



**Faculdade Hélio Rocha
Sistemas de Informação**

Manuela Dayse Fadigas Santos
Alexsandra Guimarães Moreira

**ANÁLISE DE PONTOS POR FUNÇÃO (APF): NÍVEL DE UTILIZAÇÃO
NAS EMPRESAS QUE PRODUZEM SOFTWARE, LOCALIZADAS EM
SALVADOR/BAHIA.**

Salvador
2006

Manuela Dayse Fadigas Santos
Alexsandra Guimarães Moreira

**ANÁLISE DE PONTOS POR FUNÇÃO (APF): NÍVEL DE UTILIZAÇÃO
NAS EMPRESAS QUE PRODUZEM SOFTWARE, LOCALIZADAS EM
SALVADOR/BAHIA.**

Monografia apresentada para a Faculdade Hélio Rocha, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Ernesto Massa

Salvador
2006

M56c Santos, Manuela Dayse Fadigas e Moreira, Alexsandra Guimarães Moreira.
Análise de Pontos por Função (APF): Nível de utilização nas empresas que produzem software, localizadas em Salvador/Bahia. – Salvador 2006.
113 f.: il.

Orientador: Prof. Ernesto Massa
Monografia de conclusão do curso de (Graduação) - Faculdade Helio Rocha, 2006.

2. Engenharia de Software. 3. Métricas de Estimativas de Tamanho de Software 4. Análise de Pontos por Função. I. Massa, Ernesto. II. Faculdade Hélio Rocha. III. Análise de Pontos por Função (APF): Nível de utilização nas empresas que produzem software, localizadas em Salvador/Bahia. – Salvador 2006.

CDD 341.59

**MANUELA DAYSE FADIGAS SANTOS
ALEXSANDRA GUIMARÃES MOREIRA**

**ANÁLISE DE PONTOS POR FUNÇÃO (APF): NÍVEL DE UTILIZAÇÃO
NAS EMPRESAS QUE PRODUZEM SOFTWARE, LOCALIZADAS EM
SALVADOR/BAHIA.**

Monografia apresentada ao curso de Sistemas de Informação da Faculdade Hélio Rocha, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovada em 27 de dezembro de 2006.

**Prof. Ernesto Massa – Orientador
Faculdade Helio Rocha**

**Prof. Alano dos Santos Castro Filho
Faculdade Helio Rocha**

**Prof. Marcos A. da Silveira Prado
Faculdade Helio Rocha**

DEDICATÓRIA DE MANUELA FADIGAS

Dedico este trabalho aos meus pais Antônio Gutemberg de Macedo Santos e Lucia Maria Fadigas Santos e ao meu irmão Junior pelo apoio, carinho e incentivo em todos os momentos da vida.

DEDICATÓRIA DE ALEXSANDRA MOREIRA

Dedico este trabalho à minha Mãe, Antonia Maristela Guimarães Moreira, em memória de meu pai, Ademil Gama Moreira, e aos meus filhos, Dílan Moreira dos Santos e Erick Moreira dos Santos, os quais sempre me apoiaram no crescimento pessoal e profissional.

AGRADECIMENTOS DE MANUELA FADIGAS

A Deus por me proporcionar à vida, a sabedoria e a coragem.

Aos meus pais e a meu irmão, pelo incentivo, amor e compreensão.

A toda minha família, que sempre fez parte da minha vida.

A Alexsandra Moreira, colega de pesquisa, pelo aprendizado que obtivemos durante esse período.

Ao meu orientador, Professor Ernesto Massa, pela paciência, idéias e sugestões durante a realização deste trabalho.

Aos professores Alano Castro, Alba Scheible, Éldman Nunes e Luciano Borges que contribuíram com informações fundamentais para realização deste trabalho.

As empresas que cederam informações valiosas para realização desta pesquisa.

Aos professores da banca examinadora e aos demais professores da Faculdade Helio Rocha do curso de Sistemas de Informação, pelos ensinamentos durante os quatro anos de graduação.

Aos colegas de curso, em especial Sandra, Rita e Emanuelle pela troca de conhecimento, amizade, apoio e companheirismo.

Aos colegas do Colégio Antônio Vieira pela experiência adquirida, pela amizade e carinho.

Aos demais amigos aqui não citados, mas que certamente foram imprescindíveis para essa conquista.

AGRADECIMENTOS DE ALEXSANDRA MOREIRA

A Deus por me proporcionar à vida, a sabedoria, forças e coragem para realizar esta pesquisa.

Aos meus irmãos Ademil Gama Moreira Júnior e Luciana Guimarães Moreira, pelo apoio e incentivo.

Ao orientador, Prof. Ernesto Massa, pelos ensinamentos, orientações e principalmente pela paciência durante a realização deste trabalho.

Aos professores Alano Castro, Alba Scheible, Éldman Nunes e Luciano Borges que contribuíram com informações e sugestões para este trabalho.

Aos professores do curso de Sistemas de Informação pelos conhecimentos transmitidos ao longo destes quatro anos.

A colega de curso, Manuela Fadigas pela troca de conhecimento, apoio e companheirismo.

Aos outros participantes da pesquisa que contribuíram com relevantes informações.

Aos professores membros da banca examinadora, por prestigiarem este trabalho com sua presença, críticas e sugestões de melhoria.

E a todos que colaboraram de forma direta ou indireta na realização desta pesquisa.

“O telhado plano de qualquer construção casa que possua apenas uma calha para o escoamento da água serviria de medidor das diferentes formas de precipitação da chuva”

T. Harmer, 1764

RESUMO

Cada vez mais as empresas que produzem software estão preocupadas com a qualidade do produto e com o cumprimento dos prazos e orçamentos especificados no projeto de desenvolvimento de software. A técnica de Análise de Pontos por Função (APF) é utilizada para estimar o tamanho de projetos de desenvolvimento, de manutenção ou aplicações. O objetivo maior deste trabalho é medir o nível de utilização da Análise de Pontos por Função nas empresas que produzem software em Salvador/Ba. Serão abordados alguns aspectos referentes à importância da engenharia de software no processo de desenvolvimento, aos históricos das métricas de estimativas de tamanho de software, a Análise de Pontos por Função, mostrando que o conhecimento desses assuntos implica no maior entendimento sobre a utilização de estimativas de tamanho de software. Para isso envolveu uma vasta análise bibliográfica e uma pesquisa exploratória, com aplicação de um questionário, cujo resultado permitiu obter informações relevantes sobre o nível de utilização da Análise de Pontos por Função em Salvador/Bahia. Esta investigação científica justifica-se em face de proporcionar maiores conhecimentos sob a Análise de Pontos por Função.

Palavras chave: Engenharia de Software, Gerência de Projetos, Estimativas de Tamanho, Análise de Pontos por Função.

ABSTRACT

Each more time the companies who produce software are worried about the product quality and the fulfilment of the stated periods and budgets specified in the project of software development. The technique of Function Points Analysis (FPA) is used in the whole world to estimate the size of development projects, of maintenance or applications. The biggest objective of this work is to measure the level of use of the Function Points Analysis in the companies who produce software. In Salvador/Ba. Some referring aspects to the importance of the software engineering in the development process will be approached, to the descriptions of the metric of size estimates of software, the Function Points Analysis, showing that the knowledge of these subjects implies in the biggest agreement on the use of size estimates of software. For this it involved a vast bibliographical analysis and a exploratory research, with application of a questionnaire, whose result allowed to get excellent information about the level of use of the Function Points Analysis in Salvador/Bahia. This scientific inquiry is justified in face to provide greater knowledge under Function Points Analysis.

key Words: Software Engineering, Projects Management, Size Estimates, Function Points Analysis

LISTA DE GRÁFICOS

Gráfico 01 – Tipo de empresas	72
Gráfico 02 – Faturamento anual	73
Gráfico 03 – Quantidade de funcionários	73
Gráfico 04 – Classificação das empresas	74
Gráfico 05 – Ano de fundação das empresas	74
Gráfico 06 – Atividades da organização no tratamento de software	75
Gráfico 07 – Certificações e modelos de qualidade	76
Gráfico 08 – Métricas para estimativas de tamanho de software	77
Gráfico 09 – Métricas para estimar a manutenção do software	78
Gráfico 10 - Profissionais certificados em CFPS	78
Gráfico 11 - Profissionais em processo de certificação CFPS	79
Gráfico 12 - Tipos de projetos	80
Gráfico 13 – Pontos fortes da metodologia	81
Gráfico 14 – Pontos fracos da metodologia	82
Gráfico 15 – Utilização do fator de ajuste	83
Gráfico 16 – Motivos de não utilizar a APF	83

LISTA DE FIGURAS

Figura 1 – Fatores de qualidade de software de McCall	28
Figura 2 – Métricas diretas e indiretas do software.....	32
Figura 3 – Divisão das métricas em categorias.....	34
Figura 4 – Evolução da quantidade de CFPS brasileiros.....	40

LISTA DE TABELAS

Tabela 1 – Complexidade ALI	48
Tabela 2 – Pontos por Função ALI	48
Tabela 3 – Complexidade AIE	49
Tabela 4 – Pontos por Função AIE	49
Tabela 5 – Complexidade EE	51
Tabela 6 – Pontos por Função EE	51
Tabela 7 – Complexidade SE	52
Tabela 8 – Pontos por Função SE	53
Tabela 9 – Complexidade CE	54
Tabela 10 – Pontos por Função CE	54
Tabela 11 – Pontos por Função não-Ajustados	55
Tabela 12 – Níveis de Influência	55

LISTA DE SIGLAS E ABREVIATURAS

APF	Análise de Pontos por Função
ALI	Arquivo Lógico Interno
AIE	Arquivo de Interface Externa
BFPUG	<i>Brazilian Function Point Users Group</i>
CASE	<i>Computer-Aided Software Engineering</i>
CE	Consulta Externa
CFPS	<i>Certified Function Point Specialist</i>
CGS	Características gerais de sistema
CMMI	<i>Capability Maturity Model Integration</i>
CPM	<i>Counting Practices Manual</i>
EE	Entrada Externa
ES	Engenharia de Software
FFP	<i>Full Function Points</i>
IFPUG	<i>International Function Point Users Group</i>
LOC	Linhas de Código
NESMA	<i>Netherlands Software Metrics Users Association</i>
PAMR	Plano de Administração e Monitoramento dos Riscos
PERT	<i>Program Evaluation and Review Technique</i>
PF	Pontos por Função
PMI	<i>Project Management Institute</i>
SE	Saída Externa
SQA	Garantia de Qualidade de Software
VAF	<i>Value Adjustment Factor</i>
Arq.	Arquivo
Ref.	Referenciado

SUMÁRIO

1. INTRODUÇÃO	18
2. ENGENHARIA DE SOFTWARE	22
2.1. A Crise do Software e a Engenharia de Software	22
2.2. Gerenciamento de Projetos	23
2.3. Análise e Gestão de Riscos	24
2.4. Cronograma e Acompanhamento do Projeto	26
2.5. Garantia da Qualidade de Software	27
3. MÉTRICAS DE ESTIMATIVAS DE TAMANHO DE SOFTWARE	30
3.1. Medidas de Software	30
3.2. Métricas Funcionais	34
3.2.1. Linhas de Código (LOC)	36
3.2.2. IFPUG	37
3.2.3. BFPUG	39
3.2.4. NESMA	40
3.2.5. Mark II	41
3.2.6. UCP	41
3.3. Considerações Finais.....	42
4. APF (Análise de Pontos por Função)	43
4.1. Objetivos	43
4.2. Determinação do Tipo de Contagem	45
4.2.1. Tipo de Contagem	45
4.2.1.1. Projeto de Desenvolvimento	45
4.2.1.2. Projeto de Melhoria	46
4.2.1.3. Contagem de Aplicações	46
4.3. O Escopo da Contagem e a Fronteira da Aplicação	46
4.4. Funções do Tipo Dado	47
4.4.1. Arquivo Lógico Interno	47
4.4.2. Arquivo de Interface Externa	49

4.5. Funções do Tipo Transação	50
4.5.1. Entrada Externa	50
4.5.1.1. Contagem de Itens de Dados e Arquivos Referenciados	51
4.5.2. Saída Externa	51
4.5.2.1. Contagem de Itens de Dados e Arquivos Referenciados	52
4.5.3. Consulta Externa	53
4.5.3.1. Contagem de Itens de Dados e Arquivos Referenciados	54
4.6. Pontos Por Função Não-ajustados	54
4.7. Fator de Ajuste	55
4.7.1. As Características Gerais dos Sistemas (CGS)	56
4.8. Pontos Por Função Ajustados	58
4.8.1. Contagem de Projetos de Desenvolvimento	58
4.8.2. Contagem de Projetos de Melhoria	59
4.8.3. Contagem de Aplicações	60
4.8.3.1. Fórmula para o Estabelecimento do Tamanho Inicial	60
4.8.3.2. Fórmula para Atualizar a Contagem Após a Melhoria	61
5. METODOLOGIA DA PESQUISA	63
5.1. Contexto da Pesquisa	63
5.2. Instrumento da Pesquisa	65
5.3. Coleta de Dados	65
5.4. Tratamento de Dados	66
6. PESQUISA EXPLORATÓRIA	67
6.1. Caracterização das Organizações	67
6.2. Resultado da Pesquisa Exploratória	71
7. CONSIDERAÇÕES FINAIS	85
REFERÊNCIAS	87
ANEXO	90

1. INTRODUÇÃO

A ausência de estimativas de tamanho ou estimativas incorretas no projeto de desenvolvimento do produto de software, gera por consequência cronogramas atrasados e elevação no custo do projeto. Segundo Pressman (1995) as informações coletadas para o desenvolvimento de software são julgadas importantes para a definição das estimas do projeto. Sem as medições quantitativas, o gerente de projeto não conhece a dimensão do que está sendo produzido e por consequência, não realiza um boa gestão, por essa razão o software deve ser medido.

Um dos fatores poderosos para o sucesso da organização é investir na engenharia de software para a gerência de projetos, de forma que os resultados atingidos sejam qualitativos e quantitativos. As organizações precisam compreender que tratar a questão gerencial dos projetos é fundamental para o aumento da melhoria da qualidade no processo de desenvolvimento de software.

Alguns aspectos como, adoção de normas, modelos de maturidade de processos de software, guias a exemplo de: NBR ISSO/IEC 12.207 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT), 1998); NBR ISSO/IEC 10.006 (ABNT,2000); ISSO/IEC TR 15.504 (INTERNATIONAL STANDARD ORGANIZATION (ISO) ,1998); CMM (PAULK, 2000) E PMBOOK (2000), plano de projeto, métricas para estimativas de tamanho de software e métricas para estimativas de manutenção de software, melhoram a qualidade dos produtos de software, com os custos efetivos e cumprimento de cronogramas.

As organizações estão buscando métodos para dimensionar o tamanho de um projeto de software, de forma a definir esforço, custo e prazos necessários para o desenvolvimento do software, tendo em vista a facilidade de relacionamento com o cliente, o gerenciamento dos riscos do projeto e o controle do cronograma, beneficiando a gerência de projetos para manter a qualidade dos seus produtos.

O resultado é um melhor gerenciamento dos projetos, pois estimativas no início do projeto são diferenças que as organizações precisam lançar mão para poderem competir adequadamente no mercado. Andrade (2004) afirma que:

A precisão de estimativas de tamanho é dependente de informações que nem sempre estão disponíveis no início do projeto. No entanto, essas estimativas são necessárias no início do projeto para discussão de contratos ou determinação da viabilidade do projeto em termo da análise de custos e benefícios (ANDRADE, 2004, p.2).

Verifica-se então, a grande preocupação das organizações em utilizar métricas de estimativas de tamanho de software para auxiliar o gerenciamento dos projetos, e por outro lado, percebe-se também, que os clientes buscam adquirir um software de qualidade, no prazo e custo estabelecido.

A constatação dos fatos expostos anteriormente serve de estímulo para a construção deste trabalho monográfico que traz como problema central a seguinte questão: Qual é o nível de utilização da Análise de Pontos por Função (APF) nas empresas que produzem software em Salvador/Ba?

Na intenção de responder o questionamento acima, realizou-se uma pesquisa bibliográfica, com o destaque especial para os seguintes autores e institutos: Pressman, Peters, Pedrycz, Andrade, Hazan, Vasquez *et al*, IFPUG, BFPUG, dentre outros. Para permitir avaliar e responder o questionamento realizou-se ainda, uma pesquisa de campo quantitativa.

Verificou-se ainda a necessidade de estabelecer os seguintes objetivos específicos:

- Estudar a engenharia de software.
- Analisar as métricas de estimativas de tamanho de software.
- Entender a Análise de Pontos por Função e o grau de importância para o desenvolvimento de software.
- Avaliar as considerações resultantes de uma pesquisa quantitativa nas organizações que produzem software em Salvador/ Bahia.

Esta pesquisa busca contribuir para a produção de conhecimento sobre a quantificação de organizações que utilizam técnicas para estimar o tamanho do software, em especial a APF.

Esse tema justifica-se em face de estudar de forma aprofundada e atual, a questão de estimativas de tamanho de software, pois, na era de grandes transformações no desenvolvimento de software, é necessário que as organizações saibam utilizar a engenharia de software para gerenciar de forma precisa os seus projetos. Uma forma é utilizar a APF, uma ferramenta capaz de gerar estimativas de tamanho em todo o ciclo de desenvolvimento de software.

O trabalho, seguindo orientação contida na norma NBR6024, da Associação Brasileira de Normas Técnicas, apresentar-se-á sob forma de seções e subseções conforme abaixo:

Contempla-se no segundo capítulo, conceitos relacionados à Engenharia de Software. Para isso, serão abordados conceitos, características e aplicações de software, assim como, conceitos de gestão de projetos, enfatizando o planejamento do projeto, a análise e gestão dos riscos, cronogramação e acompanhamento, garantia da qualidade de software, gestão de configuração, engenharia de requisitos, modelagem e análise de sistemas.

O terceiro capítulo, será dedicado para as métricas de estimativas de tamanho de software, a qual permite no primeiro momento introduzir a seção proposta, bem como a definição de medidas de software e por fim as métricas mais utilizadas para estimar softwares.

O quarto capítulo é dedicado à Análise de Pontos por Função, retratando os objetivos, a determinação do tipo de contagem, o escopo da contagem e a fronteira da aplicação, as funções do tipo dado e do tipo transação, os pontos por função não ajustados, o fator de ajuste e os pontos por função ajustados.

O quinto capítulo é dedicado a exposição da metodologia da pesquisa, de forma a facilitar o entendimento acerca do tema proposto, bem como, contexto da pesquisa, os instrumentos de pesquisa, a coleta de dados e o tratamento dos dados.

O sexto capítulo caracteriza-se nas organizações que produzem software em Salvador/Bahia, como também aponta os resultados obtidos com a investigação realizada entre os gerentes de projetos.

O sétimo e último capítulo, relata as conclusões e recomendações adquiridas com este trabalho científico.

2. ENGENHARIA DE SOFTWARE

O objetivo deste capítulo é descrever os principais métodos, ferramentas e procedimentos na Engenharia de Software (ES), destacando os seus principais aspectos, conceitos e importância.

2.1. A Crise do Software e a Engenharia de Software

A engenharia de software surgiu a partir da crise do software ocorrida na década de 60 devido a inadequação do software, cronograma e custos imprecisos, insatisfação dos usuários e clientes, comunicação deficiente entre as partes envolvidas, dificuldade na manutenção, surgiu a Engenharia de Software, estabelecendo padrões, ferramentas e procedimentos para tornar o processo de desenvolvimento de software sistemático e controlado.

A primeira definição atribuída a Engenharia de Software foi dada por Fritz Bauer em 1969 *apud* Pressman (1995): “O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.”. Os principais objetivos da Engenharia de Software são: melhorar a qualidade, entregar o produto dentro dos prazos e custos estabelecidos e incrementar a produtividade do desenvolvimento.

De acordo com (PRESSMAN, 2001), não existe uma única forma para resolver o problema do software. Contudo, ao combinar métodos abrangentes para todas as fases do desenvolvimento do software, melhores ferramentas para automatizar esses métodos, blocos de códigos mais poderosos para a implementação do software, melhores técnicas para a garantia da qualidade e uma filosofia de gerenciamento, controle e administração, podemos alcançar uma disciplina para o desenvolvimento do software – disciplina esta chamada de Engenharia de Software.

Os métodos da Engenharia de Software detalham como fazer para construir o software, eles envolvem um conjunto de tarefas como: planejar e estimar o projeto, analisar requisitos de software e sistemas, projetar a estrutura de dados, arquitetura de programa e algoritmo de processamento, codificar, testar e dar manutenção (PRESSMAN, 2001).

As ferramentas da Engenharia de Software oferecem apoio automatizado ou semi-automatizado aos métodos, chamadas de engenharia de software auxiliada por computador (CASE - Computer-Aided Software Engineering). O CASE combina software, hardware e um banco de dados de engenharia de software (uma estrutura que contém informações importantes sobre análise, projeto, codificação e testes) (PRESSMAN, 2001).

Os procedimentos da Engenharia de Software formam o elo que mantém juntos os métodos e as ferramentas e possibilita a evolução do software de computador. Os procedimentos definem seqüência que os métodos serão aplicados, entrega de produtos (documentos, relatório, formulários, etc.), os controles que auxiliam e asseguram a qualidade do software e os marcos de referência que ajudam aos gerentes avaliar o progresso do projeto (PRESSMAN, 2001).

2.2. Gerenciamento de Projetos

“O tempo é o mais valioso bem disponível a um engenheiro de software. Se houver tempo suficiente, um problema pode ser adequadamente analisado, uma solução pode ser compreensivelmente projetada, o código-fonte, cuidadosamente implementado e o programa, minuciosamente testado. Mas parece que nunca há tempo o bastante (PRESSMAN, 2001).”

A definição de projeto no PMBOK (2002): “Um projeto é um empreendimento temporário cujo objetivo é criar um produto ou serviço distinto e único”. O objetivo da gestão de projeto é o atendimento das demandas de clientes internos e externos, garantindo a satisfação de suas necessidades, mantendo um bom relacionamento, controlando os recursos, esforços e desempenho do projeto. De acordo com

PRESSMAN (2001), a gestão de projetos envolve planejamento, controle e acompanhamento de processos, planos de projeto, produtos e pessoas.

O planejamento de projetos de software tem como objetivo, segundo PRESSMAN (2001), fornecer uma estrutura que possibilite ao gerente fazer estimativas razoáveis de recursos, custos e prazos. Estas estimativas são realizadas dentro de um plano de tempo limitado ao início de um projeto de software e devem ser atualizadas regularmente a medida que o projeto progride. As atividades associadas ao planejamento são: definir o escopo do software, definir os riscos, estimar recursos humanos, software, hardware, reusabilidade, analisar as tarefas que serão desempenhadas e os principais referenciais durante o progresso do projeto. Durante o planejamento do projeto ocorrem as estimativas. Elas são fundamentais para a determinação dos esforços que serão exigidos, custos durante o processo de produção do software e o tempo que o processo irá durar.

De acordo com (PRESSMAN, 2001), após as estimativas a atividade a ser feita é a análise de riscos.

2.3. Análise e Gestão de Riscos

No livro sobre análise e administração de riscos, Robert Charette em 1989 *apud* PRESSMAN (1995) apresenta a seguinte definição conceitual de risco:

Primeiro, o risco preocupa-se com acontecimentos futuros, Hoje e amanhã estão além da preocupação ativa, uma vez que já estamos colhendo aquilo que foi anteriormente plantado por nossas ações passadas. A questão é, portanto: ao mudarmos nossas ações hoje, podemos criar oportunidade para uma situação diferente e esperançosamente melhor para nós mesmos amanhã? Isso significa, em segundo lugar, que o risco envolve mudanças, tais como mudanças de mentalidade, de opinião, de ações ou de lugares... [Em terceiro lugar], o risco envolve escolha e a incerteza que a própria escolha acarreta. Desta forma, paradoxalmente, o risco, como a morte e os impostos, é uma das poucas certezas da vida.

Segundo (PRESSMAN, 2001), a análise dos riscos são, de fato, compostas por quatro atividades distintas: identificação, projeção, avaliação e administração dos

riscos. A identificação dos riscos consiste em dividir em categorias macro como: riscos de projeto, riscos técnicos e riscos de negócio. Os riscos de projetos são aqueles ligados a problemas com orçamento, cronograma, de pessoal, de recursos, de clientes e de requisitos e o impacto dos mesmos no projeto. Os riscos técnicos ocorrem devido aos problemas que, durante o projeto acabam ganhando um nível de complexidade maior do que o suposto anteriormente, dificultando mais sua resolução. Por sua vez, os riscos de negócios estão relacionados a fatores como falta de mercado para o produto, *marketing* ineficiente, mudanças administrativas que podem ocasionar perda de apoio ao projeto e falta de comprometimento com os orçamentos estabelecidos e com pessoal.

A projeção dos riscos classifica cada risco de duas maneiras: a probabilidade de o risco tornar-se real e as conseqüências dos problemas associados ao risco caso ele ocorra. A avaliação dos riscos consiste em examinar com mais detalhes a precisão das estimativas que foram feitas durante a projeção dos riscos, determinar uma ordem de prioridade para os riscos que foram descobertos e escolher como controlar e/ou evitar riscos que tem a probabilidade de acontecer.

As atividades de gerenciamento e monitoração dos riscos são compostas pelo trio (descrição, probabilidade e impacto dos riscos) onde associado a cada risco é usado como uma base a partir da qual os passos de gerenciamento são desenvolvidos. Estes passos de administração de riscos estão organizados num Plano de Administração e Monitoramento dos Riscos (PAMR) que documenta todo trabalho executado como parte da análise de riscos e é usado pelo gerente de projeto como parte do plano de projeto global. Assim que o PAMR tiver sido desenvolvido e o projeto iniciado, começa a monitoração dos riscos que é uma atividade de rastreamento do projeto com três objetivos: (1) avaliar se um risco previsto, de fato, ocorre; (2) garantir que os passos de reversão definidos para o risco estejam sendo aplicados adequadamente; e (3) coletar informações para servir de histórico para serem usadas em análises de risco futuras.

A análise de riscos pode absorver boa quantidade de esforço de projeto. A identificação, projeção, avaliação, administração e monitoração, tomam muito tempo e um cronograma auxilia no cumprimento dos prazos.

2.4. Cronogramação e Acompanhamento do Projeto

A cronogramação para projetos de softwares consiste em estabelecer metas e prazos, sempre tendo em vista uma data final previamente decidida por meio de análises da distribuição dos esforços e recursos utilizados. A determinação precisa do cronograma de um projeto tem grande importância, às vezes superando a importância da determinação dos custos do projeto. Um cronograma com deficiências na elaboração ou que tenha sofrido uma série de atrasos no cumprimento das metas tem efeitos bastante negativos sobre o produto, como insatisfação de clientes, possíveis aumentos de custos, e perda de credibilidade junto ao mercado.

Segundo PRESSMAN (2001), durante um projeto de menor escala, a quantidade de pessoas pode ser bem reduzida, podendo ter somente uma pessoa responsável por praticamente todas as etapas do desenvolvimento. À medida que o tamanho de um projeto for aumentando, a quantidade de pessoas aumentará da mesma forma.

A definição das tarefas do projeto é essencial para que o mesmo seja concluído no prazo estabelecido. Em casos de paralelismo entre as tarefas, deve-se ter bem definidas as tarefas críticas, ou seja, tarefas que devem ser concluídas dentro dos prazos estabelecidos durante a cronogramação sem sofrer nenhum tipo de atraso. O conjunto de tarefas críticas formam o caminho crítico. A duração do projeto de software depende desta cadeia de tarefas, essenciais para a conclusão do projeto dentro dos prazos estabelecidos. Uma distribuição de esforço bem feita também contribui para conclusão de projetos em tempo hábil. Analisar as porcentagens de esforços que devem ser direcionados para as áreas de análises e projeto é de suma importância, tendo-se em vista que isso reduzirá o tempo de codificação e possivelmente dos testes e depurações.

“A determinação de um cronograma para um projeto de software não difere significativamente da fixação de prazos para qualquer esforço de desenvolvimento de multitarefas. Portanto, ferramentas e técnicas para determinação de um cronograma de projeto podem ser aplicadas no software com poucas modificações. (PRESSMAN, 2001).”

Métodos de determinação de cronogramas que elaboram redes de tarefas, como o PERT – *Program Evaluation and Review Technique*, facilitam o trabalho do planejador, que utiliza-se de estatísticas e estimativas de tempo para a conclusão de tarefas críticas ao projeto. Calcular de forma precisa o tempo-limite para conclusões de tarefas confere uma boa avaliação de progresso e visualização dos caminhos críticos do projeto (PRESSMAN, 2001).

2.5. Garantia da Qualidade de Software

A ES com seus métodos, ferramentas e procedimentos, segundo PRESSMAN (2001), tem como única meta produzir software de alta qualidade. A Garantia de Qualidade de Software (SQA) é uma atividade aplicada ao longo de todo o processo de engenharia. A SQA abrange:

- métodos e ferramentas de análise, projeto, codificação e teste;
- revisões técnicas formais que são aplicadas durante cada fase de ES;
- estratégia de teste de múltiplas fases;
- controle da documentação de software e das mudanças feitas nela;
- procedimento para garantir a adequação aos padrões de desenvolvimento de software; e
- mecanismos de medição e divulgação.

Muitas definições de qualidade de software têm sido propostas, porém, PRESSMAN (2001) define:

“Conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente desenvolvido (PRESSMAN, 2001).”

McCall (1977) *apud* PRESSMAN (2001) propuseram uma categorização útil dos fatores que afetam a qualidade de software. Esses fatores são mostrados na figura 1, focalizam três aspectos importantes de um software: suas características

operacionais, sua manutenibilidade de mudanças e sua adaptabilidade a novos ambientes.

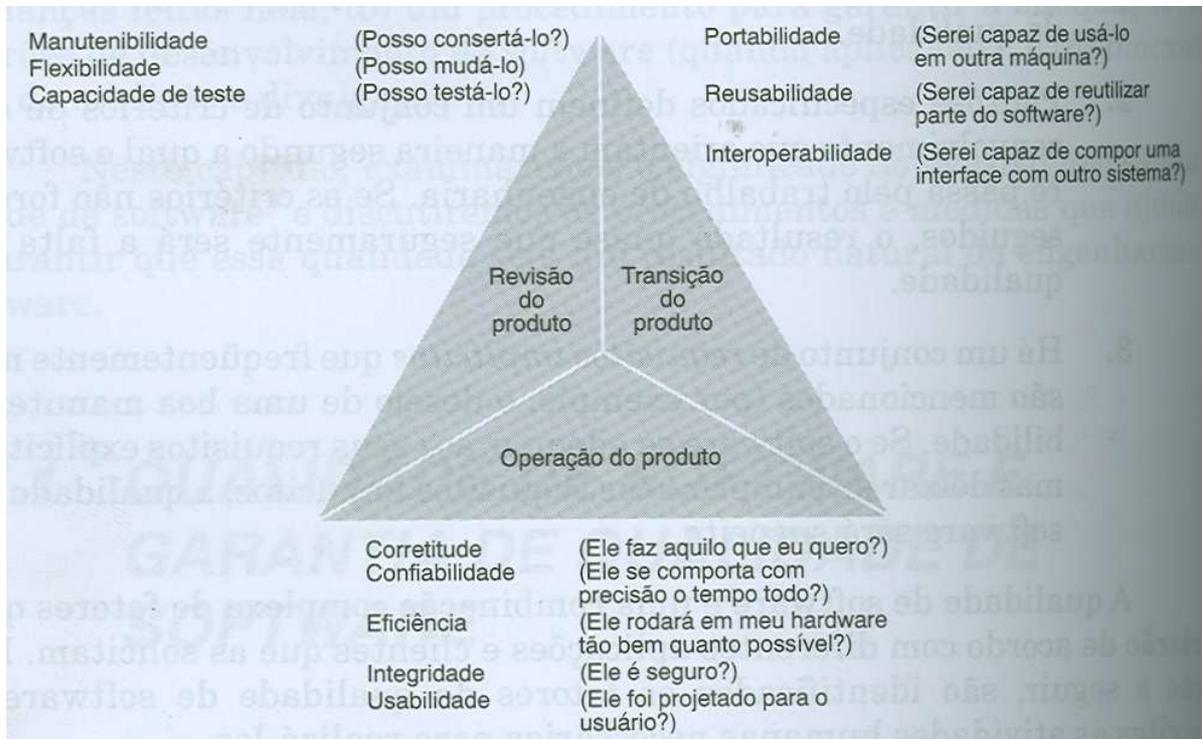


Figura 1 – Fatores de qualidade de software de McCall

Fonte: Pressman, 1995, p.726

Em relação aos fatores anotados na Figura 1, McCall apresenta as seguintes descrições:

- **Corretude:** a medida que um programa satisfaz sua especificação e cumpre os objetivos visados pelo cliente.
- **Confiabilidade:** a medida que se pode esperar que um programa execute sua função pretendida com a precisão exigida.
- **Eficiência:** a quantidade de recursos de computação e de código exigida para que um software execute sua função
- **Integridade:** a medida que o acesso ao software ou a dados por pessoas não-autorizadas pode ser controlado.
- **Reusabilidade:** a medida que um programa (ou parte de um programa) pode ser reutilizado em outras aplicações.
- **Usabilidade:** o esforço para aprender, operar, preparar a entrada e interpretar a saída de um programa.

- Manutenibilidade: o esforço exigido para localizar, entendê-lo, corrigi-lo e adaptá-lo.
- Flexibilidade: esforço exigido para modificar um programa operacional.
- Testabilidade: esforço exigido para testar um programa a fim de garantir que ele execute sua função pretendida.
- Portabilidade: esforço exigido para transportar um software de um ambiente de hardware e/ou software para outro.
- Interoperabilidade: esforço exigido para se acoplar um sistema a outro.

Podemos utilizar os recursos de métricas de estimativas de tamanho de software, que serão abordados mais profundamente no capítulo 3 a seguir, como mecanismos de medição dos fatores citados acima, que interferem diretamente na qualidade do software a ser produzido.

3. MÉTRICAS DE ESTIMATIVAS DE TAMANHO DE SOFTWARE

O comprometimento com o usuário responsável que solicita o desenvolvimento de um sistema para que a entrega do produto seja feita no prazo e com o orçamento previsto é de grande importância para o sucesso das empresas de desenvolvimento de software. Eficiência e qualidade na prestação do serviço e/ou produto pela empresa são características que identificam uma empresa responsável e comprometida com o trabalho. Para garantir o cumprimento das atividades no processo de desenvolvimento de software são utilizados mecanismos de acompanhamento e avaliação do progresso do processo, projeto e produto (ANDRADE, 2004).

Durante o projeto, informações qualitativas e quantitativas são extraídas, servindo de base para os mecanismos de acompanhamento, vistos como recursos essenciais que auxiliam a gestão de desenvolvimento de software. Esses mecanismos, assim chamados de medidas ou métricas, possibilitam ao gerente do projeto um entendimento do processo de produção e do controle do projeto de software. Portanto, a utilização destas medidas determinam os pontos fortes e os pontos fracos dos processos e dos produtos, permitem avaliar impactos no projeto, indicam ações corretivas, propiciando um melhor gerenciamento em todo o processo de desenvolvimento de software (FENTON; PFLEEGER, 1997; BASILI; CALDIERA, ROMBACH, 1994 *apud* ANDRADE, 2004).

Neste capítulo serão apresentados conceitos relacionados à medição de software e as métricas de estimativas de tamanho de software, citando métodos de medições funcionais desenvolvidos e utilizados em vários países, tais como: Linhas de Código (LOC), IFPUG, BFPUG, NESMA, Mark II, COSMIC – FFP e *Use Case Points* (UCP).

3.1. Medidas de Software

“Somente com um objetivo claro pode-se fazer uma medição adequada das medidas [de software]” (BASILI, 1994 *apud* PRESSMAN, 1995).

A medição de software é definida por Basili (1994) *apud* Pressman (1995), como um mecanismo para criar memória corporativa e auxiliar na busca de respostas em questões ligadas ao processo de software. Conforme ressalta Andrade (2004), o foco da medição está nos objetivos específicos e deve ser aplicada durante todo o processo de desenvolvimento. Quando a medição é implementada relacionando-se diretamente com atividades técnicas e de gestão de projetos, ela é mais efetiva.

De acordo com Basili (1994) *apud* Andrade (2004), cinco características da medição podem ser consideradas: objeto de medição, propósito da medição, fonte de medida, propriedades medida e contexto da medição. Objeto da medição varia de produto (por exemplo, projeto de software) a processo (por exemplo, processo de projeto arquitetônico) e projetos. A caracterização, avaliação, análise e previsão são considerados propósitos da medição. Os projetistas, testadores e gerentes de software são as fontes de medida. O custo do software, confiabilidade, manutenibilidade, tamanho e portabilidade são as propriedades medidas. Já o contexto da medição caracteriza ambientes diferentes para se medir artefatos de software.

McClure (1994) *apud* Peters e Pedrycz (2001) define medida de software como um mapeamento de um conjunto de objetos no mundo da engenharia de software em um conjunto de construções matemáticas, tais como número ou vetores de número. As medidas previnem o processo de desenvolvimento de software contra incertezas do projeto. A utilização das medidas fornece aos engenheiros de software meios que quantificam e avaliam um produto de software. Portanto, os resultados apresentados após cálculos feitos com base nessas medidas tornam-se essenciais na garantia da qualidade dos produtos de software.

Com base no IEEE *Standard Glossary of Software Engineering Terms*, Pressman (2001) define medida como “indicação quantitativa da extensão, dimensão, capacidade ou tamanho de algum atributo de um produto ou de um processo” e define métrica como “medida quantitativa do grau em que um sistema, componente ou processo possui determinado atributo. As métricas podem ser divididas em diretas ou indiretas. As métricas diretas são o custo, o esforço, a velocidade de execução das atividades e os defeitos documentados em um intervalo de tempo. As

métricas indiretas são as funcionalidades do sistema, a qualidade, a eficiência, a complexidade, a confiabilidade e a manutenibilidade, conforme é mostrado na figura 2.

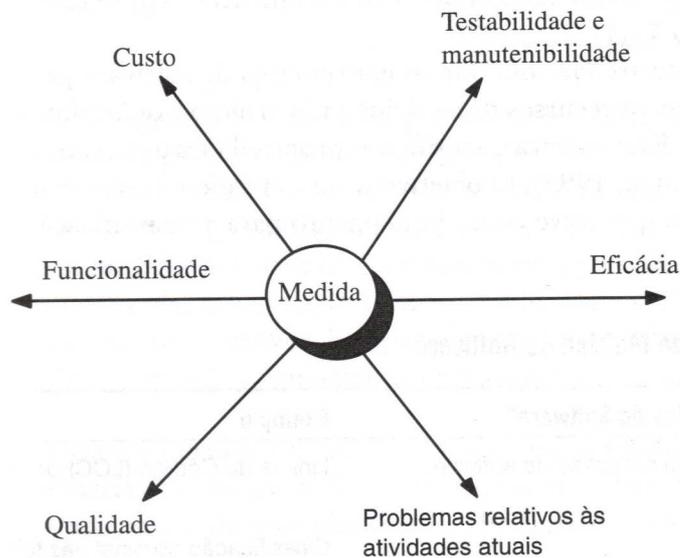


Figura 2 – Métricas diretas e indiretas do software

Fonte: Peters, Pedrycz, 2001, p. 429

Os indicadores, conhecidos como métrica ou combinação de métricas são obtidos com base nas medidas. Eles fornecem compressão de um processo, recursos e projeto ou produto de software. Com o levantamento dessas informações, os gerentes do projeto atingem eficiência e qualidade no processo de desenvolvimento de software (PRESSMAN, 1995; 2002).

As métricas podem ainda serem classificadas em outras categorias: métricas de processo, produto e recursos e objetivas e subjetivas (FENTON; PFLEEGER, 1991; PRESSMAN, 2002 *apud* ANDRADE, 2004). As atividades realizadas durante toda a fase de desenvolvimento de software são medidas pelas métricas de processo; os artefatos, produtos e documentos gerados pelo processo são medidos pelas métricas de produto; já as entidades requeridas para a execução do processo são medidas pelas métricas de recursos (FENTON; PFLEEGER, 1997 *apud* ANDRADE, 2004). Expressões numéricas ou representações gráficas de expressões numéricas calculadas a partir do código fonte, projeto e dados de testes são informações importantes fornecidos para as métricas objetivas, permitindo assim a quantificação

e medição. As métricas subjetivas baseiam-se em estimativas individuais ou de equipe. Essas estimativas normalmente são obtidas através da definição de alguns conceitos como excelente, regular, bom e ruim (MOLLER; PAULISH, 1993 *apud* ANDRADE, 2004).

Para Fenton e Pfleeger (1997) *apud* Andrade (2004), as métricas subjetivas estão diretamente ligadas às pessoas que a utilizam, ou seja, ela pode variar a depender da pessoa que a estiver mensurando, pode depender do julgamento feito pela própria pessoa e ainda do grau de imprecisão que envolve a métrica. Isso pode dificultar o consenso entre os atributos que cercam os processos, produtos ou qualidade. De acordo com esses dois autores, as métricas diretas não dependem da medida de outro atributo e sim da quantificação de um fator observado no produto. As métricas indiretas podem envolver as medidas de vários atributos.

Algumas abordagens de métricas de estimativas de projeto são destacadas dentro da categoria das métricas diretas, tais como: modelos paramétricos, analogia, julgamento de especialistas atividades baseadas em modelos e relacionamento de estimativas. Os modelos paramétricos, a analogia e o julgamento de especialistas são as abordagens mais utilizadas pelas empresas (McGARRY *et al.*, 2001 *apud* ANDRADE, 2004). Esse modelo pressupõe a existência de um relacionamento matemático entre o tamanho, esforço, cronograma e qualidade. A analogia requer um conhecimento detalhado do projeto e para isso é necessário o auxílio de um especialista (JORGENSEN, INDAHL, SJOBERG, 2003 *apud* ANDRADE, 2004).

A abordagem de julgamento de especialista trabalha baseado na experiência de profissionais em projetos semelhantes dos quais serão desenvolvidos. Na ausência de documentação que armazenam dados quantitativos, o conhecimento do especialista é útil para o processo de desenvolvimento do sistema (LONGSTREET, 2002 *apud* ANDRADE, 2004). O domínio das métricas de software pode ser dividido em mais categorias, como mostra a figura 3.

As abordagens de métricas de estimativas precisam de dados históricos similares ao projeto proposto para obter resultados significativos, tais como domínio de aplicação, o nível de maturidade do processo de desenvolvimento, a experiência da equipe,

taxas de produtividade, ferramentas utilizadas e o grau de reuso (FENTON; PFLEEGER, 1997; McGARRY et al., 2001 *apud* ANDRADE, 2004).

Na seção seguinte, serão apresentadas as métricas de estimativas de tamanho de software, o seu histórico e sua evolução.

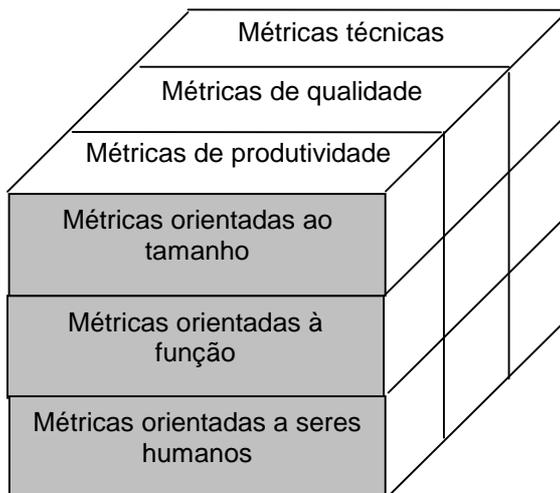


Figura 3 – Divisão das métricas em categorias

Fonte: Pressman, 1995, p.62

3.2. Métricas Funcionais

Métricas orientadas ao tamanho são medidas diretas do software e do processo por meio do qual ele é desenvolvido (PRESSMAN, 2002). O processo pelo qual uma pessoa ou um grupo de pessoas estima o tamanho de um produto de software é definido por Andrade (2004) como estimativa de tamanho de software. O tamanho influencia diretamente as soluções técnicas a serem adotadas para o projeto além de impactar na gestão do projeto, sabendo-se que estimativas errôneas levam ao fracasso do projeto (ROS, s.d *apud* ANDRADE, 2004). Conforme Peters e Pedrycz (2001) as métricas orientadas ao tamanho são óbvias e conceitualmente diretas por capturarem um grande volume de código. As medidas de tamanho de software, tanto para projetistas quanto para os usuários, são boas indicadores da grandeza de

um sistema, envolvendo recursos como os requisitos de memória, o esforço de manutenção e o tempo de desenvolvimento.

A quantidade de trabalho a ser desenvolvido no processo de desenvolvimento software deve ser estimada pelo gerente do projeto, com a finalidade de definir o tamanho do software que será produzido. Andrade (2004) ressalta que cada projeto pode ser estimado de acordo com tamanho físico (medidos através da especificação de requisitos, análise, projeto e código), com base nas funções fornecidas pelos usuários, na problematização que deverá ser resolvida pelo software e no reuso do projeto.

Entre 1950 a 1960 surgiram as primeiras métricas de estimativa de tamanho de software. A princípio essas métricas basearam-se no tamanho físico de linhas de código como o *Lines of Code* (LOC) (FENTON; NEIL, 2000 *apud* ANDRADE, 2004). Os proponentes dessa métrica afirmam que ela é dependente da linguagem de programação, que elas penalizam programas bem projetados e que o seu uso em estimativas requer um nível alto de detalhes que pode ser difícil de conseguir (PRESSMAN, 1995).

Andrade (2004) destaca duas vantagens no uso de LOC: a possibilidade de fazer estimativas automaticamente; e a facilidade de usar dados históricos, pelo fato de muitas estimativas existentes terem sido calculadas através dessa métrica. As desvantagens relacionadas ao uso de LOC são: as ambigüidades, a falta de significado da medida para o usuário final e a dependência do grau de reuso e da linguagem de programação. Fenton e Pfleeger (1997) *apud* Andrade (2004), acrescenta que a LOC penalizam programas pequenos e bem projetados, não se adapta às linguagens não procedimentais e é difícil obtenção na fase inicial de planejamento.

Nos dias atuais, há várias métricas para estimativas de tamanho de software. As principais métricas foram desenvolvidas com base nas funções de software tais como: Function Point Analysis ou Análise de Pontos de Função (APF), detalhada no capítulo 4 deste trabalho; Bang; Feature Point; Boeeng's END Function Point; Mark

II, Use Case Point ou Pontos de Casos de Usos, COSMIC Full Function Point (McPHEE, 1999; GARMUS;HERRGN, 2000 *apud* ANDRADE, 2004).

A seguir, serão discutidos alguns tópicos sobre estimativas de software: Linhas de Código (LOC), IFPUG, BFPUG, NESMA, MARK II, COSMIC – FFP e Pontos de Casos de Uso (UCP).

3.2.1. Linhas de Código (LOC)

A técnica (*Line of code* - LOC), são medidas diretas do software e uma das primeiras técnicas utilizadas para estimar tamanho do software (PRESSMAN, 1995). A contagem da linha de código é considerada por Peters e Pedrycz (2001) como sendo a forma mais dominante na classe de medidas de software orientadas ao tamanho. Como o próprio nome revela, essa métrica sugere que o tamanho de um sistema seja medido pela contagem das linhas do código fonte do sistema (LIMA, 2002). Apesar dessa métrica ser conceitualmente simples, ela apresenta certa ambigüidade em relação a forma de contagem dessas linhas, conforme levantado por Peters e Pedrycz (2001):

“Os comentários deverão ser incluídos? Como as várias instruções em uma mesma linha deverão ter tratadas?” (PETERS, 2001, p.433).

Pensando-se em solucionar os problemas, a partir das perguntas acima, Peters e Pedrycz (2001) sugere o estabelecimento de critérios de contagem, conforme descrito abaixo:

Em primeiro lugar, devemos excluir os comentários e as linhas em branco. Todas as instruções em uma única linha de código são contadas como uma única linha de código. Todas as linhas nas quais existam cabeçalhos de programa e de declarações também contribuem para a contagem geral de linhas de código (PETERS; 2001, p.433).

Tais critérios contribuem para uma apuração maior e real das linhas de código consideráveis.

Por ser considerado uma métrica frágil e imprecisa, várias desvantagens são levantadas referentes à LOC (LIMA, 2002):

- LOC não possui padrões, sendo obscura na definição de linha de código;
- Não tem significado para o usuário por ser uma medida altamente técnica;
- São inconsistentes, ou seja, algumas linhas são mais trabalhosas que outras;
- Apresenta problemas de definição para linguagens não procedurais;
- LOC na linguagem Assembly, por exemplo, não é comparável com LOC em linguagens de alto nível;
- Não possui dados para novas linguagens;
- As contagens das linhas são mascaradas quando utilizado a herança de classe;
- Não contempla as primeiras fases do ciclo de desenvolvimento de sistemas.

Em contrapartida, as LOC possuem alguns pontos relevantes e por isso não deve ser completamente descartada (LIMA, 2002):

- Possui literatura amplamente disponível;
- Serve como medida de normalização para a qualidade de software;
- Pode avaliar a produtividade;
- Pode fazer a contabilização das linhas de código automaticamente.

3.2.2. IFPUG

Após a apresentação da técnica de análise de pontos por função, no final da década de 70, como uma alternativa às métricas baseadas em linhas de código, houve um crescimento do número de usuários de pontos de função, se comparado ao surgimento dessa técnica no início da década de 70. Esse crescimento de usuários motivou em 1986 a criação do IFPUG (SALVADOR, 2005; PAVAN 2004).

O IFPUG (*International Function Point Users Group*) ou Grupo Internacional de Usuários de Pontos por Função, é uma entidade sem fins lucrativos, composta por pessoas e empresas de diversos países, com a finalidade de promover um melhor

gerenciamento dos processos de desenvolvimento e manutenção de software, utilizando a técnica de análise de pontos por função e outras técnicas de medição.

Em 1990, o IFPUG publicou a primeira versão do CPM - *Counting Practices Manual* ou Manual de Práticas de Contagem, objetivando padronizar a técnica de análise de ponto por função, sendo considerado como padrão pela maioria de indústrias de software para pontos de função. O IFPUG já lançou as versões 4.1 e 4.2 do CPM. Outras técnicas de medição funcional surgiram, tais como BFPUG, Mark II, mas, nenhuma delas possui aceitação tão ampla no mercado quanto a análise de pontos de função do IFPUG (SALVADOR, 2005; PAVAN, 2004).

O grupo Internacional de Usuários de Pontos por Função promove uma série de ações como conferências anuais, seminários e workshops educacionais, certificação profissional (CFPS) e comitês e grupos de trabalho. As conferências anuais são eventos promovidos para que especialistas renomados e usuários de pontos por função tenham a oportunidade de compartilhar experiência. Os seminários e workshops educacionais têm por objetivo divulgar a técnica de pontos por função sendo apresentados tópicos como prática de contagem, técnicas de gerenciamento de projetos e melhoria de processos. A certificação profissional, CFPS é um programa que envolve a certificação dos usuários da técnica como especialistas em pontos de função. Os comitês e os grupos de trabalho são considerados o centro das atividades promovidas pelo IFPUG. Os grupos também possuem vários comitês, como o comitê de certificação, responsável pelo programa de certificação; o comitê de conferência, responsável pelas conferências anuais; o comitê de práticas de contagens, responsável pelo CPM; o comitê de educação, responsável pelos seminários e workshops; o comitê de performance de TI, responsável pelo planejamento, gerenciamento e melhoramento das práticas e processos de engenharia de software; o comitê de novos ambientes, que desenvolve e provê diretrizes para a aplicação de pontos de função em novos ambientes; o grupo de trabalho de questões acadêmicas, que coordena os esforços entre o IFPUG e a comunidade acadêmica interessada no ensino e pesquisa de métricas de software; o grupo de trabalho do padrão ISO, responsável pela aderência do CPM no padrão de medição de tamanho funcional (ISO/IEC 14143).

3.2.3 BFPUG

No início da década de 90, o uso da análise de pontos por função começou a ser intensificada no Brasil. Entre o período de 1991 e 1994 foram realizados seis ENUPF – Encontro Nacional de Usuários de Pontos por Função, com a participação de palestrantes internacionais. Foi observado que o interesse pelas técnicas de pontos por função aumentou no Brasil pelo fato de grandes contratos públicos serem fechados baseados nessa técnica e, em paralelo, o interesse de empresas de desenvolvimento de software por modelos de qualidade, como ISO, CMM (Salvador, 2005). Para poder competir com empresas internacionais de desenvolvimento de software, os brasileiros começaram a se interessar pela técnica e viram a necessidade de aprendê-la.

Membros do IFPUG fundaram em 1998 o BFPUG (*Brazilian Function Users Group*) ou Grupo Brasileiro de Usuários de Pontos por Função – e em pouco tempo o BFPUG tornou-se o *chapter* do IFPUG no Brasil. O BFPUG possui centenas de associados, dentre estudantes, desenvolvedores, consultores e gerentes. O BFPUG promove a troca de experiência entre os profissionais, fórum, palestras, além do exame de certificação do IFPUG no Brasil.

Segundo o site do BFPUG, até dezembro de 2006, 742 pessoas fizeram o exame de certificação CFPS no Brasil. O exame para certificação CFPS já foi realizado dez vezes no Brasil e o país detém a terceira posição no *ranking* mundial com o maior número de especialistas certificados pelo IFPUG, perdendo apenas para os Estados Unidos e a Itália, na figura 3 observe a evolução da quantidade de CFPS brasileiros.

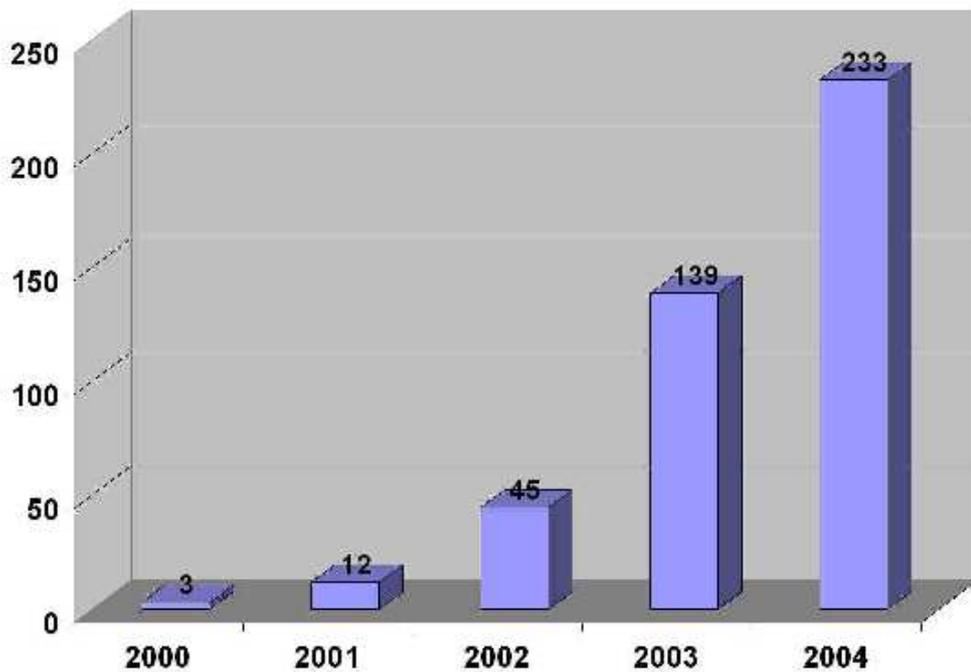


Figura 4 – Evolução da Quantidade de CFPS Brasileiros

Fonte: IFPUG (2006)

3.2.4 NESMA

NESMA (*Netherlands Software Metrics Users Association*) ou Associação de Usuários de Métricas de Software da Holanda, criada inicialmente como NEFPUG (*Netherlands Function Point Users*) ou Grupo de Usuários de Pontos por Função da Holanda, é um dos maiores grupos de usuários de pontos por função da Europa. O NESMA utiliza filosofia, conceitos, termos e regras semelhantes a do IFPUG, porém algumas diretrizes diferem um grupo do outro.

Desde 1990, a NESMA possui o seu próprio manual de contagem de pontos por função. A primeira versão desse manual foi baseada no manual do IFPUG. A versão atual do NESMA é a 2.0, sendo que sua forma de contagem apresenta resultados, bem próximos ao do IFPUG. As organizações NESMA e IFPUG possuem abordagens distintas para a aplicação da análise de pontos por função em projetos de melhoria de software. A NESMA, atualmente é um dos maiores grupos de usuários de pontos por função da Europa. Uma das propostas da NESMA é a

contagem estimada que possibilite a estimativa de tamanho a partir da identificação das funções de dados e de transação (sem a necessidade de identificação dos elementos de dados de cada função).

3.2.5 Mark II

O Mark II, também conhecido como MK II, em meados da década de 80, foi formulado por Charles Symons, inspirado na proposta de Albrecht. Após uma publicação na revista IEEE – *Transactions on Software Engineering*, o trabalho de Charles Symons ficou muito conhecido na comunidade científica no ano de 1988. Entretanto, a técnica não obteve muito sucesso, ou seja, não se tornou muito utilizada pelo fato de ser um método proprietário de uma empresa de consultoria, a KPMG.

Hoje, o método é de domínio público e a associação responsável por manter os padrões dessa técnica é a UKSMA, Associação de Métricas do Reino Unido que realizam ações semelhantes as do IFPUG. No entanto, esta técnica está muito restrita ao Reino Unido.

3.2.6 UCP

O UCP (*Use Case Points*) ou Pontos de Casos de Uso foi desenvolvido por Karner (1993) sendo baseado em cenários e casos de uso. É um método de estimativa de tamanho de projeto de software orientado a objetos, baseado na APF, Mark II e no modelo de casos de uso. O Modelo de casos de uso foi incorporado na *Unified Modelling Language* (UML). Assim como a APF, o UCP também estima o tamanho do software conforme visão do usuário e com base em uma unidade de medida da UCP. Esse modelo trabalha com quatro fatores distintos: atores, casos de uso, fatores técnicos e de ambiente (ANDRADE, 2004). Os casos de uso foram criados por Ivar Jacobson no ano de 1986. Um caso de uso é uma especificação, em forma

de narrativa, de uma seqüência de interações entre um sistema e os atores (agentes externos) que usam. Os casos de usos devem descrever em um nível de detalhes algo que o usuário solicita que o sistema faça, podendo ele ser simples ou complexos. Os casos de uso devem descrever uma tarefa de negócio, servindo a um único objetivo de negócio, não devem ser orientados a uma linguagem de programação, deve possuir um nível de detalhe apropriado e ser curto e suficiente para ser implementado por um desenvolvedor de software (ANDRADE, 2004).

3.3 Considerações Finais

As métricas para estimar o tamanho do software vêm sendo amplamente utilizadas por empresas que desenvolvem sistemas no intuito de, cumprir com os prazos estabelecidos no fechamento de contrato com o cliente e no controle do orçamento. Com a utilização dessas métricas, os gerentes de projetos podem gerar subsídios para a tomada de decisões bem informados. As medidas de software citadas neste capítulo permitem ao leitor observar as diferenças entre elas e conhecer os Institutos responsáveis pela atualização dos manuais de contagens e pela divulgação dessas técnicas em todo o mundo.

4. APF (ANÁLISE DE PONTO POR FUNÇÃO)

A técnica de análise de pontos por função foi desenvolvida em meados da década de 70 pela IBM como uma alternativa às métricas baseadas em linhas de código. Ela foi publicada em 1979 por Allan Albrecht na tentativa de reduzir as dificuldades encontradas pelos usuários das linhas de código como medida de tamanho de software (VASQUEZ, 2003).

Em 1984 Allan Albrecht refinou a primeira versão da APF que visava tratar dos fatores externos, das características importantes para o usuário. Com o aumento no número de usuários utilizando a APF viu-se a necessidade de implantar um guia que interpretasse as regras originais definidas por Allan para os novos ambientes que surgissem. O crescimento de adeptos a APF motivou a criação do *International Function Point Users Group* (IFPUG), em 1986. A partir desta data, o IFPUG se tornou o responsável pela definição das regras de contagem, pelo treinamento de usuários, pela certificação dos profissionais interessados a utilização desta técnica em ambiente de trabalho e pelo fornecimento de dados históricos de produtividade da indústria de desenvolvimento de software (IFPUG, 2000; VASQUEZ, 2003; ANDRADE, 2004). A APF foi padronizada pela norma ISO/IEC 20929 e tornou-se a métrica mais usada e estudada na história da engenharia de software.

4.1. Objetivos

A APF visa estabelecer uma medida de “tamanho” de software utilizando pontos por função (PF), através da quantificação das funções implementadas sob o ponto de vista do usuário que solicita um sistema, adotando um enfoque empresarial ao invés de técnico, o que torna o método de avaliação independente do hardware e da tecnologia de desenvolvimento (VASQUEZ *et al* 2003; ANDRADE, 2004; SALVADOR, 2005).

O princípio básico dessa métrica é o foco na fase de especificação de requisitos, visando obter estimativas de tempo, custo, esforço e recursos na fase inicial do projeto ou na manutenção de software. Os resultados das estimativas a partir da APF são independentes da tecnologia utilizada para desenvolver um software (SILVA, 2000; VASQUEZ *et al* 2003).

Tavares; Carvalho e Castros (2002) dizem que a métrica de Análise de Pontos por Função tem por objetivo medir o tamanho do software pela quantificação de sua funcionalidade externa, baseada no projeto lógico ou a partir do modelo de dados. A funcionalidade do sistema deve ser informada pelo usuário, e a funcionalidade do software desenvolvido diz “o que” será entregue ao usuário. A gerência de projeto vê na medição de software, através de pontos por função, uma ferramenta muito útil que ajuda a manter o controle sobre um projeto específico.

A APF possibilita ao gerente de projeto uma contagem indicativa na fase inicial de desenvolvimento sem que haja um detalhamento no modelo de dados. Em seguida, na fase de projeto, a contagem passa a ser mais precisa e no término do desenvolvimento do software, há uma contagem detalhada (IFPUG, 2000; ANDRADE, 2004).

As organizações aplicam a APF como: “um método para determinar o tamanho do pacote de software adquirido; para apoiar a análise da produtividade e qualidade; para estimar os custos, os recursos e esforços de projeto de desenvolvimento e manutenção de software” (GARMUS; HERRON, 2000 *apud* HAZAN, 2004; ANDRADE, 2004).

Dentre a visão dos autores acima citados podemos destacar alguns objetivos da APF, como a medição da funcionalidade dos sistemas conforme perspectiva do usuário, a comparação da produtividade no ambiente de desenvolvimento, a definição de uma unidade padrão de medida de software, a permissão de melhores estimativas no desenvolvimento de sistemas para os gerentes de projetos, a transparência para o usuário final, a obtenção de estimativas de tempo, recursos, tamanho e custos desde o início do ciclo de desenvolvimento e a melhoria da qualidade dos contratos de terceirização, permitindo basear o pagamento nos

resultados (tamanho entregue em pontos de função) e não nos insumos (horas trabalhadas).

4.2. Determinação do Tipo de Contagem

A APF pode ser utilizada para diversos tipos de necessidade por ser aplicada desde o início até o término do projeto. A utilização da APF gera estimativas mais precisas, o que acarreta em uma melhoria no gerenciamento de projetos de desenvolvimento de sistemas. Antes de iniciar a contagem de pontos por função, é preciso definir o tipo de contagem, o escopo e a fronteira da aplicação (HAZAN, 1999; SILVA, 2000; ANDRADE, 2004).

4.2.1. Tipo de Contagem

A determinação do tipo de contagem de pontos por função representa a função do objeto da contagem que será usado para estimar o projeto, podendo esse projeto ser de desenvolvimento, de manutenção ou uma aplicação já finalizada. A seguir detalharemos os tipos de contagem (IFPUG, 2000).

4.2.1.1. Projeto de Desenvolvimento

Tem por objetivo medir as funções oferecidas ao usuário na primeira instalação do software. Esse tipo de contagem mede as funcionalidades disponibilizadas por um novo desenvolvimento e pode incluir a contagem de software destinado à conversão de dados ou migração de dados (IFPUG, 2000; SILVA, 2000; VASQUEZ *et al* 2003).

4.2.1.2. Projeto de Melhoria

Tem por objetivo medir as modificações funcionais efetuadas em uma aplicação existente. As modificações efetuadas em um software já existentes podem incluir, alterar ou excluir uma funcionalidade (IFPUG, 2000; SILVA, 2000; VASQUEZ *et al* 2003).

4.2.1.3. Contagem de Aplicações

Tem por objetivo medir a funcionalidade de um software já instalado. A contagem de aplicações é utilizada quando o projeto de desenvolvimento de software já foi concluído (IFPUG, 2000; SILVA, 2000; VASQUEZ *et al* 2003).

4.3. O Escopo da Contagem e a Fronteira da Aplicação

De acordo com o IFPUG (2000), a funcionalidade que deve ser incluída na contagem de pontos por função é definida no escopo da contagem. Essa funcionalidade que o sistema deverá empenhar, é baseada na visão do usuário. A fronteira da aplicação separa o projeto a ser contado, conforme a visão do usuário, das aplicações externas que envolvem o projeto.

A fronteira da aplicação é considerada um limite conceitual que separa o software medido da visão dos usuários. Neste contexto, “usuário” é considerado qualquer pessoa ou coisa que defina os requisitos funcionais do software e que se comunique ou interaja com o software, inclusive outro sistema (IFPUG, 2000; SILVA, 2000; ANDRADE, 2004).

4.4. Funções do Tipo Dado

Os dados mantidos (atualizados) ou dados referenciados pelos processos elementares da aplicação refletem as funções de dados. Essas funções consistem de Arquivos Lógicos Internos (ALIs) e Arquivos de Interface Externa (AIEs). A diferença entre essa duas funções é a localização dos processos elementares responsáveis pela manutenção dos dados. Os ALIs estão contidos na própria aplicação que está sendo contada, já os AIEs são processos contidos em outras aplicações que interagem com a aplicação que está sendo contada (IFPUG, 2000; SILVA, 2000; VASQUEZ et al 2003; ANDRADE, 2004).

Nas seções 4.4.1 e 4.4.2, serão abordadas as funções do tipo dado: Arquivo Lógico Interno (ALI) e Arquivo de Interface Externa (AIE).

4.4.1. Arquivo Lógico Interno

Um Arquivo Lógico Interno (ALI) é um grupo lógico de dados ou informações de controle relacionados e reconhecidos pelo usuário, cuja manutenção é feita internamente pela aplicação. O objetivo principal desse arquivo é o armazenamento de dados mantidos pelos processos elementares da aplicação que está sendo contada. Na identificação de um ALI, observa-se as seguintes regras (IFPUG, 2000; ANDRADE, 2004; CARVALHO, TAVARES, CASTRO, 2002):

- Se o grupo de dados ou informações de controle é lógico e pode ser identificado pelo usuário;
- Se o grupo de dados é mantido por meio de um processo elementar dentro da fronteira da aplicação que a ser contada.

A identificação do grau de complexidade dos ALIs é feita através do número de itens de dados e de registro lógicos. O item de dados representa um segmento de um registro do ALI que tem significado único e é identificado pelo usuário (campos da tabela). Os registros lógicos são subgrupos de dados reconhecidos pelos usuários dentro do Arquivo Lógico Interno. As seguintes regras devem ser seguidas para

contar os registros lógicos e identificar os itens de dados (IFPUG, 2000; ANDRADE, 2004; SALVADOR, 2005):

- Considerar como registro lógico aqueles que não possuem subgrupos;
- Contar o item de dado para cada campo não repetitivo, reconhecido pelo usuário como único, mantido ou recuperado do ALI durante a realização de um processo elementar;
- Contar apenas os itens de dados utilizados em cada aplicação, se o arquivo mantém ou referencia itens de dados separados.
- Contar o item de dado uma única vez, se um campo aparecer como chave primária em um registro e como chave estrangeira em outro registro.

A tabela 1 deve ser utilizada para identificar o grau de complexidade de cada Arquivo Lógico Interno.

Arq.Ref/Campos	1 a 19	20 a 50	51 ou +
1	Baixa	Baixa	Média
2 a 5	Baixa	Média	Alta
6 ou +	Média	Alta	Alta

Tabela 1 – Complexidade ALI

Após determinar a complexidade de um ALI, devemos determinar a contribuição que ele vai proporcionar (VASQUEZ *et al*, 2003). Na tabela 2 verifica-se o número de pontos por função que um arquivo lógico interno proporciona em função da complexidade.

Complexidade	Pontos por Função
Simple	7
Média	10
Complexa	15

Tabela 2 – Pontos por Função ALI

4.4.2. Arquivo de Interface Externa

Um Arquivo de Interface Externa (AIE) é um grupo lógico de dados ou informações de controle relacionados e reconhecidos pelo usuário, porém mantido dentro da fronteira de uma aplicação. O objetivo principal desse arquivo é o armazenamento de dados referenciados pelos processos elementares da aplicação que está sendo contada. Um AIE nada mais é do que um dado ou um grupo de dados que são apenas referenciados pela aplicação contada, sendo mantido por uma outra aplicação (IFPUG, 2000; ANDRADE, 2004; PAVAN, 2004; SALVADOR, 2005).

A tabela 3 deve ser utilizada para identificar o grau de complexidade de cada Arquivo de Interface Externa.

Arq.Ref/Campos	1 a 19	20 a 50	51 ou +
1	Baixa	Baixa	Média
2 a 5	Baixa	Média	Alta
6 ou +	Média	Alta	Alta

Tabela 3 – Complexidade AIE

Após determinar a complexidade de um AIE, devemos determinar a contribuição que ele vai proporcionar (VASQUEZ *et al*, 2003). Na tabela 4 verifica-se o número de pontos por função que um arquivo de interface externa proporciona em função da complexidade.

Complexidade	Pontos por Função
Simple	5
Média	7
Complexa	10

Tabela 4 – Pontos por Função AIE

4.5. Funções do Tipo Transação

Para o usuário da aplicação, um processo elementar é a menor unidade de atividade com um significado próprio. Um processo elementar é independente, considerado auto-contido, deixando o negócio da aplicação em um estágio consistente após a sua execução (IFPUG, 2000, SILVA, 2000).

As funções de transação incluem os processos elementares, tratando os dados e informações de controle que entram pela fronteira da aplicação e em seguida apresentam a informação desejada ao usuário. Essas funções também mantêm os arquivos lógicos interno do sistema (IFPUG, 2000; ANDRADE, 2004).

Nas seções 4.5.1, 4.5.2 e 4.5.3, serão abordadas as funções do tipo transação: Entrada Externa (EE), Saída Externa (SE) e Consulta Externa (CE).

4.5.1. Entrada Externa

Uma entrada externa (EE) deve ser considerada quando houver um processo elementar ou informações de controle oriundas de fora da fronteira da aplicação e a intenção primária mantiver um ou mais arquivos lógicos internos ou ainda a intenção primária alterar o comportamento do sistema. Uma das opções seguintes devem ocorrer, para que, a EE seja vista como um processo elementar diferente dos demais (IFPUG, 2000; PAVAN, 2004):

- A lógica de processamento deve ser diferente das utilizadas por outras EE da aplicação;
- Os elementos de dados devem ser diferentes dos dados de outras aplicações EE;
- Os arquivos referenciados devem ser diferentes dos arquivos de outras aplicações de EE.

4.5.1.1. Contagem de Itens de Dados e Arquivos Referenciados

A partir do número de itens de dados e arquivos referenciados, determina-se o grau de complexidade de uma EE. As quantidades de itens de dados e arquivos referenciados devem ser determinadas de acordo com as regras de contagem, conforme tabela 5- Complexidade da Entrada Externa (IFPUG, 2000; VASQUEZ *et al* 2003):

Arq.Ref/Campos	1 a 4	5 a 15	16 ou +
0 a 1	Baixa	Baixa	Média
2	Baixa	Média	Alta
3 ou +	Média	Alta	Alta

Tabela 5 – Complexidade EE

Após determinar a complexidade de uma EE, deve-se determinar a contribuição que ela vai proporcionar (VASQUEZ *et al*, 2003). Na tabela 6 verifica-se o número de pontos por função que uma EE proporciona em função da complexidade.

Complexidade	Pontos por Função
Simple	3
Média	4
Complexa	6

Tabela 6 – Pontos por Função EE

4.5.2. Saída Externa

Uma saída Externa (SE) é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação. Ela deve ser considerada quando existir um processo elementar enviando dados e informações de controle para fora da fronteira da aplicação ou quando as informações apresentadas aos usuários seja

o objetivo principal do processo elementar ou quando a lógica do processo elementar contiver ao menos uma fórmula matemática ou ainda os dados gerados sejam derivados (IFPUG, 2000; SILVA, 2000; ANDRADE, 2004; SALVADOR, 2005).

O objetivo principal de uma saída externa é enviar dados para fora da fronteira da aplicação, podendo ainda manter um ou mais Arquivos Lógicos Internos e alterar o comportamento do sistema.

Um das opções abaixo devem ocorrer para que um processo elementar seja contado como uma SE (IFPUG, 2000):

- Utilização de fórmula matemática ou cálculo;
- Geração de dados derivados;
- Manutenção de Arquivos Lógicos Internos;
- Alteração do comportamento do sistema.

4.5.2.1. Contagem de Itens de Dados e Arquivos Referenciados

A partir do número de itens de dados e arquivos referenciados, determina-se o grau de complexidade de uma SE. A quantidade de itens de dados e arquivos referenciados deve ser determinada de acordo com as regras de contagem. Na tabela 7 mostra a complexidade da Saída Externa (IFPUG, 2000; VASQUEZ *et al* 2003):

Arq.Ref/Campos	1 a 5	6 a 19	20 ou +
0 a 1	Baixa	Baixa	Média
2 a 3	Baixa	Média	Alta
4 ou +	Média	Alta	Alta

Tabela 7 – Complexidade SE

Após determinar a complexidade de uma SE, deve-se determinar a contribuição que ela vai proporcionar (VASQUEZ *et al*, 2003). Na tabela 8 verifica-se o número de pontos por função que uma SE proporciona em função da complexidade.

Complexidade	Pontos por Função
Simple	4
Média	5
Complexa	7

Tabela 8 – Pontos por Função SE

4.5.3. Consulta Externa

Uma Consulta Externa (CE) é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação a partir de dados recuperados no armazenamento em ALLs ou em AIEs. Ela deve ser considerada quando existir um processo elementar que envie dados ou informações de controle para fora da fronteira da aplicação, quando a lógica de processamento, os arquivos referenciados ou os itens de dados forem diferentes dos demais utilizados pela Consulta Externa da aplicação, quando a lógica de processamento não possuir fórmula matemática, cálculo, nem criar dado derivado e quando não houver alteração do comportamento do sistema (IFPUG, 2000; SILVA, 2000).

O objetivo principal da consulta externa é apresentar informações para o usuário sem a ocorrência da atualização de arquivos lógicos internos, cálculos ou procedimentos para obter dados derivados, nem mudança de comportamento do sistema (IFPUG, 2000; ANDRADE, 2005).

4.5.3.1. Contagem de Itens de Dados e Arquivos Referenciados

A partir do número de itens de dados e arquivos referenciados, determina-se o grau de complexidade de uma CE. As quantidades de itens de dados e arquivos referenciados devem ser determinadas de acordo com as regras de contagem. A tabela 9 apresenta a complexidade da Consulta Externa, conforme o IFPUG (2000).

Arq.Ref/Campos	1 a 5	6 a 19	20 ou +
1	Baixa	Baixa	Média
2 a 3	Baixa	Média	Alta
4 ou +	Média	Alta	Alta

Tabela 9 – Complexidade CE

Após determinar a complexidade de uma CE, deve-se determinar a contribuição que ela vai proporcionar (VASQUEZ *et al*, 2003). Na tabela 10 verifica-se o número de pontos por função que uma CE proporciona em função da complexidade.

Complexidade	Pontos por Função
Simple	3
Média	4
Complexa	6

Tabela 10 – Pontos por Função CE

4.6. Pontos Por Função Não-Ajustados

Com base na classificação (simple, média ou complexa), dada às funções do tipo dado e do tipo transação, são calculadas as contribuições. O somatório das contribuições de todas as funções do aplicativo é o número encontrado dos Pontos por Função Não-Ajustados, ver tabela 11.

Parâmetro de medida	Contagem	Fator de Ponderação			=	
		Simple	Médio	Complexo		
Número de entradas do usuário	<input type="text"/>	x	3	4	6	<input type="text"/>
Número de saídas do usuário	<input type="text"/>	x	4	5	7	<input type="text"/>
Número de consultas do usuário	<input type="text"/>	x	3	4	6	<input type="text"/>
Número de arquivos	<input type="text"/>	x	7	10	15	<input type="text"/>
Número de Interfaces externas	<input type="text"/>	x	5	7	10	<input type="text"/>
Contagem - Total	→					<input type="text"/>

Tabela 11 – Pontos por Função não-Ajustados

Fonte: PRESSMAN; 1995, p.65

4.7. Fator de Ajuste

O Fator de Ajuste (VAF – *Value Adjustment Factor*) é baseado nas quatorze características gerais do sistema - CGS, descritas no subitem 4.7.1. As características gerais do sistema refletem as funções que afetam a aplicação de uma maneira geral. Cada CGS recebe um valor de influência de 0 a 5 mostrados na tabela 12.

Nível	Descrição
0	Não tem influência
1	Influência incidental
2	Influência moderada
3	Influência média
4	Influência significativa
5	Influência essencial

Tabela 12 – Níveis de Influência

Conforme o IFPUG, a partir dos níveis de influência das 14 CGS, o cálculo do Fator de Ajuste, dado pela equação (1):

$$\text{VAF} = (\text{TDI} * 0,01) + 0,65 \quad (1)$$

Onde:

- VAF (*Value Adjustment Factor*) – Fator de Ajuste.
- TDI (*Total Degree of Influence*) – Somatório dos Níveis de Influência das CGS.

O Fator de Ajuste corrige o número de pontos por função não-ajustados em no máximo + ou – 35%.

4.7.1. As Características Gerais dos Sistemas (CGS)

As características gerais do sistema avaliam a funcionalidade genérica da aplicação como um todo.

1- *Comunicação de Dados*

Essa característica descreve o grau de comunicação da aplicação com o processador. Os mecanismos de comunicação são responsáveis pelo envio e recebimento dos dados e informações de controles utilizados pela aplicação.

2- *Processamento Distribuído*

Essa característica descreve o grau de transparência de dados entre os diferentes componentes existentes na aplicação.

3- *Performance*

Essa característica descreve até que ponto as considerações sobre o *throughput* e o tempo de resposta podem influenciar o desenvolvimento da aplicação. Os objetivos de performance da aplicação, relacionado ao tempo de resposta ou *throughput*, que são declarados ou aprovados pelo usuário, influenciam o design, desenvolvimento, instalação e suporte à aplicação.

4- *Configuração Intensamente Utilizada*

Essa característica descreve até que ponto as restrições referentes aos recursos computacionais podem influenciar o desenvolvimento da aplicação.

5- *Taxa de Transação*

Essa característica descreve até que ponto a taxa de transação de negócio pode influenciar o desenvolvimento da aplicação.

6- *Entrada de Dado On-line*

Essa característica descreve até que ponto a entrada de dados pode ser efetuada através das transações interativas.

7- *Eficiência do Usuário Final*

Essa característica descreve até que ponto os fatores humanos e a facilidade de utilização do usuário podem ser considerados na aplicação.

8- *Atualização On-line*

Essa característica descreve até que ponto os ALIs podem ser atualizados on-line.

9- *Processamento Complexo*

Essa característica descreve até que ponto a lógica do processamento pode influenciar o desenvolvimento da aplicação.

10- *Reutilização*

Essa característica descreve até que ponto a aplicação e o respectivo código podem ser projetados, desenvolvidos e suportados para serem utilizados por outra aplicação.

11- *Facilidade de Instalação*

Essa característica descreve até que ponto a conversão a partir de outros ambientes pode influenciar o desenvolvimento da aplicação. A facilidade de conversão e instalação são características da aplicação.

12-Facilidade de Operação

Essa característica descreve até que ponto a aplicação atende aos aspectos operacionais como *start-up* (inicialização), *backup* (cópia de segurança) e recuperação.

13-Múltiplos Locais

Essa característica descreve até que ponto a aplicação pode ser desenvolvida para vários locais e várias organizações usuárias.

14-Facilidade de Mudança

Essa característica descreve até que ponto a aplicação pode ser desenvolvida em vista a fácil modificação de sua lógica de processamento ou estrutura de dados.

4.8. Pontos Por Função Ajustados

De acordo com o manual do IFPUG há três maneiras de calcular os pontos por função ajustados. Cada tipo de contagem tem o seu cálculo específico.

4.8.1. Contagem de Projetos de Desenvolvimento

Um projeto de desenvolvimento de sistema pode envolver três tipos de funcionalidades na contagem de pontos por função, são elas: funcionalidade da aplicação, funcionalidade de conversão e o fator de ajuste da aplicação.

A funcionalidade da aplicação existente nos requisitos do projeto, solicitado pelo usuário ao desenvolvedor do sistema, deve ser obtida a partir da aplicação instalada. Já a funcionalidade de conversão ou migração de dados, também existente nos requisitos, é disponibilizada no momento da instalação da aplicação.

Essa funcionalidade compõe programas de conversão de dados, migração de arquivos, geração de relatórios específicos da conversão ou da migração. O fator de ajuste da aplicação é obtido a partir das características gerais do sistema (IFPUG, 2000; SILVA, 2000; VASQUEZ *et al*, 2003).

A fórmula utilizada na contagem de um projeto de desenvolvimento é dado na equação (2):

$$DFP = (UFP + CFP) * VAF \quad (2)$$

Onde:

- DFP (*Development Function Points*) – Pontos de Função do Projeto de Desenvolvimento
- UFP (*Unadjusted Function Points*) – Pontos de Função Não-Ajustados
- CFP (*Conversion Function Points*) – Contagem não ajustada das funções acrescentadas pelo processo de conversão.
- VAF (*Value Adjustment Factor*) – Fator de Ajuste.

4.8.2. Contagem de Projetos de Melhoria

A contagem de pontos por função de um projeto de melhoria também envolve os três tipos de funcionalidades: funcionalidade da aplicação, funcionalidade de conversão e o fator de ajuste da aplicação.

A funcionalidade de aplicação de um projeto de melhoria é composta de pontos por função identificados a partir da funcionalidade incluída, pontos por função identificados a partir da funcionalidade alterada e pontos por função identificados a partir da funcionalidade excluída pelo projeto de melhoria.

A funcionalidade de conversão é utilizada somente na instalação do projeto de melhoria. Essa funcionalidade disponibiliza a instalação do projeto de melhoria e é composta por programas de conversão de dados, migração de arquivos, geração de relatórios específicos da conversão ou migração de dados.

O Fator de Ajuste é obtido a partir das características gerais do sistema. No projeto de melhoria do software, os fatores de ajustes são considerados antes e depois do projeto de desenvolvimento de software.

A fórmula utilizada na contagem de um projeto de melhoria é dada pela equação (3):

$$EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL * VAFB) \quad (3)$$

Onde:

- ADD (*Added*) – Contagem não ajustada das funções acrescentadas à aplicação durante um projeto
- CHGA (*Changed After*) – Contagem não ajustada das funções modificadas, após o projeto com base na complexidade funcional.
- CFP (*Conversion Function Points*) – Contagem não ajustada das funções acrescentadas pelo processo de conversão.
- VAFA (*VAF After*) – Fator de Ajuste após o projeto de melhoria.
- DEL (*Deleted*) – Contagem não ajustada das funções excluídas pelo projeto.
- VAFB (*VAF Before*) – Fator de Ajuste antes do projeto de melhoria.

4.8.3. Contagem de Aplicações

Para a contagem de Pontos por Função de uma aplicação, primeiro é necessário estabelecer a fórmula que será utilizada. Existem duas fórmulas para esse tipo de contagem: a fórmula para estabelecer o tamanho inicial da aplicação e a fórmula para atualizar a contagem após um projeto de melhoria.

4.8.3.1. Fórmula para Estabelecimento do Tamanho Inicial

A funcionalidade solicitada e recebida pelo usuário é incluída na contagem inicial. Para estabelecer o tamanho Inicial do projeto, apenas os requisitos são levados em

consideração, à funcionalidade de conversão não está incluída na fórmula equação (4):

$$AFP = ADD * VAF \quad (4)$$

Onde:

- AFP (*Application Function Points*) – Pontos de Função da Aplicação.
- ADD (*Added*) – Contagem não ajustada das funções acrescentadas à aplicação durante um projeto.
- VAF (*Value Adjustment Factor*) – Fator de Ajuste.

4.8.3.2. Fórmula para atualizar a contagem após a melhoria

A funcionalidade inicial de uma aplicação pode ser alterada após um projeto de melhoria de software. Essa alteração pode incluir uma nova funcionalidade, implicando em um aumento do tamanho da aplicação e também excluir uma funcionalidade antiga existente, acarretando em uma redução do tamanho da aplicação. Na contagem de uma aplicação, a funcionalidade de conversão, assim como na fórmula para o estabelecimento do tamanho, não é considerada.

A Fórmula para calcular o tamanho da aplicação após um projeto de melhoria é dada pela equação (5):

$$AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] * VAFA \quad (5)$$

Onde:

- UFPB – Pontos de Função Não-Ajustados, antes do projeto de melhoria.
- ADD (*Added*) – Contagem não ajustada das funções acrescentadas à aplicação durante o projeto.
- CHGA (*Changed After*) – Contagem não ajustada das funções modificadas.
- CHGB (*Changes Before*) – Contagem original não ajustada das funções modificadas.

- DEL (*Deleted*) – Contagem não ajustada das funções excluídas pelo projeto.
- VAFA (*VAF After*) – Fator de Ajuste após o projeto de melhoria.

5. METODOLOGIA DA PESQUISA

Este capítulo apresenta a metodologia utilizada neste trabalho, sendo composta por quatro partes distintas. Inicialmente, é descrito o tipo de pesquisa realizada com profissionais de TI que trabalham em organizações situadas em Salvador, estado da Bahia. Na primeira parte, apresenta-se o contexto da pesquisa, através da caracterização do universo pesquisado e da amostra coletada. Em seguida, é descrito o instrumento de mensuração usado. Na terceira parte, são apresentados os procedimentos, técnicas e ferramentas para a coleta de dados. Posteriormente, é descrito o tratamento dos dados.

5.1. Contexto da Pesquisa

Esta pesquisa tem como ponto de partida, a vivência das pesquisadoras nos seus ambientes de trabalho, nos quais constatou-se que a falta de estimativas de tamanho de software acarreta em um não cumprimento dos prazos e orçamentos especificados no início do projeto de desenvolvimento de software.

Para realização deste trabalho, foram utilizadas uma pesquisa bibliográfica e uma pesquisa de campo exploratória e quantitativa, cujo objetivo foi o de adquirir maiores informações, obter uma medição das empresas que utilizam a APF e aprimorar idéias do tema proposto. Conforme Andrade (2004), a utilização de uma métrica auxilia no gerenciamento de projetos. Através dela, os projetos podem ser medidos quantitativamente e qualitativamente, gerar estimativas de prazo, custo e recursos. Pressman (1995) afirma que, sem as medições quantitativas, o gerente de projeto não conhece a dimensão do que será produzido e, por conseqüência, não realizará um bom gerenciamento. Hazan (2005) complementa que as informações coletadas para o desenvolvimento de software são de suma importância para a definição das estimativas do projeto. Sem essas informações, o gerente de projeto fica impossibilitado de obter medições quantitativas e qualitativas.

Em relação ao modo dos objetivos da pesquisa, definiu-se a pesquisa exploratória, uma vez que proporciona maiores informações sobre o assunto abordado. Concomitante, aplicou-se a pesquisa bibliográfica ou de fontes secundárias, cuja finalidade foi o contato direto com tudo o que foi escrito, sobre determinado assunto, que consiste nas bibliografias tornada públicas em relação ao tema de estudo.

Pois, de acordo com Medeiros (2003), “Pesquisa bibliográfica de fonte secundária, é aquela que busca o levantamento de livros e revistas de relevante interesse para a pesquisa que será realizada”.

O levantamento bibliográfico foi retirado de material constituído principalmente de livros, pesquisas de textos de sites, artigos científicos, monografias de graduação, mestrado e doutorado.

Segundo Marconi e Lakatos (2003), estudo exploratório são investigações de pesquisa empírica cujo objetivo é a formação de questões ou de um problema, com tripla finalidade desenvolver hipóteses, aumentar familiaridade do pesquisador com o ambiente, fato ou fenômeno, para realização de uma pesquisa futura mais precisa ou para modificar conceitos já estabelecidos. Geralmente são empregados procedimentos sistemáticos ou para a obtenção de observações empíricas ou para a análise de dados (ou ambas, simultaneamente). A partir desse estudo são obtidas descrições tanto qualitativas quanto quantitativas do objeto de estudo.

O método de abordagem, conforme afirma Marconi (2003), “é o conjunto de procedimentos utilizados na investigação de fenômenos ou no caminho para chegar-se à verdade”. Atento ao exposto, o método utilizado para esta pesquisa foi o indutivo, o qual partindo dos dados particulares, suficientemente constatados, observou-se atentamente os fatos e fenômenos, conduzindo a uma conclusão particular. Medeiros (2003), considera que o método indutivo é o raciocínio em que chega-se a uma conclusão genérica subtraído de fatos particulares.

O estudo envolveu em observar e identificar nas organizações a quantidade de funcionários, faturamento em 2006, setor em que se enquadra. Algumas

informações importantes como o faturamento anual, número de funcionários, inibiu algumas participações potenciais. No entanto a amostra é suficiente e significativa.

5.2. Instrumento da Pesquisa

Para medir a utilização da APF em Salvador, foi utilizado um questionário, que foi submetido às empresas por meio de correio eletrônico, onde o gerente de projeto respondia de forma quantitativa às perguntas. Os modelos de qualidade, as métricas funcionais, os pontos fortes e os pontos fracos da APF foram selecionados de acordo com as informações levantadas no referencial teórico. O questionário aplicado na pesquisa está no anexo A. Nele consta o nome da empresa, o tipo da empresa, o faturamento anual, a quantidade de funcionários, a data de fundação da empresa, as atividades da organização no tratamento de software, as certificações ou avaliações de aderência a algum modelo de qualidade, as métrica(s) utilizada(s) pela empresa para realizar estimativas de tamanho de software, as métrica(s) utilizada(s) pela empresa para realizar estimativas de manutenção de software, os profissionais com certificado CFPS, os profissionais em processo de certificação CFPS, os tipos de projetos que a empresa aplica pontos por função, os pontos fortes e os pontos fracos na visão dos gerentes de projetos que utilizam a metodologia, a utilização do fator de ajuste pela empresa e o motivo pelo qual a empresa entrevistada não utiliza pontos por função.

5.3. Coleta de Dados

O estudo teve como instrumentos fundamentais para a coleta de dados, a observação e a entrevista com gerentes de projetos de organizações que produzem software, situadas na cidade de Salvador, estado da Bahia. A pesquisa de campo contou com a participação de 18 empresas. Além das entrevistas, foi utilizado correio eletrônico para a aplicação dos questionários.

A coleta dos dados da pesquisa foi realizada no período de 01/12/2006 a 20/12/2006. Após o seu recebimento, os questionários eram digitados em planilhas eletrônicas, de forma a facilitar a análise estatística e a geração dos gráficos. A coleta de dados baseou-se por meio de questionários individuais, com perguntas fechadas, mas com suficientes alternativas para abrigar uma resposta clara e pertinente sobre o tema em estudo, respaldados nos referenciais teóricos apresentados nos capítulos 2, 3 e 4 desta monografia. A partir da compilação das respostas, analisaram-se os seus resultados, os quais se encontram explicitados no capítulo 6.

5.4. Tratamento de Dados

A análise dos dados verifica a presença de questionários incompletos, de forma a retirar os questionários que não se encaixam por algum motivo.

Foram encontrados dois questionários incompletos, retirados da amostra. Vale ressaltar que a data completa da fundação da empresa não interferia para a aceitação do questionário. O ano, por si só, validava a pesquisa.

6. PESQUISA EXPLORATÓRIA

6.1. Caracterização das Organizações

- UNITECH

Criada em 1995, com a proposta de oferecer soluções e serviços com alto valor agregado, a Unitech tornou-se uma das maiores provedoras de TI do Brasil. Como *Full IT Service Provider*, nossa oferta engloba um amplo leque de soluções em Tecnologia da Informação e Comunicação (TIC), com qualificação em engenharia de software e gestão de ambientes e serviços de TI. O compromisso e o foco na satisfação de seus clientes são a marca registrada da empresa. A Unitech possui uma equipe com cerca de 1400 profissionais atuando na sua matriz em Salvador, nas filiais em São Paulo, Belo Horizonte, Brasília e Rio de Janeiro, e em clientes nas mais variadas localidades do território nacional.

- INTERATIVA INTERNET

A Interativa Internet entrou no mercado no ano de 2001 com a visão de ocupar uma posição de destaque em desenvolvimento de sistemas de internet, intranet e extranet. A infra-estrutura que envolve os sistemas da Interativa exige que os seus profissionais possuam conhecimentos em redes e sistemas operacionais, dentre os quais o Linux. Este conhecimento também é utilizado pelos clientes que atualmente contratam nossos serviços especializados de configuração de servidores e administração de sistemas.

- PA ARQUIVOS

A PA Arquivos trabalha com a tecnologia COLD ou ERM (Gerenciamento Corporativo de Relatórios) como vem sendo chamado que possibilita o gerenciamento de relatórios na forma digital. A empresa desenvolve sistema para outras empresas.

- ENTREPRISE

A Entreprise é uma empresa que oferece Soluções em Tecnologia, em atuação no mercado desde 2004. A empresa é voltada para o setor de Suporte de informática e Desenvolvimento web, tendo a sua frente profissionais experientes nesses segmentos.

- DIGINET

A DIGINET atua no mercado desde 1991, sendo destacada por ser a maior rede de escolas de informática em Salvador. A empresa voltada para o setor de Treinamento e Suporte de informática, tendo a sua frente profissionais experientes nesses segmentos, com know-how e metodologias próprias que vão desde o contato inicial com os clientes, acompanhamento, até a conclusão dos treinamentos e serviços. A empresa desenvolve software para uso próprio e para outras empresas.

- STEFANINI

Desde sua fundação em 1987, a Stefanini vêm tendo uma rápida e contínua expansão devido ao reconhecimento do mercado aos seus princípios de profissionalismo, ética e busca incessante pela qualidade. Hoje, a Stefanini é um grupo multinacional que conta com mais de 4.000 colaboradores em diversos países e é conceituada pelo mercado como uma das melhores consultorias estratégicas de Tecnologia da Informação.

- EXTREME SOFTWARE

A Extreme é uma empresa especializada em construções de softwares que tem como objetivo principal oferecer aos seus clientes um serviço de TI diferenciado pela sua qualidade técnica. Como forma de adequar-se às normas internacionais de processos para produção de software, a Extreme vem investindo na preparação e qualificação da mão-de-obra, e em consultorias especializadas para alcançar melhorias operacionais através das certificações ISO 9000:2000 e CMMI.

- CONSIST

A CONSIST nasceu do espírito empreendedor do empresário Natalio Fridman, em 1972, no Brasil. Nesse ano, era somente uma casa na Rua Peixoto Gomide, no bairro paulistano dos Jardins, em que jovens profissionais de informática começavam a difundir no Brasil a tecnologia de "Banco de Dados", a qual era muito inovadora para a época. A CONSIST registrou seu nome na história da informática brasileira pelo seu pioneirismo em criar soluções corporativas que se atualizaram, evoluíram e preservaram o investimento efetuado pelos seus clientes. Um líder de mercado não surge apenas da qualidade de seus produtos, mas também da dedicação dos 650 colaboradores no Brasil e da satisfação de seus clientes os quais são a razão da existência da CONSIST. A CONSIST em parceria com a RM Sistemas comercializam software educacionais tanto *desktop* quanto soluções via web.

- SECTI/PID

À Secretaria de Ciência, Tecnologia e Inovação do Estado da Bahia cabem ações que promovam o desenvolvimento científico e tecnológico nas áreas social, econômica, ambiental e institucional. Criada em dezembro de 2003, a SECTI vem coordenando projetos de incentivo à pesquisa científica, à inclusão digital, à qualificação de profissionais, ao desenvolvimento sustentável e ao aumento da qualidade de vida. O PID - Programa de Identidade Digital possui projetos para a pesquisa, desenvolvimento e aplicação de softwares livres, computadores e plataformas computacionais de baixo custo.

- 220i

A 220i é uma empresa que oferece Soluções em Tecnologia, com atuação no mercado desde 2002. A empresa tem como foco Desenvolvimento de pacotes para comercialização e desenvolvimento web.

- EMBASA

A Empresa Baiana de Águas e Saneamento S.A. -EMBASA é uma sociedade de economia mista de capital autorizado, pessoa jurídica de direito privado, tendo como acionista majoritário o Governo do Estado da Bahia. Criada no dia 11 de maio de 1971 pela Lei Estadual 2.929, em 1975, a Embasa incorporou as extintas COMAE e COSEB, responsáveis na época pelos serviços de distribuição de água e esgotamento sanitário no Estado, está presente em 355 municípios através de contratos de concessão firmados com as prefeituras municipais.

- FREIRE

A Freire Informática trabalha há 20 anos no mercado de informática e atendem a mais de 1.200 entidades públicas em todo o país com um único objetivo: ajudar a gerir com mais eficiência e operacionalidade. Desenvolve sistemas para a administração pública, oferecem também treinamento especializado, suporte on-line, locação de sistemas e equipamentos.

- INFOX

A INFOX é uma empresa de desenvolvimento e integração de sistemas fundada em 1986, com sede em [Aracaju](#) e negócios na Região Nordeste, reconhecida pela qualidade dos seus produtos e serviços, sendo certificada pela Norma ISO-9001:2000

- ZCR

A ZCR Informática foi fundada em 1992 com a missão de fornecer soluções de qualidade, através da Tecnologia da Informação, que visem a melhoria da produtividade e o bem-estar do ser humano nas organizações e no meio em que estão inseridas. A ZCR conquistou o laudo positivo CMMI, Nível 2, laudo internacional de qualidade em desenvolvimento de software, que visa padronizar todo o fluxo de desenvolvimento de um produto, durante todas as etapas de vida de um projeto.

- ALTIS

O Centro de Alta Tecnologia e Inovação em Software – ALTIS é uma Organização da Sociedade Civil de Interesse Público - OSCIP, com sede em Salvador – Bahia, voltada à promoção do desenvolvimento auto-sustentado de profissionais e empresas do mercado baiano de Tecnologia da Informação – TI. O ALTIS tem como principal objetivo promover a implantação e o desenvolvimento de um pólo de exportação de software e serviços de TI na Bahia.

- PRODEB

A Companhia de Processamento de Dados do Estado da Bahia - Prodeb foi criada em 1º de outubro de 1973 (lei 3.157) pelo então governador do Estado, Antônio Carlos Magalhães, sob a forma de Sociedade de Economia Mista, sendo o Governo do Estado da Bahia o acionista majoritário. O objetivo da empresa: prestar serviços de processamento eletrônico de dados aos órgãos e entidades da Administração Pública Estadual e assessorá-los no uso da informática.

- IP3

A IP3 Idéias & Projetos em Tecnologia é uma empresa que oferece Soluções em Tecnologia, com atuação no mercado desde 2001. A empresa tem como foco Desenvolvimento de pacotes para comercialização e desenvolvimento web.

- CEDASC

O CEDASC - Centro de Estudos e Desenvolvimento de Tecnologias para Auditoria, é uma autarquia criada em dezembro de 1988, através da Lei nº 4.819 e tem por finalidade o assessoramento e a execução da política de informatização das atividades meio e fim do Tribunal de Contas do Estado da Bahia.

6.2. Resultado da Pesquisa Exploratória

O objetivo desta pesquisa é o de analisar questões que estão diretamente ligadas a métodos de estimativas de tamanho de software, em especial a APF, atualmente

considerada como referência na gestão de projetos de software. Os gerentes de projetos das organizações pesquisadas responderam às perguntas do questionário, viabilizando a medição da utilização da Análise de Pontos por Função em Salvador/Bahia. Essa atividade fez com que os gerentes de projetos tivessem a oportunidade de refletir sobre a realidade vivida por eles, no que diz respeito à utilização de métricas para estimar os softwares produzidos, além de propiciar a avaliação desses líderes de projetos às vantagens e desvantagens a cerca da APF, conforme identificadas no referencial teórico.

As questões foram elaboradas consoante fundamentação teórica e aplicação do questionário em anexo, que foi respondido por uma amostra de 18 empresas, que correspondem à aproximadamente 48,7% do universo pesquisado. O questionário foi enviado para 37 empresas, 10 empresas públicas, 23 empresas privadas e 4 empresas do terceiro setor. A aplicação do questionário, em algumas organizações, foi através do correio eletrônico sendo que 7 e-mails retornaram com erro, equivalendo a 19% de e-mails inválidos.

No gráfico 01 observa-se o tipo da empresas que responderam à pesquisa, com o total de: 4 são empresas públicas, 13 são empresas privadas e 1 é empresa do terceiro setor.

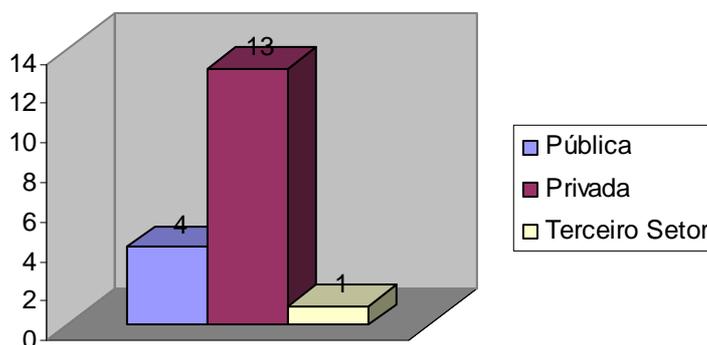


Gráfico 01 – Tipo de Empresas

Referente ao faturamento anual representada no gráfico 02, as organizações pesquisadas estão compreendidas por: 2 empresas possuem o faturamento anual menor que um milhão, 6 empresas possuem o faturamento anual de um milhão até

quinhentos milhões, outras 2 empresas possuem o faturamento anual de quinhentos e um milhões até novecentos milhões, 6 empresas possuem o faturamento anual acima de novecentos e um. As 2 empresas restantes são representadas por organizações que não apresentam faturamento anual.

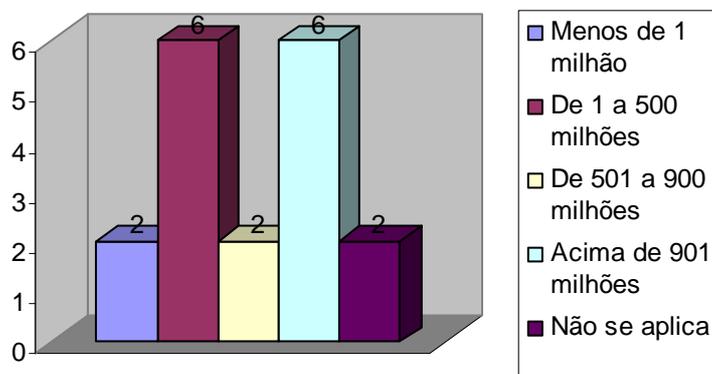


Gráfico 02 – Faturamento anual

O gráfico 03 aponta a quantidade de funcionários locados na empresa. Verifica-se que 1 empresa possui de 01 a 09 funcionários, 6 empresas possuem de 10 a 49 funcionários, 5 empresas possuem de 50 a 99 funcionários, 6 das empresas possuem acima de 100 funcionários.

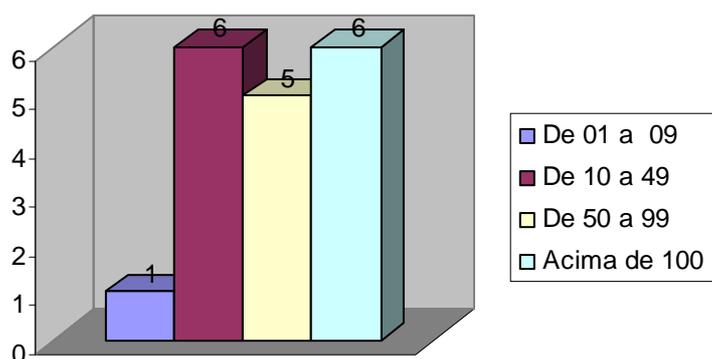


Gráfico 03 – Quantidade de Funcionários

De acordo com o SEBRAE (2006), as empresas que possuem de 0 a 9 funcionários são consideradas micros empresas, as empresas que possuem de 10 a 49 funcionários são consideradas pequenas empresas, as empresas que possuem de 50 a 99 funcionários são consideradas médias empresas, e as empresas que possuem acima de 100 funcionários são consideradas grandes empresas.

A partir dessas definições, observa-se no gráfico 04 que 1 empresa é considerada micro empresa, 6 são consideradas pequenas empresas, 5 são consideradas médias empresas e 6 são consideradas grandes empresas.

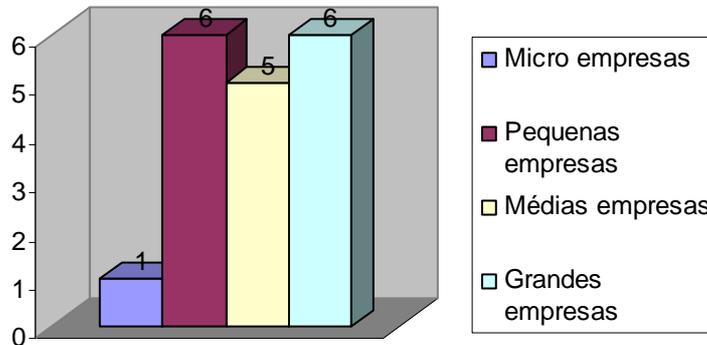


Gráfico 04 – Classificação das empresas

No gráfico 05 é apresentado o ano de fundação das empresas: 3 empresas foram fundadas no ano de 1900 até 1980, 5 empresas foram fundadas no ano de 1981 até o ano de 1990 e 10 das empresas pesquisadas foram fundadas de 1991 a 2006.

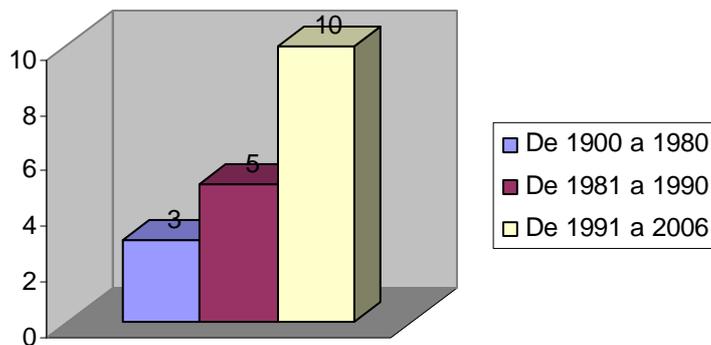


Gráfico 05 – Ano de fundação das empresas

A segunda parte da pesquisa compreendeu em abordar alguns aspectos-chaves considerados impactantes no processo de desenvolvimento de software, apresentados a seguir, através de gráficos, que permitem uma rápida e fácil assimilação dos resultados obtidos, bem como em alguns aspectos analisados, a sustentação teórica necessária para subsidiar uma informação concisa.

O gráfico 06 apresenta as atividades desenvolvidas pelas empresas, podendo uma única empresa realizar mais de uma atividade. Observa-se que 13 empresas desenvolvem software para uso próprio, 2 desenvolvem software-pacote para comercialização, 11 desenvolvem software sob encomenda para terceiros, 17 desenvolvem software para Internet, 2 empresas são distribuidoras ou editoras de software de terceiros.

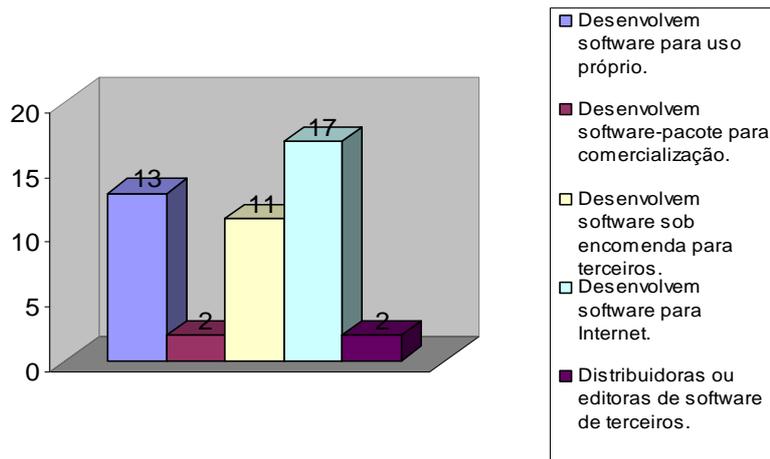


Gráfico 06 – Atividades da organização no tratamento de software

A questão de número 3 com múltiplas escolhas (anexo A), possibilitou identificar todas as atividades das organizações no tratamento de software. Constatou-se que 4 das empresas pesquisadas desenvolvem software para uso próprio e desenvolvem software para internet, 3 empresas desenvolvem software sob encomenda para terceiros e desenvolvem software para internet, apenas uma empresa desenvolve software para uso próprio, 8 empresas desenvolvem software para uso próprio, desenvolvem software sob encomenda para terceiros e desenvolvem software para internet e 2 empresas desenvolvem software-pacote para comercialização, desenvolvem software para internet e são distribuidoras ou editoras de software de terceiros.

Vale ressaltar que este item se relaciona diretamente com o atual mercado de desenvolvimento de software no país. Conforme Andrade (2004), as empresas brasileiras vêm desenvolvendo diversas atividades de software, como as apresentadas no gráfico 06, para poder competir com os grandes concorrentes

internacionais. Por conseqüência desta realidade, as organizações têm-se preocupado cada vez mais com a melhoria da qualidade dos seus produtos de software. Com base nessa afirmativa, observa-se no gráfico 07, a utilização a uma certificação ou a algum modelo de qualidade que as empresas localizadas em Salvador/Bahia têm aderido.

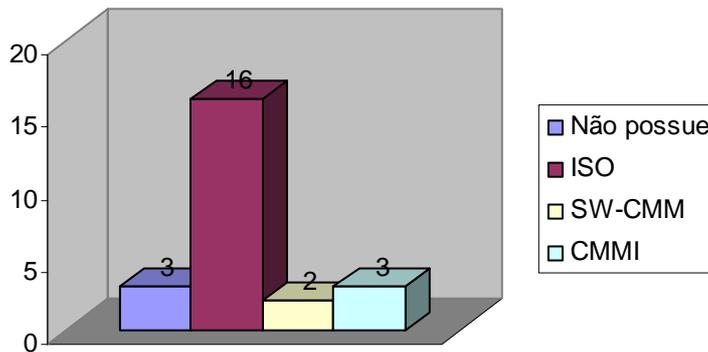


Gráfico 07 – Certificações e Modelos de Qualidade

Foi observado no gráfico 07 que, 3 empresas não possuem certificação ou avaliação de aderência a algum modelo de qualidade, 16 empresas possuem certificação ISO, 2 possuem certificação SW-CMM e 3 empresas possuem certificação CMMI. A opção “outros”, existente no questionário, não foi utilizada por nenhuma empresa.

Por ser uma questão de múltipla escolha, identificou-se que 11 das empresas pesquisadas possuem apenas a certificação ISO, apenas uma empresa possui certificação ISO, SW-CMM e CMMI, 2 empresas possuem certificação ISO e CMMI e uma empresa possui certificação ISO e SW-CMM. As três empresas restantes não possuem nenhum tipo de certificação.

Quanto às métricas utilizadas para estimar o tamanho do software, apresentadas no gráfico 08, foi observado que 16 empresas utilizam hh (homem hora)/ esforço, 1 utilizam APF – versão 4.1, 4 utilizam a APF- versão 4.2, 2 utilizam a UCP e 1 utiliza outra métrica, tal como a metodologia específica da IBM.

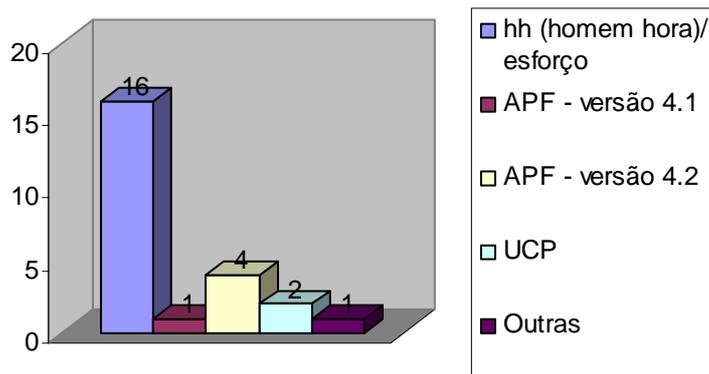


Gráfico 08 – Métricas para estimativas de tamanho de software

A questão permitiu medir o atual cenário de utilização das métricas de estimativas de tamanho de software nas empresas de Salvador. Do total pesquisado, 2 empresas utilizam hh (homem hora/ esforço), APF versão 4.1 e APF versão 4.2 para estimar o tamanho do software, 1 empresa utilizam hh (homem hora/ esforço) e APF versão 4.2, 1 utiliza apenas a APF versão 4.2, 13 utilizam apenas hh (homem hora/ esforço) e 1 utiliza outra métrica, tal como: metodologia específica da IBM.

Essas métricas são consideradas de grande importância para a engenharia de software, pois desempenham papéis fundamentais, uma vez que, estimar tamanho do software, dimensiona o que será produzido. Pressman (1995), Vasquez *et al* (2003) e Andrade (2004), afirmam que a dimensão do software previne o processo de desenvolvimento contra as incertezas do projeto, tendo em vista que, a partir das métricas haverá um melhor gerenciamento em todo o processo.

Conforme mostra o gráfico 09, quanto às métricas utilizadas pelas empresas para estimar a manutenção do software, 2 empresas utilizam o IFPUG – APF, 4 utilizam a NESMA – APF, 16 utilizam hh (homem hora) / esforço e 1 utiliza outra métrica, tal como: metodologia específica da IBM.

Atualmente, 2 empresas em Salvador utilizam três tipos de estimativas diferentes para estimar a manutenção do software: a IFPUG- APF, a NESMA-APF e a hh (homem hora)/ esforço, 1 empresa utiliza duas métricas: a IFPUG- APF e a NESMA-APF, 1 utiliza apenas a NESMA-APF, 13 empresas utilizam apenas a hh (homem

hora)/ esforço e apenas 1 empresa utiliza uma outra métrica: a metodologia específica da IBM.

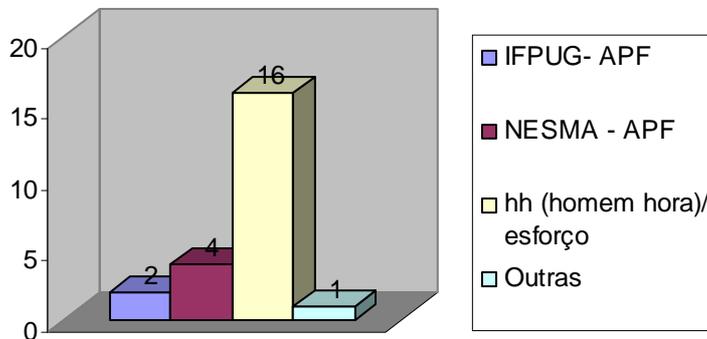


Gráfico 09 – Métricas para estimar a manutenção do software

Diante do gráfico 10, verifica-se que 14 empresas não possuem profissionais certificados em CFPS, 2 das empresas possuem de 1 a 3 profissionais certificados em CFPS, 2 empresas possuem de 4 a 6 profissionais certificados em CFPS, não havendo empresas de 7 a 10 profissionais certificados em CFPS e nem empresas com mais de 10 profissionais certificados em CFPS.

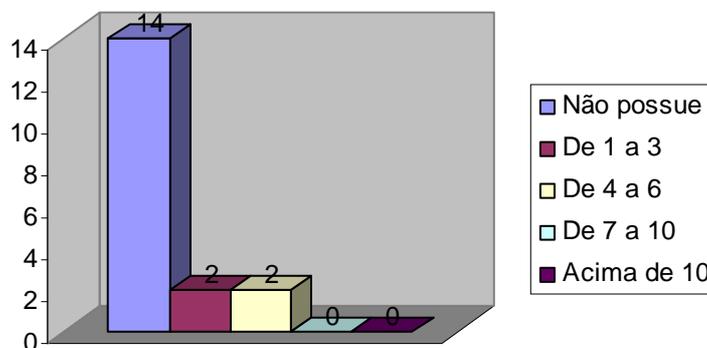


Gráfico 10 - Profissionais certificados em CFPS

Diante desses dados, observa-se que as empresas de Salvador/Bahia possuem poucos profissionais qualificados em pontos por função. Conforme notícia disponibilizada na página do Ministério da Ciência e Tecnologia (MCT, 2006), nos últimos anos, a quantidade de profissionais certificados em CFPS aumentou consideravelmente no sul e sudeste. Em consequência das empresas brasileiras preocuparem-se com a melhoria da qualidade dos produtos de software, objetivando

também atingir o mercado internacional de software, as normas e os modelos de maturidade de processo de software são cada vez mais utilizados como referência. O gráfico 11, apresenta 15 empresas que não tem profissionais em processo de certificação, 2 empresas que possuem de 1 a 3 profissionais em processo de certificação CFPS, 1 empresa que possui de 4 a 6 profissionais em processo de certificação CFPS, não havendo empresas em Salvador que possuam de 7 a 10 profissionais em processo de certificação em CFPS e nem empresas que possuam acima de 10 profissionais em processo de certificação em CFPS.

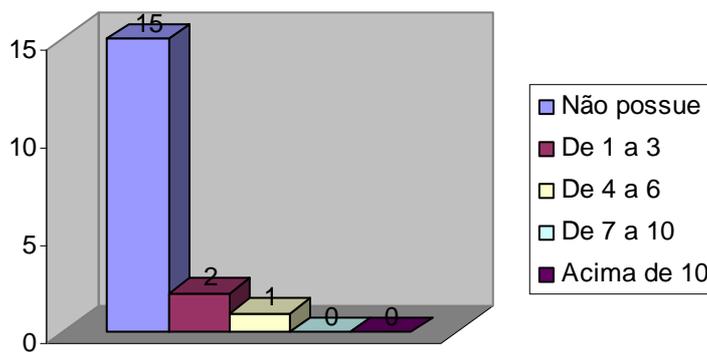


Gráfico 11 - Profissionais em processo de certificação CFPS

Conforme pesquisa realizada, a partir da questão de número 4 até a questão de número 8 (anexo A), foi constatado que empresas que adotam modelos de qualidade, tais como: SW- CMM e CMMI, utilizam pelo menos uma métrica para estimar tamanho de software ou uma métrica para estimar a manutenção de software, métricas estas que foram originadas da Análise de Pontos por Função. Daí a afirmação de que a análise de pontos por função apóia tanto a produtividade quanto a qualidade dos produtos de software.

A terceira parte da pesquisa compreendeu em abordar aspectos relacionados diretamente com a utilização da Análise de Pontos por Função, apresentados a seguir através de gráficos, que permitem uma rápida e fácil assimilação dos resultados obtidos, bem como os tipos de projetos que as empresas de Salvador/Bahia utilizam a APF, os pontos fortes e os pontos fracos observados pelos gerentes de projetos.

Quanto aos tipos de projetos desenvolvidos pelas empresas, conforme demonstra o gráfico 12, 3 empresas utilizam a APF em aplicações cliente/servidor, 4 empresas utilizam a APF em aplicações baseadas na web (Internet), 1 empresa utiliza a APF em aplicações de prospecção de dados, 2 empresas utilizam a APF em aplicações de interface gráfica, 3 empresas utilizam a APF em aplicações orientadas a objetos, 2 empresas utilizam a APF em aplicações *desktop*.

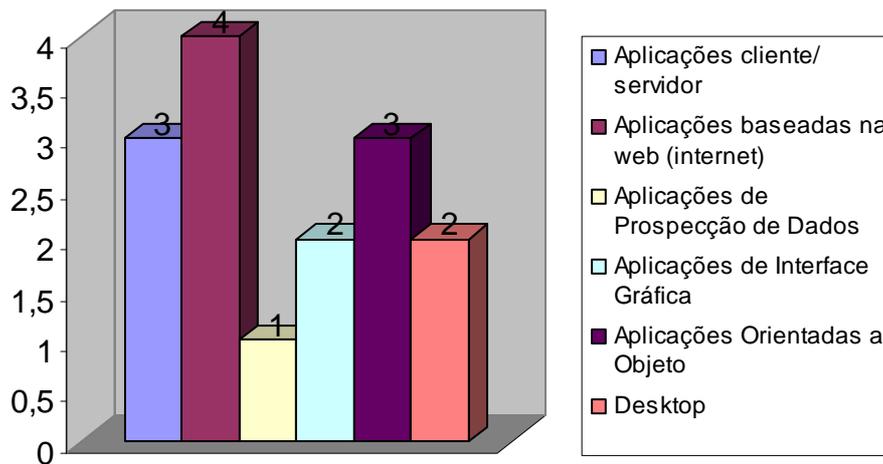


Gráfico 12 - Tipos de projetos

A questão de número 9 com múltiplas escolhas (anexo A) possibilitou identificar em quais tipos de projetos às empresas utilizam a APF para estimar o tamanho do software. Verificou-se que das 4 empresas em Salvador/Ba, 1 empresa utiliza a APF em aplicações baseadas na *web* (Internet), em aplicações de prospecção de dados e em aplicações orientadas a objeto, 1 empresa utiliza a APF em aplicações baseadas na *web* (Internet) e em aplicações cliente/ servidor e 2 empresas utilizam a APF em aplicações cliente/ servidor, em aplicações baseadas na *web* (Internet), em aplicações orientadas a objetos, em Desktop e em aplicações de Interface gráfica.

O gráfico 13 revela quais os pontos fortes da metodologia APF, levantados no referencial teórico, mais identificados pelos gerentes de projetos nas empresas que produzem software em Salvador/Bahia. Verificou-se que das 4 empresas que utilizam a APF em Salvador/Ba, 3 apontam a precisão da estimativa de tamanho de uma aplicação como fator crucial da APF, 4 apontam o fator de normalização para

comparação de software, 2 apontam o meio para estimar custo e recursos, 1 aponta a manutenção de dados históricos, 2 apontam a aplicabilidade nas diversas fases, apontam a independência da tecnologia.

A questão de número 10 com múltiplas escolhas (anexo A) possibilitou identificar os pontos fortes da metodologia, conforme visão dos gerentes de projetos de cada empresa que respondeu ao questionário. Uma empresa identificou apenas o fator de normalização para comparação de software como ponto forte da metodologia. Uma outra empresa identificou três das alternativas apresentadas no questionário como pontos fortes da metodologia APF, a precisão da estimativa de tamanho de uma aplicação, o fator de normalização para comparação de software e a manutenção de dados históricos. As outras duas últimas empresas identificaram a precisão da estimativa de tamanho de uma aplicação, o fator de normalização para comparação de software e a manutenção de dados históricos, a manutenção de dados históricos, a aplicabilidade nas diversas fases e a independência de tecnologia.

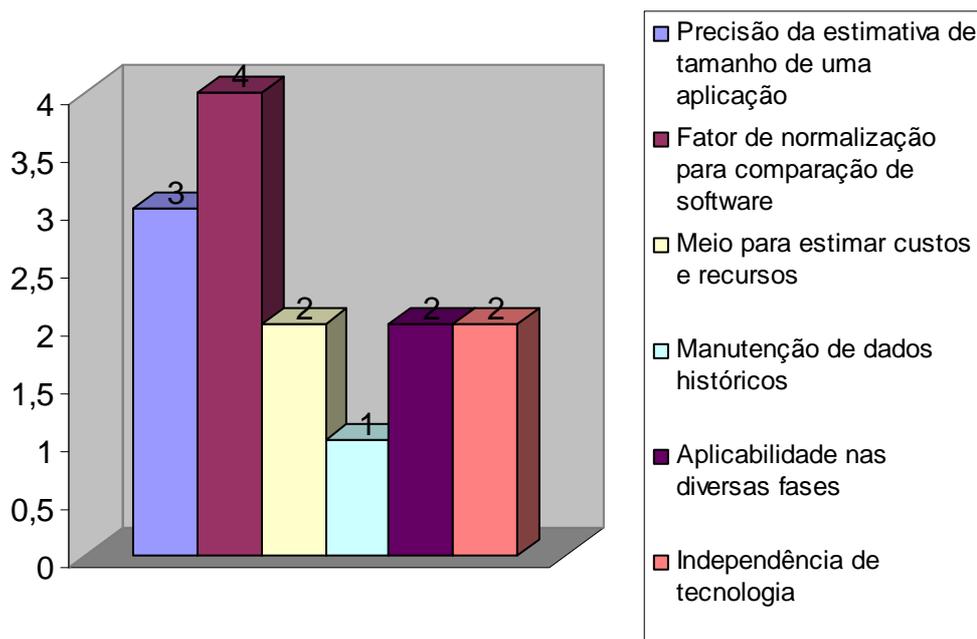


Gráfico 13 – Pontos Fortes da metodologia

O gráfico 14 revela quais os pontos fracos da metodologia APF, levantados no referencial teórico, mais identificados pelos gerentes de projetos nas empresas que produzem software em Salvador/Bahia. Verificou-se que das 4 empresas que utilizam a APF em Salvador/Ba, 3 apontam o esforço de contagens de pontos de função para cada fase como ponto fraco, 3 apontam as características gerais do

sistema, por não serem consideradas únicos fatores para cálculo do fator de ajuste, 2 apontam os aspectos algoritmos dos processos elementares.

A questão de número 11 com múltiplas escolhas (anexo A) possibilitou identificar os pontos fracos da metodologia, conforme visão dos gerentes de projetos de cada empresa que respondeu ao questionário. Uma empresa identificou as características gerais do sistema como o ponto fraco da APF, por não serem os únicos fatores a serem considerados para cálculo do fator de ajuste. Outra empresa identificou o esforço de contagens de pontos de função para cada fase, e outras duas empresas identificaram as características gerais, o esforço de contagens e o fato de não considerar suficientemente o aspecto algoritmo dos processos elementares.

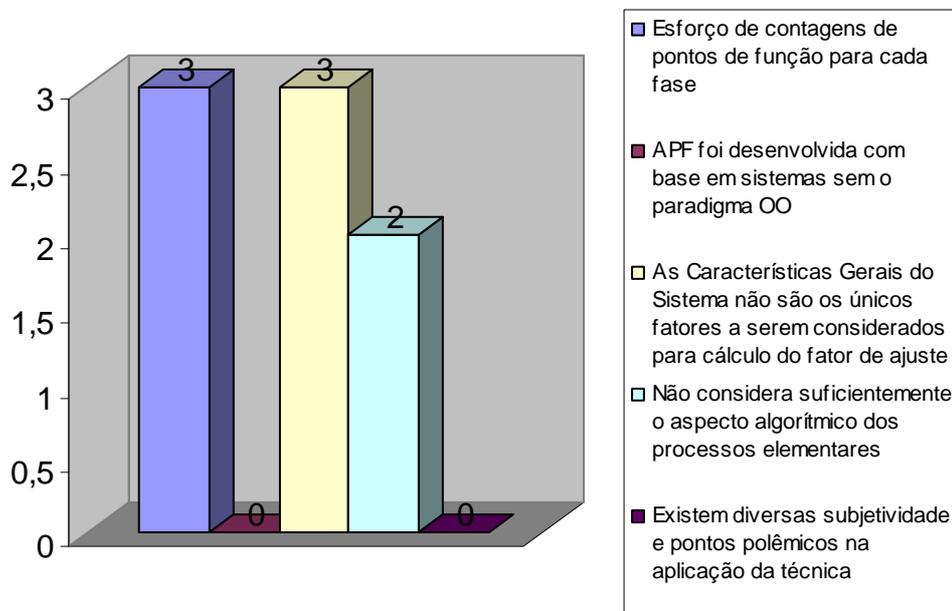


Gráfico 14 – Pontos fracos da metodologia

De acordo com o BFPUG (2006), os benefícios da APF sobrepõem os pontos fracos levantados à cerca desta metodologia. Com a utilização da APF pode-se analisar a produtividade, a qualidade dos sistemas e a medição de software em todas as fases do projeto. Após a aplicação do questionário e entrevistas com os gerentes de projetos, percebe-se que a utilização da APF trouxe melhoria no processo de desenvolvimento de sistemas. Em contrapartida, as empresas que não buscam a qualidade do software, seja como utilização da APF ou de qualquer outra técnica, tem seus cronogramas atrasados e os custos do projeto elevados.

Quanto à utilização do fator de ajuste, apresentada no gráfico 15, observa-se que as 4 empresas que utilizam a APF nas estimativas de tamanho de software em Salvador/ Bahia, trabalham com o fator de ajuste.

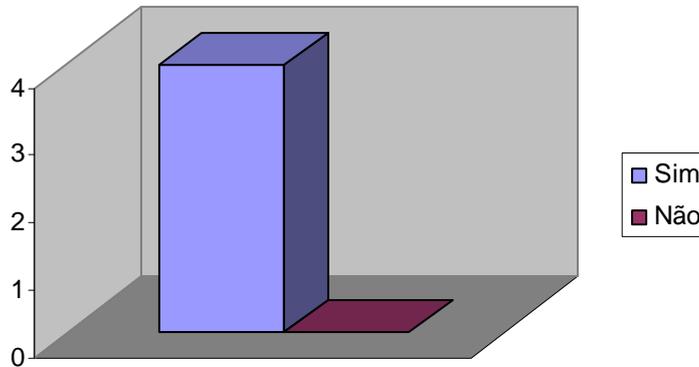


Gráfico 15 – Utilização do Fator de Ajuste

E por fim, o último item abordado foi referente a não utilização da APF. O gráfico 16 revela que 3 das empresas pesquisadas desconhecem a metodologia, 8 conhecem a metodologia mas na atual empresa onde trabalha não há profissionais capacitados e 3 empresas conhecem a metodologia mas optou por utilizar uma outra metodologia, tais como: metodologias específicas da IBM e metodologia baseada em homem hora.

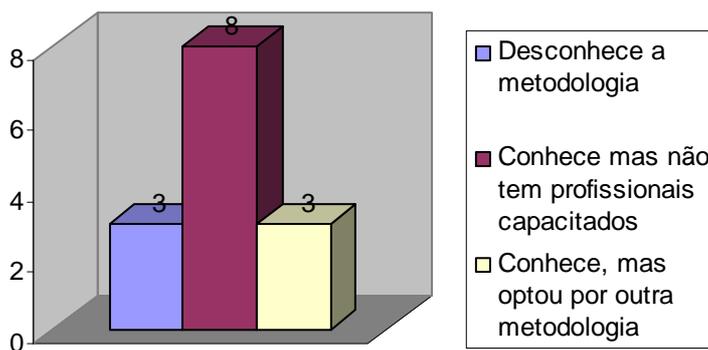


Gráfico 16 – Motivos de não utilizar a APF

Desta forma pode-se afirmar que o estado da Bahia, especificamente na cidade Salvador, as empresas que produzem software estão em busca da qualidade, com o objetivo de concorrer com empresas de âmbito nacional na comercialização de seus produtos. Entretanto, faltam profissionais capacitados para dar início a essa qualidade e manter um padrão de excelência dentro da empresa. O fato dos

investimentos nessa área serem um tanto quanto elevado e as empresas não investirem muito o seu capital na capacitação dos funcionários, espera-se muito dos profissionais recém-formados. As organizações acreditam que estudantes e/ou recém formados possam agregar novos valores a estrutura da empresa, no que diz respeito ao desenvolvimento de software, pelo fato de lidarem com assuntos mais atuais na área de tecnologia da informação. Pela falta de utilização de certas normas e padrões que objetivam um melhor processo de desenvolvimento de software, algumas empresas de Salvador/Ba acabam perdendo o mercado para empresas certificadas e ate mesmo empresas de outros estados brasileiros. Além disso, as empresas baianas acabam contratando profissionais de outros estados para atuarem no processo de desenvolvimento de software, pelo simples fato de estarem mais aptos a trabalhar na área.

7. CONSIDERAÇÕES FINAIS

O nível de utilização da APF em Salvador/Bahia é difícil de elucidar, conforme indicado em diversas partes deste trabalho, mesmo com a necessidade de utilização de métricas de estimativas de projetos de software, palestras sobre o tema, encontros nacionais. Com esta pesquisa, acredita-se ter agregado novas evidências ao conhecimento estabelecido sobre o uso da APF, provendo uma comprovação empírica do mesmo.

Um dos objetivos propostos foi identificar o percentual de utilização da APF, nas organizações que produzem software na cidade de Salvador, estado da Bahia através da utilização de um questionário. Foi encontrado uma média de utilização da APF de 22;22%, apenas 4 das 18 empresas utilizam a metodologia, comprovando que um baixo percentual de empresas utilizam a APF para estimar o tamanho do software.

A presente pesquisa viabilizou obtenção de informações sobre empresas de diferentes setores. As informações obtidas permitem caracterizar onde a APF é mais difundida. Fica comprovado que as empresas estão em busca da qualidade do software, sendo que no setor privado, onde o mercado é cada vez mais competitivo um diferencial é exigido para garantir que as necessidades dos clientes sejam atingidas.

Entende-se que as métricas são utilizadas pelas empresas para dimensionar tamanho do projeto, visando obter custo do projeto e o prazo de desenvolvimento do mesmo. As métricas são utilizadas também como indicadores de qualidade de software, permitindo um melhor gerenciamento em todo ciclo do projeto. Sendo assim, percebemos que a Análise de Pontos por Função é uma metodologia extremamente rica, que requer profissionais qualificados e certificados para sua utilização. Com as informações levantadas pela pesquisa e através de entrevistas, verificou-se ainda, que os pontos fortes da metodologia sobrepõem os pontos fracos, ou seja, mesmo com o esforço exigido para a contagem e a utilização da Análise de Pontos por Função em cada fase, ainda assim, sua precisão nas estimativas são consideradas muito altas.

Constatou-se também, que empresas com certificações ou algum modelo de qualidade, possuem procedimentos e padrões que proporcionam um bom nível de qualidade, permitindo um melhor gerenciamento organizacional.

Diante de todo levantamento teórico-prático durante esta investigação científica, ressalta-se que apesar da baixa utilização da APF nas empresas que desenvolvem software em Salvador/Ba, seu grau de precisão nas estimativas são consideravelmente bem utilizadas no controle e acompanhamento de projetos.

Para complementar o trabalho de validação proposto nesta pesquisa é necessário que mais amostras sejam coletadas para análise, pois a pequena quantidade das mesmas gerou uma limitação na pesquisa. Investigações e coletas de dados voltados a este objetivo abordado terão relevante importância para toda comunidade científica.

REFERÊNCIAS

ANDRADE , Edméia Leonor Pereira de: Pontos de Casos de Uso e Pontos de Função na Gestão de Estimativa de Tamanho de Projetos de Software Orientados a Objeto. Proposta de Tese de Mestrado defendida em 2004 na Universidade Católica de Brasília, Brasília, Brasil.

BFPUG. Brazilian Function Point User Group, www.bfpug.com.br, 2006.

COSMIC. COSMIC-FFP Measurement Manual: Versão 2.2. Common Software Measurement International Consortium. Jan, 2003.

HAZAN, C.: Método para Aplicação da Contagem de Pontos de Função como Suporte a Engenharia de Requisitos. Proposta de Tese de Doutorado defendida em Dezembro/2004 na PUC-Rio, Rio de Janeiro, RJ, Brasil

HAZAN, C., Staa, A.: Análise e Melhoria de um Processo de Estimativas de Tamanho de Projetos de Software. Monografias em Ciências da Computação nº 04/05, Departamento de Informática PUC-Rio, Rio de Janeiro, RJ, Brasil, ISSN 0103-9741, Fevereiro/2005

IFPUG. Function Point Counting Practices Manual: Versão 4.2. International Function Point Users Group, www.ifpug.org, jan., 2000.

ISBSG - International Software Benchmarking Standards Group. The Benchmarking Repository, Release 6. ISBSG. Abr,2002.

_____. ISO/IEC DTR 16326: Software Engineering:Guide for the application of ISO/IEC 12207 to project management.1999.31 p.

_____. ISO/IEC TR 15504: Information technology: Software process assessment – part 1: concepts and introductory guide. 1998.

LIMA, Osias de Souza: Análise de Pontos por Função Fuzzy. Proposta de Tese de Mestrado defendida em Junho/2002 na UNIFOR, Fortaleza, CE, Brasil.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. Fundamentos de Metodologia Científica. 5º Ed. São Paulo: Atlas, 2003.

MCT. MINISTÉRIO DE CIÊNCIA E TECNOLOGIA. Notícias de TI: mercado de software crescerá 5,3% em 2004. Notícia publicada na Computerworld em 12/11/03. 2003a. Acesso em 06/08/2006. Disponível em: http://www.mct.gov.br/Temas/info/Imprensa/Noticias_Anteriores/Noticias_2003/Software_2003.htm

MEDEIROS, João Bosco, Redação Científica. 5ª Ed. São Paulo: Atlas, 2003

NESMA. Netherlands Software Metrics Users Association, www.nesma.nl, 2006.

PAVAN, Theo Ignez: O Uso de Análise de Pontos por Função no Planejamento de Processo de Desenvolvimento de Software. Monografia em Ciências da Computação, Departamento de Computação Londrina, Brasil, 2004.

PETERS, James F.; Pedrycz, Witold. Engenharia de Software: Teoria e Prática. Rio de Janeiro, ed. Campus, 2001.

PMBOK, Project Management Body of Knowledge, 2000. Capturado em 10/12/2006. Disponível na Internet: www.usp.br/gefim/projetos/pmbok2000.pdf

PRESSMAN, Roger S. Engenharia de Software. São Paulo, Makron Books, 1995.

PRESSMAN, Roger S. Engenharia de Software: a practitioner's approach. 3th ed. McGraw Hill, 2001.

SALVADOR, Eduardo Vinicius de Figueiredo :Implementando um Método de Estimativa de Projetos de Software no Dotproject. Monografia em Ciências da Computação, Centro de Informática, Pernambuco, Brasil, 2005.

SEBRAE. Serviço Brasileiro de Apoio às Micro e Pequenas Empresas, www.sebrae.com.br, 2006. Acessado em 11/12/2006.

SILVA I. J. de M.. Delphi 5 – Análise de Pontos por Função. Rio de Janeiro, Book Express, 2000.

TAVARES, H., Carvalho, A., Castro, J.: Medição de Pontos por Função a Partir da Especificação de Requisitos. Serpro – Empresa do Ministério da Fazenda, Universidade Federal de Pernambuco (2002).

UKSMA. Mk II Function Point Analysis Counting Practises Manual: Versão 1.3.1. United Kingdom Software Metrics Association, www.ukσμα.co.uk, set. 1998.

VASQUEZ, C. E.; SIMÕES G. S.; ALBERT R. M. Análise de Pontos por Função. São Paulo, Érica, 2003.

ANEXO A


QUESTIONÁRIO DE PESQUISA PARA TRABALHO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

O objetivo deste questionário é avaliar a utilização de métricas para estimar o tamanho do software, principalmente a APF (Análise de Pontos por Função). As informações levantadas pela pesquisa são de caráter estritamente confidencial, estando vedada a divulgação ou acesso aos dados individuais da fonte informante para qualquer empresa, órgão público ou pessoa física.

1. Caracterização da Empresa

Nome da Empresa: _____

Tipo da Empresa: Pública Privada Terceiro Setor

Cidade/ Estado: _____

2. Porte

Faturamento Anual (R\$): _____

Quantidade de Funcionários: _____

Data de Fundação da Empresa: ___/___/____

3. Atividade(s) da organização no tratamento de software

- Desenvolve software para uso próprio
 Desenvolve software-pacote para comercialização
 Desenvolve software sob encomenda para terceiros
 Desenvolve software para Internet
 É distribuidora ou editora de software de terceiros
 Outra(s), Qual (is): _____

4. Possui certificação ou avaliação de aderência a algum modelo de qualidade?

- ISO
 SW - CMM
 CMMI
 Outro(s), Qual (is): _____

5. Qual(is) a(s) métrica(s) utilizada(s) pela empresa para realizar estimativas de tamanho de software?

- hh (homem hora)/esforço
 APF – versão 4.1
 APF – versão 4.2
 UCP
 Outra(s), Qual (is): _____

6. Qual(is) a(s) métrica(s) utilizada(s) pela empresa para realizar estimativas de manutenção de software?

- IFPUG - APF
 NESMA – APF
 hh (homem hora)/esforço
 Outra(s), Qual (is): _____

7. Existe(m) profissional(is) com o certificado CFPS na sua empresa?

- De 1 até 3
 De 4 a 6
 De 7 a 10
 Acima de 10, Quantos: _____

8. Existe(m) profissional(is) em processo de certificação CFPS na sua empresa?

- De 1 até 3
 De 4 a 6
 De 7 a 10
 Acima de 10, Quantos: _____

SE UTILIZA A APF:

9. Em quais tipos de projetos?

- Aplicações cliente/servidor
 Aplicações baseadas na web (internet)
 Aplicações de Prospecção de Dados
 Aplicações de Interface Gráfica
 Aplicações Orientadas a Objeto
 Desktop
 Outra(s), Qual (is): _____

10. Na sua opinião, quais os pontos fortes da metodologia?

- Precisão da estimativa de tamanho de uma aplicação
 Fator de normalização para comparação de software
 Meio para estimar custo e recursos
 Manutenção de dados históricos
 Aplicabilidade nas diversas fases
 Independência de tecnologia
 Outra(s), Qual (is): _____

11. Na sua opinião, quais os pontos fracos da metodologia?

- Esforço de contagens de pontos de função para cada fase.
 APF foi desenvolvida com base em sistemas sem o paradigma OO.
 As Características Gerais do Sistema não são os únicos fatores a serem considerados para cálculo do fator de ajuste.
 Não considera suficientemente o aspecto algorítmico dos processos elementares
 Existem diversas subjetividade e pontos polêmicos na aplicação da técnica.
 Outra(s), Qual (is): _____

12. Usa Fator de Ajuste?

- Sim
 Não

SE NÃO UTILIZA A APF:**13. Por que?**

- Desconhece a metodologia
 Conhece, mas não tem profissionais capacitados para utilizar a metodologia
 Conhece, mas optou por outra metodologia, qual: _____