Referência
PCU	Pontos de Caso de USO
PCUNA	Peso Total dos Casos de Uso Não Ajustado
PF	Pontos por Função
RUP	Rational Unified Process
SE	Saída Externa
SOFTEX	Associação para a Promoção da Excelência do Software Brasileiro
TD	Tipos de Dado.
TPNAA	Total de Pesos não Ajustados dos Atores
TPNAUC	Total de Pesos não ajustados dos casos de usos

TR	Tipo de Registro.
VFA	Valor do Fator de Ajuste

Introdução

A Engenharia de Software cada vez mais percebe a importância da Gerência de projetos, o crescimento desta área se deve ao número de incertezas que o universo de negócio possui. Este ambiente de negócio conturbado se deve ao contínuo avanço da tecnologia, mudanças políticas e a globalização entre outros fatores que deixam este ambiente arriscado e instável [1].

A Gerência de projetos está presente em todo o projeto [2], desde a coleta de requisitos, desenvolvimento e acompanhamento até a manutenção, por isso ela precisa de uma ferramenta que lhe de assistência para poder medir o projeto, programar o cronograma, definir custos e recursos.

Nesse contexto que aparecem as métricas para dimensionar o software e conseguir estruturar o projeto para diminuir riscos e estabelecer prazos com os clientes. As métricas de software são essenciais para vários modelos e certificações de qualidade, como o programa para Melhoria de Processo do Software Brasileiro (MPS.BR), que possui processo específico de medição, o qual será estudo de enfoque.

Com isso o objetivo deste trabalho é estudar o modelo GQM (Goal Question Metric) de medição para processos, produtos e recursos de software, os modelos de métricas, tanto por ponto de função como por casos de uso, quando e onde melhor aplicá-las. Fazer um estudo de caso de aplicação das métricas. Analisar como estes modelos de medição podem ajudar no gerenciamento de software e como as Métricas podem auxiliar na Implantação do nível F do MPSBR nas empresas.

Organização do trabalho

No capítulo 1 será feito um estudo sobre métricas, no início desse capítulo será feita uma introdução de engenharia de software e gerência de projetos, depois serão mostradas as métricas por ponto de função e por ponto de caso de uso, e por final será apresentado o modelo de medição GQM.

No capítulo 2, será feita a apresentação do MPS.BR, das suas características, da sua hierarquia, um pouco de cada nível, e depois a ênfase será no processo de medição do nível F.

No capítulo 3, será apresentado os Estudos de Casos, o primeiro realizado durante o desenvolvimento de um Sistema, nesse estudo foi aplicado métricas respeitando o modelo

MPS.BR. Também foi feito outro estudo de caso, medindo quesitos de uma empresa desenvolvedora de software, através do modelo GQM, quesitos estes que também estão dentro MPS.BR.

No capítulo 4, será feita a análise de resultados dos estudos de casos, trazendo de cada métrica, os seus pontos fortes e fracos, a seguir mostraremos as características marcantes do GQM, depois será comparado este modelo de medição com os resultados esperados pelo processo de medição do nível F do MPS.BR, e veremos que as métricas de ponto por função ou por ponto de caso de uso, podem ser usadas em conjunto com o modelo GQM, formando um modelo que ajuda no gerenciamento de projetos de software e que cumpre os requisitos de qualidade do MPS.BR.

Por fim, serão apresentadas as conclusões do trabalho.

1. Métricas

A importância do processo de medição pode ser evidenciada por ser um quesito obrigatório em todos os modelos de qualidade.

O processo de medição é uma ferramenta de apoio ao projeto corrente, assim como para criar um histórico que facilite ao gerente identificar a capacidade produtiva de sua equipe e o tempo gasto, com maior precisão, nos projetos subsequentes.

1.1. Engenharia e Qualidade de Software

Segundo Pressman [2], a engenharia de software surgiu por volta dos anos 80, e como o próprio nome, já traz alguns conceitos como criação, construção, análise, desenvolvimento e manutenção. E para que todos estes conceitos possam estar integrados no programa a engenharia de software procura aplicar práticas de ciência da computação, gerência de projetos e outras disciplinas, buscando organização, produtividade e qualidade.

A qualidade nos dias de hoje é requerida em todas as áreas, e a na de software não é diferente, cada vez mais é necessário satisfazer as necessidades dos clientes, tanto nos produtos e serviços, uma vez que eles exigem cada vez mais, e com isso, ela acaba sendo um fator de desempate entre as empresas. Várias organizações, como ISO (*International Organization for Standardization*), criam modelos e normas de qualidade, para que as empresas desenvolvedoras de software estejam por dentro das necessidades do cliente. Exemplo disso é a ISO 9000 uma importante norma criada em para garantir questão da qualidade para qualquer setor produtivo [17].

1.2. O Processo de Gerência de Projetos

Gerência de projetos é a primeira camada do processo de engenharia de software, é chamada de camada ao invés de etapa, porque acompanha todo o desenvolvimento do software desde o começo até o fim [3].

Para conduzir um projeto de software de maneira adequada, deve-se compreender requisitos como: o escopo do trabalho a ser feito, os riscos do projeto, os recursos necessários, o custo (esforço), a divisão das tarefas e o acompanhamento do projeto, tarefas que normalmente são consideradas difíceis pelos gerentes de projeto de software. Uma maneira de conseguir um bom planejamento para estes requisitos é a através das métricas de software que fazem a estimativa do tamanho dos projetos de software.

A tarefa de medição de projeto é muito importante, pois será através dela que todo o processo de gerência de projetos será programado, todo o cronograma de acompanhamento será feito em relação ao seu tamanho, ou seja, todas as tarefas que vão do início ao fim do projeto serão feitas através dessa consideração inicial.

Uma vez conseguindo estipular o tamanho do software, fica muito mais fácil estimar o tempo e os demais requisitos necessários já citados (riscos, custo, etc.), ou seja, fica muito mais fácil gerenciar o projeto. Como diz uma frase conhecida de Tom de Marco "Não se pode controlar o que não se pode medir" [4].

1.3. Métricas

Uma métrica é o ato de medir um atributo (propriedade ou característica) de uma entidade (produto, processo ou recursos) [5]. Exemplos:

- O tamanho de um produto, como número de linhas, ou número de pontos de funções.
- Esforço para realizar uma tarefa.

Uma medição tem quatro papéis segundo BASILI, CALDEIRA e ROMBACH [12]:

- Entender;
- Avaliar;
- Controlar;
- Prever.

As métricas também são muito importantes no desenvolvimento de software, pois:

- Avaliar e diminuir riscos;
- Avaliar benefícios;
- Avaliar os retornos;
- Aumentar a exatidão das estimativas;
- Conseguir dados quantitativos, para melhorar o processo de gerenciamento de software;

Uma boa métrica de Software deve quantificar o que queremos medir, e dar os mesmos resultados se as condições forem às mesmas, fácil de computar, e acessível. E capaz de medir recursos, processos e produtos [5].

Com as métricas pode-se entender mais sobre comportamento e a funcionalidade de produtos e processos, controlá-los, avaliar e tomar as melhores decisões e guardando-as como padrão e prever os valores dos atributos.

1.3.1. Métrica de Software Análise de Pontos de Função

A métrica de Ponto por função é um método-padrão para medir software, visando conseguir uma medida através de pontos de função (PFs), com base nas funcionalidades a serem desenvolvidas, com o ponto de vista do usuário [6].

Os objetivos da medição por Pontos de Função são:

- Medir a funcionalidade do sistema que o usuário recebe e solicita.
- Medir o desenvolvimento e manutenção de projetos de software, sem se preocupar com a tecnologia utilizada na implementação.

A análise de ponto de função (APF) é feita na ordem conforme o modelo da figura 1.1 [7]:

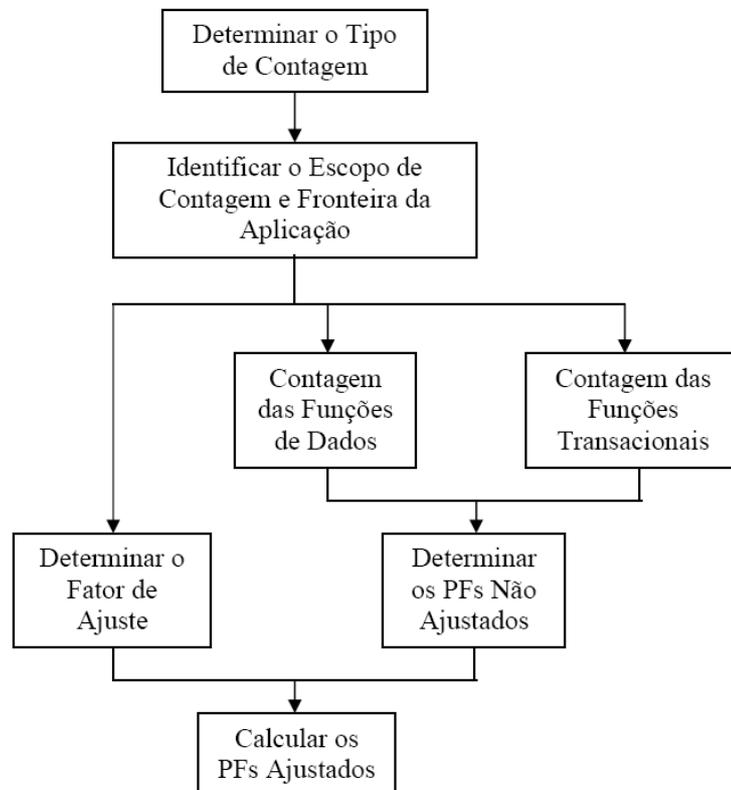


Fig. 1.1: Modelo de Análise de Ponto por função [7].

Conforme a figura 1.1, vamos ao primeiro passo da Análise de ponto de função (APF) que é determinar o tipo de contagem e será mostrado a seguir:

a) Determinação do tipo de Contagem.

Segundo Vazquez, Simões e Albert [7], deve-se identificar qual dos três tipos de contagem utilizar.

- **Projeto de desenvolvimento:** mede a funcionalidade fornecida aos usuários finais do software quando da sua primeira instalação. Isto quer dizer que abrange funções de conversões de dados, como por exemplo, uma empresa que vai substituir um sistema A, por um sistema B, e necessita importar dados do sistema anterior devem ser importados para o sistema novo.

- **Projeto de melhoria:** mede as funcionalidades modificadas, adicionadas ou excluídas do sistema, e também as conversões de dados necessárias. Depois de terminado o projeto de melhoria, deve ser atualizado o número de pontos de função do sistema, para identificar a modificação das funcionalidades.
- **Aplicação:** Mede a funcionalidade de uma aplicação instalada.

b) Determinação da Fronteira da Aplicação e do Escopo da Contagem

Nesta fase, definem-se as funcionalidades que serão incluídas na contagem específica. A fronteira da aplicação é definida estabelecendo um limite lógico entre a aplicação que está sendo medida, o usuário e outras aplicações. O escopo de contagem define a parte do sistema (funcionalidades) a ser contada.

Fronteira de Aplicação

A fronteira de aplicação delimita o que será medido (software), e o mundo exterior (seus usuários) [6]. Se existir mais de uma aplicação no escopo da contagem, várias fronteiras devem ser determinadas.

Existem algumas regras para determinação da fronteira de aplicação, como:

- A determinação deve estar de acordo com o ponto de vista do usuário. O foco deve ficar no que o usuário pode entender e descrever.
- A fronteira entre as aplicações deve estar de acordo com a separação das funções estabelecidas pelos processos do negócio, não de acordo com considerações tecnológicas.

A identificação da fronteira da aplicação é uma tarefa difícil, determinar onde uma aplicação acaba e outra começa é de grande complexidade, porém este passo é essencial para a medição funcional, pois o posicionamento incorreto da fronteira pode mudar a visão da contagem, e alguns problemas podem aparecer:

- Contagem duplicada da mesma transação por várias aplicações.

- Contagem de funções de transferência de dados.
- Dificuldade na determinação do tipo de transação.
- Duplicidade na contagem de arquivos.

Escopo de Contagem

O escopo da contagem define se a contagem será de um ou mais sistemas, ou apenas de parte de um sistema. Exemplos:

- Todas as funcionalidades disponibilizadas.
- Apenas funcionalidades que serão utilizadas pelo usuário.
- Apenas funcionalidades específicas: relatórios, etc.

c) Contagem de funções do tipo dado.

Os tipos de dados podem ser classificados como:

a. Arquivo Lógico Interno (ALI):

É um grupo de dados, ou informações de controle, identificável pelo usuário, logicamente relacionado e mantido na fronteira da aplicação [6].

A principal função é armazenar dados mantidos (adicionados, modificados ou excluídos) por meio de um ou mais processos elementares da aplicação sendo contada.

Exemplos:

- Tabelas que armazenam dados mantidos pela aplicação.
- Arquivos de configuração mantidos pela aplicação.
- Arquivos de segurança de acesso à aplicação (senhas), mantidos por ela.
- Arquivos mantidos não só pela aplicação, mas também por outra aplicação.
- Tabelas de banco de dados atualizadas pela aplicação.

b. Arquivos de Interface Externa (AIE).

É um grupo de dados ou informações de controle, identificável pelo usuário, logicamente relacionado, referenciado (lido) pela aplicação [7].

Sua principal função é a armazenagem de dados referenciados por meio de um ou mais processos elementares dentro da fronteira da aplicação sendo contada. Um AIE deve obrigatoriamente ser um ALI de outra aplicação.

Exemplos:

- Dados de referência externos utilizados pela aplicação.
- Tabelas de banco de dados lidas pela aplicação, mas atualizadas por outra aplicação.

Depois de classificados em AIE ou ALI, deve-se determinar a complexidade funcional de cada um. Esta complexidade está de acordo com o número de Tipos Dados (TD) e Tipos Registros (TR) que eles possuem, a Tab. 1.1 descreve a classificação das complexidades.

Tab. 1.1: Complexidade de AIE e ALI [7].

Tipos de Registros	Tipos de Dados		
	1-19	20-50	51 ou mais
1	Baixa	Baixa	Media
2-5	Baixa	Media	Alta
6 ou mais	Media	Alta	Alta

Então se deve encontrar o número de TDs e TRs que cada AIE e ALI possuem e classificá-los de acordo com a tabela de Complexidade de AIE e ALI.

Tipos de Dados (TD)

Um tipo dado é um campo único, reconhecido pelo usuário e não repetido [7].

Regras para contagem de Tipos Dados:

- Conte um tipo dado para cada campo único reconhecido pelo usuário e não repetido, mantido ou recuperado de um ALI ou AIE através da execução de um processo elementar.
- Quando duas aplicações mantêm ou referenciam o mesmo ALI/AIE, conte apenas os campos utilizados pela aplicação em análise.
- Conte um tipo dado para cada campo solicitado pelo usuário para estabelecer um relacionamento com outro arquivo lógico (ALI ou AIE).

Tipo de Registro (TR)

Um tipo de registro é um subgrupo de dados, reconhecido pelo usuário, e é um componente de um ALI ou AIE [6]. E eles podem ser:

- Opcionais: são aqueles que o usuário não tem a necessidade de informar no processo elementar que cria ou adiciona dados ao arquivo.
- Obrigatórios são aqueles que o usuário requer que necessariamente serão utilizados para o processo elementar que cria ou adiciona dados ao arquivo.

Regras para a contagem de Tipos de Registro

1. Conte um tipo de registro cada subgrupo obrigatório ou opcional, de cada ALI ou AIE.
2. Se não existir nenhum subgrupo, conte o próprio ALI ou AIE como um tipo de registro.

Após determinados os número de TDs e TRs e classificados pela sua complexidade, deve-se calcular a contribuição de cada arquivo, de acordo com a Tab. 1.2.

Tab. 1.2: Contribuição de ALI e AIE [7].

Tipo de Função	Baixa	Média	Alta
ALI	7 PF	10 PF	15 PF
AIE	5 PF	7 PF	10 PF

d) Contagem de funções do tipo Transação.

As funções do tipo transação podem ser classificadas como:

a. Entrada Externa (EE):

É um processo elementar, que processa dados ou informações de controle recebidas de fora da fronteira da aplicação [7].

Sua principal função é manter (incluir, alterar, excluir dados de) um ou mais ALI e ou modificar o comportamento do sistema.

Exemplos:

- Transações que recebem dados externos utilizados na manutenção de ALIs.
- Janela que permite alterar, adicionar ou excluir registros em arquivos. Nesse caso será contado como 3 entradas externas.

b. Saída Externa (SE):

É um processo elementar, que envia dados ou informações de controle para fora da fronteira da aplicação [6].

Sua principal função é apresentar informação ao usuário por meio de lógica de processamento que não seja apenas recuperação de dados ou informações de controle. Lógica de processamento deve ao menos possuir uma fórmula matemática ou cálculo, ou então criar dados derivados (esses cálculos ou dados não precisam ser mostrados ao usuário). Pode manter (incluir, alterar, excluir dados de) um ou mais ALI e ou modificar o comportamento do sistema.

Exemplos:

- Relatórios com totalização de dados.
- Telas de login com criptografia.
- Consultas com cálculos ou apresentação de dados derivados.

c. Consulta Externa (CE):

É um processo elementar, que envia dados ou informações de controle para fora da fronteira da aplicação.

Sua principal função é apresentar informação ao usuário por meio de uma simples recuperação de dados ou informações de controle de ALIs e ou AIEs. A Lógica de processamento não deve possuir uma fórmula matemática ou cálculo, nem criar dados derivados. Nenhum ALI é mantido, nem o comportamento do sistema é alterado.

Exemplos:

- Telas de help.
- Telas de login sem criptografia.

Após classificar as funções de tipo Transação em EE, SE ou CE, deve-se determinar a complexidade de cada um deles de acordo com o Número de Arquivos Referenciados (AR), e o número de Tipo de Dados (TD), conforme a Tab. 1.3.

Tab. 1.3: Complexidade funcional para entradas externas [7].

Arquivos Referenciados	Tipos de Dados		
	1-4	5-15	16 ou mais
0 ou 1	Baixa	Baixa	Media
2	Baixa	Media	Alta
3 ou mais	Media	Alta	Alta

Tab. 1.4: Complexidade funcional para consultas externas e saídas externas [7].

Arquivos Referenciados	Tipos de Dados		
	1-5	6-19	20 ou mais
0 ou 1	Baixa	Baixa	Media
2 ou	Baixa	Media	Alta
4 ou mais	Media	Alta	Alta

Então se deve encontrar o número de TDs e ARs que cada AIE e ALI possuem e classificá-los de acordo com a tabela de Complexidade de EE ou SE e CE (Tab. 1.4).

Arquivos Referenciados (AR)

Um arquivo referenciado é: um ALI lido ou mantido pela função do tipo transação ou um AIE lido pela função do tipo transação [7].

As regras abaixo são para a contagem de arquivos referenciados, as duas primeiras não se aplicam em consultas externas, pois atualizam arquivos.

- Contar um arquivo referenciado para cada ALI mantido.
- Contar apenas um arquivo referenciado para cada ALI que seja tanto mantido como lido.
- Contar apenas um arquivo referenciado para cada ALI ou AIE lido durante o processamento.

Tipo de dado (TD)

Um tipo dado em uma função transação é um campo único, reconhecido pelo usuário e não repetido, como nas funções do tipo dado [7].

Mas as regras para contagem são um pouco diferentes:

- Contar um tipo dado para cada campo reconhecido pelo usuário e não repetido, que entra ou sai pela fronteira de aplicação e que é necessário à compreensão do usuário, se o campo entrar e sair pela fronteira de aplicação ele deverá ser contado apenas uma vez.
- Contar um tipo de dado para a capacidade de especificar uma ação a ser tomada. Se a mesma ação tem várias maneiras de ser ativada, deve ser contada uma única vez.
- Contar como apenas um tipo dado para a capacidade de envio para fora da fronteira da aplicação uma mensagem de resposta do sistema, que indica erro

verificado durante o processamento, a confirmação de sua conclusão ou a verificação de seu prosseguimento.

Depois de determinada a complexidade das funções do tipo transação deve-se calcular a contribuição de cada um de acordo com a Tab. 1.5.

Tab. 1.5: Contribuição de EE, SE e CE [7].

Tipo de Função	Baixa	Média	Alta
EE	3 PF	4 PF	6 PF
SE	4 PF	5 PF	7 PF
CE	3 PF	4 PF	6 PF

e) Determinar Contagem de pontos de função não ajustados.

Neste passo é somada a contagem de funções do tipo dado com a contagem de funções do tipo Transação, ou seja, soma-se a contribuição obtida em cada uma das contagens anteriores obtendo a contagem de pontos de função não ajustados.

f) Determinar o Valor de ajuste.

O valor de ajuste é baseado em características gerais do sistema, que avalia as funcionalidades gerais do sistema. O fator de ajuste influencia os pontos de função não ajustados em +/- 35%, obtendo-se o número de PFs ajustados. Para se calcular o fator de ajuste, é usado 14 características gerais dos sistemas, a saber:

1. Comunicação de Dados.
2. Processamento de Dados Distribuído.
3. Performance.
4. Configuração altamente utilizada.
5. Volume de Transações.

6. Entrada de Dados *On-line*.
7. Eficiência do Usuário Final (Usabilidade).
8. Atualização *On-line*.
9. Processamento Complexo.
10. Reusabilidade.
11. Facilidade de Implantação.
12. Facilidade Operacional (Processos Operacionais, tais como Inicialização, Cópia de Segurança, Recuperação etc.).
13. Múltiplos Locais e Organizações do Usuário.
14. Facilidade de Mudanças (Manutenibilidade).

Cada uma destas características deve ser avaliada de acordo com a influência no projeto, e cada uma pode ter seu peso variando entre 0 a 5:

0. Nenhuma influência.
1. Influência mínima.
2. Influência moderada.
3. Influência média
4. Influência significativa.
5. Grande influência.

Depois de obtidos os pesos de cada uma das 14 características, eles devem ser somados e substituídos na fórmula do Valor do Fator de Ajuste:

$$\text{VFA} = (\text{somatório das 14 características} * 0,031) + 0,65.$$

Quando a técnica de pontos e função surgiu em 1979, ela não possuía as 14 características gerais do sistema, para fator de ajuste, o fator de ajuste era feito de maneira subjetiva podendo variar mais ou menos 25%. Mas na revisão técnica de 1984, surgiram estas características que podem proporcionar uma variação de mais ou menos 35%. Porém em 2002, o uso do fator de ajuste tornou-se opcional, como medida para que a técnica fosse aceita como um modelo padrão de medição funcional aderente a norma ISO/IEC 14143,

pois várias das características gerais do sistema contemplam características não funcionais [7].

g) Calcular o número de Pontos de função ajustados.

Neste passo multiplica-se o valor de ajuste com os pontos de função não ajustados, obtendo assim o número de pontos por função ajustados.

O tempo gasto para cada ponto de função pode variar muito com o tipo de aplicação a ser desenvolvida, com a linguagem utilizada e também com equipes de trabalho diferentes [7]. Usando softwares já desenvolvidos pela equipe para determinar o tempo de produção de cada ponto de função é uma forma mais precisa que pode ser atualizado conforme a experiência da equipe com a linguagem e tipo de aplicação. A tabela a seguir (Tab. 1.6) mostra a média de produtividade (pontos por função/ hora) em diferentes setores de negócio.

Tab. 1.6: Produtividade no desenvolvimento por setor de negócios em 1999 [7]

Setor de Negócio	Produtividade (Pontos por Função/ Hora)
Manufatura	0,34
Bancário	0,12
Seguros	0,12
Administração Pública	0,23
Atacadista	0,25

1.3.2. Métricas de Software por Ponto de Caso de Uso

A técnica de estimativa por Ponto de Caso de Uso foi primeiramente proposta por Gustav Karner em 1993 [9]. Neste Modelo a estimativa do sistema é feita com forme o modo que os usuários utilizarão o sistema, a complexidade das ações de cada usuário, e

uma análise mais abstrata que a estimativa por Pontos de Funções em relação aos passos para realização de cada tarefa [8].

O método de cálculo começa com o levantamento dos casos de uso, depois a estimativa é feita a partir de um conjunto de métricas similares à técnica de Pontos por Função.

É importante saber que o grau de detalhamento dos casos de uso influencia diretamente nos resultados da estimativa. Por isso deve-se preocupar somente em medir casos de uso em nível de sistema, onde seja possível diferenciar transações e interações com o usuário. Com isso casos de uso em nível muito alto ou muito baixo não são adequados para esta técnica. Com isso cabe aos gerentes e analistas decidir a granularidade dos casos de uso utilizados nesse método [8].

O método de estimativa por Pontos por Caso de Uso é dividido em três passos:

a) Calculando o peso dos Atores.

O primeiro passo é classificar os atores envolvidos, a tabela de pesos de atores (Tab. 1.7) é utilizada para isso. Após a classificação, é feita a soma dos produtos entre o número de atores pelo respectivo peso, com isso, conseguimos o TPNAA (Total de Pesos não Ajustados dos Atores).

Tab. 1.7: Tabela de Pesos de Atores [8].

Tipo de Ator	Peso	Descrição
Simple	1	Outro sistema que é acessado de uma API de programação.
Médio	2	Outro sistema que interage através de um protocolo de comunicação, como TCP/IP ou FTP.
Complexo	3	Um usuário que interage por uma interface gráfica (Web ou <i>stand-alone</i>).

Assim um sistema que possui os seguintes atores, um gerente e um usuário comum, e ainda que fosse acessado por outro sistema por um protocolo de comunicação, ele teria 2 atores de nível complexo (usuário comum e gerente), e um de nível médio (sistema acessado por protocolo de comunicação), totalizando um valor 8 para o TPNA A.

b) Calculando o Peso dos Casos de Uso.

O segundo passo será o cálculo do TPNAUC (Total de Pesos não ajustados dos casos de usos), para o cálculo os casos de uso são divididos em três níveis, conforme o número de transações que o caso de uso possui. A tabela de casos de uso por número de transações (Tab. 1.8) mostra isso. Transação neste caso, entende-se como uma série de processos que devem necessariamente ocorrer em conjunto, ou que devem ser totalmente cancelados, se não for possível ocorrer o caso de uso (processamento) [8].

O cálculo é feito da mesma forma que para o cálculo dos atores (TPNA A), somam-se os produtos entre os casos de uso e o seu peso classificado.

Tab. 1.8: Tabela de casos de uso por número de transações [8].

Tipo de casos de uso	Número de transações	Peso
Simple	Até 3	1
Médio	De 4 a 7	2
Complexo	7 ou mais	3

Uma outra forma de calcular o TPNAUC é levar em consideração o número de classes envolvidos no caso de uso (processo), a classificação dos casos de uso é mostrado na tabela de casos por número de entidades (Tab. 1.9), e o cálculo é feito com forme a abordagem anterior. E pode ser usado quando já é possível saber o número de entidades envolvidas em cada caso de uso.

Tab. 1.9: Tabela de casos de uso por número de entidades [8].

Tipo de casos de uso	Número de Entidades	Peso
Simple	Até 5	1

Médio	De 5 a 10	2
Complexo	Mais de 10	3

A terceira e última maneira de calcular o TPNAUC, é uma regra simples para comparar todos os casos de usos. Depois são somados os valores e se consegue o TPNAUC. Esta forma é mais rápida de se calcular que as outras, porém é menos precisa, as regras são as seguintes:

1. Se o caso de uso for simples: possuir uma interface com o usuário simples, e possuir só uma entidade no banco de dados. O caso de uso recebe peso 5.
2. Se o caso de uso for médio: possuir uma interface mais trabalhada, e o número de entidades no banco de dados forem dois ou mais. Ele receberá peso 10.
3. Se o caso de uso for complexo: possuir uma interface mais complexa, e possuir três entidades no banco de dados ou mais. Ele receberá peso 15.

O peso total dos casos de uso não ajustado (PCUNA) é conseguido através da soma entre o peso dos atores e o peso dos casos de uso, $PCUNA = TPNA + TPNAUC$.

c) Calculando os Fatores de ajuste

Este passo é bastante similar ao método de ajuste do método do ponto por funções. Ele é dividido em 2 partes, uma tendo em vista os fatores funcionais do sistema, que é feito através do cálculo do Fator de Complexidade Técnica (*FCT*), e a outra os fatores não funcionais, através do cálculo do fator de Complexidade Ambiental (*FCA*), que indica a eficiência que terá o projeto, e está relacionado à experiência da equipe e a disponibilidade de cada recurso.

1. Os Fatores de Complexidade Técnica variam de 0 a 5 de acordo com a dificuldade do sistema a ser implementado. Sendo assim 0 indica que o fator não ocorre no sistema ou não o influencia, 3 indica que tem influência moderada e 5 indica que tem grande influência no processo. Após Avaliar todos os fatores basta multiplicar o valor do

fator pelo seu respectivo peso que se encontra na tabela, depois somar todos estes produtos e substituir na seguinte fórmula: $FCT = 0.6 + (0.01 * \text{Somatório dos Produtos entre fator e peso})$.

Tab. 1.10: Fatores de complexidade técnica [8].

Fator	Descrição	PESO
T1	Sistemas Distribuídos 2,0	2
T2	Desempenho da aplicação 1,0	2
T3	Eficiência do usuário final (on-line) 1,0	1
T4	Processamento interno complexo 1,0	1
T5	Reusabilidade do código em outras aplicações 1,0	1
T6	Facilidade de instalação 0,5	0.5
T7	Usabilidade (facilidade operacional) 0,5	0.5
T8	Portabilidade 2,0	2
T9	Facilidade de manutenção 1,0	1
T10	Concorrência 1,0	1
T11	Características especiais de segurança 1,0	1
T12	Acesso direto para terceiros 1,0	1
T13	Facilidades especiais de treinamento 1,0	1

2. O Fator de Complexidade Ambiental, é calculado de maneira parecida com a anterior, os valores variam de 0 a 5 de acordo com a influência no sistema. Depois de avaliados os fatores, é feito o somatório dos produtos entre o valor do fator e seu respectivo peso na tabela, e então se deve substituir na fórmula : $TCA = 1.4 + (0.03 * \text{Somatório dos Produtos entre fator e peso})$.

Tab. 1.11: Fatores de complexidade ambiental [10].

	Descrição	Peso
F1	Familiaridade com o processo de desenvolvimento de software	1,5

F2	Experiência na aplicação	0,5
F3	Experiência com OO, na linguagem e na técnica de desenvolvimento	1,0
F4	Capacidade do líder de análise	0,5
F5	Motivação	1,0
F6	Requisitos estáveis	2,0
F7	Trabalhadores com dedicação parcial	-1,0
F8	Dificuldade da linguagem de programação	-1,0

Depois de feitos estes 3 passos, basta substituir na fórmula para obter o valor total do sistema (*Pontos de caso de Uso*, PCU), $PCU = PCUNA \times FCT \times FCA$. Segundo Karner [9], o criador do modelo, deve-se estimar o tempo com 20 horas de trabalho por Ponto de Caso de Uso (PCU). Mas Schneider e Winters [11] sugerem o seguinte melhoramento:

1. Conseguir o valor de X, que é somatório dos itens com valor menor que 3 entre os itens F1 a F6.
2. Obter o valor de Y, que é a soma dos itens com valor maior 3 que entre os itens F7 e F8.
3. Somar os valores de X e Y, e comparar:
 - a. Se $X+Y < 3$, usar 20 como horas de trabalho por PCU.
 - b. Se $X+Y=3$ ou $X+Y=4$, usar 28 como horas de trabalho por PCU.
 - c. Se $X+Y > 5$, então se deve tentar modificar o projeto, buscando diminuir o valor desta soma, pois o risco de não obter sucesso é grande.

1.4 Processo de Medição

Desenvolvimento de software requer um processo ou mecanismo de medição para *feedback* e avaliação. A medição ajuda no planejamento do projeto, alocação de recursos, estimativo de custos e tempo, além de ajudar na prevenção de erros e auxílio no suporte [12].

Vários modelos de qualidade, ou certificações requerem um processo de medição que possa mensurar objetos como recursos, processos e produtos, o MPS.BR que é o modelo estudado neste trabalho também requer esse processo de medição para estes objetos, e um processo que incorpora esses objetos é o *Goal Question Metric* (GQM), que será aprestado a seguir.

1.4.1 *Goal Question Metric* (GQM)

O GQM é um processo para medir objetos como: produtos, processos e recursos, ele é baseado numa estrutura, na qual se define o que vai ser medido e uma meta (objetivos) [12]. Depois são criadas questões para avaliar a produtividade e o tempo, e busca-se respondê-las através de métricas. E todo o processo deve ser documentado e armazenado.

Um bom processo de medição deve ter:

- 1) Foco, metas específicas;
- 2) Aplicar a todo ciclo de vida de produtos, processos e recursos;
- 3) Interpretação e entendimento do contexto organizacional, do ambiente dos objetivos.

Com isso, pode-se ver que uma boa definição do escopo é fundamental para uma boa medição, especificando os objetivos, e as maneiras de se consegui-los. Por isso Basili [12], diz que o GQM é dividido nas seguintes 3 partes:

1. *Conceptual level (GOAL)*, Nível Conceitual (Metas): Neste nível uma meta é definida para o objeto em questão, através de vários pontos de vista, relativos ao ambiente determinado. Objetos a serem medidos:
 - a. Produtos: Artefatos, documentos que são produzidos durante o ciclo de vida do processo, especificações, programas, etc.;
 - b. Processo: Atividades relacionadas ao software, geralmente ligadas ao tempo;

- c. Recursos: Itens usados durante o processo, como as pessoas, os softwares, hardwares e o espaço físico.
2. *Operational level (QUESTION)*, Nível Operacional (Questões): Criar um conjunto de questões que caracterizam o objeto a ser medido, e que sejam quantificáveis para avaliar a produtividade ou a qualidade, para cada meta.
 3. *Quantitative level (METRIC)*, Nível Quantitativo (Medidas, métricas): Escolher medidas que melhor respondam cada questão, de maneira quantitativa. Uma medida pode responder uma ou mais questão, e uma questão pode ser respondida por uma ou mais medida. A medida pode ser:
 - a. Objetiva: Quando depende apenas do objeto que esta sendo medido, e não do ponto de vista que foram tirados. Exemplos: número de versões de um documento, tamanho de um programa.
 - b. Subjetiva: Quando depende de ambos, do objeto e do ponto de vista que foi tirado. Exemplos: nível de satisfação do usuário, readaptação de um texto.

Para que o processo de levantamento de metas (*goals*) seja bem feito deve-se especificar itens importantes [13]:

- **Objeto de estudo:** especifica o que será analisado (medido). Exemplo: processo de software, projeto, documento, software, etc.
- **Objetivo:** especifica porque o objeto será analisado, a sua finalidade. Exemplo: avaliar, melhorar, monitorar, controlar, etc.
- **Enfoque:** especifica o atributo que será analisado. Exemplo: confiabilidade, custos, correção, etc.
- **Ponto de vista:** especifica quem utilizará as métricas. Exemplo: equipe de desenvolvimento, gerente de projeto, etc.
- **Contexto:** especifica o ambiente onde o processo de medição está localizado. Exemplo: projeto X, departamento Y.

Para que a especificação desses itens tenha sucesso, deve-se seguir três passos básicos para conseguir a informação.

A primeira informação vem entendendo a política e a estratégia da organização, que apóia a aproximação do GQM. Conseguimos especificar o objetivo e o enfoque da meta estabelecida, analisando a política, os planos estratégicos e os componentes relevantes da organização.

A segunda informação vem da descrição dos processos e dos produtos da organização, ou pelo menos dos que desejamos que passem pelo processo de medição. Conseguimos identificar o objeto especificando o processo e o modelo de produto da organização.

A terceira informação vem do modelo de organização, que nos ajudam a compreender o ponto de vista e o contexto. Como nem todos os processos e enfoques são relevantes para todos os pontos de vista da organização, precisamos fazer uma análise profunda antes de preencher as nossas a lista de metas, para saber se as metas definidas têm a relevância certa.

De acordo com Fontoura [13], as metas GQM podem ser feitas da seguinte maneira “Analisar o <objeto de estudo> com a finalidade de <objetivo> com respeito ao <enfoque> do ponto de vista de <ponto de vista> no seguinte contexto <contexto>”.

Exemplo:

- “Analisar o Software com a finalidade de melhorar com respeito à estimativa do tempo de desenvolvimento do ponto de vista do gerente de projeto de software no seguinte contexto do departamento de desenvolvimento”.

Com isso termina-se a especificação das metas, e o próximo passo é o Nível Conceitual para o Nível Operacional. Com a especificação de cada meta, pode-se criar questões que podem medir o objeto, geralmente é feito três grupos de perguntas que vão de acordo com as seguintes (os exemplos das questões estão de acordo com a meta do exemplo que foi traçado acima):

Grupo 1: Como podemos caracterizar o objeto (recurso, produto ou processo) com respeito à meta especificada pelo GQM?

Exemplo:

- O método de estimativa está efetivamente desenvolvido?

Grupo 2: Como podemos caracterizar os atributos do objeto que são relevantes ao enfoque especificado pelo GQM?

Exemplos:

- O método está melhorando?
- O tempo da estimativa foi próximo ao tempo real de implementação?

Grupo 3: Como avaliamos as características do objeto que são relevantes ao enfoque definido pelo GQM?

- O gerente está satisfeito com o método?
- O método está realmente melhorando?

Depois que as questões foram feitas, deve-se associar cada questão com a métrica (medida) apropriada. Vários fatores podem ajudar na escolha da métrica correta, alguns deles são:

O conjunto e a qualidade dos dados existentes: tentar maximizar o uso dos dados se eles forem validados e revalidados.

Aprendendo o processo: o modelo GQM precisa sempre de refinamento e adaptação. Entretanto a medição precisa ajudar na avaliação não só do objeto, mas também do modelo de medição que está sendo usado.

Maturidade dos objetos: apoiar medição objetiva para conseguir maturidade na medição dos objetos, e usando avaliações subjetivas quando defrontar-se com objetos informais e instáveis.

Tomando-se por base essas idéias, é possível responder as questões com mais precisão.

Exemplo:

A questão: O tempo da estimativa foi próximo ao tempo real de implementação?

Pode ser respondida com a seguinte métrica: Percentual de acerto da estimativa = $(\text{tempo real} - \text{tempo estimado} / \text{tempo estimado}) * 100$

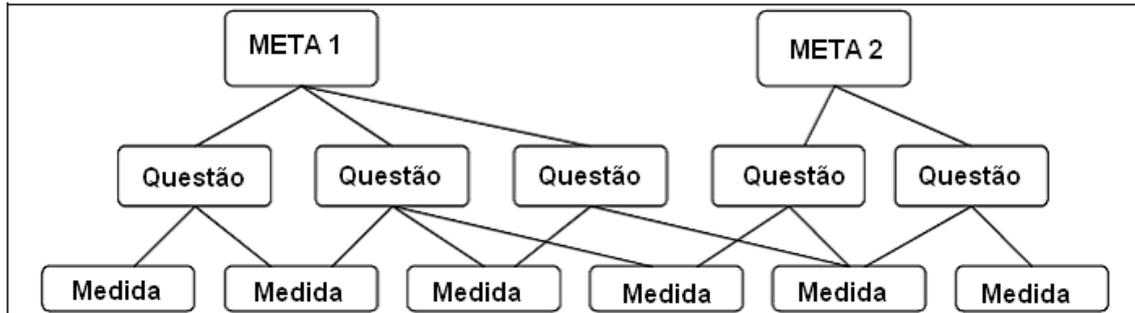


Fig. 1.2: Hierarquia do processo GQM.

Através da figura de Hierarquia do processo GQM pode-se ver que uma métrica tem a possibilidade de responder mais de uma questão da mesma meta. Em diversos modelos GQM pode-se ter questões e métricas em comum de acordo com o ponto de vista de quem estiver analisando.

Uma vez que o modelo GQM foi desenvolvido, deve-se selecionar as técnicas, ferramentas e procedimentos apropriados. Estes dados serão coletados e guardados num modelo de interpretação de acordo com as necessidades de cada empresa.

2. Melhoria de Processo do Software Brasileiro (MPS.BR)

Neste capítulo o foco será a apresentação do MPS.BR, modelo de qualidade escolhido para direcionar o trabalho desta dissertação.

Para o melhor desenvolvimento de software vários modelos de avaliação de qualidade de software são propostos para as empresas, o MPS.BR é um dos mais utilizados na região de Londrina.

O MPS.BR é um programa para a Melhoria de Processo do Software Brasileiro, é coordenado pela SOFTEX (Associação para a Promoção da Excelência do Software Brasileiro) e ainda conta com o apoio do MCT (Ministério da Ciência e Tecnologia), da FINEP (Financiadora de Estudos e Projetos) e do Banco Interamericano de Desenvolvimento (BID) [14].

Com a crescente competição por qualidade, todos os níveis devem estar adequados, tanto o produto de software, como os serviços correlatados, processos de desenvolvimento e distribuição de softwares, com isso o MPS.BR busca padronizar os processos de desenvolvimento de software para que o seu produto possa ser competitivo em nível nacional e internacional. O MPS.BR deseja ser adequado a empresas de todos os tamanhos, mas busca ser mais atencioso com empresas de micro, pequeno e médio porte. Ele tenta também ser compatível com os padrões de qualidade aceitos internacionalmente, para isso o MPS.BR aproveita conceitos de outros modelos já existentes.

O MPS.BR baseia-se no modelo de maturidade que são divididos em níveis [14], que para serem obtidos, as empresas passam por avaliações. Para ajudar nesse contexto, o MPS.BR possui três componentes, o Modelo de Referência (MR-MPS), o Método de Avaliação (MA-MPS) e o Modelo de Negócio (MN-MPS).

Assim o MPS.BR é descrito em 4 guias:

- Guia Geral: que fala de forma geral o MPS.BR e descreve o Modelo de
- Referência (MR-MPS);

- Guia de Aquisição: descreve o processo de aquisição de software e serviços correlatos. Ele apóia quem deseja adquirir produtos de software e serviços correlatos apoiando-se no MR-MPS;
- Guia de Avaliação: relata o processo e o método de avaliação MA-MPS, e os requisitos necessários;
- Guia de Implementação: possui 7 partes, cada 1 contado como implementar cada nível do MPS.BR.

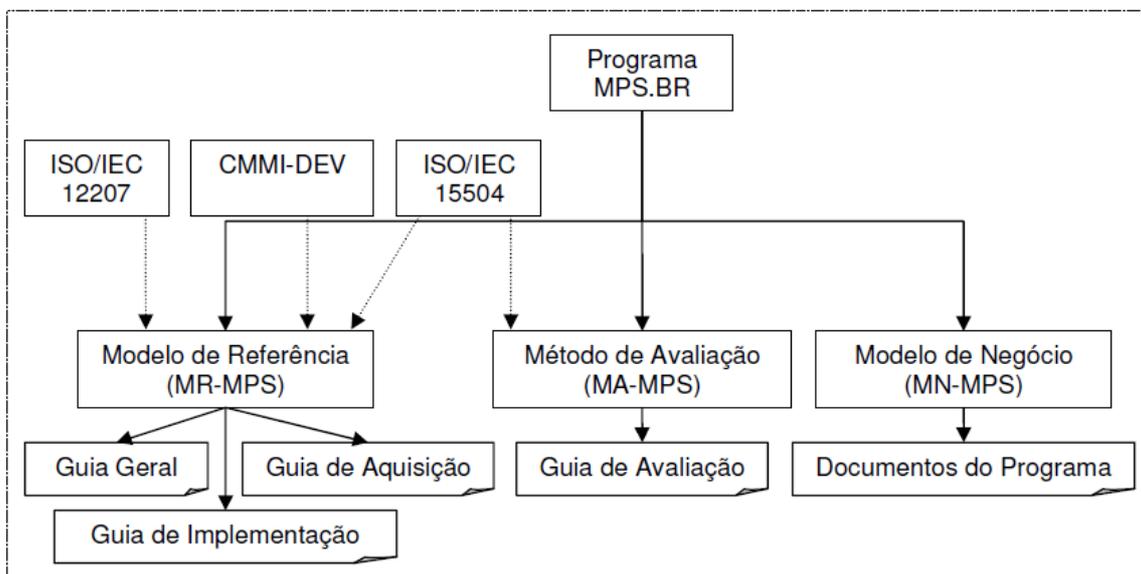


Fig. 2.1: Componentes do MPS.BR

Tab. 2.1: Níveis de maturidade do MR-MPS [14].

Nível	Processo	Capacidade
A	Inovação e implantação na organização	AP1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Análise e resolução de causas	
B	Desempenho do Processo organizacional	AP1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência Quantitativa do Projeto	
C	Análise de decisão e resolução	AP1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de riscos	
D	Desenvolvimento de requisitos	AP1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Solução técnica	
	Integração do produto	
	Instalação do produto	
	Liberação do produto	
	Verificação	
	Validação	
E	Treinamento	AP1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Avaliação e melhoria do processo organizacional	
	Definição do processo organizacional	
	Adaptação do processo para gerência de projeto	
F	Medição	AP1.1, AP 2.1 e AP 2.2
	Gerência de configuração	
	Aquisição	
	Garantia da qualidade	
G	Gerência de requisitos	AP 1.1 e AP 2.1
	Gerência de projetos	

Tabela mostra os Níveis do MPS.BR que começa no nível G e vai até o A, que é o topo de excelência do modelo. E também mostra os Atributos de Processo necessários para cada nível que será visto a seguir.

AP1.1 O Processo é executado

Este atributo mede quanto o processo atinge seus objetivos.

AP2.1 O Processo é gerenciado

Este atributo mede quanto à execução do projeto é gerenciado.

AP2.2 Os produtos de trabalho do processo são gerenciados

Este atributo mede quanto os produtos de trabalho de processo produzidos estão apropriadamente produzidos.

AP3.1 O processo é definido

Este processo mede quanto um processo padrão apóia a implementação do processo definido.

AP3.2 O Processo esta implementado

Este atributo mede o quanto um processo padrão é implementado como processo padrão para atingir os resultados, para avaliar a eficácia e o que pode ser melhorado.

AP4.1 O Processo é medido

Este atributo mede o quanto os processos de medição são usados assegurar o desempenho o alcance dos objetivos, como o apoio a objetos definidos.

AP4.2 O Processo é controlado

Este atributo mede estatisticamente quanto o processo é controlado para obter um processo estável, capaz e dentro dos limites estabelecidos.

AP5.1 O Processo é objeto de inovações

Este processo mede o quanto mudanças são identificadas durante o processo através de análises das causas como a mudança de desempenho, e de enfoques inovadores para a melhora e implementação do processo.

AP5.2 O Processo é otimizado continuamente

Este atributo mede o quanto mudanças na definição, gerência e desempenho do processo ajudam no alcance dos objetivos para a melhoria do processo.

2.1 Níveis do MPS.BR

Neste tópico será feita uma apresentação de forma rápida os níveis do MPS.BR de acordo com SOFTEX [14], e depois será feita uma abordagem mais aprofundada sobre o processo de medição do nível F, no qual serão utilizadas métricas e processos de medição de software para estudar o melhor método para a implementação do requisito de medição.

Nível G – Parcialmente Gerenciado

Este nível é composto pelo processo de Gerência de Projetos e Gerência de Requisitos, e devem atender os atributos AP 1.1 2 AP 2.1.

Processo: Gerência de Projetos

O propósito deste processo é definir e manter planos para os processos, recursos e as responsabilidades do projeto, e dar informações do projeto que permitam fazer correções quando necessário. E o propósito deste processo evolui com o aumento de maturidade da empresa.

Processo: Gerência de Requisitos

O propósito é gerenciar requisitos dos produtos e dos componentes do produto, e identificar incoerências entre requisitos, os planos e os produtos de projeto.

Nível F – Gerenciado

Este nível é composto pelos processos do nível G (anterior), e acrescido dos processos de Aquisição, Gerência de Configuração, Garantia de qualidade e Medição, todos os processos devem atender os atributos dos AP 1.1, AP 2.1 e AP 2.2.

Processo: Aquisição

O propósito desse processo é gerenciar a aquisição de serviços e/ou produtos que satisfaçam as necessidades do cliente.

Processo: Gerência de Configuração

Nesse processo deve-se conseguir estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e fornecê-los para todos os envolvidos.

Processo: Garantia de Qualidade

A finalidade deste processo é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos e recursos pré-estabelecidos.

Processo: Medição

A finalidade deste processo é coletar, analisar e relatar os dados dos produtos e processos desenvolvidos pela empresa em seus projetos, visando apoiar os objetivos da organização.

Nível E – Parcialmente Definido

Este nível é composto pelos processos do nível G e F (anteriores), e acrescido dos processos Avaliação e Melhoria do Processo Organizacional, Definição do Processo Organizacional, Gerência de Recursos Humanos e Gerência de Reutilização. Neste nível ocorre

uma evolução a primeira evolução na gerência de projetos, neste nível os projetos devem ser gerenciados de acordo com o processo definido e os planos integrados. Todos os processos devem atender os atributos dos AP 1.1, AP 2.1, AP 2.2, AP 3.1 E AP 3.2.

Processo: Avaliação e Melhoria

O propósito deste processo é determinar quanto os processos padrão da organização ajudam para alcançar os objetivos da empresa e para apoiar a empresa a planejar e realizar e implantar melhorias contínuas com base nos pontos fortes e fracos da empresa.

Processo: Processo Organizacional

Neste processo deve-se estabelecer e manter um conjunto de ativos de processo da organização e padrões de ambiente de trabalho usáveis e aplicáveis às necessidades do negocio da empresa.

Processo: Gerência de Recursos Humanos

A finalidade deste processo é prover a organização e aos projetos dela recursos humanos necessários e manter suas competências de acordo com as necessidades da organização.

Processo: Gerência de Reutilização

Neste processo deve-se gerenciar o ciclo de vida dos ativos que podem ser reutilizados.

Nível D – Largamente Definido

Este nível é composto pelos processos do nível G, F e E (anteriores), somados aos processos de Desenvolvimento de Requisitos, Integração do Produto, Projeto e Construção do Produto, Validação, e Verificação. Todos os processos devem atender os atributos dos AP 1.1, AP 2.1, AP 2.2, AP 3.1 E AP 3.2.

Processo: Desenvolvimento de Requisitos

A finalidade deste processo é definir os requisitos dos componentes do produto, do produto e do cliente.

Processo: Integração de Produtos

O propósito deste processo é formar os componentes do produto, para integrar um produto consistente e que está de acordo com os requisitos funcionais e não-funcionais do ambiente alvo ou equivalente.

Processo: Projeto e Construção do Produto

Neste processo deve-se projetar, desenvolver e implementar soluções para atender os requisitos definidos. atender os requisitos definidos.

Processo: Validação

O propósito deste processo é confirmar que o produto e seus componentes se comportarão de acordo com o estabelecido dentro do ambiente para qual ele foi feito.

Processo: Verificação

A finalidade deste processo é confirmar que todos os serviços ou produtos de trabalho do processo ou projeto estão de acordo com os requisitos já estabelecidos.

Nível C – Definido

Este nível é composto pelos processos do nível G, F, E e D (anteriores), somados aos processos de Análise de Decisão e Resolução, Desenvolvimento para Reutilização e Gerência de Riscos. Neste nível o Processo de Gerência de Reutilização evolui para ficar dentro dos resultados do processo de Desenvolvimento para Reutilização. E todos os processos devem atender os atributos dos AP 1.1, AP 2.1, AP 2.2, AP 3.1 E AP 3.2.

Processo: Análise de Decisão e Resolução

Neste processo deve-se analisar futuras decisões que podem ser tomadas, com critérios já estabelecidos num processo formal para a avaliação das melhores escolhas identificadas.

Processo: Desenvolvimento para reutilização

O propósito deste processo é verificar oportunidades de reutilização sistemática, e quando possível formar um programa de reutilização para desenvolver ativos a partir da engenharia de aplicações dos domínios.

Processo: Gerência de riscos

Neste propósito deve-se identificar, analisar, tratar, monitorar e tentar reduzir os riscos da organização e dos seus projetos.

Nível B – Gerenciado Quantitativamente

Este nível é composto pelos processos do nível anteriores (G, F, E, D e C), e no processo de Gerência de Projetos são requeridos novos resultados. Todos os processos devem atender os atributos de processo AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2 e os RAP 16 e RAP 17 do AP 4.1, sendo que os processos selecionados para análise de desempenho devem satisfazer totalmente os atributos AP 4.1 e AP 4.2.

Nível A – Em Otimização

Este nível é composto pelos processos do nível anteriores (G, F, E, D, C e B), acrescido do processo Análise de Causas de Problemas e Resolução. Todos os processos devem atender os atributos de processo AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2 e os RAP 16 e RAP 17 do AP 4.1, sendo que os processos selecionados para análise de desempenho devem satisfazer totalmente os atributos AP 4.1, AP 4.2, AP 5.1 e AP 5.2.

Processo: Análise de Causas de Problemas e Resolução

A finalidade deste processo é encontrar causas, defeitos e problemas em geral, e tomar ações para prevenir que eles não ocorram no futuro.

Na seqüência, será abordado mais detalhadamente o processo de medição nível F do MPS.BR, que é foco do trabalho.

2.2 O Processo de Medição do nível F do MPS.BR

Como a gerência de projetos que acompanha todo o desenvolvimento, ela precisa ter um apoio nos processos de gerência de processo e melhoria de processo e produto, este apoio é conseguido através do processo de medição, que é um dos principais processos para gerenciar as atividades do ciclo de vida de softwares e avaliar os riscos e viabilidades de projeto [15].

O processo de medição possui uma clara relação com todos os processos do MPS.BR, uma vez que deve ser aplicado no contexto dos tanto dos projetos e como dos processos, para a integração de dados da empresa.

O processo de medição deve ser implantado de maneira evolutiva, pois ele demonstra a maturidade da empresa, no começo as medições podem ser difíceis, a coleta de dados também, que deve ser reflexo da falta de maturidade de processos.

Alguns resultados são esperados de acordo com SOFTEX [15] com o processo de medição:

MED1 - Objetivos de medição são estabelecidos e mantidos a partir dos objetivos da organização e das necessidades de informação de processos técnicos e gerenciais.

Nesta etapa deve ser identificado o que será medido, então se tem a necessidade de informação (prioridade, derivadas de objetivos da empresa, legislação ou produtos), que pode ser passado pelos dirigentes da empresa, e dos processos técnico e gerencial. Os aspectos que se podem medir são recursos, produtos e processos.

- Recursos: elementos de entrada para a produção. Exemplos são pessoas, ferramentas, métodos que podem ser medidos, como a produtividade de um programador.

- Produtos: são os resultados (programas, projetos, código fonte) do processo. E as medidas seriam o tamanho (linhas, ponde função) e o tipo (comercial, científico, de sistemas).
- Processos: atividades relacionadas ao software (atividade de análise e projeto). Neste caso seriam quantificados atributos como: tempo, esforço, etc.

Depois definir o objetivo da medição, que documentam as finalidades para qual a medição e as análises foram feitas, e para estabelecer ações e rumo para o projeto.

MED2 - Um conjunto adequado de medidas, orientado pelos objetivos de medição, é identificado e/ou definido, priorizado, documentado, revisado e atualizado.

Com relação aos objetivos definidos, deve-se encontrar medidas que os satisfaçam. Alguns critérios podem ajudara na seleção das medidas, como: relevância em relação às necessidades de informação, disponibilidade de recursos, facilidade de coleta entre outros.

As medidas devem ser documentadas, com o seu nome, unidade de medida, descrição e a relação com a necessidade de medição, também devem ser revisadas pelos gerentes para ver se estão dentro da necessidade de informação e objetivos da organização.

Medidas podem ser classificadas em básicas ou derivadas, básicas são aquelas que são conseguidas através de um método de medição apenas, enquanto medidas derivadas são conseguidas através da combinação de duas ou mais medidas básicas ou derivadas.

MED3 - Os procedimentos para a coleta e o armazenamento de medidas são especificados.

Os dados coletados e os processos de coleta devem ser armazenados junto com o contexto em que foram coletados, para que se possa verificar e entendê-los. É interessante existir um repositório para a armazenagem dos dados, mas ele não necessário neste nível.

MED4 - Os procedimentos para a análise da medição realizada são especificados.

Para cada medida de MED2, deve estar documentadas as atividades de análise das medições através da definição da frequência, fase, responsável, dados de origem, ferramenta utilizada, verificações, e como os resultados serão demonstrados aos interessados.

MED5 - Os dados requeridos são coletados e analisados.

Os dados devem ser coletados de acordo como foi planejado, e dentro do tempo estipulado, pois coletas tardias podem trazer dados não confiáveis e não relatar a realidade. O procedimento de análise deve ocorrer da maneira planejada e pelas pessoas que possuem esta responsabilidade dentro da empresa.

MED6 - Os dados e os resultados de análises são armazenados.

Os dados e os resultados da análise, incluindo os dados e as especificações da medição, resultados da análise, indicadores e interpretações, ou seja, todas as informações necessárias para entender, interpretar recuperar as medidas devem ser armazenadas, e da maneira que foi estipulado na MED3.

MED7 - As informações produzidas são usadas para apoiar decisões e para fornecer uma base objetiva para comunicação aos interessados.

A comunicação entre os dados armazenados e as pessoas que vão utilizá-las deve ser clara, fácil e despersonalizada, de maneira que todos possam entendê-los e para evitar erros. E a comunicação também deve estar relacionada com as necessidades e os objetivos, e as medições também podem servir para ações corretivas e avaliar riscos.

3. Estudo de Casos de Métricas

O estudo foi realizado durante o estágio na empresa Guenka de Desenvolvimento de Software LTDA, situada em Londrina, Paraná, no projeto de desenvolvimento do sistema para a entidade ALGA (Associação Londrinense de Ginástica Artística).

Na busca de fomentar a disseminação da ginástica artística, a Guenka associou-se a ALGA, que tem o ginásio localizado nas proximidades da GUENKA e é declarado de utilidade pública LEI Nº 13945 - 06/12/2002, Publicado no Diário Oficial Nº 6375 de 10/12/2002. A Guenka fornece gratuitamente a ALGA soluções em tecnologia que ajudam na gestão da associação, bem como na avaliação dos novos talentos e nos resultados alcançados. Além disso, a empresa está desenvolvendo o web site da ALGA, no intuito de aumentar a divulgação da mesma e aumentar o número da procura de participantes para prática da ginástica artística.

O projeto tinha as seguintes características:

- Sistema para gerenciamento e um web site da entidade;
- 3 gerentes de projeto;
- 1 programador;
- Linguagem *ruby on rails*;
- Ambiente do cliente, a empresa ALGA, que atende mais de 100 alunos.

Foram medidas as duas aplicações, sistema de gerenciamento e web site, com as métricas por ponto de caso de uso e por ponto de função, detalhes deste estudo pode ser encontrado no relatório de estágio [16].

Abaixo estão relacionadas as tabelas, de pontos de caso de usos de uso do web site (Tabela 3.1), e por pontos de função do sistema de gerenciamento (Tabela 3.2).

Tab. 3.1: Análise por pontos de caso de uso do web site.

Métrica de Pontos por Caso de Uso	Colunas1	Colunas2	Colunas3	Colunas4
Casos de Uso	Nº transações	Total pontos caso uso	Atores	Total pontos Atores
		9		6
Home	3	1	Usuario	3
Parceiros	2	1	Administrador	3
Eventos	2	1		
Horários de aula	1	1		
Equipe Técnica	1	1		
Calendário	2	1		
Historia	3	1		
Contato	3	1		
Intranet	2	1		
Fatores Técnicos		Pesos	Valor	Produto fat x Pesos
Sistemas Distribuídos		2	0	0
Desempenho da aplicação		1	2	2
Eficiência do usuário final (on-line)		1	2	2
Processamento interno complexo		1	0	0
Reusabilidade do código em outras aplicações		1	4	4
Facilidade de instalação		0,5	4	2
Usabilidade (facilidade operacional)		0,5	5	2,5
Portabilidade		2	4	8
Facilidade de manutenção		1	3	3
Concorrência		1	0	0
Características especiais de segurança		1	1	1
Acesso direto para terceiros		1	4	4
Facilidades especiais de treinamento		1	3	3
TOTAL FT		0,915		31,5
Fatores Ambientais		Pesos	Valor	
Familiaridade com o processo de desenvolvimento de software		1,5	3	4,5
Experiência na aplicação		0,5	2	1
Experiência com OO, na linguagem e na técnica de desenvolvimento		1	3	3
Capacidade do líder de análise		0,5	4	2
Motivação		1	4	4
Requisitos estáveis		2	4	8
Trabalhadores com dedicação parcial		-1	3	-3
Dificuldade da linguagem de programação		-1	2	-2
TOTAL FA		0,875		17,5
Total pontos não ajustados		15		
Total ponto ajustados		12,009375		

Tab. 3.2: Análise de pontos por função do sistema de gerenciamento.

Arquivos		TD	AR/TR	C	Complex.	PF
Atleta	ALI	24	1	L	Baixa	7
Associado	ALI	17	3	L	Baixa	7
Avaliação do Atleta	ALI	22	2	A	Média	10
Controle de Pagamento	ALI	3	1	L	Baixa	7
Análise de Desempenho	ALI	2	2	L	Baixa	7
Gerar Documentos	ALI	2	3	L	Baixa	7
Atleta						
Cadastro de Atleta	EE	26	1	A	Média	4
Editar Atleta	EE	27	1	A	Média	4
Excluir Atleta	EE	4	1	L	Baixa	3
Associado						
Cadastro de Associado	EE	19	3	H	Alta	6
Editar Associado	EE	20	3	H	Alta	6
Excluir Associado	EE	4	3	A	Média	4
Avaliação do Atleta						
Cadastro de nova Avaliação	EE	23	2	H	Alta	6
Editar Avaliação	EE	23	2	H	Alta	6
Excluir Avaliação	EE	4	2	L	Baixa	3
Controle de Pagamento						
Efetuar Pagamento	EE	6	1	L	Baixa	3
Consulta de Pagamento	SE	1	1	L	Baixa	4
Editar Pagamento	EE	5	1	L	Baixa	3
Gerar Relatório	SE	1	1	L	Baixa	4

Nas tabelas mediu-se a quantidade de pontos de caso de uso e de pontos por função de cada uma das aplicações conforme descrito nos itens 1.3.1 e 1.3.2. Então foi realizada uma comparação de quanto tempo a equipe gastava desenvolvendo cada ponto por função, através de projeto anteriormente implementado. Para cada ponto de caso de uso foi utilizado o tempo padrão de 20 horas. Após determinado esses tempos, calculou-se a estimativa do tempo de desenvolvimento dos projetos. Para o projeto do sistema de gerenciamento foi feito um acompanhamento e constatou-se um tempo real de desenvolvimento 20% maior que a estimativa de pontos por função e de 10% menor que a estimativa por pontos casos de uso.

Ainda numa empresa é importante a mensuração de outras variáveis que interferem diretamente ou indiretamente no desenvolvimento de software, que pode ser realizado baseado no modelo GQM, e que possui aspectos a serem medidos de acordo com modelo MPS.BR, como recursos, processos e produtos. Na Guenka o GQM foi desenvolvido para medir: o tempo de desenvolvimento de software, a satisfação do Cliente, o aproveitamento de visitas para venda e o tempo de Implantação do sistema na empresa que o requisitou. Além disso foi feito um estudo do GQM, comprando-o com os resultados esperados pelo processo de medição do MPS.BR.

Exemplo de medição feita com o modelo GQM:

Objeto: Cliente.

Objetivo: medir

Enfoque: satisfação

Ponto de Vista: do gerente de vendas.

Meta: Analisar o cliente com a finalidade de medir sua satisfação em relação ao software adquirido do ponto de vista do gerente de vendas.

Questão 1: Qual percentual de clientes está satisfeito com o sistema?

Métrica 1: Percentual de clientes entrevistados que pretendem manter relação com a empresa (CS).

$CS = (\text{numero de clientes satisfeitos} / \text{número de clientes entrevistados}) * 100.$

Questão 2: Qual a satisfação em relação às funcionalidades?

Métrica 2: Percentual de funcionalidades que satisfizeram o cliente.

$FS = ((\text{n}^\circ \text{ de funcionalidades total} - \text{n}^\circ \text{ de funcionalidades insatisfeitas}) / \text{n}^\circ \text{ de funcionalidades total}) * 100.$

4. Análise de Resultados

Através da análise do Estudo de Casos chegamos aos seguintes resultados:

- **Análise de Pontos por Função e Pontos de Caso de uso**

Os resultados das estimativas das medições foram considerados bons para empresa Guenka. E dentre as causas do maior erro da estimativa de Pontos por Função, foi considerado o pequeno número de funcionalidades desenvolvidas para comparação de quanto tempo a equipe gastava desenvolvendo cada ponto por função, e a pouca experiência do desenvolvedor com a linguagem.

Analisando cada uma das métricas e as medições feitas, chegou-se aos seguintes resultados:

a) Análise de Pontos por Função

A medição por pontos de função pode ser muito eficaz, desde que o processo de contagem seja conhecido pela equipe. Deve-se tomar cuidado com experiências passadas pela empresa, pois em contextos diferentes eles não podem ser repetidos. Também se deve avaliar a disponibilidade de recursos e como melhorar ao máximo a utilização pela empresa.

Esta métrica embora pareça ser um pouco extensa e meio complicada, ela pode ser mais fácil de ser utilizada, quando o projeto a ser implementado for parecido com outro já implementado, ou quando já possuir protótipos, pois assim ficará mais fácil de visualizar entradas, saídas e consultas que o software terá bem como os arquivos que o integrarão.

b) Pontos por caso de uso

Uma vantagem desta técnica em relação à por Ponto de Função é que se pode conseguir resultados mais cedo no ciclo de desenvolvimento. E ainda a técnica utiliza-se de

um documento que é essencial em várias metodologias que utilizam casos de uso (*Use Case Driven*), como o RUP (*Rational Unified Process*), com isso seria fácil calcular mudanças nas estimativas do sistema a cada alteração pequena de requisito, refazendo alguns cálculos.

Porém o método possui problemas visíveis, como a diferença da estimativa por gerentes ou analistas de software diferentes, principalmente por causa do ponto de vista sobre a granularidade sobre os Casos de Uso. E por basear-se em requisitos que o sistema irá possuir, para levantar o cálculo da estimativa do tamanho, num nível que normalmente só é estimável depois do levantamento dos requisitos, feito os primeiros protótipos ou após uma análise detalhada junto ao cliente.

A metodologia de estimativa por Pontos de Casos de Uso vale a pena ser vista e usada, principalmente por pessoas que necessitam de uma métrica formal para estimativa de software, ou para quem não conseguiu bons resultados com outros tipos de estimativas.

- **GQM**

O *Goal Question Metric* é um mecanismo de aproximação para definição, interpretação e medição. Ele pode ser usado isoladamente, ou em conjunto com um contexto geral de aproximação de software para aumentar a qualidade. No último caso a experiência do time de projeto no ambiente e no gerenciamento ajudará no processo de descrição das metas.

O GQM por combinar no processo vários modelos atuais para medir e generalizar processos, produtos e recursos, faz com que este modelo seja adaptável em diferentes ambientes, como por exemplo, as organizações: NASA, Hewlett Packard, Motorola, Coopers & Lybrand que trabalham em vários e diferentes ambientes.

Depois de implantado o modelo GQM, as técnicas, ferramentas e procedimentos apropriados, são coletados e guardados num modelo de interpretação da maneira que a empresa tiver necessidade, com isso, ficam mais fáceis de serem acessados poupando tempo e esforço para atividades mais árduas.

- **Métricas de Software e o Processo de Medição do nível F do MPS.BR**

Através do estudo do MPS.BR pode-se ver que os resultados esperados pelo processo de medição do nível F , neste trabalho seguiu o modelo abaixo:

1. Identificar o que será medido.
 - a. Necessidade de informação (prioridade, derivadas de objetivos da empresa, legislação ou produtos);
 - b. Definir os objetivos da medição. Depois se deve julgar se os resultados são proporcionais aos recursos alocados para a tarefa.
2. Identificar o conjunto de medidas adequadas para a medição, e documentá-las. E em todas as medidas devem ser especificados os procedimentos de:
 - a. Coleta e avaliação dos dados;
 - b. Armazenamento de dados;
 - c. Análise dos dados.
3. Coletar os dados, da maneira e dentro do prazo estipulado.
4. Medir os dados coletados.
5. Analisar os dados obtidos.
6. Armazenar os dados e os resultados obtidos como foi estipulado.
7. Utilizar as informações para ajudar a tomar decisões e para comunicação dos interessados. Disponibilizar os dados de maneira que o usuário entenda para evitar que o uso desses dados o prejudique.

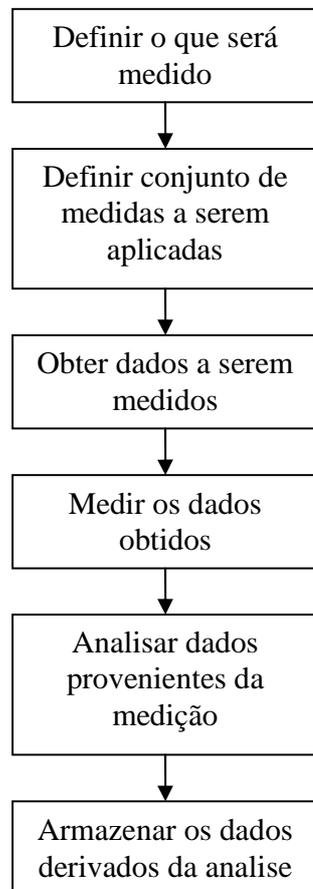


Fig. 4.1: Estrutura de medição

Esta estrutura, que foi montada através dos resultados esperados pelo processo de medição do nível F do MPS.BR é bem parecida com a do modelo GQM, desde a definição do objeto que será medido, que em ambos os casos podem ser processos, produtos e recursos, incluindo como conseguir a informação sobre o objeto, estabelecer objetivos para a medição, coleta, análise e até o armazenamento de dados, de maneira que se possam usar as informações obtidas facilmente, para melhor contextualização e ajudar em novos projetos.

Com isso se pode ver que a utilização do modelo GQM para a implantação do processo de Medição do nível F está dentro dos padrões esperados pelo MPS.BR, para a medição dos objetos, coleta de dados, definição dos objetivos, armazenagem e documentação. E na hora da medição do produto, o software a ser implementado, no momento de responder a questão definida, ou seja, quantificar o produto (software), as

métricas de Ponto por Função ou de Pontos por Caso de Uso são ótimas candidatas, podendo escolher entre elas, de acordo com a familiaridade com cada uma das métricas, com o ambiente de trabalho, com o tipo de desenvolvimento de cada equipe, ou se o projeto é parecido com algum anterior implementado, ou se já possui protótipos, entre outros fatores.

Conclusão

Através do estudo, constatou-se que as métricas podem ajudar muito no gerenciamento de projetos, que é a primeira camada do processo de engenharia de software e está presente durante todo o desenvolvimento do software. Através das estimativas das métricas é possível estipular o tempo de implementação, estimar o valor, encontrar os riscos e avaliar se realmente o desenvolvimento do software é compensatório.

Nos estudos das métricas por ponto de função e por caso de uso, pode-se concluir que a por ponto de caso de uso traz uma estimativa mais cedo no ciclo de desenvolvimento, pois o subsídio para cálculo da estimativa é obtido ainda na fase de levantamento dos casos de uso, porém por basear-se nesses dados que ainda irão ser integrados ao sistema, e por se tratar de um nível ainda inicial acabam não tendo uma maior precisão, pois alguns dados podem não ser realmente inseridos no sistema ou dados que ainda não foram percebidos podem ser integrados, ao contrário de um método, como a análise por pontos de função, que conclui a estimativa depois de levantado os requisitos, feito os primeiros protótipos ou após uma análise detalhada junto ao cliente. A análise por ponto de função é um pouco mais extensa, mas quando o projeto é semelhante a outro já implementado, ou, se esse já possui protótipos, ela pode ser utilizada mais facilmente, pois ficará mais fácil de visualizar entradas, saídas e consultas que o software terá, bem como, os arquivos que o integrarão, com isso poupará esforço e conseqüentemente tempo.

O modelo de medição GQM também foi analisado e constatou-se que além de possuir implementação rápida e relativamente fácil, oferece um modelo de medição para recursos, produtos e processos, parecido com os resultados esperados pelo processo de medição do MPS.BR. E com a ajuda das métricas por ponto de função ou por ponto de caso (dependendo da ocasião e da familiaridade com cada métrica) na hora de quantificar o software, trará um modelo de medição que além de conter os requisitos do processo de medição do MPS.BR, ajudará muito na gerência de software, pois esse além de trazer estimativas para todo o desenvolvimento do software, é adaptável a diferentes ambientes, pode guardar os dados de cada projeto, para experiência, interpretação, avaliação e reutilização deles em próximos projetos.

Então concluímos este trabalho com resultados satisfatórios, pois esse proporcionou a mim um conhecimento mais aprofundado sobre métricas, para o departamento um trabalho que pode ser utilizado para novos estudos de métricas e do MPS.BR. Para a empresa uma ferramenta que auxilia em suas estimativas, para que essa possa propor aos seus clientes um tempo mais preciso de entrega do software, ganhando credibilidade para que eles continuem adquirindo seus produtos. E como trabalho futuro pretende-se implantar o processo de medição do nível F do MPS.BR na empresa GUENKA de desenvolvimento de software.

Referências Bibliográficas

- [1] Campêlo, G. M. C., A utilização de métricas na Gerência de Projetos de Software Uma abordagem focada no CMM Nível 2, RECIFE, 2002.
- [2] Pressman, Roger S. ,Engenharia de Software, 2006.(livro)
- [3] Gimenes, I. M. S., O Processo de Software: Ambientes e Formalismos. Em: Congresso da Sociedade Brasileira de Computação, 14., 1994, Caxambu. Uma Introdução ao Processo de Engenharia de Software. Maringá: Sociedade Brasileira de Computação, 1994. Páginas 1 - 40. Disponível em: <<http://www.din.uem.br/~itana/downfiles/>>. Acessado em: 20/06/2008.
- [4] Demarco, T. *Controlling software projects*, Yourdon PressPrentice-Hall, 1982.
- [5] Vasoncelos, Alexandre , Introdução a Métricas de Software, 2005.
- [6] Hazan. Análise de Pontos de Função, 2001.
- [7] Vazquez, Carlos Eduardo; Simões, Guilherme Siqueira; Albert, Renato Machado. Análise De Pontos De Função: Medição, Estimativas E Gerenciamento De Projetos De Software (livro).
- [8] Freire, Herval Calculando Estimativas: o Método de Pontos de Caso de Uso, Developer's Magazine número 78, Fevereiro de 2003.
- [9] Karner, G. Resource Estimation for Objectory Projects. Objectory Systems, 1993 apud [FREIRE, 2003].
- [10] Heimberg, Viviane; Grahl, Everton. Estudo de Caso de Aplicação da Métrica de Pontos de Casos de Uso numa Empresa de Software.
- [11] Schneider, Geri; Winters, Jason. Applying Use Cases: A Practical Guide, 2/e, Addison Wesley Professional, 2001 apud [FREIRE, 2003].
- [12] Basili, R. B.; Caldeira, G.; Rombach, H.D., The Goal Question Metric Approach.
- [13] Fontoura, L. M.; Price, R. T. Usando GQM para Gerenciar Riscos em Projetos de Software, 18º Simpósio Brasileiro de Engenharia de Software, 2004.
- [14] SOFTEX, Guia Geral - Versão 1.2, Guia Geral do MPS-BR, 2007, disponível em www.softex.br, acesso em 27/03/2008.

[15] SOFTEX, Guia de Implementação – Parte 2: Nível F(Versão 1.1), 2007, disponível em www.softex.br, acesso em 27/03/2008.

[16] Luvizotto, Daniel C. Romano. Métricas de Software e o Mapeamento do Processo de Medição do nível F do MPS.BR.

[17] Côrtes, Mario Lúcio. Modelos de Qualidade de SW. Campinas: IC-UNICAMP. 83 slides. Disponível em: <<http://www.ic.unicamp.br/~cortes/mc726/>>, acesso em 07/08/008.

Bibliografia Consultada

CARVALHO, A. E.; TAVARES, H. C., CASTRO, J.F. B.; Medição de Pontos por Função a Partir da Especificação de Requisitos, WER, 2002.

Universidade do vale do Itajaí. Relatório de Pesquisa: Projeto Gemetrics. Itajaí, 2000.

VAVASSORI, F. B.; Souza E. W. , INTEGRANDO MÉTRICA DE SOFTWARES O GERENCIAMENTO DE PROJETOS Itajaí, 2000.