UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RUBIA MARLI SCHROEDER

APLICAÇÃO DA TÉCNICA *FUNCTION POINT ANALYSIS* PARA MENSURAÇÃO DE SOFTWARE E MELHORIA DA QUALIDADE

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

VITÓRIO BRUNO MAZZOLA

Florianópolis, Fevereiro de 2003

APLICAÇÃO DA TÉCNICA *FUNCTION POINT ANALYSIS* PARA MENSURAÇÃO DE SOFTWARE E MELHORIA DA QUALIDADE

RUBIA MARLI SCHROEDER

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Professor Fernando A. O. Gauthier, Dr.
Coordenador do Curso de Pós-Graduação

Banca Examinadora

Professor Vitório Bruno Mazzola, Dr.
Presidente da Banca (Orientador)

Professor Rosvelter João Coelho da Costa, Dr.
Membro da Banca

Professor João Bosco da Mota Alves, Dr.
Membro da Banca

Dedico este trabalho ao meu marido Wilson, meus pais Arno e Carmen, meu irmão Elias e aos meus amigos Nestor, Daltro e Marco Aurélio, compartilhando minha felicidade e gratidão.

AGRADECIMENTOS

Minha sincera gratidão:

Ao Diretor da Softcon Consultoria e Sistemas Ltda., Daltro Ferreira da Silva, pelas contribuições e informações prestadas para a efetivação deste trabalho;

Ao Diretor da AltoQi Tecnologia em Informática Ltda., Rui Luis Gonçalves, que viabilizou a aplicação prática deste trabalho;

Ao meu Orientador, Dr. Vitório Bruno Mazzola, pela paciência e compreensão;

Aos membros do fórum do BFPUG e IFPUG, pelos artigos fornecidos e inúmeras dúvidas esclarecidas;

Ao meu amigo Nestor Charles Fernandes, pelo apoio amplo e decisivo em muitos momentos;

À minha família e ao meu marido, pela compreensão e apoio;

A todos os demais que, de alguma forma, contribuíram para a realização deste trabalho.

SUMÁRIO

LISTA DE TABELAS ..... **x**

LISTA DE ILUSTRAÇÕES ..... **x**

LISTA DE ABREVIATURAS E SIGLAS ..... **xii**

RESUMO ..... **xiiii**

ABSTRACT ..... **xiii**

1. INTRODUÇÃO **1**

1.1. Justificativas 4

1.2. Objetivos 5

1.3. Premissas 6

1.4. Metodologia de Trabalho 6

1.5. Estrutura do Trabalho 8

2. QUALIDADE DE SOFTWARE **10**

2.1. A Qualidade e suas Origens 10

2.2. Os Fundamentos da Qualidade 12

2.2.1. O que é Qualidade ? 13

2.2.2. O que é Controle da Qualidade ? 14

2.2.3. O que é Garantia da Qualidade ? 14

2.2.4. O que é Controle da Qualidade Total ? 14

2.2.5. O que é Sistema da Qualidade ? 15

2.2.6. O que é Melhoria da Qualidade ? 16

2.3. Os Benefícios da Qualidade 16

2.4. A Qualidade de Software 19

2.4.1. Fatores de Qualidade de Software 20

2.4.1.1. Fatores Implícitos 20
2.4.1.2. Fatores Explícitos 21
2.4.2. A Crise do Software 23
2.5. Conclusões 25

3. NORMAS E MODELOS DE QUALIDADE 26
3.1. Benefícios da Normatização 27
3.2. As Normas da ISO 28
3.2.1. ISO 9000 29
3.2.2. ISO/IEC 12207 31
3.3. Modelo CMM 32
3.3.1. Níveis de Maturidade do CMM 33
3.3.2. As Métricas no CMM 36
3.4. SPICE 37
3.4.1. A Abordagem da Norma 38
3.4.1.1. Dimensão de Processo 38
3.4.1.2. Dimensão de Capacidade 38
3.5. Algumas Normas Complementares 40
3.6. Conclusões 41

4. MENSURAÇÃO DE SOFTWARE 42
4.1. A Dimensão do Software 43
4.2. Classificação de Métricas 44
4.2.1. Medidas, Métricas e Indicadores 44
4.2.2. Classificação de Métricas 45
4.2.2.1. Classificação Conforme o Tipo de Medida 46
4.2.2.2. Classificação Conforme o Objetivo da Medição 46
4.2.2.3. Classificação Conforme a Dimensão do Software 49
4.3. A Importância das Métricas de Software 52
4.4. Conclusões 54

5. MÉTODOS E TÉCNICAS DE MENSURAÇÃO DE SOFTWARE 56

5.1. Modelo GQM 57
5.1.1. As Fases do Modelo GQM 58
5.1.1.1. Estudo Prévio 58
5.1.1.2. Identificação de Metas GQM 59
5.1.1.3. Desenvolvimento do Plano GQM 60
5.1.1.4. Desenvolvimento do Plano de Mensuração 60
5.1.1.5. Coleta de Dados 61
5.1.1.6. Análise dos Dados 62
5.1.1.7. Interpretação dos Dados 62
5.1.1.8. Captura de Experiências 62
5.1.2. Princípios Básicos do GQM 63
5.2. COCOMO 63
5.2.1. Modelos COCOMO 64
5.2.1.1. COCOMO Básico 64
5.2.1.2. COCOMO Intermediário 64
5.2.1.3. COCOMO Detalhado 65
5.2.2. As Fases do COCOMO 65
5.2.2.1. Determinar o Tipo de Projeto 65
5.2.2.2. Determinar o Número Estimado de Instruções Fontes 66
5.2.2.3. Determinar a Estimativa de Esforço 66
5.2.2.4. Determinar a Estimativa de Prazo 67
5.2.2.5. Determinar a Estimativa de Quantidade de Pessoal 67
5.2.2.6. Determinar a Estimativa de Custo 67
5.3. Modelo de Estimativa de Putnam 68
5.4. Contagem de linhas de código fonte 68
5.5. Sistema Métrico de Halstead 69
5.6. Análise de Pontos de Função 69
5.7. Feature Points 71
5.8. Outras Técnicas 71
5.9. Quadro Comparativo das Técnicas 72
5.10. Conclusões 72

6. TÉCNICA FPA - ANÁLISE DE PONTOS DE FUNÇÃO 74
6.1. Histórico 74
6.2. Principais Objetivos 75
6.3. Terminologia 76
6.4. Tipos de Funções 77
6.4.1. Arquivo Lógico Interno 79
6.4.2. Arquivo de Interface Externa 80
6.4.3. Entrada Externa 80
6.4.4. Saída Externa 81
6.4.5. Consulta Externa 81
6.5. Cálculo de Pontos de Função – Metodologia 1 82
6.5.1. Determinar o Tipo de Contagem 82
6.5.2. Identificar as Fronteiras da Contagem 82
6.5.3. Identificar as Funções do Sistema 83
6.5.4. Classificar as Funções 83
6.5.5. Determinar os Pontos de Função não Ajustados 85
6.5.6. Determinar o Valor de Ajustamento 87
6.5.7. Determinar os Pontos de Função Ajustados 88
6.6. Cálculo de Pontos de Função – Metodologia 2 88
6.7. Precisão da Técnica FPA 90
6.8. Aplicações da Técnica FPA 91
6.9. Conclusões 91

7. APLICAÇÃO PRÁTICA DA TÉCNICA FPA 93
7.1. Descrição do Sistema 95
7.1.1. Levantamento de Funcionalidades e Requisitos 95
7.1.1.1. Cadastros e Configurações 97
7.1.1.2. Processos 99
7.1.1.3. Consultas 107
7.1.1.4. Relatórios 108
7.1.2. Características Especiais 109
7.2. Cálculo de Pontos de Função – FPA Estimativa 109

7.3. Cálculo de Pontos de Função – FPA Detalhada 111
7.4. Conclusões 121

8. CONCLUSÃO 123
8.1. Recomendações para Trabalhos Futuros 125

ANEXO 1 – MODELO PRELIMINAR DE DADOS 126

REFERÊNCIAS BIBLIOGRÁFICAS 129

LISTA DE TABELAS

Tabela 4.1 – Métricas de recursos 50
Tabela 4.2 – Métricas de processo 50
Tabela 4.3 – Métricas de projeto e produto 51
Tabela 5.1 – Quadro comparativo de técnicas de mensuração de tamanho 72
Tabela 6.1 – Classificação dos Arquivos Lógicos Internos 84
Tabela 6.2 – Classificação dos Arquivos de Interfaces Externas 84
Tabela 6.3 – Classificação das Entradas Externas 84
Tabela 6.4 – Classificação das Saídas Externas 85
Tabela 6.5 – Classificação das Consultas Externas 85
Tabela 6.6 – Determinação dos Pontos de Função não ajustados 86
Tabela 6.7 – Relação entre conhecimento do projeto e o percentual de erro 90
Tabela 7.1 – Determinação dos Pontos de Função não ajustados na FPA Estimada ...111
Tabela 7.2 – Identificação e classificação de Arquivos de Interface Externa 112
Tabela 7.3 – Identificação e classificação de Arquivos Lógicos Internos 113
Tabela 7.4 – Identificação e classificação das Entradas Externas 115
Tabela 7.5 – Identificação e classificação das Saídas Externas 117
Tabela 7.6 – Identificação e classificação das Consultas Externas 118
Tabela 7.7 – Determinação dos Pontos de Função não ajustados na FPA Detalhada ..119
Tabela 7.8 – Características gerais dos sistemas e nível de influência 120

LISTA DE ILUSTRAÇÕES

Figura 3.1 – Níveis de maturidade do modelo CMM 33
Figura 4.1 – Componentes básicos e ciclo de vida do software 43
Figura 6.1 – Componentes da FPA 78

LISTA DE ABREVIATURAS E SIGLAS

ABNT	- Associação Brasileira de Normas Técnicas
BFPUG	- Brazilian Function Point Users Group
CAD	- Computer Aided Design
CASE	- Computes Aided Software Engineering
CMM	- Capability Maturity Model
CMMI	- Copability Maturity Model Integrated
COCOMO	- Constructive Cost Model
COSMIC FFP	- Cosmic Full Function Points
CPM	- Counting Pratices Manual
FPA	- Function Point Analysis
GQM	- Goal/Question/Metrics
IEC	- International Electro Technical Commission
IEEE	- Institute of Electrical and Electronics Engineers
IFPUG	- International Function Point Users Group
ISO	- International Organization for Standardization
KPA	- Key Process Area
LOC	- Lines of Code
MK II	- Mark II Function Point Analysis
PDCA	- Plan, Do, Check and Action
SPICE	- Software Process Improvement and Capability Determination
SPR	- Software Productivity Research
TI	- Tecnologia da Informação
UKSMA	- United Kingdom Software Metrics Association
3DFP	- Boeing 3D Function Points

RESUMO

O cenário das organizações que atuam na área de software vem se alterando rapidamente nos últimos anos. O mercado globalizado está mais exigente e competitivo, fazendo com que as empresas procurem alternativas para uma melhoria na qualidade do processo e dos seus produtos. Para melhorar a qualidade, é preciso primeiro conhecer a situação real da empresa. A mensuração vem de encontro a essa necessidade, pois fornece informações sob diferentes aspectos do processo e dos produtos.

Várias técnicas vêm sendo pesquisadas para auxiliar nas atividades de mensuração. Este trabalho apresenta uma proposta de aplicação da técnica FPA - Análise de Pontos de Função para o dimensionamento do tamanho de um software.

A principal motivação deste trabalho reside no fato de que a ausência de práticas formais de mensuração de tamanho é uma das causas de sérios problemas enfrentados pelas empresas. A FPA vem crescendo no mercado de software, pois contribui de forma significativa para a melhoria do gerenciamento de projetos, permitindo a elaboração de estimativas de tamanho e a derivação de outras métricas importantes durante as fases iniciais do projeto.

Como pôde ser comprovado pelos experimentos relatados neste trabalho, mediante a aplicação das metodologias da FPA é possível obter métricas de dimensionamento de tamanho em qualquer etapa do ciclo de vida do software e com um razoável grau de precisão. Além disso, a aplicação freqüente da técnica FPA permite criar uma base de conhecimento histórica, visando a melhoria contínua do processo e refletindo em maior qualidade do produto e satisfação dos clientes.

ABSTRACT

The scenario of organizations that act in the software area has been changing quickly in the last years. The globalization of the market has been turning it out more demanding and competitive. Thus, companies have to look for alternatives of quality improvement in their processes and products. In order to improve quality in the companies it is needed, first and foremost, to know their real situation. The measurement meets such a need, because it provides information about different aspects concerning processes and products.

Various techniques have been under research in order to help the accomplishment of this goal. This dissertation aims to present an application proposal to the FPA Technique – Function Point Analysis for the size measurement of software products.

The main motivation for such a dissertation lays on the fact that the absence of formal size measurement practice has been considered as one of the causes of serious problems faced by companies. The FPA has been growing in the software market for the sole reason that it contributes in a significant way to the systems management improvement, allowing the managers to estimate the product size and the deriving process of other important metrics during the initial project phases.

As it could have been confirmed by the reported experiments in this dissertation through the FPA methodologies application, it is possible to obtain sizing metrics in any stage of the software life cycle, with a reasonable precision degree. Besides, by applying the FPA techniques, on a constant basis, to different projects allows the creation of a basis of historic knowledge aiming the continuous improvement process, resulting in a higher product quality and client satisfaction.

1. INTRODUÇÃO

Nos últimos anos, as organizações vêm criando uma dependência cada vez maior em relação à tecnologia da informação. Com o avanço da área de informática no mercado mundial, a geração de produtos de software com alta qualidade e com um custo compensador, vêm se tornando fatores essenciais para o sucesso das empresas fornecedoras e a garantia de um elevado nível de competitividade.

Como o volume de produtos de software vem aumentando, os custos de desenvolvimento e de manutenção de sistemas e a garantia da qualidade dos produtos são parâmetros críticos principalmente para a área de gerenciamento dentro das empresas.

Vários autores afirmam que a tão falada crise do software é eminentemente gerencial. Além disso, o desenvolvimento dos sistemas de informação ainda está fortemente focado somente em aspectos tecnológicos, estruturais e funcionais, ficando os aspectos de qualidade e de gerenciamento, com bastante freqüência, em segundo plano.

Como este trabalho procura demonstrar, a qualidade de produtos de software está fortemente relacionada à qualidade do processo aplicado para o desenvolvimento do produto. Para muitas organizações, a qualidade do processo de software é tão importante quanto a qualidade do produto [ROCHA 01].

Os processo e produtos de qualidade são aqueles que efetivamente agregam valor para a empresa e para os clientes que os utilizam. Para a empresa, isto significa mais flexibilidade, redução de custos de seus processos empresariais, manutenção dos clientes atuais, criação de novos nichos de mercado, aumento na participação do mercado e lucratividade. Para os clientes representa a garantia de aquisição de produtos confiáveis e que atendem as suas necessidades.

Com isso, a preocupação com a modelagem e melhoria do processo de software vem ganhando seu devido espaço nos últimos anos. Algumas abordagens importantes como as Normas ISO 9000 e a ISO/IEC 12207, o modelo CMM - Capability Maturity Model e o SPICE - Software Process Improvement and Capability Determination sugerem que melhorando o processo de software, automaticamente é possível melhorar a qualidade dos produtos.

Basicamente, a preocupação com o processo de software está relacionada à necessidade de entender, avaliar, controlar, aprender, comunicar, melhorar, prever e certificar as atividades envolvidas. Mas para isso é preciso documentar, definir, medir, analisar, avaliar, comparar e alterar os processos da empresa.

Algumas dessas necessidades podem ser alcançadas em uma empresa através da modelagem e da mensuração explícita dos fatores que influenciam o processo de desenvolvimento de software.

A mensuração ajuda as organizações e os tomadores de decisões fornecendo informações significativas com relação à qualidade, adequação e progresso evolutivo de processos, produtos e projetos de software. Além disso, a mensuração tem provado ser uma ferramenta eficaz no auxílio à gerência de projetos de software, ajudando os gerentes a identificar riscos e rastrear problemas específicos, verificar os impactos desses problemas nos custos, no cronograma e nos objetivos do projeto, desenvolver soluções alternativas e selecionar a melhor abordagem para corrigir e prevenir esses problemas.

Como em qualquer outra área de engenharia, a medição não pode garantir que um projeto tenha êxito. Entretanto, auxilia os gerentes de projeto de software a utilizar uma abordagem pró-ativa em relação aos projetos.

Procurando desenvolver formas de medidas de sistemas que fossem aceitáveis e consistentes, muitos esforços e pesquisas estão sendo realizados na tentativa de formalizar esses processos de mensuração. As organizações envolvidas com a engenharia de software vêm, durante anos, lutando em busca de técnicas e métodos quantitativos aceitáveis e consistentes, para medir a eficiência e eficácia dos processos e produtos e para gerenciar os sistemas por elas adquiridos, desenvolvidos, melhorados ou mantidos [AZEVEDO 01].

Os principais métodos e técnicas para mensuração de software conhecidos são o Modelo GQM, COCOMO, Contagem de Linhas de Código Fonte e FPA – Análise por Pontos de Função.

A técnica de Análise de Pontos de Função é uma ferramenta de mensuração utilizada para o dimensionamento do tamanho de um software, podendo ser aplicada em qualquer fase do projeto e levando em consideração a funcionalidade geral que o produto proporciona aos usuários. Sua grande vantagem está na base de sua contagem: a visão do usuário. Como está baseada em conceitos que podem ser compreendidos tanto pelo desenvolvedores quanto pelos seus usuários, permite a quantificação dos produtos em termos significativos para ambos.

A técnica FPA – Análise de Pontos de Função é o tema central deste trabalho de pesquisa, pois, devido a crescente exigência de produtos e serviços de informática de alta qualidade e baixo custo, a utilização de técnicas e métodos de mensuração vêm ganhando espaço no mercado de software e contribuindo de forma significativa para a melhoria do gerenciamento de projetos.

1.1. JUSTIFICATIVAS

Uma pesquisa abordando a aplicação de técnicas de medição em projetos de desenvolvimento e manutenção de sistemas de software pode ser justificada através de algumas questões importantes, como:

- aumento significativo do uso de software em todas as áreas de negócios;
- a busca por uma satisfação plena do cliente;
- a grande necessidade de cumprimento de prazos e cronogramas;
- a análise de economia e custo da empresa;
- a elaboração de estimativas de projeto mais próximas da realidade;
- a necessidade de uso de técnicas e metodologias visando uma padronização do processo;
- a globalização do mercado que vem exigindo o uso de normas, técnicas e metodologias no desenvolvimento de sistemas;
- a necessidade de indicadores para compreender, avaliar e otimizar os processos de desenvolvimento e manutenção em uma empresa;
- a necessidade de remediar a crise do software;
- a busca constante de melhorias na área de gerenciamento de projetos.

A partir das justificativas citadas, um estudo detalhado sobre a técnica de Análise de Pontos de Função se torna viável, pois a técnica permite estimar e medir o tamanho de um projeto de desenvolvimento ou manutenção de software, em qualquer fase do projeto, principalmente nas fases iniciais de planejamento.

Além disso, os pontos de função resultantes da aplicação da técnica podem ser utilizados em combinação com outras medidas, permitindo determinar os níveis de produtividade, o esforço de desenvolvimento do software, o custo do projeto, a taxa de produção, o tempo médio de desenvolvimento, entre outros. Essas informações são imprescindíveis para a área de gerenciamento, permitindo a criação de uma base histórica, o estabelecimento de uma linha básica (baseline) e a elaboração de estimativas para novos projetos com características iguais ou semelhantes.

1.2. OBJETIVOS

Esta dissertação tem como objetivos principais a realização de um estudo detalhado da técnica de mensuração de software FPA – Análise por Pontos de Função e a sua aplicabilidade no mercado atual de software.

A aplicabilidade da técnica será relatada através dos seguintes itens:

- um estudo de caso prático, visando determinar o tamanho do sistema através da aplicação das técnicas FPA Estimativa e Detalhada em um determinado projeto de uma empresa, cujo ramo de negócios é o desenvolvimento de sistemas;

- uma análise dos resultados obtidos após a aplicação prática da técnica FPA no projeto. Essa análise visa estabelecer alguns pontos positivos e negativos das duas metodologias da técnica FPA e apresentar sugestões para possíveis melhorias no gerenciamento de projetos da empresa.

O estudo de alguns temas complementares se faz necessário, permitindo enquadrar este trabalho de pesquisa no seu contexto mais amplo. Dessa forma, os seguintes aspectos serão analisados:

- os principais conceitos pertinentes à qualidade de produtos e serviços;
- um levantamento das normas e modelos de qualidade de software existentes, visando identificar as propostas de mensuração;
- um estudo do processo de mensuração dentro do contexto de qualidade de software;
- um levantamento das principais técnicas de mensuração de software disponíveis e a sua aplicabilidade.

1.3. PREMISSAS

O padrão reconhecido atualmente pela indústria de software para a técnica FPA é o Manual de Práticas de Contagem de Pontos de Função (CPM - Counting Practices Manual) divulgado pelo IFPUG - International Function Point Users Group. A versão mais recente do CPM é a 4.1.1, divulgada em 2000.

O IFPUG é uma entidade sem fins lucrativos, composta por pessoas e empresas de diversos países cuja finalidade é promover um melhor gerenciamento dos processos de desenvolvimento e manutenção de software através do uso da FPA. No Brasil, a representação oficial do IFPUG é através do BFPUG - Brazilian Function Point Users Group.

Futuros estudos que porventura venham compartilhar os resultados aqui obtidos, devem considerar que este trabalho de pesquisa está fortemente baseado nas regras de normatização da técnica FPA definidas pelo IFPUG, e divulgadas em território nacional pelo BFPUG. Algumas variações dessa técnica definidas por outras instituições serão apresentadas apenas para fins informativos.

O Capítulo 7 deste trabalho relata um estudo de caso de aplicação da técnica FPA em uma empresa catarinense, visando determinar o tamanho do sistema proposto em pontos de função. Por motivos confidenciais e de âmbito restrito à empresa, a derivação de outras métricas a partir dos resultados obtidos com a aplicação da FPA e a determinação de uma linha de base dentro da empresa serão apresentadas como sugestões de melhoria, porém não fazendo parte do escopo deste trabalho.

1.4. METODOLOGIA DE TRABALHO

A metodologia adotada para este trabalho de pesquisa seguirá as seguintes etapas de estudo:

- serão estudados os conceitos relativos à qualidade de software;

- serão estudados os modelos e as principais normas de qualidade existentes no mercado, buscando evidenciar familiaridades e particularidades no que diz respeito ao processo de mensuração;

- serão estudados os conceitos e processos de mensuração dentro do contexto de qualidade de software;

- será efetuado um levantamento e um estudo das técnicas de mensuração de software disponíveis, procurando apresentar seus objetivos e as principais vantagens e desvantagens;

- será efetuado um estudo detalhado da técnica FPA e a aplicação prática da técnica em um sistema seguindo as regras definidas no Manual de Práticas de Contagem de Pontos de Função publicada pelo IFPUG;

Para evidenciar a aplicabilidade da técnica FPA, a metodologia englobará as seguintes etapas:

- Seleção de um determinado sistema de uma empresa catarinense, cujo ramo de negócios é o desenvolvimento de sistemas
- Análise e levantamento de informações do sistema junto ao usuário
- Descrição do sistema
- Elaboração de documento contendo os requisitos e funcionalidades do sistema
- Elaboração do Modelo Preliminar de Dados
- Cálculo da FPA Estimativa
- Análise detalhada do sistema
- Cálculo da FPA Detalhada
- Conclusões e Análise de Resultados

1.5. ESTRUTURA DO TRABALHO

O escopo desta dissertação está dividido nos seguintes capítulos:

Capítulo 2 - Qualidade de Software

Para um melhor entendimento do contexto deste trabalho, serão apresentados diversos estudos que tratam de critérios para a garantia da qualidade. Apesar da dimensão do tema, este módulo tem como objetivo principal abordá-lo de uma forma resumida, porém buscando a maior abrangência possível no que diz respeito à área de software.

Capítulo 3 - Normas e Modelos de Qualidade

Neste módulo, serão abordados as principais normas e modelos usados no desenvolvimento de software. Segundo os autores, antes de adotar ou estabelecer uma norma, é importante verificar se ela atende a alguns requisitos básicos, garantindo assim que a normatização trará benefícios. Dessa forma, os requisitos básicos e benefícios esperados também serão devidamente descritos neste capítulo.

Capítulo 4 - Mensuração de Software

De uma forma geral, existem quatro motivos principais para adoção de programas de mensuração: para compreender, avaliar, estimar e melhorar. Neste capítulo, serão apresentados os principais conceitos e aplicações relacionadas à mensuração de software, bem como a classificação das métricas de software sob diferentes aspectos, dependendo do enfoque dado.

Capítulo 5 – Métodos e Técnicas de Mensuração de Software

As organizações envolvidas com a engenharia de software vêm, durante anos, lutando em busca de métodos quantitativos aceitáveis e consistentes, para medir a eficiência e eficácia dos processos e produtos e para gerenciar os sistemas por elas adquiridos, desenvolvidos, melhorados ou mantidos. Neste módulo, serão estudados os principais métodos e técnicas para mensuração de software conhecidos e utilizados no mercado de informática.

Capítulo 6 – Técnica FPA - Análise de Pontos de Função

Um fator considerado crítico no processo de desenvolvimento de software é a elaboração de estimativas de prazo, esforço, custo do projeto e tamanho do software. A Análise por Pontos de Função - FPA é uma das técnicas que vem sendo pesquisada para auxiliar no dimensionamento do tamanho de produtos de software e será apresentada em detalhes neste capítulo.

Capítulo 7 – Aplicação Prática da Técnica FPA

Neste capítulo, será apresentado um estudo de caso de aplicação da técnica FPA em uma empresa catarinense, cujo ramo de negócios é o desenvolvimento de sistemas. Para a aplicação da técnica, foi realizada uma análise de um sistema desta empresa, em fase final de testes, e aplicada a técnica FPA para obter os valores de tamanho do projeto utilizando a metodologia estimativa e detalhada. Após a aplicação da técnica, também foi realizada uma análise de resultados, visando confrontar os valores obtidos via FPA Detalhada e FPA Estimativa. O objetivo deste capítulo da dissertação é evidenciar a aplicabilidade da técnica FPA em uma empresa do mercado de informática e exemplificar o processo de contagem de pontos de função em uma aplicação prática.

2. QUALIDADE DE SOFTWARE

Para melhorar o entendimento do contexto deste trabalho de pesquisa, é conveniente abordar um assunto bastante extenso e excitante: Qualidade.

Apesar da dimensão do tema, este módulo tem como objetivo principal abordá-lo de uma forma resumida, porém, buscando a maior abrangência possível no que diz respeito à área de software.

2.1. A QUALIDADE E SUAS ORIGENS

A aplicação e a exigência da qualidade sempre esteve intrínseca na natureza humana. Desde há muitos séculos atrás, a busca pela qualidade em serviços e produtos tem sido feita de forma formal ou informal.

A História conta que até mesmo na época dos artesões já era constatada a existência informal do processo de qualidade, pois os próprios artesões analisavam as necessidades do mercado, produziam, vendiam o produto, realizavam a inspeção do produto e mantendo contato direto com o cliente, também conseguiam avaliar o grau de satisfação.

Mesmo depois, com o desenvolvimento do comércio e da indústria, esta situação informal ainda perdurou por muito tempo. Porém, o grande diferencial neste período foi o aparecimento das primeiras funções de supervisão e inspeção. Ainda que informalmente, a responsabilidade de garantia de qualidade dos produtos era dos supervisores e inspetores.

Finalmente, o início de uma aplicação mais formal da qualidade ocorreu no Japão a partir da década de 50, quando, na tentativa de auxiliar na reconstrução do país após a guerra, o Dr. Eduard Deming implantou várias técnicas de qualidade no comércio e na indústria japonesa.

Segundo a bibliografia, Deming é considerado o precursor na disseminação de conhecimentos e de técnicas formais de qualidade. Além disso, Deming também é conhecido como o responsável pelo milagre japonês [FERNANDES 95].

A partir da década de 60, o nível de exigência dos serviços e produtos cresceu de uma forma gigantesca, principalmente em razão do início da globalização da economia. Desde então o assunto tem tomado sua devida importância e a busca por técnicas e melhorias na qualidade vem se tornando relevante nas organizações que enfrentam a competitividade do mercado.

Muitos pesquisadores vêm contribuindo de forma incisiva para o desenvolvimento de conceitos, métodos, técnicas e aplicações práticas da Qualidade. Entretanto, as maiores contribuições vieram dos seguintes autores:

- Deming
- Juran
- Feigenbaum
- Ishikawa
- Crosby
- Imai
- Osada

Deming é considerado o “pai da qualidade”, pois foi um dos pesquisadores que mais contribuiu para o desenvolvimento da qualidade nos ambientes industriais. A sua abordagem está fundamentada em 4 itens principais:

- Os 14 Pontos de Deming, relacionando as principais questões que devem ser consideradas durante a implantação de um processo de qualidade;

- A Teoria do Saber Profundo, indicando as principais áreas de conhecimento que precisam ser exploradas profundamente para o sucesso da gestão da qualidade, como por exemplo, os aspectos relativos às pessoas;

- A Reação em Cadeia, que define o fluxo natural da empresa e de seus negócios a partir da implantação da qualidade;

- O Ciclo de Deming, mais conhecido como PDCA (Plan, Do, Check and Action). O ciclo define o fluxo de implantação de um processo de qualidade, cuja característica principal é a melhoria contínua do processo.

Da mesma forma, Juran também deixou sua grande contribuição. Ele foi responsável pela elaboração da famosa trilogia, freqüentemente usada nas empresas para definir as etapas de uma gestão da qualidade, e composta por:

- Planejamento da Qualidade
- Controle da Qualidade
- Melhoria da Qualidade

2.2. OS FUNDAMENTOS DA QUALIDADE

Existem inúmeras definições para o conceito de Qualidade, que variam de acordo com o enfoque dado ao assunto. Alguns autores sugerem que qualidade é atender ou exceder as expectativas dos clientes. Outros, por sua vez, colocam que qualidade é um modo de gerenciar a empresa.

Neste item, serão apresentadas algumas definições de Qualidade sob a ótica de vários autores e os principais conceitos relacionados.

2.2.1. O QUE É QUALIDADE ?

As seguintes definições de qualidade foram levadas em consideração:

- A boa qualidade significa um grau previsível de uniformidade e confiabilidade a baixo custo, estando a qualidade adequada ao mercado. Em outras palavras, a qualidade implica em atender as expectativas e requisitos dos clientes.

- A qualidade é a composição total das características de marketing, engenharia, fabricação e manutenção de um produto ou serviço, através das quais o mesmo produto ou serviço, em uso, atenderá as expectativas do cliente. Em resumo, a qualidade é um modo de gerenciar a empresa.

- A qualidade é a adequação ao uso do ponto de vista do cliente.

- A qualidade é conformidade com os requisitos.

- Um produto ou serviço de qualidade é aquele que atende perfeitamente as necessidades do cliente, de forma confiável, de forma segura e no tempo certo.

Apesar de existirem inúmeras definições, o foco central sempre considera a qualidade em termos da satisfação do cliente. O cliente representa todo aquele que está envolvido no processo, podendo ser tanto interno quanto externo. Portanto, qualquer produto ou serviço que satisfaça plenamente as necessidades e requisitos de todos aqueles envolvidos no processo apresenta qualidade [LISTER 98].

Entretanto, as necessidades dos clientes e os requisitos dos produtos variam com o tempo e com os avanços da tecnologia. Por isso é que nenhuma empresa pode afirmar que chegou no topo em termos de qualidade. O processo é dinâmico e evolutivo, e para qualquer empresa a qualidade deve ser uma busca contínua [FERNANDES 95].

2.2.2. O QUE É CONTROLE DA QUALIDADE ?

O controle da qualidade envolve a aplicação de técnicas e atividades operacionais objetivando o monitoramento de um processo e a eliminação das causas de desempenho insatisfatório, de modo a se obter uma eficiência econômica [FERNANDES 95].

A partir do momento que uma organização visa garantir a qualidade do processo e de seus produtos, o controle se torna uma atividade indispensável e constante. A partir deste controle é possível coletar informações estatísticas, analisar e corrigir possíveis desvios, bem como garantir a plena aplicação dos procedimentos de qualidade.

2.2.3. O QUE É GARANTIA DA QUALIDADE ?

Garantia da qualidade significa garantir a qualidade de um produto para que o consumidor possa comprá-lo com confiança e usá-lo por um longo período de tempo com satisfação e confiança.

Uma visão mais abrangente e atualizada considera que a garantia da qualidade tem como finalidade assegurar que todas as atividades da qualidade (qualidade do projeto, processo, produção e outros) estão sendo conduzidas da forma requerida.

De uma forma geral, a garantia da qualidade verifica as atividades realizadas em cada fase do projeto, tendo como procedimento padrão uma série de requisitos definidos previamente para cada etapa. Normalmente, a qualidade é garantida através de auditorias de produtos, de processos, de desenvolvimento e da própria qualidade.

2.2.4. O QUE É CONTROLE DA QUALIDADE TOTAL ?

Existem duas abordagens para definir o Controle da Qualidade Total:

A primeira abordagem, mais conhecida como o Controle da Qualidade Total Por Toda a Empresa, é japonesa e tem como meta melhorar o desempenho das empresas utilizando uma metodologia apropriada e científica e envolvendo todas as pessoas da empresa, desde o topo até a base, em todos os departamentos. Isso significa que todos os envolvidos precisam conhecer com clareza a importância das atividades que executam e como cada etapa do trabalho interfere nos resultados finais. Além disso, esta abordagem também inclui a participação dos fornecedores, filiais e redes de distribuição.

A segunda abordagem, denominada ocidental, tem como ponto focal a existência de uma área dentro da organização especializada no controle da qualidade.

2.2.5. O QUE É SISTEMA DA QUALIDADE ?

Os Sistemas da Qualidade tem como objetivo a determinação de métodos e procedimentos que auxiliem a empresa no controle de processos e prevenção de qualquer não conformidade, garantindo a aceitabilidade de seus produtos ou serviços pelo cliente.

Para que um Sistema da Qualidade seja implantado com sucesso, é necessário que a empresa esteja preparada e receptiva, e principalmente, conquiste o apoio de toda a equipe de colaboradores. Isso significa dizer que todas as pessoas devem estar envolvidas e realmente comprometidas com este objetivo.

No capítulo 3, serão apresentados alguns modelos e normas de qualidade, com ênfase na área de software, que viabilizam e auxiliam a implementação de Sistemas da Qualidade em uma organização.

2.2.6. O QUE É MELHORIA DA QUALIDADE ?

O ciclo PDCA, também conhecido como ciclo de Deming, é uma abordagem para a gestão de qualidade de uma empresa, podendo ser utilizado tanto para a manutenção como para a melhoria contínua da qualidade dos processos e produtos. A idéia básica do ciclo PDCA é justamente a busca incessante de melhorias, através da execução contínua das seguintes etapas:

- **Plan** ou Planejamento: consiste no estabelecimento de um plano de melhoria, definição das metas de qualidade e definição dos métodos que permitirão a execução destas metas. Os recursos disponíveis também devem ser levados em conta nesta etapa;

- **Do** ou Execução: consiste na realização efetiva das atividades previstas no planejamento. O acompanhamento do processo e a coleta de dados são imprescindíveis para uma análise posterior;

- **Check** ou Verificação: consiste na análise dos resultados coletados na etapa anterior. Uma comparação da meta alcançada com a planejada faz parte desta etapa;

- **Action** ou Ação Corretiva: consiste na tomada de ações corretivas para os problemas, não-conformidades e grandes variações encontrados no processo. Nesta etapa, os desvios devem ser corrigidos em definitivo para que não se repitam, procurando sempre identificar e atacar a causa do problema.

2.3. OS BENEFÍCIOS DA QUALIDADE

Os clientes são extremamente beneficiados quando recebem produtos ou serviços com qualidade, pois têm a garantia que estão adquirindo um produto ou serviço confiável e que satisfaz as suas necessidades.

Além disso, a existência de qualidade nos produtos e serviços também gera uma série de benefícios para a empresa:

- conquista e fideliza os clientes, pois sempre que houver novas necessidades, o cliente procurará os fornecedores que outrora lhe trouxeram produtos ou serviços confiáveis;
- garante um nicho de mercado para a empresa;
- a empresa pode apresentar um custo competitivo;
- a satisfação dos clientes;
- o aumento da produtividade, pois os processos passam a ser controlados e melhor gerenciados.

Mas não é apenas a empresa e o cliente que são favorecidos por um Sistema da Qualidade. De uma forma genérica, é possível identificar cinco grupos que interagem entre si e que são beneficiados de alguma maneira: os clientes, os colaboradores, os fornecedores, os proprietários e acionistas e a sociedade.

Mas, a qualidade só traz benefícios ?

É óbvio que muitas vezes os interesses dos grupos se tornam conflitantes: os colaboradores, por exemplo, podem desejar maiores salários, o que implicaria em maiores custos, que afetariam o preço final do produto, o que não seria interessante para o cliente. A Gestão da Qualidade tem como missão administrar estes conflitos, buscando adotar estratégias e processos que equilibrem os interesses dos vários grupos.

À medida que tais conflitos são resolvidos e gerenciados, o aumento de produtividade é notório. O aumento de produtividade não significa somente um aumento de faturamento, ou de vendas, ou de serviços prestados. Se estes aumentos implicam em um aumento significativo de investimentos, então não houve uma melhora na produtividade. É possível dizer que uma empresa é produtiva se ela obtém mais resultados com menos custos.

Outro fator importante que deve ser considerado é a redução dos custos da má qualidade. Neste caso específico, quando se fala em custos, não são os custos para se obter qualidade, mas sim, os custos decorrentes da falta de qualidade, ou seja, a localização de defeitos no produto ou serviço e o retrabalho para corrigi-los. Esses tipos de custos da má qualidade são agrupados como segue:

- **Custos de Falhas Internas**: são os custos decorrentes de defeitos encontrados antes mesmo da entrega do produto ou serviço ao cliente, como por exemplo, retrabalho, refugo, reinspeções e análise de falhas;

- **Custos de Falhas Externas**: relativos aos defeitos encontrados após a entrega do produto ou serviço ao cliente, como por exemplo, ajustes de reclamações, devoluções e concessões;

- **Custos de Avaliação**: são os custos para determinar o grau de conformidade do produto ou serviço aos requisitos da qualidade que foram previamente definidos, como por exemplo, inspeções, auditorias e testes;

- **Custos de Prevenção**: são custos para manter os custos de falhas internas, externas e de avaliação o mais baixo possível, como por exemplo, planejamento da qualidade, auditorias, planejamento e controle do processo, treinamentos e avaliações.

Para alcançar os benefícios da qualidade, além das barreiras citadas, várias outras também precisam ser enfrentadas. Dentre elas, a necessidade de adequar ou modernizar o capital material (equipamentos e processos), pode esbarrar em limitações financeiras.

Outro problema é a necessidade de despertar a motivação e alocação de tempo a todas as pessoas envolvidas.

Certamente o emprego de um Sistema da Qualidade traz vantagens, mas não se pode esquecer que a implantação da qualidade requer um investimento, e conseqüentemente existem custos associados. Em um primeiro momento, certamente haverá apenas custos, mas a partir do momento que existe um retorno, e este superar os custos do investimento, pode-se dizer que o Sistema da Qualidade implantado trouxe benefícios e melhorias.

2.4. A QUALIDADE DE SOFTWARE

Após uma abordagem dos conceitos relativos à qualidade, é possível focar o assunto em uma área específica: a área de software. Os conceitos fundamentais e os benefícios apresentados anteriormente foram expostos de uma maneira genérica, porém, são totalmente aplicáveis à área de software.

Aplicar os conceitos de qualidade aos produtos de software e ao processo de desenvolvimento de software não é uma tarefa fácil, pois muitas pessoas acreditam que desenvolver sistemas é uma arte que não precisa seguir regras, normas ou padrões [INTHURN 01].

Apesar de ser uma área bastante refratária à implantação da qualidade, o desenvolvimento de software com qualidade é um assunto amplo, complexo e que vem sendo muito discutido e pesquisado nos últimos anos.

De uma forma resumida, os processos e produtos de software com qualidade são aqueles que efetivamente agregam valor para a empresa e para os seus clientes [FERNANDES 95].

Além disso, a qualidade de software também pode ser vista como um conjunto de fatores ou características que devem ser alcançadas em um determinado grau para que o produto atenda às necessidades de seus usuários [ROCHA 01].

É através desse conjunto de características que a qualidade de um processo e produto de software pode ser definida e avaliada. Mais adiante, serão apresentados os principais fatores considerados críticos quando se trata de qualidade de software.

É importante ressaltar que a qualidade do software não deve ser restrita ao produto, mas deve estar concentrada também no próprio projeto e processo de desenvolvimento. Portanto, a melhoria da qualidade de um produto está fortemente baseada em uma compreensão detalhada e um controle sistemático de todo o processo de desenvolvimento [INTHURN 01].

2.4.1. FATORES DE QUALIDADE DE SOFTWARE

São vários os fatores que precisam ser considerados para se obter um resultado satisfatório em termos de qualidade de software. Estes fatores são categorizados em explícitos e implícitos [FERNANDES 95].

2.4.1.1. FATORES IMPLÍCITOS

Os fatores implícitos, por sua vez, são apresentados pelos desenvolvedores, gerentes e outras pessoas internas da organização. Correspondem aos elementos necessários para atender os requisitos dos clientes e principalmente para atender a produção do software de forma econômica.

Exatidão das Estimativas: fator que considera a extensão na qual as estimativas do projeto em termos de prazo, esforço, custo e tamanho do software são atingidos.

Eficiência: corresponde à quantidade de recursos computacionais e de códigos requeridos pelo sistema para desempenhar uma função.

Manutenibilidade: é o esforço requerido para localizar e remover um defeito em um módulo ou no programa do sistema, bem como fazer modificações específicas no software.

Testabilidade: é o esforço requerido para testar um programa ou módulo assegurando que o mesmo desempenha a função designada.

Flexibilidade: corresponde ao esforço necessário para modificar um programa ou módulo.

Portabilidade: este fator indica o esforço requerido para transferir um programa, módulo ou o sistema como um todo de uma plataforma de hardware e/ou software para outra.

Reusabilidade: é a extensão na qual um programa pode ser usado em outras aplicações; este fator está relacionado ao empacotamento e escopo das funções que o programa desempenha.

Interoperabilidade: indica o esforço requerido para interagir ou integrar sistemas entre si.

Estabilidade do Software: é a extensão na qual os fatores acima são mantidos ao longo da vida útil do produto de software.

2.4.1.2. FATORES EXPLÍCITOS

Os fatores explícitos normalmente são apresentados pelo cliente e consistem basicamente nas expectativas do cliente em relação ao produto ou serviço que lhe foi entregue. Estes fatores abrangem a qualidade do projeto, do processo e do produto final, bem como o atendimento após a entrega e manutenção do produto. Os fatores explícitos são:

Prazo do Projeto: é o prazo desejado pelo cliente ou estabelecido em termos contratuais.

Informações sobre Progresso: é importante que as informações sobre o andamento do projeto sejam disponibilizadas para o cliente. A periodicidade pode ser também estabelecida em termos contratuais.

Atendimento Funcional: corresponde a extensão na qual o sistema satisfaz suas especificações e atinge os objetivos, necessidades e requisitos dos usuários.

Confiabilidade: corresponde à capacidade do software em manter seu nível de desempenho, executando com precisão as funções requeridas, sob condições previamente estabelecidas e por um determinado período de tempo.

Integridade: é o fator que indica o nível de controle de acesso ao sistema ou dados do mesmo, por pessoas não autorizadas.

Usabilidade: é o esforço requerido para aprender, operar, preparar entradas e interpretar as saídas do sistema, considerando o julgamento individual desse uso por um conjunto de usuários.

Retorno sobre o Investimento: corresponde aos benefícios econômicos obtidos pelo cliente através do sistema.

Tempo de Atendimento: correspondem aos tempos de atendimento para manutenções ou melhorias dos produtos requeridos pelo cliente.

No desenvolvimento de um produto de software não se deve almejar a qualidade perfeita e total, mas sim um nível de qualidade necessário e suficiente para o uso especificado. A qualidade tem o seu custo e não é possível atingir todas as características de qualidade no mais alto nível. É importante identificar quais são as

características de qualidade necessárias para um determinado produto de software e o respectivo nível que satisfazem as necessidades dos usuários.

Entretanto, para atingir os níveis de qualidade considerados aceitáveis é crucial que sejam aplicadas medições e consecutivamente, comparações dos resultados obtidos com padrões previamente estabelecidos pela empresa. A avaliação da satisfação do cliente é um exemplo de medição, e quando realizada com um certo grau de freqüência, ajuda a definir um padrão, permitindo posteriormente comparações e análise de resultados.

2.4.2. A CRISE DO SOFTWARE

Apesar da existência de modernas ferramentas de desenvolvimento, a área de software tem vivido a famosa **crise do software**, que pode ser caracterizada por produtos e serviços de baixa qualidade, prazos e orçamentos ultrapassados, métodos de gerenciamento artesanais e processos baseados em “apagar incêndios”.

Além disso, é muito comum se ouvir reclamações da baixa qualidade de software e de seu elevado custo de desenvolvimento.

Os problemas mais comuns encontrados na área de desenvolvimento e que evidenciam a crise do software são [HOWER 99]:

- **Especificação dos Requisitos:** na maioria das vezes não é realizada uma documentação formal especificando os requisitos dos sistemas e submetida à aprovação dos usuários.

  Mesmo assim, quando esta documentação é feita, pode ainda pecar pela falta de clareza e objetividade, estar incompleta ou muito genérica e causar dúvidas e inconsistências na interpretação do conteúdo;

- **Cronogramas:** muitas vezes os cronogramas são definidos sem levar em consideração os requisitos reais do sistema e os recursos disponíveis. Normalmente, não existe um embasamento histórico de projetos anteriores e as estimativas de tempo são projetadas em função das disponibilidades econômicas. Fatalmente, se existe muito trabalho para pouco tempo, os problemas são inevitáveis.

- **Testes:** a fase de testes muitas vezes é desconsiderada pelos gerentes e desenvolvedores. Com isso, as falhas no sistema são detectadas pelos próprios clientes, o que caracteriza falta de qualidade no produto.

- **Novas Necessidades:** a detecção de novas necessidades por parte dos usuários durante o processo de desenvolvimento é muito comum, tornando o produto infindável.
  Uma boa alternativa para minimizar este problema é a especificação de requisitos do sistema sob forma contratual, aprovada e assinada pelos usuários. Qualquer inovação requer um novo contrato e projeto.

- **Comunicação:** a falta de comunicação entre clientes e fornecedores pode evidenciar produtos que não atendem as necessidades dos clientes. Muitas vezes, os desenvolvedores desconhecem as reais necessidades dos usuários. Por outro lado, os clientes criam expectativas errôneas em relação ao produto, causando frustrações no momento da entrega do software.

A crise de software é uma crise eminentemente gerencial, muitas vezes transcendendo as responsabilidades dos gerentes de software, pois sem o patrocínio da administração da empresa, dificilmente uma inovação ou implantação de métodos de gestão da qualidade é bem sucedida [FERNANDES 95].

2.5. CONCLUSÕES

Este capítulo apresentou algumas definições de Qualidade e os principais conceitos relacionados. Apesar de existirem inúmeras definições, muitas pessoas acham difícil definir o que é Qualidade.

Por outro lado, é muito simples detectar quando um produto ou serviço não possui qualidade: reclamações dos clientes, retrabalho, desgastes no relacionamento entre clientes e fornecedores, insatisfação mútua, discussões e rompimento de contrato são apenas algumas das conseqüências.

Mesmo assim, um produto pode satisfazer plenamente um determinado cliente, enquanto que, um outro cliente pode garantir que a qualidade do produto é insatisfatória. Quem estará com a razão?

Estas e muitas outras questões são levantadas quando se trata de definir qualidade. O ideal é que se busque uma definição que seja a mais adequada para o momento e o meio em que se vive. Uma rápida comparação entre as definições apresentadas evidenciou que, apesar de existirem diferentes enfoques, o objetivo final é atender satisfatoriamente as necessidades e expectativas de todos os envolvidos no processo.

Com relação à qualidade de software, neste capítulo foi ressaltada a importância de se estabelecer um processo de software bem definido, visando o desenvolvimento de produtos de alta qualidade e com custo reduzido. Nos últimos anos, as empresas têm dado excessiva importância aos aspectos técnicos e sendo os aspectos gerenciais e de qualidade freqüentemente negligenciados. Este foi um dos fatores que acabou desencadeando a conhecida crise do software.

Felizmente, a rápida disseminação dos conceitos de qualidade para processos e produtos de software e a importância econômica e globalizada que vem sendo atribuída à área de qualidade de software, têm contribuído para minimizar uma série de problemas e acelerar as tentativas de reversão da crise.

3. NORMAS E MODELOS DE QUALIDADE

A globalização da economia tem influenciado as empresas produtoras e prestadoras de serviços de software a alcançar um patamar de qualidade e produtividade internacional para enfrentarem a competitividade cada vez maior [ROCHA 01].

Em alguns ramos de negócios, onde a competição é extremamente acirrada, a empresa necessita ter uma competência estabelecida para sobreviver e crescer no seu negócio [FERNANDES 95].

Estas empresas estão implantando modelos de qualidade em seus ambientes de desenvolvimento de software ou procurando a certificação e melhoria do processo e produto através do uso de normas.

A implantação de modelos e normas de qualidade é uma tarefa em longo prazo, pois envolve aspectos de mudança cultural dentro da empresa. Além disso, os novos procedimentos de trabalho trazem total transparência ao processo de desenvolvimento, apontando claramente as falhas e a sua localização. As mudanças e a adoção desses procedimentos podem ocasionar resistências internas, estendendo ainda mais o processo.

As principais normas e modelos adotados pelas empresas de desenvolvimento de software e que serão abordadas neste trabalho são:

- Normas da Família ISO 9000
- Normas da Família ISO/IEC 12207
- Modelo SEI CMM (Capability Maturity Model)
- Modelo SPICE (Software Process Improvement and Capability dEtermination)

No final deste capítulo, serão apresentadas algumas normas complementares e específicas, consideradas de grande importância para a área de software.

3.1. BENEFÍCIOS DA NORMATIZAÇÃO

Para muitas pessoas, criar normas, procedimentos e padrões a serem seguidos significa simplesmente burocratizar o processo interno da empresa e restringir a criatividade. Esta é uma visão distorcida da realidade, comumente resultante de um uso inadequado da normatização ou da falta de comunicação entre os vários grupos.

O sucesso da implantação de normas ou modelos depende de um bom planejamento, especialmente nos aspectos relativos às pessoas, que precisam ser envolvidas com os novos desafios e se sentirem comprometidas com as mudanças [ROCHA 01]. Para tanto, é imprescindível que sejam estabelecidos programas de treinamento, esclarecimentos e o envolvimento efetivo das pessoas nos novos procedimentos.

A normatização deve ser usada como um parâmetro, orientando as atividades, facilitando a rotina diária e garantindo um padrão uniforme no processo interno da empresa.

Um sistema de normatização, quando bem definido e aplicado, traz vários benefícios [WEBER 97]:

- Cria uniformidade para os processos, produtos e serviços, que passam a ser executados dentro de um conjunto de padrões previamente determinados;
- Facilita o treinamento e transferência de conhecimento. Existindo um guia prático de procedimentos é muito mais simples e rápido qualificar os profissionais, evitando critérios individuais;
- À medida que a empresa vai aprimorando seus procedimentos, vai incorporando a experiência e criando automaticamente os caminhos a serem seguidos por todos, evitando a repetição de erros e reduzindo os desperdícios;
- Facilita a comparação e aumenta a previsibilidade de resultados;

- Cria uma linguagem comum entre todos os funcionários e hierarquias dentro de uma empresa;
- Melhora o relacionamento entre clientes e fornecedores, garantindo principalmente a qualidade dos produtos e serviços.

Porém, antes de adotar ou estabelecer uma norma ou modelo, é importante verificar se ela atende a alguns requisitos básicos, garantindo assim que a normatização trará os benefícios citados anteriormente. Os principais requisitos de uma norma são [HOWER 99]:

- Precisa trazer alguma vantagem;
- Ser dinâmica e permitir melhorias contínuas;
- Poder ser registrada para que todos tenham acesso;
- Ser elaborada levando em consideração quem vai fazer uso dela;
- Ser viável e facilmente adotada;
- Possuir um conteúdo claro e de fácil entendimento
- Sua importância deve ser reconhecida por todos os envolvidos;
- Indicar *o que* deve ser feito, e não *como* deve ser feito.

Estabelecer normas que não trazem nenhum benefício ou que são inviáveis de serem seguidas significa fantasiar a realidade e burocratizar o processo, perdendo tempo e criando um clima desagradável na empresa, o que contraria qualquer objetivo de uma Gestão de Qualidade.

3.2. AS NORMAS DA ISO

A série de normas internacionais ISO 9000 e ISO 12207, elaboradas e editadas pela ISO (International Organization for Standardization) e adotadas no Brasil pela ABNT (Associação Brasileira de Normas Técnicas), especificam os procedimentos necessários para atender aos requisitos da qualidade e podem ser usadas como um guia pelas empresas para implantação de Sistemas da Qualidade.

As normas da ISO são modelos flexíveis, que podem ser incorporados e adaptados às necessidades específicas de cada organização. É importante ressaltar que as normas não especificam *como* deve ser implantado um Sistema da Qualidade, mas sim, indicam *o que* deve ser feito para alcançar os requisitos de qualidade.

3.2.1. ISO 9000

A série ISO 9000 existe desde 1987 e consiste em um modelo para gerenciamento e garantia da qualidade dentro da empresa. Em 1994, esta série foi revisada esclarecendo várias questões duvidosas em relação à versão inicial e facilitando o seu entendimento. Devido ao crescente uso da norma nos mais variados setores, incluindo o setor de desenvolvimento de software, a sua mais recente revisão foi em 2000, com mudanças bastante significativas.

A série é composta por 5 conjuntos de normas: ISO 9000, 9001, 9002, 9003 e 9004.

Na prática, somente as normas ISO 9001, 9002 e 9003 são utilizadas como requisitos entre clientes e fornecedores. Portanto, para fins de certificação do Sistema da Qualidade, a empresa pode escolher entre estas três. As demais normas ISO 9000 e 9004 são destinadas a orientar a empresa na escolha da norma ideal a ser utilizada e como deve ser a Gestão da Qualidade na empresa.

A ISO 9001 é a mais completa, abrangendo todo o ciclo de vida de um produto ou serviço, e pode ser utilizada por empresas que projetam, desenvolvem, produzem e instalam produtos ou prestam serviços de assistência técnica. Como é bastante abrangente, esta norma apresenta o maior número de requisitos: são 20 itens.

Devido ao crescente uso desta norma no setor de software, a ISO reconheceu que os processos de software são diferentes daqueles da maioria dos outros tipos de indústrias [WEBER 97]. Por isso, em 1991 foi publicada a ISO 9000-3 e revisada em 1997, que estabelece algumas diretrizes para a aplicação da ISO 9001 ao desenvolvimento, fornecimento e manutenção de software.

A ISO 9002 é menos rigorosa que a ISO 9001, abrangendo requisitos para a garantia da qualidade em produção, instalação e serviços associados. Esta norma pode ser aplicada para as empresas cuja atividade principal é a montagem de produtos, ou seja, não definem ou especificam o projeto. O número de requisitos é um pouco menor que a anterior: são 19 itens.

A ISO 9003, por sua vez, abrange somente requisitos para a garantia da qualidade em inspeção e ensaios finais. Esta norma se aplica a situações onde o fornecedor deve garantir a capacidade no que se refere a inspeções e ensaios finais. Um exemplo de aplicação desta norma é para as empresas distribuidoras de produtos. Esta norma possui apenas 16 requisitos.

Obter a certificação não é uma exigência da implementação das normas ISO 9000 dentro de uma empresa. Na verdade, o mais importante na adoção do modelo é a mudança de mentalidade e do processo, com a disseminação de uma cultura organizacional voltada para a qualidade. Mas com certeza, depois de investir tanto tempo e dinheiro ajustando os procedimentos, normalmente a empresa deseja obter o reconhecimento de todo seu esforço.

A certificação ISO 9000 é reconhecida praticamente em todos os países e por todos os setores, incluindo o setor de software. Para uma empresa, conquistar a certificação ISO 9000 significa alcançar um padrão internacional em seus processos [WEBER 97].

3.2.2. ISO/IEC 12207

Em 1995, a ISO (International Organization for Standardization) e a IEC (International Electrotechnical Commision) aprovaram a norma internacional ISO/IEC 12207 – Tecnologia da Informação, que define os processos do ciclo de vida de um software e que vem sendo usada como referência em muitos países, inclusive no Brasil.

O objetivo desta norma é auxiliar os envolvidos na produção de software a definir seus papéis, por meio de processos bem definidos, e assim proporcionar às organizações que a utilizam um melhor entendimento das atividades a serem executadas nas operações que envolvem, de alguma forma, o software [ROCHA 01].

A norma estabelece uma arquitetura de alto nível para o ciclo de vida do software, abrangendo desde a concepção até a descontinuidade do produto. Nesta arquitetura, as fases que envolvem o ciclo de vida do software são agrupadas nos seguintes processos, de acordo com sua natureza:

- **Processos Fundamentais:** correspondem aos processos iniciais de contratação entre o cliente e fornecedor, bem como os processos de desenvolvimento, operação e manutenção de produtos de software;

- **Processos de Apoio:** são os processos que contribuem para o sucesso e qualidade do produto e incluem os processos de documentação, gerência de configuração, garantia da qualidade, verificação, validação, revisão conjunta, auditoria e resolução de problemas;

- **Processos Organizacionais:** normalmente são processos aplicados fora do escopo dos projetos, mas que contribuem de forma considerável para a melhoria da organização. São eles: processos de gerenciamento, infra-estrutura, melhoria e treinamento;

- **Processo de Adaptação:** este processo define as atividades necessárias para adaptar a utilização da norma na organização e nos projetos.

A norma ISO/IEC 12207 é bastante flexível para as abordagens da engenharia de software, podendo ser aplicada para qualquer modelo de ciclo de vida, para qualquer método ou técnica de programação e para qualquer linguagem de programação. Estes critérios dependem do projeto e da tecnologia necessária, mas independem da aplicação da norma.

3.3. MODELO CMM

O CMM (Capability Maturity Model) desenvolvido pelo SEI (Software Engineering Institute), nos Estados Unidos, é um dos modelos mais antigos com relação à qualidade de processos em empresas de desenvolvimento de software. Mesmo assim, o CMM é um modelo que vem recebendo grande destaque na área de qualidade de software.

O modelo CMM foi concebido em 1987, com o intuito de estabelecer um padrão de qualidade dos produtos de software desenvolvidos especificamente para as Forças Armadas dos Estados Unidos. Pouco tempo depois, o modelo obteve aceitação mundial.

O CMM foi baseado nos conceitos de qualidade total estabelecidos por Philip Crosby, que demonstrou que a implantação da qualidade em empresas segue um amadurecimento gradativo em patamares [PAULK 96].

Este modelo permite que a organização saiba em qual nível de maturidade está seu processo de software e também que sejam realizadas determinadas práticas visando atingir níveis superiores e a busca de uma melhoria contínua [DYMOND 98].

3.3.1. NÍVEIS DE MATURIDADE DO CMM

No modelo CMM foram estabelecidos cinco níveis de maturidade. O objetivo básico das metas definidas em cada nível é a realização de um processo controlado e mensurado. Além disso, cada nível possui um conjunto de metas almejando um nível superior e a melhoria contínua do processo.

A Figura 3.1 mostra os níveis de maturidade do modelo CMM e o respectivo relacionamento entre eles [ROCHA 01].

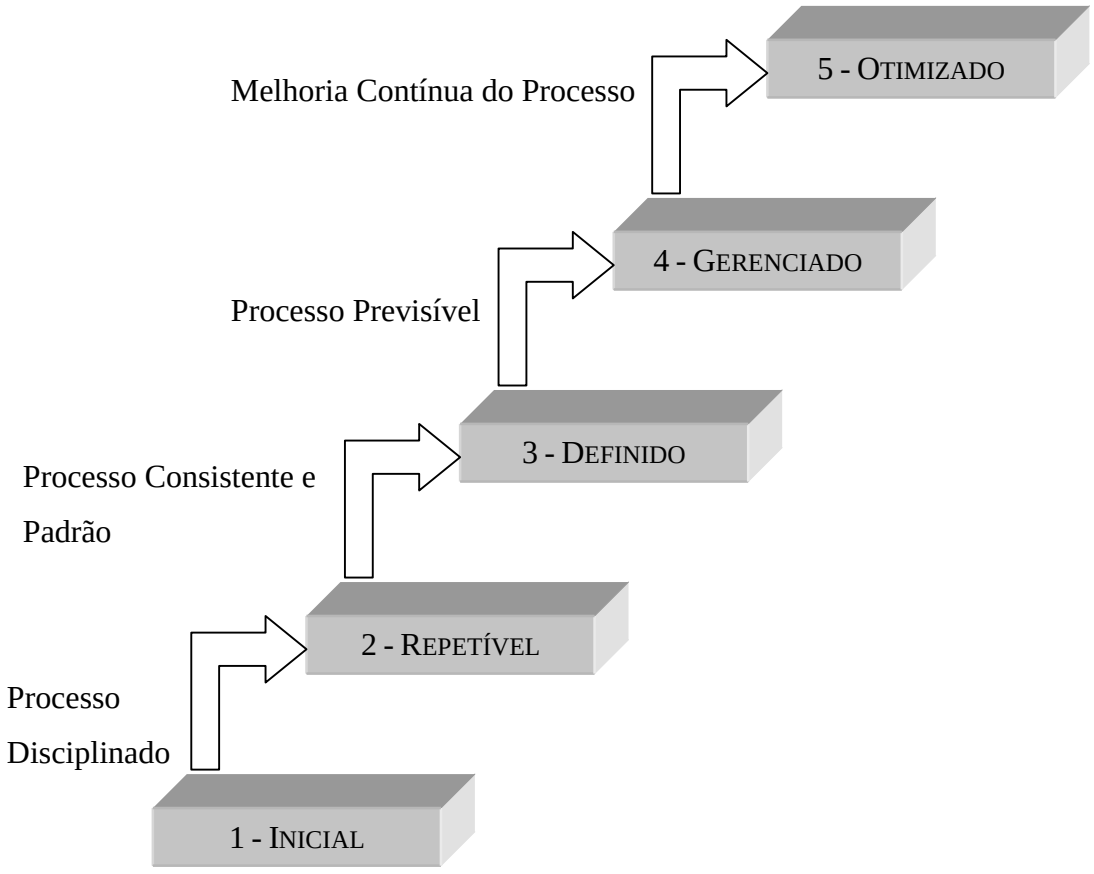


Figura 3.1 – Níveis de Maturidade do Modelo CMM

Cada nível de maturidade do CMM está dividido em áreas chave de processo ou KPA - Key Process Area, que estabelecem grandes temas a serem abordados e consistindo em um total de 18 áreas chave. Cada área chave, por sua vez, é detalhada nas práticas chave que são os requisitos a serem cumpridos na implantação do modelo, totalizando 316 práticas chave no modelo todo [ROCHA 01].

No **nível 1 – Inicial**, não existe área chave. Este nível é caracterizado pelas organizações que não apresentam um ambiente estável. Normalmente as empresas desenvolvem produtos de software de alta qualidade, mas seu desempenho depende totalmente da competência das pessoas. Mudando as pessoas, a qualidade pode cair. Os maiores problemas das organizações que se encontram neste nível são gerenciais, e não de ordem técnica.

O **nível 2 – Repetível** é o primeiro degrau a ser galgado pela empresa. Este nível se caracteriza pela existência de um sistema de gerenciamento de software, com documentação e acompanhamento, e que garante o cumprimento de custos e prazos dos projetos. As práticas bem sucedidas são repetidas em novos projetos que possuem características semelhantes. No nível 2 existem ao todo seis áreas chaves que devem ser levadas em consideração, que são:

- gerenciamento de requisitos;
- planejamento de projeto de software;
- acompanhamento de projeto de software;
- gerenciamento de contrato de software;
- garantia da qualidade de software;
- gerenciamento da configuração de software.

No **nível 3 - Definido**, a organização deve possuir um processo de desenvolvimento de software bem definido, compreendido, melhorado, documentado, e principalmente, praticado por todos. Existe uma preocupação com um processo padronizado para a empresa e customizado para cada projeto. Neste nível foram definidas sete áreas chaves, que são:

- enfoque no processo da organização;
- definição do processo da organização;
- programa de treinamento;
- gerenciamento integrado de software;
- engenharia de produto de software;
- coordenação intergrupos;
- revisões.

No **nível 4 - Gerenciado**, a empresa possui bases objetivas para a tomada de decisão, pois o processo é medido e gerenciado quantitativamente, sendo possível prever o desempenho dentro de limites quantificados. Deve existir uma base de dados histórica contendo informações do comportamento de projetos anteriores, e que vão servir de subsídios para previsões e estimativas de projetos novos. Neste nível existem duas áreas chave:

- gerenciamento quantitativo do processo;
- gerenciamento da qualidade de software.

No **nível 5 - Otimização**, o enfoque principal é a melhoria contínua do processo. Neste nível, a mudança de tecnologia e as mudanças no próprio processo são gerenciados de tal forma que não causam impacto na qualidade do produto final. No nível 5 existem três áreas chave, que são:

- prevenção de defeito;
- gerenciamento da mudança tecnológica;
- gerenciamento da mudança de processo.

Embora o CMM seja um tanto complexo, principalmente quando se trata de pequenas e médias empresas, muitas vezes é possível interpretar e aplicar o CMM a empresas de qualquer tamanho. O importante é interpretar o CMM de acordo com a

realidade da organização, definindo o processo padrão da forma mais simples possível e criando mecanismos para que esse processo evolua continuamente [PAULK 98].

Devido a sua complexidade, a implantação do CMM é uma tarefa a longo prazo. Atualmente, a maioria das organizações que adotam o modelo, encontram-se nos níveis inferiores do CMM, notadamente nos níveis 1, 2 e 3.

O CMM é um modelo bastante antigo e em conseqüência, resultou no desenvolvimento de uma série de variações. Este fator levou a SEI em direção a um processo de evolução do CMM, gerando o **CMMI – CMM Integrated**. Esta nova abordagem visa abranger a integração das variações de CMM existentes e a compatibilização com a norma ISO 15504. Além disso, o CMMI vem sendo caracterizado pelo grande aumento da flexibilidade na implantação de melhorias nos processos de software [BELLOQUIM 02].

3.3.2. AS MÉTRICAS NO CMM

É especialmente importante conhecer e compreender o nível atual de processos e produtos de software em uma empresa. Dessa forma, é possível estabelecer as metas de negócios da empresa e posteriormente avaliar se estas metas estão sendo cumpridas ou não. Esse objetivo pode ser alcançado em uma empresa através da modelagem e da mensuração de fatores que influenciam o processo de desenvolvimento de software [BURNETT 00]. Este assunto será apresentado em maiores detalhes no capítulo 4.

Especificamente com relação a métricas, o modelo CMM apresenta as seguintes características em cada nível de maturidade [DYMOND 98]:

- **No nível 1:** não existem métricas consistentes;

- **No nível 2:** existem métricas de projetos e alguma infra-estrutura básica para suportar rastreamento e retenção de dados de projeto;

- **No nível 3:** as métricas de produto são integradas aos projetos. Existe uma base de dados de produtos padronizada através dos projetos;

- **No nível 4:** o processo é medido e controlado e as métricas são coletadas e analisadas. Existe uma base de dados de processo;

- **No nível 5:** o foco é na melhoria de desenvolvimento de processos e produtos baseado em dados quantitativos de bases históricas. As métricas são coletadas, analisadas e utilizadas para a melhoria dos processos.

3.4. SPICE

Em janeiro de 1993, a ISO iniciou os trabalhos do projeto SPICE – Software Process Improvement and Capability dEtermination e a última versão do SPICE foi aprovada em 1998, como um relatório técnico da ISO.

O projeto teve sua origem decorrente da pressão existente em virtude do aparecimento de diversos modelos de qualidade para o desenvolvimento de software, dentre eles o CMM, Trillium e Bootstrap [WEBER 97].

O foco inicial do projeto SPICE era produzir um relatório técnico que fosse mais abrangente que os modelos existentes e ao mesmo tempo mais específico que as normas ISO 9001 e ISO 12207. Conseqüentemente, o projeto herdou a arquitetura dos processos do ciclo de vida do software da ISO/IEC 12207, enquanto que, do CMM herdou o conceito de níveis de maturidade de processos.

O projeto SPICE foi elaborado inicialmente como um relatório técnico composto por um conjunto de nove documentos, mas em 2002 foi revisado e substituído em sua totalidade pela norma internacional ISO/IEC 15504, especificamente para avaliação de processos de software.

Os objetivos da norma ISO/IEC 15504 prevê a realização de avaliações de processos de software considerando dois pontos principais: a melhoria dos processos e a determinação da capacidade de processos de uma organização.

Caso o foco da avaliação é a melhoria dos processos, a empresa deve gerar um perfil dos processos que será usado para a elaboração de um plano de melhorias. No segundo caso, é possível avaliar um fornecedor obtendo o seu perfil de capacidade. Este perfil permite ao contratante estimar o risco associado à contratação do fornecedor.

3.4.1. A ABORDAGEM DA NORMA

Diferentemente do modelo CMM, que associa áreas de processo aos níveis de maturidade, a norma ISO/IEC 15504 tem uma abordagem de duas dimensões, que associa níveis de maturidade a qualquer processo do ciclo de vida de software.

As duas dimensões estabelecidas no modelo de referência da norma são [ROCHA 01]:

3.4.1.1. DIMENSÃO DE PROCESSO

Esta dimensão corresponde à definição de um conjunto de processos considerados universais e fundamentais para a boa prática da engenharia de software.

3.4.1.2. DIMENSÃO DE CAPACIDADE

Corresponde à definição de um modelo de medição, através da identificação de um conjunto de atributos que determina o nível de capacidade de um processo para atingir suas metas.

A dimensão da capacidade estabelece uma graduação do processo baseado nos seguintes níveis de capacidade:

- **Nível 0 – Incompleto**: o processo não é implementado ou falha no cumprimento de seu propósito;

- **Nível 1 – Executado**: o propósito do processo geralmente é alcançado, mas sua execução não é rigorosamente planejada e acompanhada;

- **Nível 2 – Gerenciado**: o processo gera produtos de trabalho de acordo com procedimentos específicos e sua execução é planejada e controlada;

- **Nível 3 – Estabelecido**: o processo é definido por meio de princípios de engenharia de software e de um processo padrão da empresa;

- **Nível 4 – Previsível**: o processo definido é executado de forma consistente e dentro de limites previamente estabelecidos, atingindo as metas definidas;

- **Nível 5 – Em Otimização**: o desempenho do processo é continuamente melhorado para atingir as necessidades dos produtos.

A abordagem de duas dimensões, de processo e capacidade de processo, é bastante flexível. Em conseqüência, uma das principais vantagens da norma é a grande diversidade de formatos de apresentação dos resultados, normalmente com muito mais riqueza de detalhes que o modelo CMM, que fornece um único número para representar o nível de maturidade dos processos da organização [ROCHA 01].

3.5. ALGUMAS NORMAS COMPLEMENTARES

Além dos modelos e normas citados, que são utilizados normalmente para a gestão da qualidade nas empresas e melhoria do processo de desenvolvimento de um software, existem ainda outras normas que são complementares ou específicas para os produtos de software.

Dentre elas, a norma internacional **ISO/IEC 9126** que foi publicada em 1991. Esta norma representa a atual padronização mundial para a avaliação das características de qualidade dos produtos de software [INTHURN 01]. A versão brasileira desta norma foi editada pela ABNT em 1998 como sendo a norma NBR 13596.

Basicamente, esta norma relaciona um conjunto de características que devem analisadas em um software para que ele seja considerado um produto de qualidade. Esta norma abrange seis grandes grupos de características:

- Funcionalidade
- Confiabilidade
- Utilizabilidade
- Eficiência
- Manutenibilidade
- Portabilidade

Além disso, as organizações internacionais de normalização ISO/IEC tem disponibilizado outras normas relativas à qualidade dos produtos de software, como por exemplo:

- **ISO/IEC 14598:** usada como referência para avaliação da qualidade dos produtos de software;

- **ISO/IEC 12119:** responsável pela determinação de requisitos de qualidade e testes em pacotes de software.

3.6. CONCLUSÕES

À medida que a área de desenvolvimento de software vem se tornando mais competitiva e exigente, a melhoria da qualidade dos produtos de software tem sido não apenas um diferencial para as empresas, mas sim, um fator crítico para a sobrevivência das mesmas. Isto tem levado as empresas à procura de abordagens que auxiliem e garantam a melhoria do processo e dos produtos de software.

Neste capítulo, foram apresentados os principais modelos e normas que atendem a essa necessidade, como por exemplo, as normas ISO 9000, a ISO/IEC 12207, o modelo CMM e o SPICE. Apesar das semelhanças e diferenças existentes entre essas várias abordagens, uma evidência é conclusiva: todas elas sugerem que melhorando o processo de software, é possível automaticamente melhorar a qualidade dos produtos.

Além disso, a implantação de modelos ou normas de qualidade nas empresas traz como conseqüência natural a satisfação dos clientes e uma notória melhoria no relacionamento entre clientes e fornecedores.

Um outro aspecto conclusivo é que, independente de qual seja a abordagem adotada pela empresa, é importante considerar que a normatização sempre envolve aspectos de mudança cultural dentro da empresa. Essas mudanças nem sempre são fáceis de serem implantadas. Por isso, o sucesso da implantação de uma norma ou modelo de qualidade depende de um bom planejamento, especialmente nos aspectos relativos às pessoas, que precisam ser envolvidas de forma efetiva com os novos desafios e se sentirem comprometidas com as mudanças decorrentes da normatização.

4. MENSURAÇÃO DE SOFTWARE

A mensuração é considerada uma atividade fundamental em qualquer área de engenharia, e a engenharia de software não é uma exceção.

Na área de engenharia de software, a mensuração visa obter níveis de qualidade cada vez maiores. Nesta área, o termo métricas de software é relativo a um conjunto de mensurações tanto para projetos e produtos como para processos, visando a satisfação dos clientes ou usuários, a um custo economicamente viável.

Métricas de software é o nome usado para descrever uma gama de atividades relacionadas com mensuração, como por exemplo: estimativas de tamanho, previsão de custos e esforços, coleta de dados, medidas e modelos de qualidade, medidas de confiabilidade, avaliação de capacitação e maturidade, gerenciamento suportados por métricas e ferramentas e métodos de avaliação [FENTON 99].

Os programas de mensuração de software ajudam as organizações e os tomadores de decisões fornecendo informações significativas com relação à qualidade, adequação e progresso evolutivo de processos, produtos e projetos de software [BURNETT 00]. De uma forma geral, existem vários motivos para adoção de programas de mensuração: para compreender, avaliar, estimar, planejar, melhorar e controlar os projetos de software com rigor e precisão.

Tom de Marco é responsável pela famosa frase: “You cannot control what you can not measure”, ou seja, sem efetuar as medições necessárias, é impossível obter um gerenciamento eficaz.

Porém, para uma implantação eficaz de um processo de medição, é necessária a motivação de todos os profissionais que atuam no processo de desenvolvimento de sistemas [HAZAN 99].

4.1. A DIMENSÃO DO SOFTWARE

Antes de implantar um programa de mensuração é importante conhecer claramente a dimensão do software e as atividades que o compõem.

Todo desenvolvimento de software pode ser definido como sendo uma seqüência básica de atividades, conhecida formalmente na Engenharia de Software como Ciclo de Vida do Software, conforme ilustra a Figura 4.1.

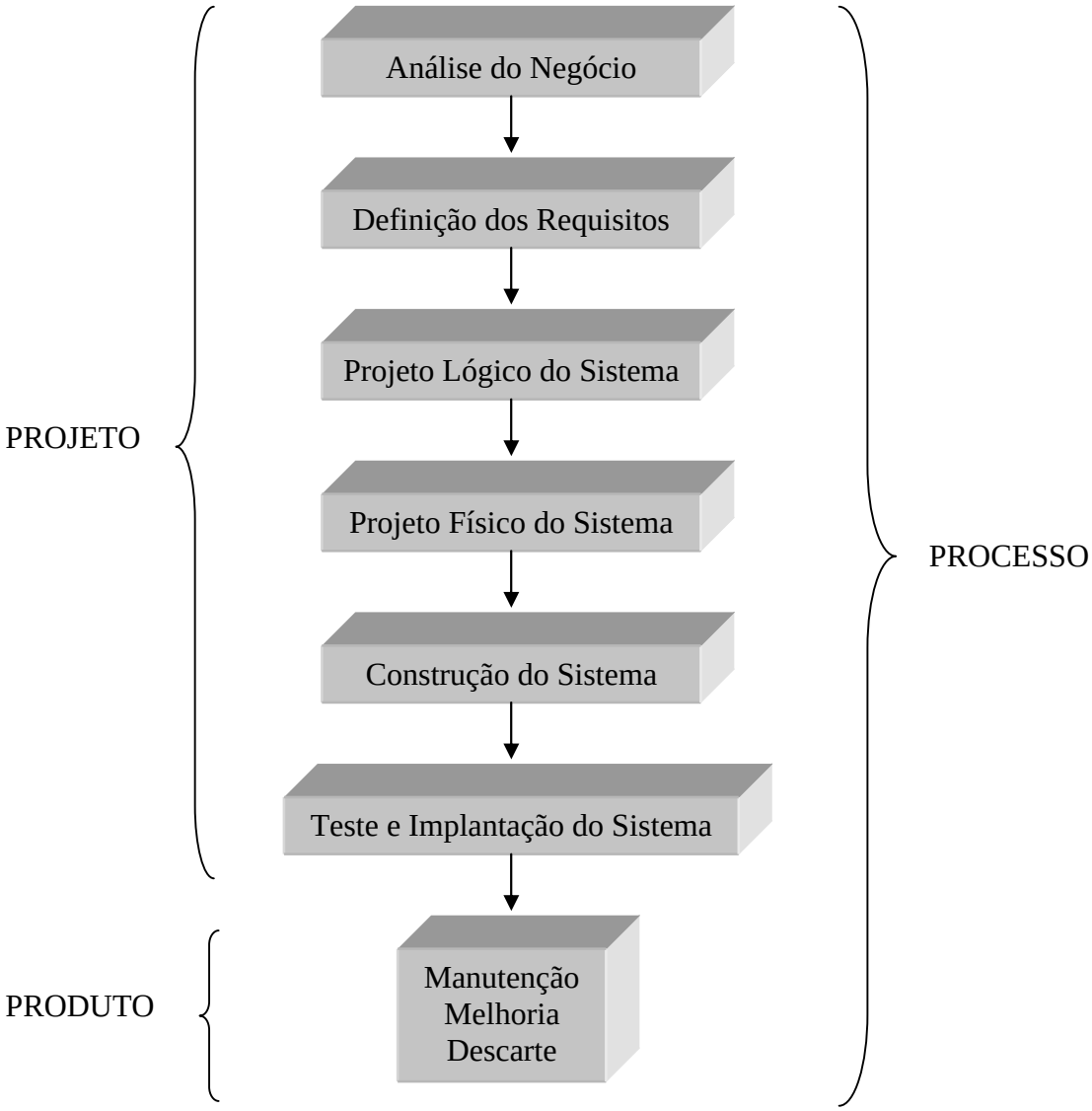


Figura 4.1 – Componentes Básicos e Ciclo de Vida do Software

O desenvolvimento de um software ocorre no contexto de um **projeto**, o qual é executado de acordo com um **processo**, gerando **produtos intermediários** e um **produto final**. Todo o processo é efetivado pelos **recursos** [FERNANDES 95].

Os componentes básicos do software (processos, recursos, projetos e produtos) e as etapas do ciclo de vida, permitem estabelecer diferentes domínios e estratégias para a mensuração de software. Ao mesmo tempo eles sevem como ponto de partida para uma classificação detalhada de métricas de software.

4.2. CLASSIFICAÇÃO DE MÉTRICAS

As métricas de software podem fornecer dados sobre diferentes aspectos do software. Neste item, serão apresentadas algumas definições úteis e em seguida uma classificação detalhada das principais métricas usadas na área de software.

4.2.1. MEDIDAS, MÉTRICAS E INDICADORES

Os especialistas vêm, até hoje, discutindo as diferenças entre medidas, métricas e indicadores, que são os termos mais freqüentes quando se fala em mensuração. Algumas definições foram atribuídas a cada um deles, visando facilitar a compreensão e tendo como intenção padronizar a nomenclatura [BURNETT 00]. São elas:

- **Medida**: atribuição de um número ou símbolo a uma entidade para caracterizar um determinado atributo (IEEE). Cinco metros de largura é um exemplo de medida. O número de pontos de função de um sistema é uma medida de tamanho de um software.

- **Métrica**: medida quantitativa do grau que um sistema, componente ou processo possui um dado atributo (IEEE). Um exemplo de métrica pode ser: 10 erros encontrados pelo usuário durante 1 ano de operação do sistema.

- **Indicador**: dispositivo ou variável que pode ser setado para um determinado estado baseado nos resultados ou na ocorrência de uma condição específica, como por exemplo, um flag ou semáforo (IEEE). O aumento significativo do número de defeitos encontrados na última versão de um sistema é um exemplo de indicador.

Alguns pontos relevantes com relação a estas definições são [RAGLAND 95]:

- Sem uma tendência a seguir ou um valor esperado para comparação, uma medida dá pouca ou nenhuma informação.
- Uma métrica provê uma informação mais significativa que simplesmente uma afirmação (medida).
- Um indicador genericamente compara uma métrica com uma baseline ou resultado esperado.

4.2.2. CLASSIFICAÇÃO DE MÉTRICAS

As métricas de software podem ser classificadas sob diferentes aspectos, dependendo do enfoque dado. A seleção das métricas mais apropriadas deve estar de acordo com as necessidades de cada área dentro da empresa.

[FERNANDES 95] classifica as métricas considerando o tipo de dado a ser coletado e os seus objetivos e nível de utilização, enquanto que [BURNETT 00] classifica de acordo com os componentes básicos de desenvolvimento de software.

4.2.2.1. CLASSIFICAÇÃO CONFORME O TIPO DE MEDIDA

Nesta classificação é considerado o tipo de dado coletado na medição, sendo:

- **Objetiva/Subjetiva:** as medidas que não possuem julgamento humano são consideradas objetivas;

- **Absoluta/Relativa:** as medidas absolutas não independentes das demais e não variam com a alteração de algum item. O tamanho do software é considerado uma medida absoluta, enquanto que a taxa de erros mensal é uma medida relativa;

- **Explícita/Derivada:** as medidas explícitas são obtidas diretamente enquanto que as derivadas são obtidas a partir de outras medidas;

- **Dinâmica/Estática:** as medidas dinâmicas têm a influência de um componente temporal enquanto que as estáticas não variam com o tempo;

- **Preditiva/Explanatória:** as medidas explanatórias são obtidas logo após a ocorrência de um evento enquanto que as preditivas são estimativas obtidas a partir da transformação de outras medidas;

4.2.2.2. CLASSIFICAÇÃO CONFORME O OBJETIVO DA MEDIÇÃO

Nesta classificação são considerados os objetivos da medição e a sua utilidade dentro da empresa, podendo ser: operacionais, táticas e estratégicas.

As **medições operacionais** são em nível de cada projeto ou produto de software, visando fornecer informações de controle para o planejamento do projeto e para a gestão do processo e do produto dentro da empresa. As métricas operacionais são aplicadas em cada etapa do ciclo de vida do software.

Alguns exemplos destes tipos de métricas são:

- Estimativa do Tamanho do Software
- Estimativa do Prazo do Projeto
- Estimativa do Esforço do Projeto;
- Custo Estimado do Projeto;
- Estimativa do Número de Instruções Fontes;
- Estimativa da Distribuição de Esforço e Prazo por Fase do Projeto;
- Estimativa do Número de Defeitos Pré-Release e Pós-Release;
- Estimativa do Esforço e Custo de Retrabalho;
- Estimativa do Número de Profissionais;
- Exatidão das Estimativas;
- Origem dos Defeitos;
- Taxa de Inspeção;
- Densidade Média de Defeitos;
- Eficiência na Remoção de Defeitos;
- Média de Reutilização de Código;
- Complexidade do Projeto de Software;
- Distribuição de Esforço por Fase;
- Produtividade Média de Desenvolvimento;
- Verificação da Exatidão das Estimativas;
- Crescimento Funcional do Software;
- Custos de Manutenção;
- Intensidade de Falhas;
- Índice de Manutenção no Software;
- Custo Médio do Ponto de Função;

As **medições táticas** auxiliam a gestão do ambiente de software, indicando o impacto da introdução de novas tecnologias. Este impacto pode ser avaliado tanto no processo de desenvolvimento quanto no pessoal envolvido. Normalmente, as métricas táticas são usadas para verificar tendências, analisar impactos e analisar atributos.

Os exemplos mais comuns destes tipos de métricas são:

- Tendência da Produtividade;
- Tendência da Densidade de Defeitos;
- Tendência da Exatidão das Estimativas;
- Tendência da Distribuição de Defeitos por Fase;
- Tendência da Distribuição de Esforço por Fase;
- Tendência do Custo do Ponto de Função;
- Tendência do Crescimento Funcional do Software;
- Tendência dos Defeitos Pré-Release e Pós-Release;
- Impacto na Introdução de Ferramentas CASE;
- Impacto na Introdução de nova Metodologia de Desenvolvimento e Gestão;
- Impacto na Introdução de nova Ferramenta de Desenvolvimento;
- Impacto da Introdução de Inspeção Formal de Software;
- Impacto no Aumento da Capacitação do Pessoal;
- Análise da Densidade de Defeitos;
- Análise de Custos por Processo;
- Análise de Produtividade conforme Método de Desenvolvimento;
- Análise de Produtividade entre Plataformas de Desenvolvimento.

As **medições estratégicas** são importantes para a gestão estratégica da empresa, visando à realização de benchmarking (comparação entre projetos ou empresas), melhorias contínuas e avaliações econômicas. As medições estratégicas normalmente são obtidas a partir da combinação das métricas operacionais e táticas.

Alguns exemplos dessas medições são:

- Benchmarking da produtividade de desenvolvimento;
- Retorno sobre o Investimento;
- Custos Médios de Desenvolvimento;
- Nível de Satisfação dos Usuários;
- Melhoria da Qualidade do Software;
- Melhoria da Produtividade de Desenvolvimento;
- Redução de Custos de Processos.

4.2.2.3. CLASSIFICAÇÃO CONFORME A DIMENSÃO DO SOFTWARE

A classificação que considera a dimensão do processo de software e os recursos necessários para sua construção, pode ser dividida nos seguintes grupos:

- **Métricas de Recursos:** pessoais, de software e de hardware;
- **Métricas de Processos:** maturidade, gerenciamento e ciclo de vida;
- **Métricas de Projeto e Produto:** tamanho, arquitetura, complexidade e qualidade.

As tabelas a seguir mostram uma classificação detalhada de métricas de software para os grupos mencionados [BURNETT 00].

MÉTRICAS DE RECURSOS		
Métricas Pessoais	**Métricas de Software**	**Métricas de Hardware**
Métricas de experiência de programação	Métricas de performance	Métricas de performance
Métricas de níveis de comunicação		Métricas de confiabilidade
Métricas de produtividade		Métricas de disponibilidade
Métricas de estrutura de equipe		

Tabela 4.1 – Métricas de Recursos

MÉTRICAS DE PROCESSO		
Métricas de Maturidade	**Métricas de Gerenciamento**	**Métricas de Ciclo de Vida**
Métricas organizacionais	Métricas de milestones	Métricas de análise especificação de requisitos
Métricas pessoais, de recursos e de treinamento	Métricas de risco	Métricas de design
Métricas de gerenciamento	Métricas de revisão	Métricas de implementação
Métricas de documentação	Métricas de produtividade	Métricas de manutenção
Métricas de análise e gerenciamento de dados	Métricas de custos de desenvolvimento	
Métricas de controle de processo	Métricas de tempo de desenvolvimento	

Tabela 4.2 – Métricas de Processo

MÉTRICAS DE PROJETO E PRODUTO			
Métricas de Tamanho	**Métricas de Arquitetura**	**Métricas de Complexidade**	**Métricas de Qualidade**
Pontos de Função, Linhas de Código, Número de Casos de Testes e outros	Número de componentes	Métricas de complexidade computacional	Métricas de Funcionalidade: adequação, exatidão, segurança de acesso, interoperabilidade, conformidade
	Número de linguagens no produto		Métricas de confiabilidade: maturidade, tolerância a falhas, recuperabilidade
			Métricas de usabilidade: inteligibilidade, apreensibilidade, operacionalidade
			Métricas de eficiência: comportamento do sistema através do tempo, utilização de recursos
			Métricas de manutenibilidade: analisabilidade, modificabilidade, estabilidade, testabilidade
			Métricas de portabilidade: adaptabilidade, capacidade para ser instalado, conformidade, capacidade de substituição

Tabela 4.3 – Métricas de Projeto e Produto

4.3. A IMPORTÂNCIA DAS MÉTRICAS DE SOFTWARE

As quatro razões básicas para medir recursos, processos, projetos e produtos relacionados ao software são: para compreender, avaliar, estimar e melhorar [BURNETT 00].

A primeira razão que justifica o uso de medições é para que se possa **compreender** e entender melhor os processos atuais dentro da empresa, permitindo estabelecer linhas-base quantitativas para comparações com avaliações futuras. Esta compreensão dos processos e práticas na organização é crucial para planejar, controlar e melhorar o desenvolvimento e a manutenção do software.

As **avaliações** são usadas para determinar a situação atual do processo com relação ao que foi previamente planejado, evidenciando possíveis desvios e possibilitando a correção em tempo hábil. Também se avalia para verificar se os objetivos de qualidade foram atingidos e para verificar os impactos de tecnologias e melhorias sobre os produtos e processos.

O conhecimento obtido com a análise e interpretação dos dados mensurados pode ser usado para planejar novos projetos, controlar o projeto corrente através da comparação dos valores estimados com os valores coletados e principalmente validar os modelos de mensuração adaptando-os sempre que se for necessário.

As **estimativas**, por sua vez, são realizadas para que se possa planejar. As estimativas ajudam a compreender os relacionamentos entre processos e produtos e devem ser melhoradas à medida que os valores observados apresentarem grandes variações. Isto é importante para se estabelecer metas tangíveis e próximos à realidade, para custo, cronograma e qualidade dos produtos. Projeções e estimativas baseadas em dados históricos também ajudam a analisar riscos e fazer relações de custo / benefício.

Finalmente, a última razão para realizar medições é com intuito de **melhorar**. As informações quantitativas são obtidas para auxiliar na identificação de barreiras e problemas, ineficiências, fraquezas e outras oportunidades para melhoria da qualidade do produto e da performance dos processos.

O propósito subjacente de qualquer programa de mensuração deve ser o alcance dos resultados específicos de coleta e interpretação dos dados mensurados. Sem estes objetivos, nenhum benefício será alcançado através do esforço da mensuração. Geralmente é aceito que a mensuração não é um fim em si, mas um meio para o fim [GRESSE 98].

Por outro lado, a inexistência de mensuração no desenvolvimento de software resulta em uma série de dificuldades [ZUSE 95]:

- as estimativas de tamanho de software, cronogramas, recursos, esforço e custo normalmente são realizados com base em julgamentos pessoais;
- não é possível introduzir melhorias quando não existe mensuração;
- a qualidade de planejamento dos projetos não pode ser avaliada;
- a qualidade dos produtos intermediários e do produto final não pode ser avaliada;
- a produtividade e a evolução da qualidade dentro da empresa não é mensurada;
- os fatores que influenciam na qualidade dos produtos não são determinados;
- as ocorrências de não conformidade do produto não são mensuradas;
- o nível de satisfação dos clientes não é identificado.
- não é possível identificar os riscos do projeto e rastrear problemas específicos;
- não é possível verificar os impactos dos problemas no custo, cronograma e performances do projeto;
- não é possível selecionar a melhor abordagem para corrigir os problemas;
- não é possível determinar soluções alternativas;

Um aspecto importante que deve ser observado na implantação de métricas é a sua utilidade dentro do contexto do projeto ou da organização. Além da definição e aplicação de uma métrica, é necessário determinar as diretrizes de utilização e a interpretação dos resultados obtidos.

Os principais fatores que devem ser considerados na escolha das métricas são [SIMOES 99]:

- detalhamento dos objetivos que precisam ser atingidos com a utilização da métrica;
- as métricas devem prover resultados consistentes;
- devem ser de fácil entendimento;
- precisam ser objetivas, minimizando a influência de julgamentos pessoais;
- as métricas devem ser efetivas no custo, ou seja, o resultado obtido deve compensar e exceder o esforço de utilização da métrica;
- precisam fornecer informações que evidenciem a situação atual e sejam relevantes para a tomada de decisões;
- as métricas devem ser apropriadas para cada etapa do ciclo de vida de um software;
- devem servir para a realização de estimativas,
- devem possibilitar a obtenção de séries históricas.

4.4. CONCLUSÕES

Conforme foi relatado neste capítulo, as medições têm provado serem uma ferramenta eficaz no gerenciamento de processos e produtos de software, fornecendo informações quantitativas que ajudam na identificação de áreas de risco, ineficiências, fraquezas e uma série de oportunidades de melhoria da qualidade e da performance dos processos em uma empresa.

Porém, para o sucesso da mensuração é necessário elaborar objetivos muito bem definidos, selecionar as métricas mais adequadas para a organização e que atendam esses objetivos, e sobretudo, manter o alinhamento das atividades de mensuração e análise dos resultados com os objetivos definidos.

Portanto, é possível concluir que em um programa de mensuração, a definição de quais as métricas consideradas importantes e quais os resultados esperados pela empresa, devem ser baseados nas necessidades reais de informação de cada nível da organização, caso contrário, os resultados obtidos e o esforço investido se tornam inúteis. Essas necessidades podem ser obtidas a partir de um levantamento de informações junto às áreas interessadas.

Apesar dos grandes benefícios que podem ser obtidos com a utilização de práticas de mensuração no gerenciamento dos processos de software, o fator humano é um aspecto extremamente sensível que deve ser levado em consideração e que, se negligenciado, pode conduzir ao fracasso as tentativas bem intencionadas de mensuração.

Nestes casos, é importante lembrar o enfoque central do programa de mensuração e das técnicas utilizadas: a melhoria da qualidade dos processos e produtos de uma empresa. Conseqüentemente, jamais se devem avaliar os desempenhos individuais, mas sim, enfocar os processos e produtos os quais são desenvolvidos por equipes.

5. MÉTODOS E TÉCNICAS DE MENSURAÇÃO DE SOFTWARE

Uma das maiores dificuldades encontradas no gerenciamento de projetos de software em uma empresa é conhecer e definir a dimensão do processo e do projeto que está sendo gerenciado, de uma forma quantitativa. Muitos projetos que a princípio parecem pequenos, quando em fase de desenvolvimento, tornam-se muitas vezes maior que o previsto inicialmente e, em alguns casos, tornam-se tão grandes e complexos, que se perde o controle [SIMOES 99].

Este problema pode ser minimizado em uma empresa através de um programa de modelagem e de mensuração explícita dos fatores que influenciam o processo de software.

Um ponto considerado crítico em um programa de mensuração é a elaboração de estimativas na fase de planejamento de um software, abrangendo principalmente as estimativas de prazo, esforço, custo do projeto e tamanho do software. Estes fatores e suas derivações são cruciais nesta etapa inicial, pois estimativas não realistas afetam seriamente a credibilidade do projeto perante o cliente e implicam em prejuízos para a empresa.

A utilização de técnicas ou modelos de estimativas é imprescindível, porém, é necessário calibrá-los ou mesmo customizá-los conforme a realidade da organização [FENTON 99]. Esta realidade constitui-se em uma série histórica baseada em observações acerca de prazos, esforço, custos, tamanho de software e outros atributos específicos de cada projeto.

As organizações envolvidas com a engenharia de software vêm, durante anos, lutando em busca de técnicas e métodos quantitativos aceitáveis e consistentes, para medir a eficiência e eficácia dos processos e produtos e para gerenciar os sistemas por elas adquiridos, desenvolvidos, melhorados ou mantidos.

Os principais métodos e técnicas para mensuração de software conhecidos e que serão abordados neste capítulo, são:

- Modelo GQM
- COCOMO
- Modelo de Estimativa de Putnam
- Contagem de Linhas de Código Fonte
- Sistema Métrico de Halstead
- FPA – Function Point Analysis
- Feature Point

5.1. MODELO GQM

A busca de modelos e técnicas específicas para a aplicação prática de métricas, sobretudo visando o gerenciamento, resultou em pelo menos dois trabalhos importantíssimos. O primeiro deles foi de Grady e Caswell, que em 1987 publicaram um livro relatando suas experiências na aplicação de um programa de métricas em uma grande organização. O outro trabalho de extrema relevância foi o de Basili, Rombach e acadêmicos de Maryland, nos Estados Unidos, que desenvolveram uma abordagem de definição de métricas que recebeu o nome de GQM (Goal/Question/Metrics).

O paradigma GQM é uma abordagem orientada a metas e utilizada para a mensuração de produtos e processos de software, suportando a definição top-down para o programa de mensuração e a análise e interpretação bottom-up dos dados de mensuração [GRESSE 98].

O requisito básico dessa abordagem é que a mensuração deve ser orientada a metas, ou seja, toda coleta de dados deve estar baseada em um fundamento lógico muito bem definido e documentado. Por este motivo, o modelo serve como base para a grande maioria dos métodos de mensuração desenvolvidos por especialistas da área.

Uma das principais vantagens do modelo GQM é a identificação das métricas consideradas relevantes e úteis dentro de uma empresa, uma vez que suporta a análise e interpretação dos dados coletados.

Além disso, permite uma melhor compreensão dos produtos e processos da empresa, a identificação de áreas consideradas de alto risco ou problemáticas, a determinação de linhas-base para os assuntos de interesse e a consciência dos pontos fortes e fracos na empresa, bem como a sua variação ao longo do tempo.

5.1.1. AS FASES DO MODELO GQM

O escopo básico para implantação do método GQM inclui o planejamento, a execução do programa de mensuração e a captura e interpretação das experiências obtidas durante a aplicação do método.

A seguir serão descritos detalhadamente os vários passos deste escopo [GRESSE 98]:

5.1.1.1. ESTUDO PRÉVIO

O objetivo na fase inicial é a coleta de informações que são pertinentes à introdução de um programa de mensuração.

Nesta etapa, é muito importante caracterizar a empresa e definir as suas metas de negócio, pois com essas informações, é realizado um estudo prévio para encontrar modelos de experiência relevantes e que estejam de acordo com as metas e características da empresa e dos projetos.

Em seguida, um projeto piloto deve ser selecionado para a aplicação do programa de mensuração e a definição de suas metas.

5.1.1.2. IDENTIFICAÇÃO DE METAS GQM

Após a definição das metas organizacionais e do projeto piloto, as metas a serem alcançadas pelo programa de mensuração são determinadas, como uma base para o desenvolvimento de um programa de mensuração efetivo.

Normalmente, as metas GQM são derivadas das metas de negócio, das metas estratégicas da empresa ou mais diretamente das metas de melhoria.

Para se identificar quais são as metas GQM mais importantes e que serão efetivamente aplicadas no programa de mensuração, elas são priorizadas de acordo com as necessidades da organização e do projeto piloto. É importante começar com poucas metas, para manter os custos baixos e mostrar a usabilidade da mensuração antes de ampliar o programa.

Para cada meta selecionada, é preciso especificar e dimensionar os seguintes itens:

- **Objeto de Estudo:** indicando o que será analisado, como por exemplo, o processo de desenvolvimento ou testes;

- **Objetivo:** indicando porque o objeto será analisado;

- **Enfoque de Qualidade:** indicando qual atributo do objeto será analisado, como por exemplo, confiabilidade, custos ou manutenção;

- **Ponto de Vista:** indicando quem irá utilizar os dados coletados, como por exemplo, o gerente, desenvolvedor ou usuário

- **Contexto:** indicando qual o ambiente em que a meta será aplicada, como por exemplo, projeto A ou departamento X.

5.1.1.3. DESENVOLVIMENTO DO PLANO GQM

Os planos GQM devem conter a informação necessária para planejamento da mensuração, bem como garantir posterior análise e interpretação dos dados coletados. Um plano GQM precisa ser desenvolvido para cada meta GQM selecionada na etapa anterior.

O plano GQM consiste de uma meta GQM e um conjunto de perguntas, modelos e medidas. O plano deve definir precisamente as perguntas e as medidas e como elas serão usadas no programa de mensuração. As perguntas são usadas como meio de captação de informações para atingir a meta selecionada. As medidas definem de forma operacional os dados a serem coletados em resposta às perguntas. O modelo, por sua vez, utiliza os dados de coletados como entrada para gerar respostas significativas às perguntas.

Uma forma de determinar as perguntas que são relevantes à meta é através da aplicação de entrevistas. O objetivo das entrevistas é adquirir o conhecimento implícito das pessoas envolvidas, descritas no item ponto de vista da meta. Este conhecimento é fundamental para a elaboração das perguntas, modelos e medidas consideradas úteis e relevantes à meta.

5.1.1.4. DESENVOLVIMENTO DO PLANO DE MENSURAÇÃO

O objetivo principal do desenvolvimento do plano de mensuração é efetivar a captura das medidas no contexto do projeto. Para cada medida identificada nos planos GQM deve ser determinado **quando**, **como** e **quem** executará a coleta dos dados no processo de software.

Nesta etapa, dois aspectos são cruciais e precisam ser considerados: a confiabilidade dos dados coletados deve ser o máximo possível, enquanto que a intrusividade durante a coleta dos dados deve ser mínima.

Para cada medida, o instante de tempo precisa ser determinado de forma coerente, podendo ser periodicamente, no começo e fim das fases ou durante fases de transição.

Após a definição do cronograma, é determinado quem pode ou deve coletar os dados. A determinação do responsável pela coleta de dados é muito importante, devendo ser considerado a sua experiência, nível de acesso aos dados, custo, disponibilidade, existência de algum preconceito e motivação.

Finalmente, para definir a forma de aquisição dos dados, deve-se buscar como primeira opção um instrumento que permita automatizar a coleta. Caso não exista, um certo nível de subjetividade nos dados não poderá ser evitado. Basicamente, existem três categorias de instrumentos de coleta de dados: ferramentas ou técnicas, questionários e entrevistas estruturadas.

5.1.1.5. COLETA DE DADOS

Durante a fase de desenvolvimento ou execução do programa GQM, os dados são coletados de acordo com o que foi especificado no plano GQM. Quando os dados foram coletados, eles precisam passar por um processo de garantia de qualidade antes de serem armazenados ou analisados. Este processo de garantia de qualidade dos dados deve evidenciar a:

- completude: verificar se todos os dados necessários foram fornecidos;
- plausibilidade: verificar se os dados se apresentam de forma coerente.

Caso sejam detectados dados defeituosos ou faltantes, estes devem ser corrigidos assim que possível. Os dados válidos são armazenados no banco de dados de mensuração ficando disponíveis para análise.

5.1.1.6. ANÁLISE DOS DADOS

O objetivo da análise é identificar padrões e relações entre atributos permitindo o estabelecimento de linhas-base e a identificação de áreas problemáticas e de risco.

A análise de dados deve ser orientada de acordo com o que foi definido no plano GQM e os dados atuais são comparados com as hipóteses de linhas-base especificadas no plano. Os resultados da análise de dados são uma entrada essencial para a etapa seguinte: a interpretação dos dados.

5.1.1.7. INTERPRETAÇÃO DOS DADOS

Os dados coletados e analisados são interpretados no contexto da meta de mensuração. Normalmente, a interpretação destes dados é realizada através de sessões de feedback, ou seja, reuniões contando com a presença de todos os envolvidos no programa de mensuração.

O objetivo dessas reuniões é interpretar os dados juntamente com o pessoal que possui a expertise necessária. Assim, os dados são apresentados aos participantes da reunião, possíveis interpretações são discutidas e modificações para melhoramento podem ser planejadas.

5.1.1.8. CAPTURA DE EXPERIÊNCIAS

A última etapa consiste em capturar explicitamente as experiências obtidas durante o programa de mensuração para reutilizar esse conhecimento em novos projetos de software.

Estas experiências são continuamente sintetizadas a partir de vários projetos, adquirindo-se uma compreensão geral dos processos e produtos na organização.

5.1.2. PRINCÍPIOS BÁSICOS DO GQM

Os seguintes princípios devem ser levados em consideração e executados para se obter sucesso e vantagens de um programa de mensuração baseado em GQM [BASILI 94]:

- A execução da tarefa de análise precisa ser especificada precisamente e explicitamente através de uma meta de mensuração;
- As medidas precisam ser derivadas de uma forma top-down baseada em metas e perguntas. Uma estrutura de metas e perguntas não pode ser adaptada de forma retroativa a partir de um conjunto de medidas existente;
- Cada medida precisa ter um fundamento lógico relacionado e devidamente documentado. O fundamento lógico é usado para justificar a coleta de dados e guiar a análise e interpretação posterior;
- Os dados coletados precisam ser interpretados de uma forma bottom-up no contexto das metas GQM e das perguntas;
- As pessoas que vão utilizar os resultados do programa de mensuração precisam estar envolvidas profundamente na definição e interpretação do programa.

5.2. COCOMO

O método COCOMO (Constructive Cost Model) foi desenvolvido por Barry Boehm para estimar esforços de desenvolvimento, prazo, custo e tamanho da equipe para um projeto de software [SIMOES 99].

O COCOMO foi derivado a partir da análise de informações provenientes de 63 projetos diferentes, compreendendo as áreas de negócio, controle, científica, suporte e sistema operacional [FERNANDES 95].

O método é baseado no número de instruções fonte (número de linhas de código) e fundamentado em fórmula matemática, podendo ser aplicado nas diversas fases do ciclo de desenvolvimento. Em contrapartida, o método apresenta algumas desvantagens, pois depende da tecnologia aplicada, depende de experiências anteriores e não produz indicadores.

5.2.1. MODELOS COCOMO

O método COCOMO está subdividido nos seguintes modelos:

- COCOMO Básico
- COCOMO Intermediário
- COCOMO Detalhado

5.2.1.1. COCOMO BÁSICO

O COCOMO Básico é um modelo que pode ser aplicado à grande maioria dos projetos de software de pequeno até médio porte, em termos de tamanho do software.

As estimativas fornecidas pelo COCOMO Básico são limitadas, pois não consideram alguns fatores como restrições de hardware, qualificação e experiência do pessoal envolvido, uso de técnicas e ferramentas modernas e outros atributos pertinentes ao projeto.

5.2.1.2. COCOMO INTERMEDIÁRIO

O modelo COCOMO Intermediário é uma versão melhorada do modelo anterior, abrangendo as limitações do modelo anterior e proporcionando estimativas mais acuradas.

5.2.1.3. COCOMO DETALHADO

O COCOMO Detalhado é uma técnica que permite realizar estimativas em nível de módulo, sub-sistema e sistema, inclusive evidenciando os atributos de custo em cada fase do projeto.

5.2.2. AS FASES DO COCOMO

A seguir serão apresentados os passos para a realização de estimativas de prazo, esforço, custo e tamanho da equipe usando o método COCOMO [FERNANDES 95]:

5.2.2.1. DETERMINAR O TIPO DE PROJETO

Nesta etapa inicial deve ser realizada uma caracterização do projeto. O COCOMO categoriza os projetos de software em três tipos fundamentais:

- **Modo Orgânico ou Convencional:** este tipo de projeto se caracteriza pela existência de equipes relativamente pequenas de desenvolvimento, cujas pessoas envolvidas possuem alguma experiência prévia com sistemas similares; um ambiente estável e familiar; e projetos de tamanho normalmente pequeno e algoritmos simples, tornando-os flexíveis para introdução de novas funcionalidades e requisitos.

- **Modo Difuso:** este modo representa um estágio intermediário entre os modos orgânico e restrito, apresentando uma mistura das características desses modos. O tamanho dos projetos normalmente são de médio porte.

- **Modo Restrito:** este modo caracteriza os projetos que apresentam alta complexidade de hardware, software, regras de negócio e procedimentos operacionais. Normalmente são projetos que requerem altos custos para verificação e validação, bem como, são projetos restritos a implementação ou alteração de novas funcionalidades e requisitos. Alguns exemplos deste tipo são os sistemas de transferência eletrônica de fundos e os sistemas de controle de tráfego aéreo.

5.2.2.2. DETERMINAR O NÚMERO ESTIMADO DE INSTRUÇÕES FONTES

Esta é uma das limitações do método, pois não é possível determinar quantas instruções fontes serão produzidas. Vários autores sugerem uma alternativa bastante viável para este problema: a utilização combinada do método FPA com o COCOMO.

A partir da aplicação da técnica FPA é possível determinar o número de pontos de função do projeto, e em seguida determinar um número de instruções fontes estimado, aplicando uma tabela de transformação. Várias pesquisas atualizam essa tabela de transformação, que apresenta uma correlação média entre as linguagens de programação e o número de instruções fontes geradas para desenvolver um ponto de função.

5.2.2.3. DETERMINAR A ESTIMATIVA DE ESFORÇO

Para determinar os valores estimados de esforço, cuja unidade de medida é H/M (Homem/Mês), a técnica propicia as equações específicas de acordo com o modo do projeto previamente determinado e o modelo COCOMO utilizado. Estas equações levam em consideração o número estimado de instruções fontes calculado na etapa anterior.

No caso do modelo Intermediário, as equações consideram também os fatores de custo aplicáveis ao projeto. Estes fatores correspondem a determinados atributos do projeto, do produto, da equipe e computacionais. Cada um dos fatores possui um peso específico, de acordo com a tabela desenvolvida por Boehm, que são os multiplicadores e que ajustam para mais ou para menos as estimativas iniciais de esforço.

5.2.2.4. DETERMINAR A ESTIMATIVA DE PRAZO

Da mesma forma como na etapa anterior, para determinar as estimativas de prazo, a técnica possui equações específicas de acordo com o modo do projeto previamente determinado e o modelo COCOMO utilizado. Neste caso, as equações levam em consideração a estimativa de esforço calculado na etapa anterior.

5.2.2.5. DETERMINAR A ESTIMATIVA DE QUANTIDADE DE PESSOAL

Para determinar o número médio de profissionais necessários para o projeto, basta dividir o esforço estimado pelo prazo estimado.

5.2.2.6. DETERMINAR A ESTIMATIVA DE CUSTO

O custo estimado do projeto é a multiplicação do esforço previsto pelo custo de um homem/mês ou homem/hora.

5.3. MODELO DE ESTIMATIVA DE PUTNAM

O modelo de estimativa Putnam é um modelo dinâmico e que permite equacionar matematicamente a distribuição de esforço necessário para o desenvolvimento de um projeto. O modelo possui múltiplas variáveis e pressupõe uma distribuição de esforço específica para cada fase do ciclo de vida de um software.

Da mesma forma como o COCOMO, o modelo de estimativa de Putnam se baseia em estimativas do número de linhas de código. A equação de software que relaciona linhas de código com o tempo e esforço de desenvolvimento foi derivada a partir da curva de distribuição matemática definida por Rayleigh-Norden.

O modelo foi elaborado inicialmente a partir de informações de distribuição de esforço oriundas de grandes projetos, porém, a sua aplicação em projetos menores é totalmente viável.

5.4. CONTAGEM DE LINHAS DE CÓDIGO FONTE

A partir da segunda metade da década de 60, a técnica de contagem de linhas de código era usada como base para medir qualidade, produtividade e esforço de programação. As linhas de código eram utilizadas tanto para medir a qualidade do programa, normalmente avaliando o número de defeitos por mil linhas de código, quanto a produtividade do programador, utilizando números de linhas por mês.

Além disso, a técnica também é a mais antiga forma existente para medir um fator que é considerado crítico em um programa de mensuração: o tamanho de um sistema.

A técnica tem como vantagem a simplicidade, porém está totalmente ligada à linguagem de programação utilizada, tendo praticamente nenhum significado para o usuário final. Além disso, ela só pode ser aplicada após a codificação dos programas, sendo inviável para estimativas de projetos.

5.5. SISTEMA MÉTRICO DE HALSTEAD

O Sistema Métrico de Halstead é uma técnica que permite a mensuração do tamanho de um software, e foi desenvolvida em 1977 por Maurice Halstead, da Universidade de Purdue.

Esta técnica consiste em calcular o tamanho do programa e o esforço de programação baseando-se no número de operadores (comandos e palavras reservadas) e operandos (itens de dados) utilizados.

A técnica é independente da linguagem de programação, mas está baseada na sintaxe dos programas. Uma desvantagem é que seu processo é incompreensível ao usuário final e só pode ser aplicado após a codificação, portanto, não permitindo a realização de estimativas em projetos.

5.6. ANÁLISE DE PONTOS DE FUNÇÃO

Como já foi citado anteriormente, um dos maiores obstáculos encontrados no gerenciamento de processos de software é a determinação do tamanho do software.

Apesar de ser considerado um fator crítico em um programa de mensuração, vários métodos de medição de software foram propostos no passado, mas a grande maioria destes métodos sofre de limitações, pois não podem:

- Ser aplicados no início do processo de desenvolvimento de um software;
- Ser aplicados uniformemente ao longo da vida do software;
- Ser facilmente interpretados nos termos do negócio;
- Ser compreendidos pelos usuários do software;
- Fornecer estimativas realistas para os usuários e para a empresa;
- Prover resultados consistentes;
- Possibilitar a obtenção de séries históricas.

A técnica FPA – Análise de Pontos de Função vem suprir a grande maioria dessas limitações. Desenvolvida por Albrecht em 1974 e evoluída por Carpers Jones juntamente com o IFPUG, a técnica FPA mede o tamanho do software através da quantificação das funcionalidades de um projeto, baseando-se no projeto lógico ou no modelo preliminar de dados.

As principais vantagens da técnica de Análise por Pontos de Função são [FERNANDES 95]:

- Medir exatamente o que o cliente requisitou e recebeu;
- Medir o tamanho do software independente da tecnologia utilizada para sua implementação;
- Propiciar um fator de normalização para comparação entre software;
- Ser geradora de outros indicadores e estimativas quando usada em conjunto com o tempo, como por exemplo, análises de qualidade, estimativas de prazos, produtividade e custos;
- Melhoria da qualidade dos processos e produtos de software;
- Facilidade de aprendizagem e aplicação da técnica;
- Aplicabilidade nas diversas fases do ciclo de vida de um software.

Devido às crescentes exigências dos requisitos de qualidade no mercado de software, as organizações estão começando a perceber as aplicações e o potencial das medidas funcionais de tamanho para gerenciamento e melhoria dos processos e produtos de software.

A técnica FPA é o tema central deste trabalho, portanto, será explorada detalhadamente nos capítulos consecutivos.

5.7. FEATURE POINTS

Em 1986, a Software Productivity Research desenvolveu um método experimental para aplicar a lógica dos Pontos de Função em um software básico, como por exemplo, sistemas operacionais e sistemas de roteamento telefônico. Para evitar confusão com a Análise de Pontos de Função, esta variação experimental foi denominada de Feature Points.

A diferença básica entre a técnica do Feature Point e a Análise de Pontos de Função é a inclusão de um sexto componente na ponderação do tamanho do software: os algoritmos do sistema

Como os resultados iniciais da utilização dessa técnica foram positivos, o método vem sendo aplicado em caráter experimental em vários projetos de software básico, sistemas em tempo real, CAD e inteligência artificial.

5.8. OUTRAS TÉCNICAS

Para mensuração do tamanho funcional de um software, nos últimos anos foram desenvolvidas algumas variações da técnica FPA original [SYMONS 01]. São elas:

- **MK II Function Point Analysis**, introduzida em 1984 por Charles Symons visando uma melhoria da técnica original. Este método vem sendo divulgado pela UKSMA (United Kingdom Software Metrics Association) através do MK II Function Point Analysis Counting Practices Manual;

- **3DFP – Boeing 3D Function Points**, introduzida em 1992 por Whitmire da Boeing como uma extensão da FPA para sistemas em tempo real;

– **COSMIC FFP – COSMIC Full Function Points**, vem sendo desenvolvida desde 1998 por um grupo de especialistas em métricas de software e corresponde à uma evolução da FPA para sistemas em tempo real.

5.9. QUADRO COMPARATIVO DAS TÉCNICAS

A Tabela 5.1 apresenta um resumo das principais características das técnicas cujo objetivo final é o dimensionamento do tamanho do software [AZEVEDO 00].

CARACTERÍSTICAS	LINHAS DE CÓDIGO	SISTEMA HALSTEAD	FPA	FEATURE POINT
1. Independência de tecnologia	Não	Sim	Sim	Sim
2. Produção de resultados consistentes	Sim	Sim	Sim	Sim
3. Avaliação por usuários sem conhecimento de Tecnologias de software e desenvolvimento	Não	Não	Sim	Sim
4. Significância para o usuário final	Não	Não	Sim	Sim
5. Utilização em estimativas	Não	Não	Sim	Sim
6. Contabilização automática	Sim	Sim	Sim	Sim

Tabela 5.1 – Quadro Comparativo de Técnicas de Mensuração de Tamanho

5.10. CONCLUSÕES

Com a implantação de uma nova mentalidade nas organizações voltada para a qualidade e melhoria do processo e a disponibilidade de métodos formais, não será mais admissível que o planejamento e desenvolvimento de produtos de software sejam baseados em “feeling”, principalmente no que diz respeito às variáveis de mensuração.

Apesar da grande dificuldade em definir técnicas e métodos de mensuração que sejam aceitáveis e consistentes, muitas pesquisas vêm sendo realizadas nesse sentido. Neste capítulo, foram apresentados alguns métodos e técnicas de mensuração que auxiliam na implantação de um programa de mensuração.

Como foi possível observar, a grande maioria das técnicas foram desenvolvidas a partir de análises e pesquisas efetuadas em um conjunto amplo de projetos e dados históricos, visando identificar as características particulares e comuns em cada conjunto.

Fazendo uma correlação entre as técnicas mais utilizadas atualmente no mercado de informática, o GQM demonstrou ser bastante útil em um programa de mensuração para a identificação das métricas relevantes em uma empresa, uma vez que suporta a análise e interpretação dos dados coletados.

O COCOMO e o Modelo de Estimativa de Putnam, substancialmente fundamentados em fórmulas matemáticas, contribuem para a determinação de estimativas de esforços de desenvolvimento.

Para o dimensionamento do tamanho de um software, as técnicas de contagem de linhas de código (LOC) e análise de pontos de função (FPA) são as mais conhecidas. Com os avanços tecnológicos e das linguagens de programação, a técnica LOC caiu em desuso.

Por outro lado, a técnica FPA, baseada nos requisitos funcionais do software e independente da tecnologia, vem garantindo o seu espaço no mercado. Além disso, várias pesquisas vêm sendo realizadas no sentido de customizar a técnica FPA para áreas específicas de software, como por exemplo, sistemas em tempo real, CAD, inteligência artificial e software básico.

Cada tipo de técnica de medição apresenta suas vantagens e desvantagens. O ideal é que a empresa busque aquela que mais de adapte com as necessidades do momento e objetivos da organização.

6. TÉCNICA FPA - ANÁLISE DE PONTOS DE FUNÇÃO

O cenário das organizações de informática vem mudando radicalmente nos últimos anos. O mercado está mais exigente, a competição mais acirrada e como forma de sobrevivência as organizações estabelecem objetivos de fazer mais, com melhor qualidade e com menos custos.

Muitos desses objetivos podem ser alcançados através da aplicação apropriada de um conjunto coordenado de métricas, algumas das quais podem estar baseadas em medidas do tamanho dos produtos.

A técnica de Análise de Pontos de Função - FPA é uma abordagem para a mensuração de software e que vem sendo aplicada nas organizações para auxiliar no dimensionamento do tamanho do software [RULE 01].

A técnica FPA mede o tamanho de um software levando em consideração as suas funcionalidades, ou seja, é baseada em uma avaliação padronizada e normalizada dos requisitos funcionais especificados pelos usuários.

6.1. HISTÓRICO

As pesquisas dessa técnica iniciaram na década de 70, quando os pesquisadores do Serviço de Processamento de Dados da IBM - International Business Machine começaram a analisar centenas de programas com o intuito de identificar os fatores e as variáveis críticas que determinavam a produtividade da programação. Descobriram que poderiam basear a avaliação de um sistema, medindo o valor das funções executadas pelos programas, ao invés de utilizar como base o volume ou a complexidade do código dos programas [AZEVEDO 00].

Em 1979, dando continuidade a essas pesquisas, o pesquisador da IBM Allan J. Albrecht introduziu a técnica de avaliação conhecida como FPA – Function Point Analysis. Porém, somente em 1984 os conceitos foram refinados em uma metodologia formal e disponibilizados para domínio público.

Em 1986, foi fundado o IFPUG – International Function Point User's Group, que juntamente com Capers Jones evoluíram a técnica inicialmente desenvolvida por Albrecht. Atualmente, o IFPUG possui como objetivo divulgar informações e pesquisar novas implementações e melhorias da técnica.

Segundo o IFPUG, a abordagem desenvolvida por Albrecht é conhecida como a Metodologia 1 ou **FPA Detalhada**, e está fortemente calcada na medição de funções do software a partir dos requisitos funcionais e do Projeto Lógico.

A Metodologia 2 ou **FPA Estimativa** foi desenvolvida pelo Software Productivity Research em parceria com o próprio IFPUG e considera a medição a partir dos requisitos funcionais e de um modelo de dados preliminar, antes mesmo do início do Projeto Lógico.

6.2. PRINCIPAIS OBJETIVOS

A técnica FPA tem como objetivo básico dimensionar o software quantificando a funcionalidade que ele proporciona aos usuários. Portanto, é uma técnica que considera a visão externa do usuário, independente da linguagem de programação, ferramentas ou tecnologia utilizada para o desenvolvimento do software.

Fazendo uma analogia, esta situação é semelhante à medição do tamanho de uma casa, definida através do número de metros quadrados contidos em uma planta baixa. O tamanho da casa em metros não muda, independentemente de como a casa venha a ser construída [DEKKERS 98].

A contagem de Pontos de Função tem como objetivos [LONGSTREET 02]:

- Medir o que o cliente requisitou e recebeu;
- Medir independentemente da tecnologia utilizada para implementação;
- Propiciar um veículo para realização de estimativas de software;
- Propiciar um fator de normalização para comparação entre software;
- Facilidade de aprendizagem e aplicação da técnica;
- Aplicabilidade nas diversas fases do ciclo de vida de um software.

Para atingir os objetivos descritos, o processo de contagem de Pontos de Função deve ser simples, para minimizar o trabalho adicional do processo de mensuração; e conciso, para permitir consistência ao longo do desenvolvimento dos projetos e entre a utilização da técnica em diferentes projetos [AZEVEDO 00].

6.3. TERMINOLOGIA

Para facilitar o entendimento dos tipos de funções e o cálculo de pontos de função, alguns conceitos básicos precisam ser diferenciados, pois existem termos que são utilizados tanto na metodologia de pontos de função quanto na tecnologia da informação. A confusão surge porque a definição desses termos na tecnologia de informação freqüentemente possui um significado diferente daquele utilizado na técnica FPA.

[DEKKERS 98a] relaciona alguns termos e apresenta a seguinte diferenciação:

- **Lógico:** em TI, normalmente representa tanto considerações conceituais quanto de projeto. Por exemplo, o modelo lógico de dados;
  em FPA, refere-se a funções lógicas e requisitos conceituais dos usuários, a partir de uma perspectiva do negócio. Não inclui considerações de projeto ou qualidade. Reflete o que o software deve fazer e não como.

- **Usuário:** em TI, corresponde à pessoa que utiliza o software;
  em FPA, é a pessoa, coisa, outro aplicativo, departamento, etc. que provê requisitos funcionais do usuário para o software.

- **Aplicativo (sistema):** em TI, é a implementação física do software. A fronteira do aplicativo ou sistema freqüentemente coincide com a do hardware ou software;
  em FPA, é uma coleção coesa de procedimentos e dados automatizados, dando suporte a um objetivo do negócio.

- **Projeto:** em TI, dependendo da organização pode significar novos desenvolvimentos, alterações ou melhorias em diversos aplicativos;
  em FPA, diz respeito ao trabalho efetuado sobre um único aplicativo. Se for de desenvolvimento, corresponde à especificação, construção, teste e entrega de um novo sistema. Se for de melhoria, corresponde a modificação de um aplicativo existente.

- **Arquivo:** em TI, representa um Dataset ou conjunto físico de dados, como por exemplo, um arquivo de entrada ou arquivo de dados;
  em FPA, é um grupo de dados logicamente relacionados e não a implementação física desse grupo de dados.

- **Tamanho Funcional:** em FPA, é uma medida de tamanho do software considerando as funcionalidades do software. O tamanho funcional é definido através de uma unidade de medida específica: os pontos de função.

6.4. TIPOS DE FUNÇÕES

Uma aplicação vista sob a ótica do usuário, é um conjunto de funções ou atividades do negócio que o beneficiam na realização de suas tarefas. Essas funções podem ser classificadas em Funções de Dados ou Funções Transacionais.

As Funções de Dados representam a funcionalidade proporcionada ao usuário para atender seus requisitos de dados internos e externos. As Funções Transacionais, por sua vez, representam a funcionalidade proporcionada aos usuários para processar os dados da aplicação [LONGSTREET 02].

Estas funções estão subdivididas em vários componentes, conforme mostra a Figura 6.1.

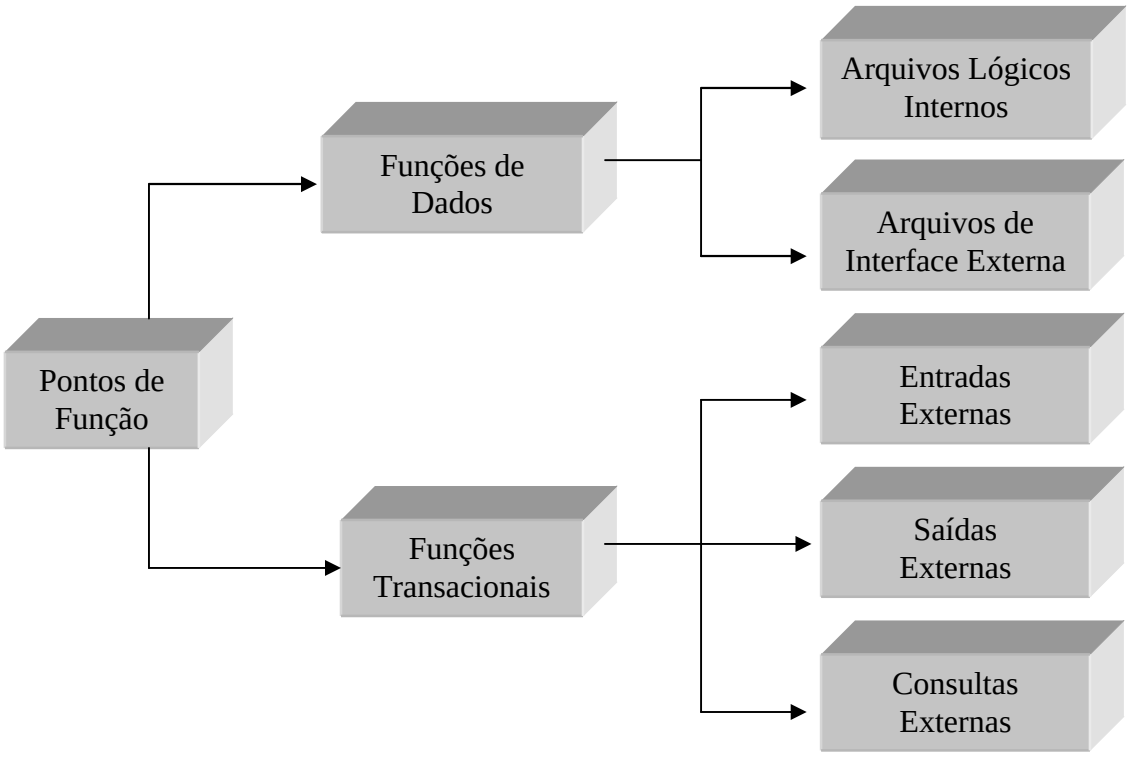


Figura 6.1 – Componentes de FPA

Basicamente, para o cálculo dos Pontos de Função é necessário identificar e enumerar as funções da aplicação, ou seja, determinar o número de Arquivos Lógicos Internos, Arquivos de Interface Externa, Entradas Externas, Saídas Externas e Consultas Externas.

As regras de contagem e identificação de cada um desses componentes estão definidas no Manual de Práticas de Contagem da Análise de Pontos de Função versão 4.1.1, publicada pelo IFPUG, e descritas em detalhes nos itens a seguir.

6.4.1. ARQUIVO LÓGICO INTERNO

Os Arquivos Lógicos Internos representam os requisitos de armazenamento de dados cuja manutenção é realizada pela própria aplicação. Esses arquivos influenciam no cálculo de FPA de acordo com sua quantidade e complexidade funcional de cada um dos arquivos.

Um Arquivo Lógico Interno é definido com sendo um grupo de dados logicamente relacionados ou um conjunto de informações de controle especificadas pelos usuários e mantidos dentro das fronteiras da aplicação. O grupo de dados refere-se a dados relacionados logicamente em um nível no qual o usuário consegue perceber como sendo importante para que a aplicação realize uma atividade específica. As informações de controle são dados utilizados pela aplicação para garantir total conformidade com os requisitos das funções especificadas pelos usuários.

Os Arquivos Lógicos Internos podem ser exemplificados por:

- Dados da aplicação;
- Arquivos de dados de segurança da aplicação;
- Arquivos de dados de auditoria;
- Arquivos de mensagens de auxílio;
- Arquivos de mensagens de erro;
- Arquivos de backup, quando forem solicitados explicitamente nos requerimentos do cliente;
- Um arquivo que sofre manutenção pela aplicação em questão e também por outras aplicações deve ser contabilizado como um Arquivo Lógico Interno em todas as aplicações.

6.4.2. ARQUIVO DE INTERFACE EXTERNA

Os Arquivos de Interface Externa representam as necessidades de dados externos à aplicação. Esses arquivos influenciam no cálculo de FPA de acordo com sua quantidade e complexidade funcional de cada um dos arquivos.

Um Arquivo de Interface Externa pode ser definido com sendo um grupo de dados logicamente relacionados ou um conjunto de informações de controle especificadas pelos usuários e que são utilizados pela aplicação, porém, sua manutenção é realizada por outra aplicação externa. O grupo de dados refere-se a dados relacionados logicamente em um nível no qual o usuário consegue perceber como sendo importante para que a aplicação realize uma atividade específica. As informações de controle são dados utilizados pela aplicação para garantir total conformidade com os requisitos das funções especificadas pelos usuários.

Alguns exemplos de Arquivos de Interface Externa são:

- Arquivos mestres de outras aplicações;
- Tabelas criadas para atender outras aplicações;
- Arquivos de referência, cujos dados externos são utilizados pela aplicação mas não sofrem qualquer tipo de manutenção pela aplicação.

6.4.3. ENTRADA EXTERNA

As Entradas Externas representam as atividades de manutenção de dados e processamentos de controle da aplicação. Este componente influencia no cálculo de FPA de acordo com sua quantidade e complexidade funcional de cada uma das entradas.

Uma Entrada Externa processa dados ou informações de controle oriundas da fronteira da aplicação. Os dados atualizam os Arquivos Lógicos Internos através de um processo lógico, enquanto que, as informações de controle podem ou não atualizar diretamente um Arquivo Lógico Interno.

As entradas de dados On-Line e via Batch são consideradas exemplos de Entradas Externas.

6.4.4. SAÍDA EXTERNA

Ao contrário das Entradas Externas, as Saídas Externas representam as atividades da aplicação que tem como resultado à saída de dados. As Saídas Externas contribuem para o cálculo dos Pontos de Função com base na sua quantidade e complexidade funcional de cada uma delas.

Uma Saída Externa pode ser definida como uma atividade ou processo que envia dados ou informações de controle para fora das fronteiras da aplicação.

Uma Saída Externa pode ser exemplificada por:

- Dados transferidos para outra aplicação;
- Relatórios
- Relatórios On-Line;
- Gráficos gerados em forma de texto.

6.4.5. CONSULTA EXTERNA

As Consultas Externas representam as necessidades de recuperação imediata de dados da aplicação. As Consultas Externas contribuem para o cálculo dos Pontos de Função com base na sua quantidade e complexidade funcional de cada uma delas.

A Consulta Externa é uma combinação de entrada e saída de dados, onde uma entrada de dados ocasiona uma recuperação e saída imediata de dados. Uma Consulta Externa não contém dados derivados e não atualiza nenhum Arquivo Lógico Interno.

Alguns exemplos de Consultas Externas são:

- Recuperação de dados;
- Consulta de dados;
- Consulta de mensagens de auxílio.

6.5. CÁLCULO DE PONTOS DE FUNÇÃO – METODOLOGIA 1

A Metodologia 1 também é conhecida como FPA Detalhada, e os procedimentos básicos para o cálculo dos Pontos de Função de acordo com o Counting Practice Manual Release 4.1.1 do IFPUG são:

6.5.1. DETERMINAR O TIPO DE CONTAGEM

A contagem de Pontos de Função pode ser aplicada aos seguintes tipos de projetos:

- Projetos novos de desenvolvimento;
- Projetos de melhoria;
- Aplicações existentes e em produção.

6.5.2. IDENTIFICAR AS FRONTEIRAS DA CONTAGEM

Uma fronteira indica os limites entre a aplicação ou o projeto que está sendo medido e as aplicações externas, ou seja, o escopo do projeto.

A fronteira determina as funções que deverão ser medidas. Estas funções são determinadas através de um levantamento preliminar feito juntamente com o cliente e a respectiva especificação dos requisitos do sistema [FERNANDES 95].

6.5.3. IDENTIFICAR AS FUNÇÕES DO SISTEMA

Nesta etapa é necessário identificar e enumerar as funções da aplicação, ou seja, determinar os Arquivos Lógicos Internos, Arquivos de Interface Externa, Entradas Externas, Saídas Externas e Consultas Externas.

6.5.4. CLASSIFICAR AS FUNÇÕES

Após a identificação das funções, é necessário classificar cada uma delas conforme seu nível de complexidade relativo, que pode ser baixo, médio ou alto.

O nível de complexidade das Funções de Dados é determinado de acordo com o número de itens de dados no arquivo e o número de registros lógicos associados para cada Arquivo Lógico Interno e Arquivo de Interface Externa identificado.

O nível de complexidade das Funções Transacionais é determinado pelo número de itens de dados no arquivo e o número de arquivos referenciados associados para cada Entrada Externa, Saída Externa e Consulta Externa.

As tabelas a seguir apresentam a classificação do nível de complexidade para cada uma das respectivas funções.

Número de Registros Lógicos	Itens de Dados Referenciados		
	De 1 a 19	De 20 a 50	51 ou mais
Apenas 1	Baixa	Baixa	Média
De 2 a 5	Baixa	Média	Alta
6 ou mais	Média	Alta	Alta

Tabela 6.1 – Classificação dos Arquivos Lógicos Internos

Número de Registros Lógicos	Itens de Dados Referenciados		
	De 1 a 19	De 20 a 50	51 ou mais
Apenas 1	Baixa	Baixa	Média
De 2 a 5	Baixa	Média	Alta
6 ou mais	Média	Alta	Alta

Tabela 6.2 – Classificação dos Arquivos de Interfaces Externas

Número de Arquivos Referenciados	Itens de Dados Referenciados		
	De 1 a 4	De 5 a 15	16 ou mais
1	Baixa	Baixa	Média
2	Baixa	Média	Alta
3 ou mais	Média	Alta	Alta

Tabela 6.3 – Classificação das Entradas Externas

Número de Arquivos Referenciados	Itens de Dados Referenciados		
	De 1 a 5	De 6 a 19	20 ou mais
Apenas 1	Baixa	Baixa	Média
De 2 a 3	Baixa	Média	Alta
4 ou mais	Média	Alta	Alta

Tabela 6.4 – Classificação das Saídas Externas

Número de Arquivos Referenciados	Itens de Dados Referenciados		
	De 1 a 5	De 6 a 19	20 ou mais
Apenas 1	Baixa	Baixa	Média
De 2 ou 3	Baixa	Média	Alta
4 ou mais	Média	Alta	Alta

Tabela 6.5 – Classificação das Consultas Externas

6.5.5. Determinar os Pontos de Função não Ajustados

Após a classificação das funções conforme o seu nível de complexidade, é necessário determinar o número de Pontos de Função não ajustados através da aplicação de uma tabela de transformação. A técnica pressupõe que cada nível de complexidade de uma função tem um determinado peso, como mostra a Tabela 6.6.

Para o cálculo de Pontos de Função não ajustados, é necessário multiplicar o número de ocorrências de cada função e de acordo com sua complexidade, pelo respectivo peso. Os pontos de função não ajustados correspondem ao somatório desses resultados.

A contagem de pontos não ajustados reflete as funções específicas e mensuráveis do negócio, provida ao usuário pela aplicação.

FUNÇÃO	OCORRÊNCIAS	COMPLEXIDADE	PESO	RESULTADO
ENTRADAS EXTERNAS	N	BAIXA	3	N * 3
	N	MÉDIA	4	N * 4
	N	ALTA	6	N * 6
			TOTAL 1	
SAÍDAS EXTERNAS	N	BAIXA	4	N * 4
	N	MÉDIA	5	N * 5
	N	ALTA	7	N * 7
			TOTAL 2	
ARQUIVOS LÓGICOS INTERNOS	N	BAIXA	7	N * 7
	N	MÉDIA	10	N * 10
	N	ALTA	15	N * 15
			TOTAL 3	
ARQUIVOS DE INTERFACES EXTERNAS	N	BAIXA	5	N * 5
	N	MÉDIA	7	N * 7
	N	ALTA	10	N * 10
			TOTAL 4	
CONSULTAS EXTERNAS	N	BAIXA	3	N * 3
	N	MÉDIA	4	N * 4
	N	ALTA	6	N * 6
			TOTAL 5	
TOTAL DE PONTOS DE FUNÇÃO NÃO AJUSTADOS				

Tabela 6.6 - Transformação para Determinação dos Pontos de Função Não Ajustados

6.5.6. DETERMINAR O VALOR DE AJUSTAMENTO

Esta etapa gera o valor do fator de ajustamento, que representa a funcionalidade geral proporcionada ao usuário pela aplicação. A técnica FPA mede a funcionalidade geral de uma aplicação através da avaliação do nível de influência de 14 características gerais dos sistemas.

O nível de influência dessas características é determinado por uma escala que pode variar de 0 a 5, onde:

0 = não está presente ou não tem influência;
1 = pouca influência;
2 = moderada influência;
3 = média influência;
4 = significante influência;
5 = forte influência.

As principais características dos sistemas levadas em consideração são:

- Comunicação de dados;
- Processamento de dados distribuído;
- Desempenho;
- Configuração pesadamente utilizada;
- Taxa de transação;
- Entrada de dados On-Line;
- Eficiência do usuário final;
- Atualização On-Line;
- Processamento complexo;
- Reutilização;
- Facilidades de instalação;
- Facilidade operacional;
- Múltiplas instalações;
- Facilidade de mudança.

Para cada uma dessas características é necessário aplicar o nível de influência. No Manual de Práticas de Contagem do IFPUG podem ser encontrados os textos explicativos associados a cada característica e respectivo nível de influência, visando facilitar a classificação. Se nenhum dos textos se adequar à aplicação avaliada, deve ser escolhido o que melhor se aproxima do caso.

Em seguida, deve ser feito o somatório dos níveis, gerando assim o nível de influência total. Finalmente, para cálculo do fator de ajuste, basta utilizar a fórmula:

Fator de Ajuste = 0,65 + (0,01 * Nível de Influência Total)

6.5.7. DETERMINAR OS PONTOS DE FUNÇÃO AJUSTADOS

A última etapa resulta na contagem de Pontos de Função ajustados, onde se aplica o fator de ajuste ao número de Pontos de Função não ajustados, calculados nas etapas anteriores:

Pontos de Função Ajustados = Fator de Ajuste * Pontos de Função Não Ajustados

6.6. CÁLCULO DE PONTOS DE FUNÇÃO – METODOLOGIA 2

A Metodologia 2 também é conhecida como FPA Estimativa e foi desenvolvida pelo IFPUG em conjunto com a empresa Software Productivity Research.

A Software Productivity Research coletou dados a partir de 120 projetos e verificou a existência de uma correlação entre os vários tipos de funções de um sistema. Essa correlação foi expressa da seguinte forma:

- a quantidade de Arquivos Lógicos Internos representa 25% do total de funções de um sistema;
- a quantidade de Arquivos de Interfaces Externas representa 3% do total de funções de um sistema;
- a quantidade de Entradas Externas representa 30% do total de funções de um sistema;
- a quantidade de Saídas Externas representa 28% do total de funções de um sistema;
- a quantidade de Consultas Externas representa 14% do total de funções de um sistema;

Para a metodologia 2, os procedimentos básicos para o cálculo dos Pontos de Função de acordo com o Counting Practice Manual Release 4.1.1 do IFPUG, são:

- Determinar o tipo de contagem;
- Determinar as fronteiras da aplicação;
- Determinar a quantidade de cada tipo de função. Baseado na análise estatística apresentada, esta etapa pode ser realizada identificando-se apenas a quantidade de um dos tipos de funções do sistema, para então derivar os demais;
- Determinar a classificação dos tipos de função. Neste caso, deve ser considerada sempre uma complexidade funcional média para cada tipo de função;
- Determinar os pontos de função não ajustados. Para tanto, é necessário multiplicar o número de ocorrências de cada função pelo respectivo peso, considerando sempre a complexidade média;
- Determinar os pontos de função ajustados considerando o valor do fator de ajuste igual a 1.

A aplicabilidade da técnica FPA estimativa é interessante nas fases iniciais de um projeto, quando a única informação disponível são os requisitos funcionais do sistema e um modelo preliminar de dados.

É importante ressaltar que este método presume um relacionamento lógico e previsível entre variáveis independentes, quando na realidade este relacionamento pode não existir. Nesses casos, à medida que aumenta o grau de conhecimento sobre o sistema, pode-se identificar qual e em que quantidade um determinado tipo de função ficou fora destas médias, procedendo-se os ajustes necessários no planejamento das atividades e dos recursos [AZEVEDO 00].

6.7. PRECISÃO DA TÉCNICA FPA

Uma das principais vantagens da técnica FPA é a possibilidade de estimar a dimensão do projeto em todas as fases, incluindo principalmente as fases iniciais. Porém, a precisão da estimativa de tamanho varia de acordo com o grau de conhecimento adquirido sobre o sistema. A empresa SPR – Software Productivity Research realizou uma pesquisa apresentando a relação existente entre o conhecimento do projeto e a margem de erro da técnica FPA, conforme mostra tabela 6.7.

Fases do Projeto	Conhecimento do Projeto	Percentual de Erro
0 – Planejamento	Até 10%	
1 – Requerimentos	Até 25%	Até +/- 35%
2 – Desenho Inicial	Até 40%	Até +/- 25%
3 – Desenho Detalhado	Até 60%	Até +/- 20%
4 – Codificação	Até 80%	Até +/- 10%
5 – Testes	Até 90%	Até +/- 3%
6 – Instalação	100%	0%

Tabela 6.7 - Relação entre o Conhecimento do Projeto e o Percentual de Erro

6.8. APLICAÇÕES DA TÉCNICA FPA

A técnica FPA pode ser utilizada para [HELLER 02] e [LONGSTREET 00]:

- Dimensionar o tamanho de aplicações já implantadas;
- Dimensionar o tamanho de projetos novos de desenvolvimento;
- Dimensionar o tamanho de um projeto de manutenção;
- Determinar o nível de produtividade da equipe;
- Determinar o esforço de desenvolvimento de software;
- Determinar estimativas de cronograma e custo de software;
- Determinar a taxa de produção de software;
- Determinar a taxa de manutenção de software;
- Estimar casos de testes;
- Definir quando e onde fazer reengenharia;
- Ajudar a entender faixas de produtividade amplas;
- Entender e dimensionar o aumento do escopo do sistema;
- Ajudar em negociações contratuais;
- Desenvolver uma padronização de métricas na empresa;
- Prover um fator de comparação entre produtos de software;

6.9. CONCLUSÕES

A técnica FPA tem alcançado uma expressiva utilidade na gerência de projetos de software, sobretudo devido à sua independência tecnológica, simplicidade de aprendizagem, consistência e possibilidade de aplicação em qualquer fase do projeto.

Conforme foi relatado neste capítulo, a FPA é utilizada para medir o tamanho de um produto de software através da quantificação dos requisitos funcionais fornecidos pelos usuários.

A princípio, a informação de tamanho do software em pontos de função parece ter pouco significado prático. No entanto, quando utilizada de forma eficiente, confiável e em combinação com outras medidas, a técnica possui várias aplicações.

A partir da informação de tamanho do software e de alguns outros atributos coletados ao final do projeto é possível derivar várias métricas referentes ao projeto. Essas métricas são todas relativas a um ponto de função, como por exemplo, o tempo médio de desenvolvimento por ponto de função, o esforço requerido, a produtividade média, o custo médio por ponto de função, e outros.

Essas métricas, por sua vez, podem ser utilizadas para a criação e manutenção de uma série histórica, que servirá de base para a elaboração de estimativas em projetos novos e com características comuns. Ao final de cada projeto, é importante coletar essas métricas para complementação da base histórica. Com isso, se cria um ciclo de melhoria e customização do programa de mensuração dentro da empresa.

Diante do estudo realizado neste capítulo, constatou-se que a possibilidade de dimensionar o tamanho de um software na fase de planejamento e, junto com uma série histórica na empresa, elaborar as estimativas do projeto com um certo grau de precisão, é um dos aspectos mais importantes que viabiliza a utilização da FPA. Infelizmente, projetos importantes são conduzidos ao fracasso devido à elaboração de estimativas considerando apenas o “feeling” e experiências pessoais, sem a utilização de uma técnica formal.

Apesar das inúmeras vantagens que a FPA pode trazer, é possível concluir que a aplicação da técnica requer alguns investimentos, como:

- o esforço requerido para contagem dos pontos de função;
- a disponibilidade de um meio eficiente de armazenagem dos resultados obtidos após a contagem;
- o acompanhamento de alguns atributos para, em combinação com os pontos de função, derivar as métricas do projeto.

7. APLICAÇÃO PRÁTICA DA TÉCNICA FPA

Após um estudo detalhado, a técnica FPA demonstrou ser uma ferramenta importante para o gerenciamento de processos de software, principalmente para a determinação de estimativas na fase de planejamento de projetos.

Com o intuito de avaliar a aplicabilidade da técnica FPA no mercado atual de informática e exemplificar o processo de contagem de pontos de função, este trabalho de pesquisa foi complementado através da aplicação prática da técnica FPA em um determinado projeto de uma empresa catarinense, cujo ramo de negócios é o desenvolvimento de sistemas.

O projeto selecionado foi o Sistema Auditor, desenvolvido pelas empresas AltoQi Tecnologia em Informática Ltda. em parceria com a Softcon Consultoria e Sistemas Ltda., ambas sediadas em Florianópolis – SC. O Sistema se encontra atualmente em fase final de testes e será descrito em maiores detalhes neste capítulo.

A aplicação da técnica FPA no Sistema Auditor foi motivada por uma série de fatores, dentre elas, a introdução de uma melhoria no processo de planejamento de produtos dessas empresas.

O objetivo específico deste trabalho prático é a definição do tamanho do Sistema Auditor através da aplicação da técnica FPA. Como o sistema se encontra em fase final, é possível determinar o seu tamanho com uma baixa margem de erro. De posse dessa informação, a empresa tem condições de coletar e armazenar dados gerenciais importantes, como por exemplo, tempo médio de desenvolvimento por ponto de função, custo médio por ponto de função, esforço médio de desenvolvimento por ponto de função, entre outros. Essas informações, por sua vez, servirão de base para a elaboração de estimativas para novos projetos da empresa com características tecnológicas iguais ou semelhantes.

A metodologia adotada para a aplicação da técnica FPA no Sistema Auditor, abrangeu as seguintes atividades:

- Análise e levantamento de informações junto ao usuário
- Descrição do Sistema
- Elaboração de documento contendo os requisitos e funcionalidades do Sistema
- Aprovação dos requisitos
- Elaboração do Modelo Preliminar de Dados
- Cálculo da FPA Estimativa
- Estudo do Sistema Auditor existente
- Cálculo da FPA Detalhada

Apesar do Sistema se encontrar em fase final de testes dentro da empresa, foi necessário realizar a análise e estudo do mesmo, para a devida aquisição de conhecimento sobre as suas funcionalidades. Isto viabilizou a aplicação das técnicas Estimativa e Detalhada, uma vez que elas foram aplicadas em momentos distintos, conforme o grau de conhecimento sobre o Sistema.

Nos próximos itens, será apresentado uma breve descrição e um detalhamento dos requisitos e funcionalidades do Sistema Auditor. Em seguida, os passos realizados para a contagem dos pontos de função do sistema da forma estimativa e detalhada serão descritos, bem como, os respectivos resultados.

Ao final deste capítulo, será apresentada uma conclusão contendo uma análise dos resultados obtidos. O intuito é demonstrar os impactos positivos e negativos de utilização da técnica FPA Detalhada e Estimativa, bem como, evidenciar possíveis melhorias no planejamento de projetos da empresa através do uso de FPA.

7.1. DESCRIÇÃO DO SISTEMA

O **Sistema Auditor** tem como objetivos principais controlar as requisições de compras dentro da empresa e a importação de informações oriundas do Sistema de Vendas. Em paralelo a essas funcionalidades, o Sistema é responsável pelo controle orçamentário e principalmente pelo gerenciamento financeiro e contábil da empresa e de seus centros de custo. O gerenciamento financeiro abrange todas as atividades de contas a pagar e receber, enquanto que, o controle contábil é efetuado à medida que as atividades financeiras são registradas no Sistema.

Em resumo, o Sistema agiliza e integra os processos de Compras, de Vendas, de Contas a Pagar, de Contas a Receber e de Contabilidade, e principalmente, evita a duplicidade de informações e o retrabalho para lançamento das informações em cada um dos módulos.

7.1.1. LEVANTAMENTO DE FUNCIONALIDADES E REQUISITOS

De uma forma geral, as funcionalidades do Sistema contemplam os seguintes módulos:

- **Módulo de Compras:** este módulo mantém as atividades de previsão orçamentária, solicitação de compra, autorização da solicitação, tomada de preços, autorização de compra, execução e liquidação da compra.

- **Módulo de Vendas:** este módulo permite a integração entre este Sistema e o Sistema de Vendas da empresa. Esta integração é realizada através da exportação de dados do Sistema de Vendas e a importação no Sistema Auditor. Os dados exportados são relativos às vendas de produtos e serviços prestados pela empresa e importados no Sistema como lançamentos de receitas, servindo de entrada para o módulo de contas a receber e para atualização do controle financeiro e contábil da empresa.

- **Módulo Financeiro de Contas a Pagar:** este módulo abrange os processos de contas a pagar e baixas, empenho e pagamento rápido, impressão de cheques e transferências bancárias.

- **Módulo Financeiro de Contas a Receber:** este módulo é responsável pelo controle financeiro das receitas da empresa, abrangendo os processos de contas a receber, baixas e desconto rápido.

- **Módulo Contábil:** este módulo é atualizado de forma paralela, à medida que as demais atividades vão sendo executadas no Sistema. Com isso, são evitadas as redundâncias e o retrabalho de digitação de informações, permitindo a integração do módulo contábil com os demais módulos. Este processo é viabilizado através da determinação dos códigos do Plano de Contas para cada atividade realizada no Sistema.

  Em algumas situações, se faz necessário o registro explícito de determinados tipos de lançamentos contábeis. Neste caso, o Sistema disponibiliza um processo específico para cadastro destes lançamentos contábeis.

- **Módulo de Segurança:** todos os processos do Sistema são suportados por um módulo de segurança, permitindo que determinadas atividades sejam acessadas e executadas somente por pessoas devidamente autorizadas. Além disso, antes de efetuar uma tarefa é necessário que seja fornecida a assinatura digital do operador, que fica devidamente registrada, e podendo ser impressa na emissão de documentos.

Neste documento, as funcionalidades e requisitos do Sistema serão divididas nos seguintes grupos: Cadastros e Configurações, Processos, Consultas e Relatórios.

7.1.1.1. CADASTROS E CONFIGURAÇÕES

Esta funcionalidade permite a configuração e padronização de uma série de informações que serão utilizados nos processos do Sistema.

a) **Cadastro de Empresas:** esta funcionalidade permite manter a lista de empresas que serão administradas pelo Sistema Auditor e definir os centros de custos de cada empresa.

Este módulo tem uma grande importância para o Sistema, pois também permite a configuração do Plano de Contas para a empresa selecionada. Esta configuração viabiliza a integração entre os vários processos do Sistema e o processo de Contabilidade, evitando o retrabalho no lançamento dos dados. Dessa forma, para determinadas atividades realizadas pelo Sistema, automaticamente é registrado o respectivo código de Plano de Contas, utilizado para controle contábil.

Na configuração de Plano de Contas da empresa, é possível definir o código do plano de contas para cada um dos itens abaixo:

- para o ativo e passivo da empresa
- para as despesas e receitas lançadas no Sistema
- para as disponibilidades
- para os grupos de fornecedores
- para os grupos de retenções
- para os grupos de cauções
- para as contas de juros passivos
- para os descontos obtidos e concedidos
- para os cheques pré-datados
- para os grupos de clientes
- para os juros recebidos
- para as multas recebidas
- para as despesas de cartório

b) **Cadastro de Centros de Custo:** os centros de custos auxiliam no controle financeiro das empresas. Este módulo permite cadastrar todos os centros de custo das empresas, definir os responsáveis de cada centro e os seus principais atributos.

c) **Cadastro de Plano de Contas:** este cadastro permite que sejam registrados todos os códigos de plano de contas utilizados nas empresas, com o respectivo tipo e natureza da conta. Para cada plano de conta é possível associar determinados atributos do plano, facilitando o agrupamento e a pesquisa de lançamentos contábeis.

d) **Cadastro de Produtos e Serviços:** este módulo é responsável pelo cadastro dos mais variados produtos e serviços adquiridos e comercializados pelas empresas. Esta lista de produtos e serviços é utilizada posteriormente no processo de solicitação de compra e para a integração com o Sistema de Vendas da empresa.

e) **Cadastro de Atributos:** neste módulo são definidos os mais variados atributos. Estes atributos são associados a determinados centros de custos e planos de conta, permitindo o agrupamento e facilitando a pesquisa dos lançamentos contábeis.

f) **Cadastro de Índices de Rateio:** existem certos tipos de despesas que precisam ser divididos entre os vários centros de custos de uma empresa. A partir dessa necessidade, foi criado este módulo que permite definir essas despesas e o respectivo percentual a ser debitado para cada centro de custo.

g) **Cadastro de Históricos Padronizados:** este módulo permite que sejam registradas e padronizadas as descrições dos mais variados tipos de lançamentos. Esta padronização facilita o gerenciamento financeiro e contábil, evitando redundâncias e interpretações duvidosas nos lançamentos contábeis.

h) **Cadastro de Fontes:** neste módulo é possível definir as possíveis fontes de recursos financeiros a serem utilizados nas solicitações de compra do Sistema.

i) **Cadastro de Tipos de Documentos:** com o intuito de padronizar a descrição dos tipos de documentos utilizados nos processos de compra, de venda, financeiro e contábil, neste módulo é possível cadastrar os diversos tipos de documentos, como por exemplo: nota fiscal, recibos, faturas, pedidos, orçamento, entre outros.

j) **Cadastro de Caixa/Banco:** este módulo mantém todas as contas bancárias da empresa, permitindo especificar o banco, a agência, o número da conta corrente e os dados do banco como CGC e endereço. Além disso, é possível especificar o tipo da conta (conta corrente, conta caução, conta desconto, empréstimo) e o limite bancário concedido para a empresa. Um controle de numeração do último talão de cheques recebido também é incluído neste módulo, para fins de segurança.

k) **Cadastro de Entidades:** o cadastro de entidades mantém informações relevantes dos fornecedores, representantes e clientes. Estas informações incluem a Razão Social, CPF/CGC, endereço completo e dados para contato. Para cada entidade cadastrada, é necessário especificar o seu tipo, que pode ser Cliente, Fornecedor, Representante ou Todos.

7.1.1.2. PROCESSOS

Neste item será apresentada uma descrição mais detalhada dos processos contemplados pelo Sistema.

a) **Solicitação de Compras:** este processo permite que os usuários autorizados de cada centro de custo (solicitantes) realizem as solicitações de compra de mercadorias ou serviços, obedecendo ao orçamento previamente estabelecido.

Para cada nova solicitação é necessário informar:

- a data da solicitação,
- o valor total da solicitação de compra ou serviço,
- o número estimado de parcelas de pagamento,
- a data de vencimento estimada da primeira parcela,
- a descrição da solicitação,
- a lista de partidas ou lançamentos.

Para cada partida cadastrada na solicitação é necessário especificar:

- os centros de custo beneficiados na solicitação do produto ou serviço,
- o respectivo plano de contas para permitir o lançamento contábil,
- o valor total da partida,
- a descrição da partida,
- a lista de produtos e serviços da partida, contendo o tipo de produto, unidade de medida, quantidade solicitada, valor unitário, desconto e valor total do produto ou serviço.

Para auxiliar no controle orçamentário, durante o processo de solicitação o Sistema apresenta o valor total mensal solicitado, licitado, orçado, executado e o orçamento mensal previsto. Estes valores mensais são apresentados para cada tipo de regime utilizado na empresa: Regime de Competência ou Regime de Caixa.

b) **Autorização de Solicitações:** as solicitações cadastradas no processo anterior só poderão ser atendidas mediante o aval da diretoria da empresa. O Sistema permite que os responsáveis pela autorização consultem todos os dados das solicitações pendentes e o respectivo extrato orçamentário, optando pela aprovação ou cancelamento da solicitação. Em seguida, o Sistema registra a data da autorização ou cancelamento e o responsável.

c) **Tomada de Preços:** após a aprovação da solicitação de compra de mercadoria ou serviço, o usuário autorizado pode dar continuidade ao processo realizando um levantamento de fornecedores e preços. Neste caso, o Sistema apresenta uma tela listando as solicitações já autorizadas com os respectivos produtos e serviços solicitados, quantidades e valores estimados.

O processo de cotação de preços consiste em contatar vários fornecedores e colher informações financeiras sobres os produtos ou serviços solicitados. Nesta etapa, o usuário deve registrar os seguintes dados no Sistema:

- o fornecedor
- a descrição da cotação
- o valor total cotado
- a lista de parcelas com o respectivo valor da parcela e data de vencimento
- a lista de produtos com o respectivo valor unitário e desconto
- as instruções de cobrança, se houver necessidade

Para cada solicitação podem ser realizadas tantas cotações quanto forem necessárias, que serão analisadas posteriormente e uma delas será aprovada e executada.

d) **Autorização de Compras:** as compras efetivas dos produtos ou serviços, levando em consideração os preços orçados na etapa anterior, só poderão ser executadas mediante o aval da diretoria da empresa ou responsável. O Sistema permite que os responsáveis pela autorização consultem os dados das solicitações pendentes e os respectivos orçamentos de cada fornecedor, optando pela aprovação de uma das cotações ou cancelamento da solicitação.

Para auxiliar nessa tarefa, o Sistema mostra:

- os dados da solicitação e autorização
- os preços dos produtos e serviços de cada fornecedor

- os dados de parcelamento de cada fornecedor
- algumas instruções ou observações, quando for necessário
- um extrato orçamentário mensal de acordo com o tipo de regime, permitindo visualizar a situação financeira geral da empresa, com relação aos gastos e despesas orçadas e executadas.

Nesta etapa, o Sistema registra a data da autorização ou cancelamento da compra e o responsável.

e) **Execução de Compra ou Empenho:** após a aprovação da compra de mercadoria ou serviço, o usuário autorizado pode dar continuidade ao processo realizando efetivamente a compra com o fornecedor escolhido. Neste caso, o Sistema apresenta uma tela listando as solicitações devidamente orçadas e autorizadas, os respectivos produtos e serviços, quantidades e valores orçados. Esta solicitação pode ser impressa para que seja enviado ao fornecedor, oficializando o pedido de compra.

f) **Liquidação:** este processo consiste em avaliar a qualidade do produto no ato da entrega ou execução do serviço, e autorizar o pagamento pela área financeira, liquidando a compra. Para consultar a lista de compras pendentes de liquidação ou já liquidadas, o Sistema mostra uma tela contendo:

- Número do empenho
- Número da cotação de preços
- Número da autorização
- Número da solicitação
- Valor da parcela
- Número de parcelas
- Fornecedor

Quando é realizada a liquidação da compra, devem ser informados o tipo de documento usado, a data e alguma observação, caso seja necessário.

g) **Previsão Orçamentária:** este processo permite que seja registrada a previsão orçamentária anual da empresa e de cada centro de custo, divididas em períodos mensais e de acordo com o tipo de regime (por competência ou por caixa).

Com isso, o Sistema exibe as informações orçamentárias previstas e executadas durante o processo de compras de mercadorias ou serviços, auxiliando os responsáveis na tomada de decisão para autorizar ou cancelar as solicitações.

h) **Contas a Pagar e Baixas:** este processo é responsável pela atualização financeira e contábil dos lançamentos de despesas da empresa, como por exemplo, o pagamento das parcelas referentes às compras de mercadorias ou serviços.

O Sistema permite consultar as contas a pagar, contas pagas ou ambas, indicando:

- o número do empenho,
- o número de parcelas,
- o tipo de documento,
- a data de vencimento,
- o valor da parcela, valor pago e valor em aberto,
- o fornecedor que receberá o pagamento.

Para registrar um pagamento, é necessário primeiro selecionar a parcela em aberto. Em seguida, o Sistema mostra todas as informações financeiras relativas a essa parcela, bem como os dados do fornecedor e os pagamentos já efetuados nessa compra. Finalmente, deve ser informado o valor a pagar, a data de pagamento, a descrição deste lançamento e os acréscimos ou descontos, se houver. Se o pagamento for feito através de cheque, a impressão do cheque também pode ser solicitada neste momento. Com isso, o Sistema realiza a baixa da parcela, atualizando as informações financeiras e gerando os lançamentos contábeis devidos.

Em determinadas situações, é necessário reverter uma baixa que já foi lançada no Sistema. Neste caso, é possível consultar a lista de baixas efetuadas em um determinado período e proceder com o cancelamento.

i) **Compra e Pagamento Rápido:** este processo faz parte do módulo financeiro de Contas a Pagar. O objetivo principal é o lançamento dos dados financeiros e contábeis em situações de compra e pagamento rápido, que dispensam o fluxo de atividades do módulo de compras.

Para estes casos, devem ser registradas as seguintes informações:

- Data da compra;
- Data do vencimento;
- Data do pagamento;
- Fornecedor;
- Código do Plano de Contas relativo à despesa;
- Código do Caixa/Banco;
- Número do Cheque;
- Lista de centros de custos beneficiados com a compra e o respectivo valor a pagar;
- Acréscimos e descontos do pagamento, indicando o código do Plano de Contas, uma descrição e o valor;
- O tipo de documento usado para pagamento.

Neste processo, o Sistema executa de forma rápida as principais atividades de compra de mercadoria e serviço, lançando e liquidando a compra e efetuando o pagamento. O controle orçamentário e contábil também é realizado com base nos dados informados.

j) **Impressão de Cheques:** caso a impressão do cheque não seja efetuada no momento em que é realizada a baixa no Sistema, ela pode ser realizada posteriormente através do processo de impressão.

Em algumas situações, é interessante agrupar vários pagamentos em um único cheque, evitando gastos desnecessários. Nestes casos, o Sistema permite visualizar e selecionar os lançamentos que serão pagos através deste cheque.

k) **Transferência Bancária:** esta funcionalidade permite que sejam realizadas transferências monetárias entre as contas financeiras, para eventuais ajustes. Para tanto, é necessário informar a data da transferência, código do Plano de Contas para crédito, código do Plano de Contas para débito, o valor e a descrição do lançamento. A impressão de cheque para efetivar a transferência também pode ser efetuada neste processo.

l) **Contas a Receber:** este processo é responsável pelo cadastro dos lançamentos relativos às receitas da empresa e a respectiva atualização financeira e contábil. De acordo com o tipo de receita, estes lançamentos podem ser importados a partir do Sistema de Vendas ou lançados manualmente no Sistema. O cadastro de contas a receber mantém as seguintes informações:

- Número do lançamento
- Data de entrada;
- Tipo de documento;
- Representante ou Vendedor;
- Cliente;
- Lista de receitas contendo o código do Plano de Contas e o respectivo valor recebido;
- Lista de parcelamentos contendo a data de vencimento e o respectivo valor;
- Descrição do lançamento contábil;

m) **Contas a Receber e Baixas:** este processo é responsável pela atualização financeira e contábil dos pagamentos recebidos pela empresa, como por exemplo, o recebimento de pagamentos referentes às vendas de mercadorias ou serviços realizados pela empresa para seus clientes.

O Sistema permite consultar as contas a receber, contas recebidas ou ambas, indicando:

- o número do lançamento,
- o número de parcelas,
- o tipo de documento,
- a data de vencimento,
- o valor da parcela, valor pago e valor em aberto,
- o cliente que efetuou o pagamento.

Para registrar o recebimento de um pagamento, é necessário primeiro selecionar a parcela em aberto. Em seguida, o Sistema mostra todas as informações financeiras relativas a essa parcela, bem como os dados do cliente e os pagamentos já recebidos nessa determinada venda. Finalmente, deve ser informado o valor a receber, a data de pagamento, a descrição deste lançamento e os acréscimos ou descontos, se houver. Com isso, o Sistema realiza a baixa da parcela para o cliente, atualizando as informações financeiras e gerando os lançamentos contábeis devidos.

Em determinadas situações, é necessário reverter uma baixa que já foi lançada no Sistema. Neste caso, é possível consultar a lista de baixas efetuadas em um determinado período e proceder com o cancelamento.

n) **Desconto Rápido:** este processo faz parte do módulo financeiro de Contas a Receber. O objetivo principal é o lançamento dos dados financeiros e contábeis, de forma rápida, em situações que seja necessário fornecer um determinado desconto. Para aplicar um desconto rápido, devem ser registradas as seguintes informações:

- o número da operação;
- a data do desconto;

- o código do banco;
- o código do Plano de Contas;
- o valor do desconto;
- a descrição do lançamento;
- o tipo de documento usado para fornecer o desconto.

o) Lançamento Contábil: este processo faz parte do módulo contábil. O objetivo principal é o cadastro de informações em situações que seja necessário registrar alguns tipos de lançamentos contábeis específicos ou para efetuar ajustes de contabilidade. Para estes casos, devem ser registradas as seguintes informações:

- o número do lote;
- a data do lote;
- a lista de lançamentos com o respectivo valor e data;
- o código do Plano de Contas para Débito;
- a lista de Centros de Custo para Débito;
- o código do Plano de Contas para Crédito;
- a lista de Centros de Custo para Crédito;
- a descrição do lançamento.

Para cada lote, o Sistema apresenta o total de créditos, o total de débitos, o número de lançamentos e o saldo.

7.1.1.3. CONSULTAS

As seguintes consultas fazem parte do escopo do Sistema:

- Consulta de Lançamentos
- Consulta de Lançamentos por Plano de Contas
- Consulta de Lançamentos por Vencimento
- Consulta de Compras

- Consulta de Compras Liquidadas
- Consulta de Compras Pagas
- Consulta de Contas a Pagar - Duplicatas a Pagar
- Consulta de Contas a Pagar - Pagamentos e Baixas Realizadas
- Consulta de Contas a Pagar - Cheques
- Consulta de Contas a Receber - Vendas Realizadas
- Consulta de Saldo

7.1.1.4. RELATÓRIOS

Os seguintes relatórios estão disponíveis no Sistema:

- Relatório de Contas a Receber – Títulos por Data de Emissão
- Relatório de Contas a Receber – Baixa
- Relatório de Contas a Receber por Data de Lançamento
- Relatório de Contas a Receber por Data de Vencimento
- Relatório de Movimento Financeiro
- Relatório de Razão
- Relatório de Razão Individualizado
- Relatório de Balancete do Plano de Contas
- Relatório de Balancete dos Centros de Custo
- Relatório de Balancete Individualizado
- Relatório de Rateio
- Relatório de Lançamentos
- Relatório Diário
- Relatório de Extrato dos Bancos
- Relatório de Vencimentos por Data de Vencimento
- Relatório de Vencimentos por Credor
- Relatório de Ficha do Credor
- Relatório de Fluxo de Caixa

- Relatório de Lista de Plano de Contas
- Relatório de Lista de Centros de Custo
- Relatório de Lista de Históricos Padronizados

7.1.2. CARACTERÍSTICAS ESPECIAIS

As seguintes características são destacadas com relação à funcionalidade geral do Sistema:

- O acesso e utilização do Sistema são de forma online;
- A atualização das informações é online e em tempo real;
- Critérios contra perda e segurança de dados são considerados;
- Para a instalação do Sistema pode ser mantida a configuração atual de equipamentos;
- O Sistema apresenta fácil navegação entre as telas;

7.2. CÁLCULO DE PONTOS DE FUNÇÃO – FPA ESTIMATIVA

Após o levantamento dos requisitos e funcionalidades do Sistema Auditor, foi elaborado um modelo preliminar de dados, disponibilizado no Anexo 1 deste documento. Com essas informações, foi possível determinar o total de pontos de função do sistema utilizando a técnica FPA estimativa, conforme os seguintes passos:

PASSO 1 - DETERMINAR O TIPO DE CONTAGEM

O Sistema Auditor é uma aplicação existente que atualmente se encontra em fase final de testes.

PASSO 2 - DETERMINAR AS FRONTEIRAS DA APLICAÇÃO

As funcionalidades e requisitos do Sistema que delimitam a fronteira da aplicação foram descritos no item 7.1.

PASSO 3 - DETERMINAR O NÚMERO DE OCORRÊNCIAS DOS TIPOS DE FUNÇÃO

Considerando a correlação existente entre os vários tipos de funções, foram determinadas o total de ocorrências para cada tipo de função, conforme as seguintes regras:

- Se 43 arquivos lógicos internos = 25% do total de funções, então total de funções = 172

- Se arquivos de interfaces externas = 3% do total de funções, então arquivos de interfaces externas = 5,16 (arredonda-se para 5)

- Se entradas externas = 30% do total de funções, então entradas externas = 51,6 (arredonda-se para 52)

- Se saídas externas = 28% do total de funções, então saídas externas = 48,16 (arredonda-se para 48)

- Se consultas externas = 14% do total de funções, então consultas externas = 24,08 (arredonda-se para 24)

O número de arquivos lógicos internos (43) informado como valor de entrada para a determinação da quantidade dos demais tipos de funções foi obtido após a elaboração do modelo preliminar de dados do Sistema Auditor, apresentado no Anexo 1 deste documento.

PASSO 4 – DETERMINAR OS PONTOS DE FUNÇÃO NÃO AJUSTADOS

Para determinar a classificação dos tipos de função deve ser considerada sempre uma complexidade funcional média para cada tipo de função. Dessa forma, o total de pontos de função não ajustados foi obtido conforme mostra a Tabela 7.1

FUNÇÃO	QUANTIDADE	COMPLEXIDADE	PESO	TOTAL
Arquivos Lógicos Internos	43	MÉDIA	10	430
Arquivos de Interfaces Externas	5	MÉDIA	7	35
Entradas Externas	52	MÉDIA	4	208
Saídas Externas	48	MÉDIA	5	240
Consultas Externas	24	MÉDIA	4	96
TOTAL DE PONTOS DE FUNÇÃO NÃO AJUSTADOS				1009

Tabela 7.1 – Determinação dos Pontos de Função Não Ajustados na FPA Estimativa

PASSO 5 – DETERMINAR OS PONTOS DE FUNÇÃO AJUSTADOS

Para determinar o total de pontos de função do Sistema Auditor foi considerado o valor do fator de ajuste igual a 1 e aplicado a fórmula abaixo:

Pontos de Função Ajustados = Fator de Ajuste * Pontos de Função Não Ajustados

Pontos de Função Ajustados = 1 * 1009 = 1009

7.3. CÁLCULO DE PONTOS DE FUNÇÃO – FPA DETALHADA

Levando em consideração os requisitos e funcionalidades do sistema e realizando um estudo detalhado do Sistema Auditor existente, foi possível aplicar a técnica FPA Detalhada ,conforme os passos descritos a seguir.

PASSO 1 - DETERMINAR O TIPO DE CONTAGEM

O Sistema Auditor é uma aplicação existente que atualmente se encontra em fase final de testes.

PASSO 2 - IDENTIFICAR AS FRONTEIRAS DA CONTAGEM

As funcionalidades e requisitos do Sistema que delimitam a fronteira da aplicação foram descritos no item 7.1.

PASSO 3 - IDENTIFICAR E CLASSIFICAR AS FUNÇÕES DO SISTEMA

Nesta etapa, as funções foram identificadas conforme os seguintes tipos: Arquivos Lógicos Internos, Arquivos de Interface Externa, Entradas Externas, Saídas Externas e Consultas Externas. Em seguida, cada uma delas foi classificada de acordo com o nível de complexidade, podendo ser: Simples, Médio e Complexo.

As tabelas a seguir apresentam os resultados encontrados.

Arquivos de Interface Externa					
Descrição da Função	Registros	Itens	S	M	C
Lançamentos das Vendas	1	13	X		
Parcelas das Vendas	1	3	X		
Total de Arquivos de Interface Externa = 2			2		

Tabela 7.2 – Identificação e Classificação de Arquivos de Interface Externa

Arquivos Lógicos Internos					
Descrição da Função	Registros	Itens	S	M	C
Unidades de Federação	1	2	X		
Municípios	1	3	X		
Empresas e Endereços	2	25		X	
Cartão de Crédito e Datas de Vencimento	2	8	X		
Plano de Contas e Atributos	2	10	X		
Plano de Contas da Empresa	1	36	X		
Centros de Custo e Atributos	2	10	X		
Centros de Custo da Empresa	1	10	X		
Atributos	1	2	X		
Índices de Rateio por Centros de Custo	2	5	X		
Tipos de Documento	1	3	X		
Históricos Padronizados	1	2	X		
Solicitações	6	20			X
Situação dos Lançamentos	1	2	X		
Partidas	1	15	X		
Itens Solicitados	1	6	X		
Unidades de Medida	1	2	X		
Produtos e Serviços	1	3	X		
Fontes	1	3	X		
Cotações	1	6	X		
Entidades e Endereços	2	24		X	
Itens Cotados	1	5	X		
Parcelamentos	1	3	X		
Lançamentos Contas a Pagar	1	20	X		
Lançamento Contábil Contas a Pagar	2	24		X	
Notas Fiscais Contas a Pagar	1	15	X		
Balancetes Contas a Pagar	1	20	X		
Parcelas Contas a Pagar	1	5	X		

Arquivos Lógicos Internos (continuação)					
Descrição da Função	Registros	Itens	S	M	C
Baixas Contas a Pagar	2	26		X	
Lançamentos Contas a Receber	1	13	X		
Lançamento Contábil Contas a Receber	2	23		X	
Notas Fiscais Contas a Receber	1	15	X		
Balancetes Contas a Receber	1	20	X		
Parcelas Contas a Receber	1	3	X		
Baixas Contas a Receber	2	26		X	
Caixa/Banco e Movimentos	2	22		X	
Controle de Cheques	1	6	X		
Dotação Orçamentária (Caixa e Competência)	2	20		X	
Parâmetros	1	5	X		
Usuários	1	5	X		
Grupos	1	2	X		
Elementos de Segurança	1	2	X		
Elementos dos Grupos	1	2	X		
Total de Arquivos Lógicos Internos = 43			34	8	1

Tabela 7.3 – Identificação e Classificação de Arquivos Lógicos Internos

Para a identificação e classificação das entradas externas, foi utilizado o conceito de manutenção de um arquivo, que engloba as atividades de inclusão, alteração e exclusão. Portanto, o total de entradas externas corresponde ao total de funções identificadas multiplicado por 3.

Entradas Externas					
Descrição da Função	Arquivos	Itens	S	M	C
Manutenção de Unidades de Federação	1	3	X		
Manutenção de Municípios	1	4	X		
Manutenção de Empresas e Endereços	1	28		X	
Manutenção de Plano de Contas da Empresa	1	38		X	
Manutenção de Centros de Custo da Empresa	1	16		X	
Manutenção de Cartão de Crédito e Vencimentos	1	9	X		
Manutenção de Plano de Contas	1	11	X		
Manutenção de Atributos do Plano de Contas	1	4	X		
Manutenção de Centros de Custo	1	9	X		
Manutenção de Atributos dos Centros de Custos	1	4	X		
Manutenção de Atributos	1	3	X		
Manutenção de Índices de Rateio	1	4	X		
Manutenção de Índices de Rateio dos CC	1	3	X		
Manutenção de Tipos de Documento	1	3	X		
Manutenção de Históricos Padronizados	1	3	X		
Manutenção de Solicitações	1	20		X	
Manutenção de Situação dos Lançamentos	1	3	X		
Manutenção de Partidas	1	9	X		
Manutenção de Itens Solicitados	1	6	X		
Manutenção de Unidades de Medida	1	3	X		
Manutenção de Produtos e Serviços	1	4	X		
Manutenção de Fontes	1	4	X		
Manutenção de Autorizações de Solicitações	1	16		X	
Manutenção de Cotações	2	17			X
Manutenção de Itens Cotados	1	6	X		
Manutenção de Entidades e Endereços	1	25		X	
Manutenção de Parcelamentos	1	3	X		
Manutenção de Autorizações de Compras	1	16		X	
Manutenção de Empenhos	2	20			X

Entradas Externas (continuação)					
Descrição da Função	Arquivos	Itens	S	M	C
Manutenção de Liquidações	4	30			X
Manutenção de Lançamentos Contas a Pagar	1	16		X	
Manutenção de Lançamento Contábil Contas a Pagar	2	16			X
Manutenção de Notas Fiscais Contas a Pagar	1	16		X	
Manutenção de Parcelas Contas a Pagar	1	5	X		
Manutenção de Baixas Contas a Pagar	5	30			X
Manutenção de Acréscimo/Desconto Contas a Pagar	1	5	X		
Manutenção de Transferência Bancária	7	32			X
Manutenção de Cheques	1	8	X		
Manutenção de Empenho Rápido	8	35			X
Manutenção de Lançamentos Contas a Receber	1	16		X	
Manutenção Lançamento Contábil Contas a Receber	1	16		X	
Manutenção de Notas Fiscais Contas a Receber	1	16		X	
Manutenção de Parcelas Contas a Receber	1	3	X		
Manutenção de Baixas Contas a Receber	5	30			X
Manutenção Acréscimo/Desconto Contas a Receber	1	5	X		
Manutenção de Importação de Vendas	3	26			X
Manutenção de Caixa/Banco	2	16			X
Manutenção de Orçamento Caixa	1	16		X	
Manutenção de Orçamento Competência	1	16		X	
Manutenção de Parâmetros	1	6	X		
Manutenção de Usuários	1	6	X		
Manutenção de Grupos de Usuários	1	3	X		
Manutenção de Elementos de Segurança	1	3	X		
Manutenção de Elementos dos Grupos	1	3	X		
Total Parcial = 54			30	14	10
Total de Entradas Externas = 54 * 3 = 162			90	42	30

Tabela 7.4 – Identificação e Classificação das Entradas Externas

Saídas Externas					
Descrição da Função	Arquivos	Itens	S	M	C
Relatório de Títulos por Data de Emissão	2	20			X
Relatório de Baixas	2	22			X
Relatório de Contas a Receber por Data de Lançamento	2	15		X	
Relatório de Contas a Receber por Data de Vencimento	2	15		X	
Relatório de Movimento Financeiro	2	20			X
Relatório de Razão	4	21			X
Relatório de Razão Individualizado	4	21			X
Relatório de Balancete do Plano de Contas	3	11		X	
Relatório de Balancete dos Centros de Custo	3	11		X	
Relatório de Balancete Individualizado	3	21			X
Relatório de Rateio	3	15		X	
Relatório de Lançamentos	4	18			X
Relatório Diário	2	6		X	
Relatório de Extrato dos Bancos	2	10		X	
Relatório de Vencimentos por Data	4	12			X
Relatório de Vencimentos por Credor	4	12			X
Relatório de Ficha do Credor	4	16			X
Relatório de Fluxo de Caixa	1	30		X	
Relatório de Lista de Planos de Conta	1	6	X		
Relatório de Lista de Centros de Custo	1	5	X		
Relatório de Lista de Históricos Padronizados	1	3	X		
Consulta de Saldo	2	6		X	
Consulta de Movimento Bancário	2	15		X	
Total de Saídas Externas = 23			3	10	10

Tabela 7.5 – Identificação e Classificação das Saídas Externas

Consultas Externas					
Descrição das Funções	Arquivos	Itens	S	M	C
Consulta de Municípios	1	3	X		
Consulta de Empresas	1	15	X		
Consulta de Usuários	1	5	X		
Consulta de Centros de Custo	1	10	X		
Consulta de Planos de Conta	1	10	X		
Consulta de Produtos e Serviços	1	3	X		
Consulta de Entidades	1	10	X		
Consulta de Caixa/Banco	1	20		X	
Consulta de Históricos Padronizados	1	2	X		
Consulta de Tipos de Documento	1	2	X		
Consulta de Fontes	1	2	X		
Consulta de Lançamentos	10	18			X
Consulta de Lançamentos por Planos de Contas	2	20			X
Consulta de Lançamentos por Vencimento	2	22			X
Consulta de Solicitações	2	15		X	
Consulta de Autorizações de Solicitações	2	15		X	
Consulta de Tomada de Preços	2	20			X
Consulta de Solicitações	2	15		X	
Consulta de Autorizações de Compras	2	20		X	
Consulta de Compras Executadas	2	15		X	
Consulta de Compras Liquidadas	2	8		X	
Consulta de Compras Pagas	2	7		X	
Consulta de Duplicatas a Pagar	2	6		X	
Consultas de Parcelas	2	11		X	
Consulta de Pagamentos e Baixas Realizadas	2	6		X	
Consulta de Cheques	2	6		X	
Consulta de Vendas Realizadas	2	20			X
Total de Consultas Externas = 27			10	12	5

Tabela 7.6 – Identificação e Classificação das Consultas Externas

PASSO 4 - CALCULAR OS PONTOS DE FUNÇÃO NÃO AJUSTADOS

Após a identificação e classificação das funções, foi aplicada a Tabela de Transformação para determinar o total de pontos de função não ajustados, conforme mostra a Tabela 7.7.

FUNÇÃO	OCORRÊNCIAS	COMPLEXIDADE	PESO	RESULTADO
ENTRADAS EXTERNAS	90	BAIXA	3	270
	42	MÉDIA	4	168
	30	ALTA	6	180
			TOTAL 1	618
SAÍDAS EXTERNAS	3	BAIXA	4	12
	10	MÉDIA	5	50
	10	ALTA	7	70
			TOTAL 2	132
ARQUIVOS LÓGICOS INTERNOS	34	BAIXA	7	238
	8	MÉDIA	10	80
	1	ALTA	15	15
			TOTAL 3	333
ARQUIVOS DE INTERFACES EXTERNAS	2	BAIXA	5	10
	0	MÉDIA	7	0
	0	ALTA	10	0
			TOTAL 4	10
CONSULTAS EXTERNAS	10	BAIXA	3	30
	12	MÉDIA	4	48
	5	ALTA	6	30
			TOTAL 5	108
TOTAL DE PONTOS DE FUNÇÃO NÃO AJUSTADOS				1201

Tabela 7.7 - Determinação dos Pontos de Função Não Ajustados na FPA Detalhada

PASSO 5 - DETERMINAR O FATOR DE AJUSTE

Para determinar o fator de ajuste, que representa a funcionalidade geral que a aplicação proporciona ao usuário, foi avaliado o nível de influência de 14 características gerais dos sistemas, conforme mostra a Tabela 7.8.

Características Gerais dos Sistemas	Nível de Influência
01. Comunicação de Dados	4
02. Processamento Distribuído	0
03. Desempenho	1
04. Utilização do Equipamento	2
05. Volume de Transações	1
06. Entrada de Dados On-line	5
07. Eficiência do Usuário Final	5
08. Atualização On-line	5
09. Processamento Complexo	3
10. Reutilização de Código	2
11. Facilidade de Implantação	3
12. Facilidade Operacional	1
13. Múltiplos Locais	4
14. Facilidade de Mudanças	2
Nível de Influência Total	38

Tabela 7.8 – Características Gerais dos Sistemas e Nível de Influência

Fator de Ajuste = 0,65 + (0,01 * Nível de Influência Total) = 1,03

PASSO 6 - DETERMINAR OS PONTOS DE FUNÇÃO AJUSTADOS

Finalmente, o total de pontos de função do Sistema Auditor utilizando a técnica FPA detalhada foi determinado, conforme a fórmula abaixo:

Pontos de Função Ajustados = Fator de Ajuste * Pontos de Função Não Ajustados

Pontos de Função Ajustados = 1,03 * 1201 = 1237

7.4. CONCLUSÕES

Neste capítulo foi apresentada uma experiência de aquisição de conhecimento e aplicação prática da técnica de Análise de Pontos de Função, de acordo com as regras de contagem do IFPUG, em um determinado sistema de uma empresa de informática.

Este estudo de caso permitiu concluir algumas questões importantes:

– a FPA Estimativa foi aplicada na fase inicial deste trabalho prático e baseada somente nas informações fornecidas pelo usuário e no modelo preliminar de dados. Nesta fase, o nível de conhecimento sobre o Sistema era baixo e a margem de erro prevista é de +/- 25%;

  Após a instalação de uma versão do Sistema e um estudo detalhado, foi aplicada a FPA Detalhada. A diferença de tamanho do Sistema encontrada nos resultados de cada uma das metodologias foi de 228 pontos de função a mais na FPA Detalhada;

  Isto representou uma margem de erro de -18,43% na aplicação da FPA Estimativa. Com esse percentual, foi possível concluir que a utilização da técnica FPA Estimativa é totalmente viável nas fases iniciais do projeto, pois a margem de erro apresentada ficou abaixo do percentual previsto, conforme a Tabela 6.7;

- apesar da FPA Detalhada ser mais precisa que a FPA Estimativa, o esforço requerido para a contagem de pontos de função na metodologia detalhada é notoriamente mais elevado. Dependendo da amplitude do sistema, essa contagem pode requerer um tempo adicional sensível, devendo portanto ser considerada no cronograma do projeto;

- como foi comprovada neste trabalho, uma das grandes vantagens da FPA é a independência de tecnologia, pois para a contagem de pontos de função, em nenhum momento foi necessário o conhecimento prévio das ferramentas utilizadas para o desenvolvimento do Sistema Auditor;

- um outro aspecto importante concluído nesta experiência prática é com relação ao fator humano. A implantação de um programa de mensuração em uma empresa requer atenção especial no que diz respeito ao fator humano e, para o seu sucesso é imprescindível o envolvimento efetivo das pessoas e a divulgação clara dos objetivos da mensuração e métodos utilizados.

8. CONCLUSÃO

No Brasil, embora muitas empresas reconheçam a importância da qualidade de software, a grande maioria ainda não conhece ou não utiliza normas ou modelos de qualidade. Umas das razões é que a implantação de um Sistema da Qualidade não é trivial e muitos esforços são necessários.

Para a implantação de um Sistema da Qualidade, um ponto de extrema importância são as atividades de mensuração. Como foi possível concluir neste trabalho, os programas de métricas não devem ser implementados por conformismo – quando demandado como parte de certificações, e nem por pragmatismo – quando usados para resolver problemas imediatos apenas, mas devem sim ser embutidos na cultura organizacional, definidos conforme as necessidades e percebidos como a chave para um gerenciamento cada vez melhor nas empresas.

Outra questão conclusiva é que para o sucesso da implantação de um programa de mensuração, é imprescindível que todas as pessoas sejam plenamente informadas e envolvidas no processo, evitando assim problemas com relação ao fator humano. Esta conclusão vem em encontro com a Teoria do Saber Profundo de Deming, apresentada no Capítulo 2.

Dentro de um programa de mensuração, um tópico extremamente relevante é a determinação de estimativas nas fases iniciais do projeto. A freqüente determinação de estimativas nas empresas baseadas no “feeling” e nas experiências pessoais, muitas vezes gera problemas sérios e leva projetos importantes ao fracasso.

Estes fatores estimularam a realização de um estudo detalhado e aplicação prática da técnica FPA – Análise de Pontos de Função, que vem sendo utilizada nas empresas para dimensionamento do tamanho do software. Nos capítulos 6 e 7 foram apresentados os conceitos e um estudo de caso de aplicação das metodologias FPA Detalhada e Estimativa em um determinado sistema de uma empresa.

As seguintes conclusões e sugestões de melhoria foram obtidas neste caso prático:

- devido ao esforço requerido para a contagem dos pontos de função em projetos com um elevado nível de complexidade, pode ser mais conveniente aplicar a FPA Estimativa na fase de planejamento desses projetos, enquanto que, a FPA Detalhada se torna mais viável para as fases de manutenção, melhoria ou projetos concluídos com o intuito de criar uma base histórica;

- a aplicação da FPA Estimativa nas fases iniciais é viável uma vez que foi possível comprovar neste trabalho de pesquisa que a margem de erro é inferior a 25%;

- a partir da informação de tamanho do sistema apresentada neste trabalho e obtida na fase final do projeto, a empresa tem condições de derivar as demais métricas pertinentes ao projeto. Em seguida, essas métricas poderão ser utilizadas para estimar novos projetos com características semelhantes;

Como resultado final deste trabalho, foi possível concluir que a técnica FPA permite a determinação de estimativas de projeto com um grau de precisão admissível, evitando assim a aplicação de mensurações baseadas em aspectos pessoais. Com isso, é possível elaborar cronogramas mais próximos da realidade, impactando de forma positiva no processo de desenvolvimento e melhoria da qualidade dos produtos de software.

8.1. RECOMENDAÇÕES PARA TRABALHOS FUTUROS

As seguintes propostas de continuação deste trabalho poderão ser consideradas:

- a técnica FPA poderia ser aplicada em empresas diferentes e que possuem sistemas semelhantes em termos de funcionalidades, mas que foram desenvolvidos utilizando ferramentas diversas. Analisando apenas as funcionalidades comuns, a aplicação da FPA deve resultar em tamanhos equivalentes, uma vez que a técnica é independente da tecnologia utilizada;

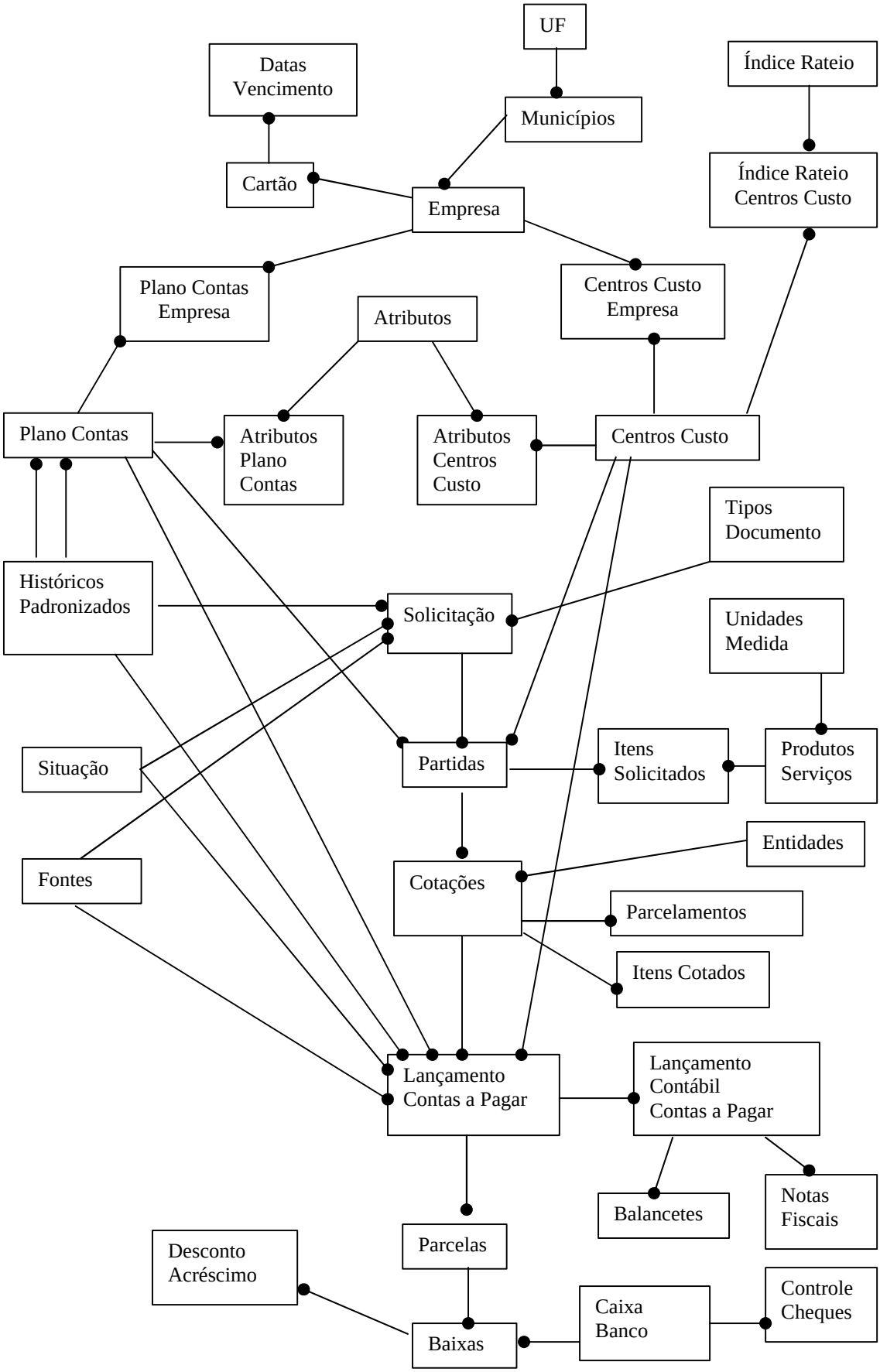
- as variações da técnica FPA apresentadas no Capítulo 5 poderiam merecer um estudo específico e comparativo. As variações da técnica FPA foram desenvolvidas a partir de necessidades específicas de alguns tipos de software, como por exemplo, software básico, CAD, sistemas em tempo real e outros.

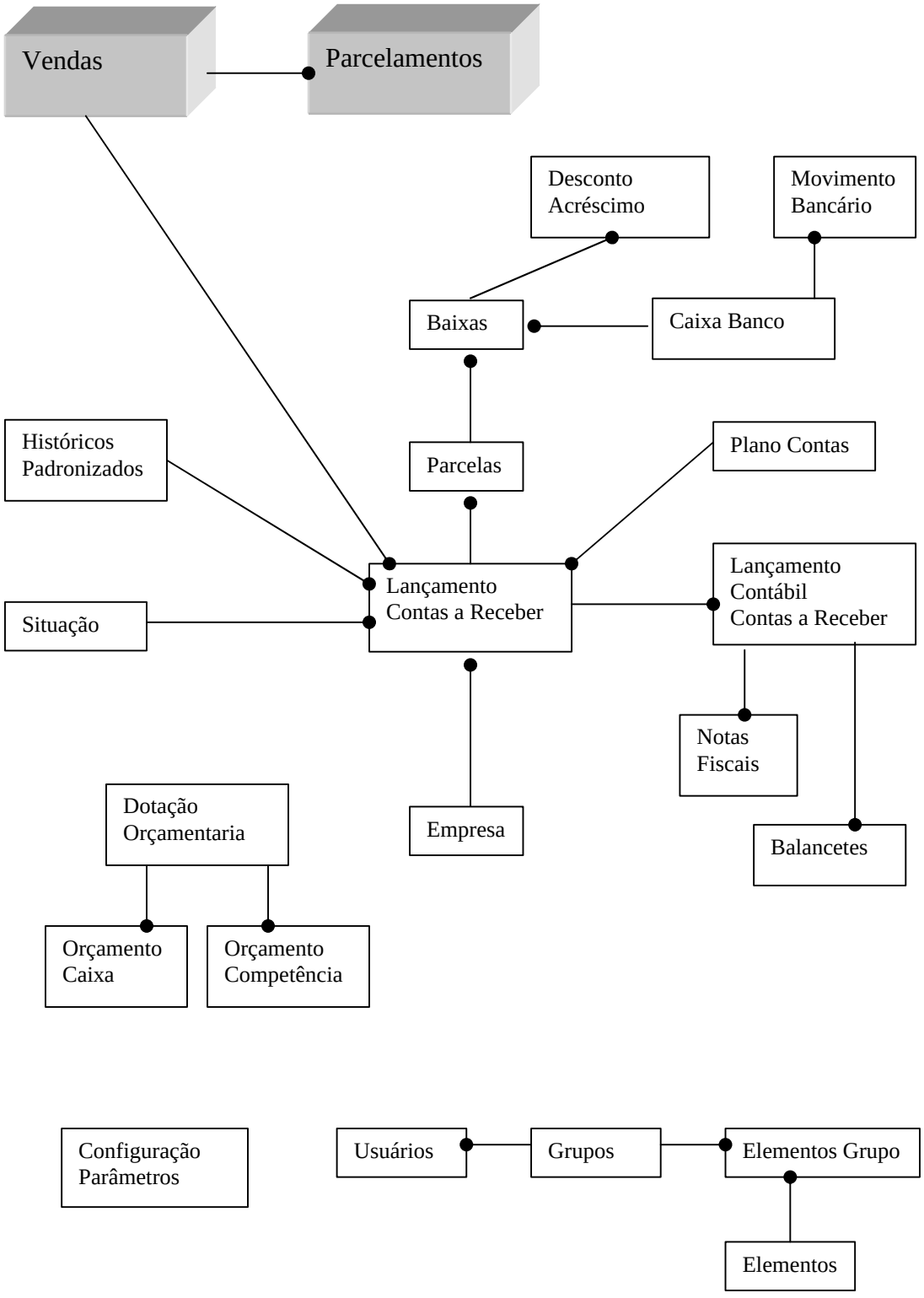
- entre as variações da FPA, as técnicas Feature Point e COSMIC Full Function Points vêm recebendo destaque nas pesquisas e poderiam ser estudadas em maiores detalhes, inclusive aplicadas em casos práticos na área de software básico;

- uma das maiores dificuldades encontradas no início da implantação de um programa de mensuração é a definição de quais métricas são importantes e úteis em uma empresa. Como foi mostrado no Capítulo 5, o modelo GQM foi desenvolvido para este fim, podendo ser alvo de estudo e utilizado em uma empresa para a definição das métricas relevantes.

- para a implantação de um Sistema da Qualidade, as empresas necessitam buscar informações acerca das normas e modelos existentes, para escolher aquela que mais se adapte às suas necessidades e situação atual. Um estudo comparativo entre essas abordagens visando auxiliar as empresas na tomada de decisão seria bastante útil e interessante.

ANEXO 1 – MODELO PRELIMINAR DE DADOS


UF
Datas Vencimento
Índice Rateio
Municípios
Índice Rateio Centros Custo
Cartão
Empresa
Plano Contas Empresa
Centros Custo Empresa
Atributos
Plano Contas
Atributos Plano Contas
Atributos Centros Custo
Centros Custo
Tipos Documento
Históricos Padronizados
Solicitação
Unidades Medida
Situação
Partidas
Itens Solicitados
Produtos Serviços
Entidades
Fontes
Cotações
Parcelamentos
Itens Cotados
Lançamento Contas a Pagar
Lançamento Contábil Contas a Pagar
Balancetes
Notas Fiscais
Parcelas
Desconto Acréscimo
Caixa Banco
Controle Cheques
Baixas


Vendas
Parcelamentos
Desconto
Acréscimo
Movimento
Bancário
Baixas
Caixa Banco
Históricos
Padronizados
Parcelas
Plano Contas
Lançamento
Contas a Receber
Lançamento
Contábil
Contas a Receber
Situação
Notas
Fiscais
Dotação
Orçamentaria
Empresa
Balancetes
Orçamento
Caixa
Orçamento
Competência
Configuração
Parâmetros
Usuários
Grupos
Elementos Grupo
Elementos

REFERÊNCIAS BIBLIOGRÁFICAS

[AGUIAR 02] AGUIAR, M.. **Practical Software Measurement: O CMM da Mensuração ?** Axcel, 2002.

[ALLEN 01] ALLEN, E. B.; KHOSHGOFTAAR, T. M.; JONES, W. D.; HUDEPOHL, J. P.. Cost-benefit Analysis of Software Quality Models. **Software Quality Journal**, 2001.

[AZEVEDO 00] AZEVEDO, D. J. P.. **Curso de FPA – Function Point Analysis**. São Paulo: 2000.

[BASILI 92] BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D.. **The Experience Factory.** 1992.

[BASILI 94] BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D.. **The Goal Question Metric Approach**. 1994.

[BELLOQUIM 02] BELLOQUIM, A.. Qualidade de Software: O que Há de Novo no Mercado? **Developers' Magazine**, Abril 2002.

[BELCHIOR 01] BELCHIOR, A. D.; FARIAS, P. P. M.; LIMA, O. S.. **Modelo Fuzzy para Extensão da Análise de Pontos de Função.** XII CITS, Curitiba: 2001.

[BLASCHEK 01] BLASCHEK, J. R.; SANTOS, J.; MICCOLIS, A. G.; COSTA, M. L.; SANTOS, F. C.. **Implantação da Métrica Análise de Pontos de Função segundo uma abordagem de Aprendizagem Organizacional.** XII CITS, Curitiba: 2001.

[BOEHM 96] BOEHM, B.. Anchoring The Software Process. **IEEE Transactions On Computers**, July 1996.

[BURNETT 00] BURNETT, R.; VALLE, A.; MARCINIUK, M.; MELHORETTO, S.. Um Roadmap para Métricas de Software: Definições e Histórico. **Developers' Magazine**, n.49, Setembro 2000.

[BURNETT 01] BURNETT, R.; VALLE, A.; MARCINIUK, M.; MELHORETTO, S.. **O Papel da Mensuração no Gerenciamento de Processos.** XII CITS, Curitiba: 2001.

[DEKKERS 98] DEKKERS, C. A .. **Pontos de Função e Medidas. O que é um Ponto de Função ?** Quality Plus Technologies Inc., Dezembro 1998.

[DEEKERS 98a] DEEKERS, C. A.. **Desmistificando Pontos de Função: Entendendo a Terminologia.** Quality Plus Technologies Inc., Outubro 1998.

[DYMOND 98] DYMOND, K. M.. **A Guide to CMM.** USA: Process Transaction International, 1998.

[EBERT 92] EBERT, C.; LIEDTKE, T.. **Software Realiability Prediction for Large Systems.** Alcatel SELAG, Stuttgart: 1992.

[FENTON 99] FENTON, N. & NEIL, M.. **Software Metrics: Roadmap**. 1999.

[FERNANDES 95] FERNANDES, A. A.. **Gerência Efetiva de Software através de Métricas.** São Paulo: Atlas, 1995.

[FROSSARD 00] FROSSARD, R. S.. **PSP – Qualidade e Produtividade no Desenvolvimento de Software.** XI CITS, Curitiba: Junho 2000.

[GIBBS 94] GIBBS, W. W.. **Software's chronic Crisis.** Scientific American, 1994.

[GUERRA 02] GUERRA, A. C.; SANTANA M. L.. Qualidade de Produtos ou Qualidade de Processo de Software ? **Developers' Magazine**, Abril 2002.

[GRESSE 98] GRESSE, C.; ROMBACH, H. D.; RUHE, G.. **Melhoramento da Qualidade de Software Baseado em Mensuração: Como estabelecer um Programa de Mensuração na Prática ?** IX CITS, Curitiba: 1998.

[HASAN 99] HASAN, C.. **Introdução da Gerência Pela Qualidade Total em Organizações de Desenvolvimento de Software**. 1999.

[HELLER 02] HELLER, R.. **An Introduction to Function Point Analysis.** Q/P Management Group Inc., 2002.

[HERRON 00] HERRON, D.; GARMUS, D.. **Function Point Analysis: Measurement Practices for Sucessful Software Projects.** Addison Wesley, 2000.

[HOLMES 02] HOLMES, L.. **Evaluating COTS Using Function Fit Analysis.** Q/P Management Group Inc., 2002.

[HOWER 99] HOWER, R.. **Software QA And Testing Frequently-Asked-Questions.** Software QA/Test Resource Center, December 1999.

[HUMPHREY 99] HUMPHREY, W.S.. **Introduction to The Personal Software Process.** Addison-Wesley, January 1999.

[INTHURN 01] INTHURN, C.. **Qualidade & Teste de Software.** Visual Books, 2001.

[IFPUG 00] IFPUG. **International Function Point Users Group Frequently Asked Questions**. July 2000.

[ISBSG 02] ISBSG. **ISO Standard for Functional Size Measurement**. June 2002.

[KITCHENHAM 96] KITCHENHAM, B.; PFLEEGER, S. L.. Software Quality: The Elusive Target. **IEEE Transactions On Computers**, 1996.

[LIPTON 02] LIPTON, D.. **Function Points and the SEI Capability Maturity Model.** Q/P Management Group Inc., 2002.

[LISTER 98] LISTER, T.. Defining Quality Intuitively. **IEEE Transactions On Computers,** 1998.

[LONGSTREET 00] LONGSTREET, D.. **The Utility of Function Points**. Longstreet Consulting Inc., 2000.

[LONGSTREET 02] LONGSTREET, D.. **Function Point Analysis Training Course.** Longstreet Consulting Inc., March 2002.

[LONGSTREET 02a] LONGSTREET, D.. **Fundamentals of Function Point Analysis.** Longstreet Consulting Inc., March 2002.

[MAUTH 96] MAUTH, R.. Software Bugs. **Byte Magazine,** July 1996.

[MOLINARI 02] MOLINARI, L.. Os Principais Pontos da Nova Qualidade de Software. **Developers' Magazine**, Abril 2002.

[PAULK 96] PAULK, M.C.. **Questions and Answers on The CMM.** March 1996.

[PAULK 98] PAULK, M.C.. **The Capability Maturity Model for Software.** Version l.l, 1998.

[PAULK 99] PAULK, M.C.. **Using the Software CMM in Small Organizations.** 1999.

[PARIKH 90] PARIKH, G.. **Técnicas de Manutenção de Programas e Sistemas.** Rio de Janeiro: Livros Técnicos e Científicos Ed., 1990.

[PRESSMAN 95] PRESSMAN, R. S.. **Engenharia de Software.** São Paulo: Makron Books, 1995.

[RAGLAND 95] RAGLAND, B.. **Measure, Metric or Indicator: What's the difference.** Cross Talk, 1995.

[ROCHA 01] ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C.. **Qualidade de Software: Teoria e Prática.** São Paulo: Prentice Hall, 2001.

[ROSSI 99] ROSSI, R..F. R.. **A verdadeira crise do software.** May 1999.

[RULE 01] RULE, G.. **The importance of the Size of Software Requirements.** In: Nasscom Conference. India: February 2001.

[SCHMAUCH 96] SCHMAUCH, C.H.. ISO 9000 For Software. **Developers' Magazine**, 1996.

[SIMOES 99] SIMOES, C. A.. Sistemática de Métricas, Qualidade e Produtividade. **Develpers Magazine**, 1999.

[SYMONS 01] SYMONS, C.. **Come Back Function Point Analysis (modernised) - All is Forgiven** ! 2001.

[WEBER 97] WEBER, K. C.; ROCHA, A. R. C.; LUCA, J. C. M.. **Qualidade e Produtividade em Software.** 2.ed São Paulo: Makron Books, 1997.

[UKSMA 98] UKSMA. **MK II Function Point Analysis Counting Practices Manual.** Version 1.3.1, United Kingdom: United Kingdom Software Metrics Association, September 1998.

[ZUSE 95] ZUSE, H.. **Foundations of Software Measurement**. August 1995.