

**NIELSEN OLIVEIRA DE MORAES**

**ESTUDO DE CASO - USO DE MÉTRICAS PARA DESENVOLVIMENTO  
DE SISTEMAS INFORMATIZADOS**

Dissertação apresentada ao Curso de Pós-Graduação em Sistema de Gestão da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistema de Gestão pela Qualidade Total

Aprovada em março de 2005

**BANCA EXAMINADORA**

---

José Rodrigues de Farias Filho, D.Sc.  
Universidade Federal Fluminense

---

Stella Regina Reis da Costa, D.Sc.  
Universidade Federal Fluminense

---

Eugênio Kahn Epprecht, D.Sc.  
Pontifícia Universidade Católica do Rio de Janeiro

Niterói, Rio de Janeiro

2005

## DEDICATÓRIA

Dedico este trabalho:

A Deus por ter me escolhido.

À minha Esposa Simone, que sempre me incentivou a não permanecer parado, a sempre me movimentar a buscar meus ideais de vida, a ser correto e justo, sempre me motivando a andar segundo a vontade de Deus. Devo muito a ela.

Às minhas Filhas Felissa e Vitória, que lhes sirvam de incentivo na caminhada longa que terão pela vida em busca de estudo e de aperfeiçoamento.

Aos meus Pais Nilton e Noemia e à minha Avó Irene, que sempre incentivaram os meus estudos, e por seu exemplo, de retidão e caráter, o grande legado que me deixam e que espero deixar para minhas filhas.

À minha Sogra Eliene que sempre nos apoiou, nunca deixando faltar nada na nossa operação doméstica.

Aos meus cunhados Susana e Ronaldo e sobrinha Isabela sempre perto e sempre colaboradores.

Ao meu Tio Vicente que, talvez sem saber, sempre foi o meu exemplo.

## AGRADECIMENTOS

Ao Professor José Rodrigues, que com sua orientação tornou meu trabalho mais fácil.

Aos colegas do Serviço de Informática do Inmetro, por suas contribuições, paciência, compreensão e apoio.

Aos colegas de grupo de trabalho, pela ótima e divertida companhia.

## SUMÁRIO

<b>LISTA DE QUADROS E TABELAS .....</b>	<b>7</b>
<b>LISTA DE FIGURAS .....</b>	<b>8</b>
<b>LISTA DE SIGLAS E SÍMBOLOS .....</b>	<b>9</b>
<b>RESUMO .....</b>	<b>11</b>
<b>ABSTRACT .....</b>	<b>12</b>
<b>1 INTRODUÇÃO .....</b>	<b>13</b>
1.1 CONTEXTUALIZAÇÃO DO TEMA .....	13
1.2 PROBLEMA .....	14
1.3 OBJETIVOS .....	15
1.4 ASPECTOS TEÓRICOS .....	17
1.5 ASPECTOS METODOLÓGICOS .....	17
1.6 JUSTIFICATIVA .....	17
1.7 HIPÓTESES E QUESTÕES .....	19
1.8 DELIMITAÇÃO DO TRABALHO .....	20
1.9 ESTRUTURA DO TRABALHO .....	20
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>22</b>
2.1 CONSIDERAÇÕES INICIAIS DO CAPÍTULO .....	22
2.2 A MEDIÇÃO NA TERCEIRIZAÇÃO E NA GESTÃO DE CONTRATOS .....	22
2.3 MÉTRICAS DE CONSTRUÇÃO DE SOFTWARE .....	24
2.3.1 CONTAGEM DE LINHAS DE CÓDIGO .....	25
2.3.2 CONTAGEM DE SÍMBOLOS – SOFTWARE SCIENCE HALSTEAD .....	26
2.3.3 CONSTRUTIVE COST MODEL – COCOMO .....	29
2.3.4 MODELO DE ESTIMATIVA PUTNAM .....	32
2.3.5 MODELO DE PONTOS DE FUNÇÃO .....	33
2.3.6 MODELO FEATURE POINTS .....	37
2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	39
<b>3 METODOLOGIA CIENTÍFICA .....</b>	<b>40</b>

3.1 CONSIDERAÇÕES INICIAIS DO CAPÍTULO .....	40
3.2 ESCOLHA DO MÉTODO DE PESQUISA .....	40
3.3 DELINEAMENTO DA PESQUISA .....	41
3.4 DEFINIÇÃO DO INSTRUMENTO DE PESQUISA .....	41
3.5 DEFINIÇÃO DA AMOSTRA E FORMA DE COLETA DE DADOS .....	42
3.6 LIMITAÇÕES DA PESQUISA .....	43
3.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	43
<b>4 PERFIL DO ESTUDO DE CASO .....</b>	<b>44</b>
4.1 CONSIDERAÇÕES INICIAIS DO CAPÍTULO .....	44
4.2 O INSTITUTO .....	44
4.3 ANÁLISE DE SOLUÇÕES .....	45
4.4 VISÃO GLOBAL .....	47
4.5 PRINCIPAIS PROCESSOS .....	47
4.5.1 PLANEJAMENTO TECNOLÓGICO .....	47
4.5.2 DISPONIBILIZAÇÃO DAS SOLUÇÕES .....	48
4.5.3 ACOMPANHAMENTO DAS SOLUÇÕES E AVALIAÇÃO DOS RESULTADOS ..	49
4.6 ESTRUTURA E FUNÇÕES NECESSÁRIAS .....	51
4.7 A ÁREA DE ATENDIMENTO .....	60
4.8 APOIO AO ATENDIMENTO .....	61
4.9 CONSULTORES DE TI .....	61
4.10 AMBIENTE DE PRODUTIVIDADE .....	61
4.11 APOIO AO USUÁRIO .....	62
4.12 A ÁREA DE SUPORTE E PRODUÇÃO .....	62
4.12.1 PLANEJAMENTO E PROJETO DE INFRA-ESTRUTURA .....	63
4.12.2 SUPORTE E REDE .....	63
4.12.3 PRODUÇÃO .....	63
4.13 METODOLOGIA PARA GESTÃO DE PROJETOS DE TECNOLOGIA DA INFORMAÇÃO .....	64
4.14 METODOLOGIA PARA DESENVOLVIMENTO E MANUTENÇÃO DE SISTEMAS .....	64
4.15 METODOLOGIA DE MANUTENÇÃO DE EQUIPAMENTOS .....	65
4.16 CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	66
<b>5 ANÁLISE COMPARATIVA DAS MÉTRICAS .....</b>	<b>67</b>

5.1 CONSIDERAÇÕES INICIAIS DO CAPÍTULO .....	67
5.2 DADOS EM ANÁLISE .....	67
5.2.1 LINHA DE CÓDIGO .....	67
5.2.2 <i>COCOMO</i> .....	68
5.2.3 <i>PUTMAN</i> .....	69
5.2.4 <i>FEATURE POINTS</i> .....	69
5.2.5 <i>SOFTWARE SCIENCE</i> .....	69
5.2.6 <i>PONTOS POR FUNÇÃO</i> .....	69
5.3 ANÁLISE COMPARATIVA .....	70
5.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	71
<b>6 CONCLUSÃO .....</b>	<b>72</b>
6.1 RESULTADOS .....	72
6.2 CONCLUSÃO E CONSIDERAÇÕES FINAIS .....	72
6.3 SUGESTÕES PARA NOVOS TRABALHOS .....	73
<b>7 REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>75</b>
<b>GLOSSÁRIO .....</b>	<b>81</b>

## LISTA DE QUADROS E TABELAS

QUADRO 1	Estrutura e suas Funções .....	49
QUADRO 2	Funções e Atribuições .....	52
QUADRO 3	Comparativo de Métricas .....	70
TABELA 1	Análise de Símbolos da Sub-rotina de Ordenação .....	28
TABELA 2	Histórico de Esforço de uma Empresa .....	31
TABELA 3	Método Feature Points .....	31

## **LISTA DE FIGURAS**

FIGURA 1	Organograma para o Gerenciamento do Contrato de Terceirização .....51
----------	---



## LISTA DE SIGLAS E SÍMBOLOS

ABNT	Associação Brasileira de Normas Técnicas
BFPUG	Brazilian Function Point Users Group
CMM	Capability Maturity Model
CNN	Comitê Nacional de Normalização
COCOMO	Constructive Cost Model ou Modelo de Custo Construtivo
Conmetro	Conselho Nacional de Metrologia, Normalização e Qualidade Industrial
CPM	Couting Practices Manual ou Manual e Contagem Prática
DFJUG	Brasília Java Users Group
DIRAF	Diretoria de Administração e Finanças
EA	European Accreditation
ENUPF	Encontros Nacionais de Usuários de Pontos de Função
EUA	Estados Unidos da América
FNPQ	Fundação para o Prêmio Nacional da Qualidade
IBM	Empresa Multinacional Americana na Área de Tecnologia da Informação
IFPUG	International Function Point Users Group
ILAC	International Laboratory Accreditation Cooperation
Inmetro	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial
INPI	Instituto Nacional de Propriedade Industrial
INPM	Instituto Nacional de Pesos e Medidas
ISO	International Organization for Standardization
MDIC	Ministério do Desenvolvimento, Indústria e Comércio Exterior
MIT	Massachusets Institute of Technology
MPOG	Ministério do Planejamento, Orçamento e Gestão
NIST	National Institute of Standards and Technology
OMC	Organização Mundial do Comércio
PAT	Plano Anual de Treinamento
PBQP	Programa Brasileiro da Qualidade e Produtividade
PIB	Produto Interno Bruto
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PNM	Plano Nacional de Metrologia
PGSI	Plano Global do Sistema de Informações do Inmetro

PTB	Phisikalisch Technische Bundesanstalt
RBC	Rede Brasileira de Calibração
SEI	Software Engineering Institute
SIG	Sistema de Informações Gerenciais do Inmetro
SIPLAN	Sistema de Planejamento e Orçamento do Inmetro
SICAP	Sistema de Acompanhamento e Controle de Projetos do Inmetro
Sinmetro	Sistema Nacional de Metrologia, Normalização e Qualidade Industrial
SOFTEX	Sociedade para promoção da Excelência do Software Brasileiro
TI	Tecnologia da Informação
TIB	Tecnologia Industrial Básica
UFF	Universidade Federal Fluminense

## RESUMO

O presente trabalho tem como objetivo analisar as métricas mais comuns de desenvolvimento e manutenção de sistemas e, levando-se em consideração a ótica de um Gerente de TI de um órgão do Governo Federal. Uma vez que a velocidade de informação imposta pela globalização remete as empresas a terem sistemas informatizados isto impulsiona as empresas prestadoras de serviço a desenvolverem técnicas para aumento de produtividade. Neste processo de desenvolvimento é fundamental medir os custos de toda a produção e não só do produto final assim como é necessário às empresas controlarem os seus processos de desenvolvimento. Já para as empresas contratantes há uma necessidade muito grande de planejar e executar melhor os gastos com a informatização, assim com acompanhar o desenvolvimento e fazer as correções necessárias. Atrelado a isto vem a necessidade de conhecer as empresas prestadoras de serviço, e em particular conhecer as empresas participantes das licitações, podendo avaliar quais entenderam o problema proposto e quais tem condições de chegar ao produto desejado. Para isto apresentamos as métricas mais conhecidas, realizando um breve estudo sobre cada uma delas e ao término apontando as vantagens e desvantagens de cada uma. Foi observado ao longo do projeto que muito se fala a respeito de métricas, porém pouco é efetivamente utilizado, o que causou grande dificuldade à pesquisa. Isto remete ao pensamento do por que não são usadas as métricas uma vez que parecem dimensionar com maior clareza os trabalhos.

Palavras-chaves: Métricas. Contrato de Serviços. Desenvolvimento de Sistemas, Gestão de TI.

## ABSTRACT

The goal of this piece of work is to analyze the most common metrics used for the development and maintenance of information systems, under the viewpoint of a Federal Government Institution IT manager. The increasing speed of information that comes from globalization makes it necessary for the companies to have their own information systems. This encourages the service supplying firms to develop techniques for the increase of productivity. In this developing process it is fundamental to take into account the cost of the whole production, not only the cost of the final product. It is also necessary for the companies to control their developing processes. As to the companies that are contracting the services of a supplying firm, there is a great necessity of improving the way the computerization expenses are planned and carried out. It is also necessary to follow-up the developing process and make the necessary corrections. Together with that comes the necessity of being familiar with the service supplying firms and particularly being familiar with the firms that take part in the public bidding, being also in a position to evaluate them, deciding which ones have understood the proposed problem and which ones are able to achieve the aimed product. For that purpose we are presenting the most well-known metrics, making a brief study of each one of them and, at the end, pointing out the advantages and disadvantages of each one. It has been noticed throughout the project that much is said about metrics, but little is actually used, which caused a great difficulty to the research. This leads us to the thought of the reasons why the metrics are not always used, when they seem to bring more clarity to the work.

Key Words: Metrics, Services Contract, Systems Development, It Management.

# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO DO TEMA

A globalização e a maior competitividade no mercado tornaram as empresas mais dependentes dos seus sistemas de informações. Construir estes sistemas, em tempo hábil e com a qualidade adequada para serem úteis, é o desafio que fornecedores de soluções de software estão enfrentando. Isto tem levado as empresas de desenvolvimento de software a aplicarem muitas técnicas de desenvolvimento orientadas à qualidade, buscando a melhoria contínua de seus processos, procurando orientações em modelos e padrões já consagrados.

Em um processo de desenvolvimento de software é preciso medir o custo, a produtividade e a qualidade não só do produto final, mas também de todo o processo. O desenvolvimento e manutenção de sistemas são funções que mais consomem recursos financeiros de uma empresa, chegando a ser um dos pontos estratégicos desta.

A indústria de software está lidando com clientes cada vez mais sofisticados que exigem qualidade, baixo custo e suporte total após a entrega. Em muitos casos, depois de feita a compra, os clientes necessitam de melhorias contínuas na funcionalidade o que requer uma atualização constante do produto, por sua vez, as organizações, ou empresas de construção de softwares precisam maximizar os lucros por meio de um processo eficiente, utilizando menos recursos e obtendo maior produtividade.

Para sobreviverem neste mercado, as organizações devem poder controlar de modo preciso o processo de desenvolvimento de software. Para isto foram implementadas métricas de construção e manutenção de software, que são ferramentas indicadas para avaliar a qualidade e a produtividade destes processos, fornecendo dados quantitativos e qualitativos, dando suporte para decisões sobre os investimentos, com uma margem de erro menor.

As organizações envolvidas com a engenharia de software têm, durante anos, lutado em busca de métodos quantitativos aceitáveis, para medir a eficiência dos processos, e para gerenciar os custos de software dos sistemas por ela adquiridos, desenvolvidos, melhorados ou mantidos. Um aspecto crítico e especialmente difícil deste requisito de medida tem sido a necessidade de se determinar o tamanho de software. Numerosos métodos de medição de software têm sido propostos. Tais métodos incluíram a quantidade de linhas dos programas, bem como várias medidas derivadas das características técnicas do software.

Segundo o Clipping Pesquisa MIT/SOFTEX da SOFTEX em sua página na internet (<http://www.tecsoft.softex.br/clipp-mes04--06.php>):

O Brasil é o sétimo mercado de software no mundo, com vendas de US\$ 7,7 bilhões em 2001, competindo diretamente com a Índia e a China, com respectivamente US\$ 7,9 bilhões e US\$ 8,2 bilhões. Entre 1991 e 2001, a participação do segmento no PIB triplicou, alcançando 0,71%. Além disso, a oferta de soluções foi versátil, com empresas de software atuantes em vários setores.

Esses são resultados de uma pesquisa conduzida pelo MIT, um dos institutos de tecnologia de maior renome em todo o mundo. Segundo o estudo, os produtos gerados no Brasil, nesse setor, são sofisticados e equiparáveis, em volume, ao mercado chinês e indiano.

Segundo o Boletim nr. 121 de 2001 do DFJUG, em sua página na internet ([http://www.dfjug.org/DFJUG/boletins/bolet\\_121.htm](http://www.dfjug.org/DFJUG/boletins/bolet_121.htm)):

O diretor da Rational, empresa sediada em Cupertino, na Califórnia, com faturamento anual de US\$ 811 milhões, estima que o mundo emprega cerca de 13 milhões de pessoas na produção de programas de computador. No Brasil esse contingente está estimado em 65 mil profissionais, o que corresponde a 0,5% do total. Ele estima que o País tem potencial para pelo menos dobrar e até triplicar essa participação, segundo informações da página do Centro Nacional de Pesquisa em Informática.

## **1.2 PROBLEMA**

Fica mais evidenciado, a cada dia, a necessidade por parte dos gerentes de planejar e executar melhor os gastos com informatização e um dos maiores fatores que mais contribui para o distanciamento do que é executado em relação ao planejado é o desenvolvimento e manutenção de sistemas informatizados, pois as estimativas são falhas e na grande maioria das empresas não existe uma cultura de medir este tipo de projeto, a não ser pelos gastos financeiros, ou seja, quanto se gasta.

Ao se fazer esta análise invariavelmente chega-se na conclusão que os gastos com a área de informática são muito elevados. O problema do Inmetro não é diferente.

Numa situação real ao realizarmos uma contratação para desenvolvimento ou até mesmo de manutenção de sistemas necessitamos ter ferramentas que nos auxiliem a determinar

se uma empresa é melhor ou pior do que a outra, ou ainda, como julgar entre as várias concorrentes, qual compreendeu corretamente as especificações do sistema ou manutenção pretendido e que assim supostamente teria mais condições de atender aos requisitos e assim chegar ao produto final mais próximo às nossas necessidades.

Atrelado às necessidades relacionadas acima vem a de comprar, desenvolver ou manter sistemas com qualidade e custos compatíveis, assim como a de dar transparência, ou seja, de mostrar à alta gerência como são feitos os investimentos e quais os retornos em termos de melhoria da qualidade, qual o tamanho do patrimônio de sistemas e o seu crescimento, além de permitir a obtenção de indicadores consistentes de desempenho, tais como: produtividade, qualidade do produto e etc., melhoria da capacidade de estimar e de gerenciar projetos, assim como melhorar a capacidade de comunicação com os clientes internos, uma vez que as regras são definidas previamente e que todos tem ciência, inclusive quem solicita o serviço.

A comparação com as práticas do mercado e a avaliação do impacto de novas tecnologias também são necessidades que carecem ser tratadas, assim como a identificação e gerenciamento de riscos, identificação e resolução de problemas, a harmonização do conhecimento visando à comunicação com a equipe, a avaliação do desempenho da área de informática e as justificativas objetivas para as decisões são fatores inequívocos e sólidos que devem ser alcançados com a adoção de medição do software.

É inquestionável que a necessidade de medir é inerente à necessidade de se controlar, de pagar e da mesma forma de se acompanhar o que está sendo contratado, porém como executar tal tarefa com imparcialidade, precisão e eficácia?

O problema em questão passa por estes aspectos e tem como ponto chave a temática de medir para poder controlar. A questão que consideramos com principal é: Como alcançar este estágio?

### **1.3 OBJETIVOS**

#### **Objetivo Geral:**

Genericamente, o objetivo deste trabalho é o de apontar uma métrica de desenvolvimento e manutenção de software que melhor atenda aos anseios de um gerente na área de desenvolvimento e manutenção de sistemas, mostrando as vantagens e desvantagens

de cada uma e, apontando quais, do ponto de vista da qualidade, atendem as necessidades de uma instituição seja ela pública ou privada com as características do Inmetro.

### **Objetivo Específico:**

Relacionar, sob a ótica da melhoria contínua do processo de desenvolvimento e manutenção de software, os pontos básicos para alcançarmos o controle. Tomando como parâmetro o que diz Tom DeMarco em seu livro Controle de Projetos de Software, “*Não se pode controlar o que não se pode medir*”, para se manter um processo sob controle, devemos instituir uma métrica eficaz, implantada de forma correta e objetiva.

Melhorar o processo de desenvolvimento, adotando uma métrica, visualizando de maneira antecipada a todo e qualquer custo que esteja associado ao sistema, tais como: construção, instalação, operação e manutenção. O custo da construção é um tópico importante, visto que é graças a ele que sabemos o total de pessoas envolvidas no desenvolvimento do sistema, como: administradores, diretores, gerentes, membros da comunidade usuária, consultores e programadores, membros da auditoria, do controle de qualidade ou da equipe de operações. O fornecimento de um sistema pode ser um ato simples como por exemplo a entrega de disquetes ou CD-ROM e a instalação ficar por conta do próprio usuário, porém em caso de sistemas grandes o processo de instalação é mais complexo e envolve outros fatores, como: custo de treinamento do usuário, custo de conversão de banco de dados e povoamento de tabelas, custo da aprovação legal, custo do processamento em paralelo no caso de migração de sistemas, custo da equipe de desenvolvimento durante a instalação. Naturalmente, toda a comunidade de usuários precisará de treinamento para maior familiarização com o sistema. O custo operacional entra em ação após a instalação do sistema, que nada mais é do que o custo para manter o sistema à disposição do usuário. Isso também deve representar uma área em que seu novo sistema economizará dinheiro, pois ele presumivelmente será mais barato. Os custos operacionais mais comuns são: custos de hardware e de suprimentos, custos de software, custos de pessoal e custos de manutenção. O custo de falhas, ou manutenção, como geralmente é chamado, diz respeito às diversas formas de erros que podem tornar o sistema completamente indisponível até que seja corrigido, enquanto que em outros casos o sistema continua funcionando, porém uma ou mais saídas podem estar incorretas. Este custo é muito importante e não deve ser desprezado, uma vez que não temos sistemas perfeitos, sempre há algo a ser feito.



## **1.4 ASPECTOS TEÓRICOS**

Existem várias metodologias de medição de softwares, isto porque nenhuma é suficientemente boa para que se possa afirmar que há unanimidade de escolha. Tais métricas defendem pontos de vistas e assim formam-se correntes e tendências em torno de teóricos e de suas teorias. Podemos destacar Barry W. Boehm e Roger Presmann como dois dos principais teóricos de criação de métricas, que iniciaram o processo de tentar medir, visando o controle.

Algumas técnicas são levantadas aqui e nelas identificadas quais os seus pontos fortes e fracos.

## **1.5 ASPECTOS METODOLÓGICOS**

O presente trabalho de dissertação pode ser classificado quanto aos fins como pesquisa aplicada e descritiva.

- Aplicada - pois se caracteriza por seu interesse prático, no qual os resultados podem ser utilizados na solução de problemas reais.
- Descritiva - pois aborda os aspectos descrição, registro, análise e interpretação do problema objetivando seu funcionamento no presente.
- Quanto aos meios, este pode ser classificado como pesquisa de campo com caráter exploratório.
- De campo – pois é utilizada com o objetivo de conseguir informações ou conhecimentos sobre o problema, para o qual procuramos uma resposta, através de fatos ou fenômenos tal qual ocorrem espontaneamente.
- Exploratória – pois visa à formulação do problema, com a finalidade de desenvolver hipóteses, aumentar a familiaridade do pesquisador com um ambiente, fato ou fenômeno, para a realização de pesquisas futuras ou modificar e clarificar conceitos.

## **1.6 JUSTIFICATIVA**

Este trabalho de dissertação torna-se justificável, além do aspecto científico, pelos seguintes motivos:

### **Ordem Pessoal:**

As constantes contratações de desenvolvimento, a condição de profissional da área de TI, assim como a busca de uma métrica que melhor atenda as necessidades do Inmetro, são os interesses do pesquisador, que ora desempenha a função de gerente de TI.

Além disto, os resultados obtidos no estudo permitirão ao pesquisador desenvolver ainda mais sua experiência na coordenação dos projetos de desenvolvimento e manutenção de aplicativos, assim como na gestão de contratação de terceiros.

### **Ordem Institucional:**

A racionalização e diminuição dos custos com o desenvolvimento e a manutenção de aplicativos informatizados são objetivos à serem alcançados pelo Inmetro, visando uma melhor gestão da área de TI.

### **Ordem Prática:**

O resultado deste refletirá na qualidade de atendimento às demandas das diversas áreas do Inmetro, tendo em vista que cada vez mais os demandantes de sistemas informatizados esperam ser atendidos em prazos curtos e com a maior abrangência possível. Os demandantes dos sistemas querem a cada dia mais objetividade nas ações de desenvolvimento e manutenção dos seus sistemas, visando assim uma redução dos gastos, podendo assim fazer uma alocação melhor dos recursos financeiros disponíveis, que são finitos e em alguns casos escassos.

### **Ordem Teórica:**

Uma vez que a tecnologia da informação vem sofrendo grandes mudanças, que alteram não só a forma de administração das empresas, mas também as exigências dos clientes com relação à qualidade dos seus produtos, tornou-se necessário a otimização dos recursos, sejam financeiros ou técnicos na hora de desenvolver qualquer aplicativo.

Visando uma excelência de gestão, cada vez mais, as empresas investem em modelos para obter o controle de seus processos para desenvolver e manter softwares e aplicativos.

### **Ordem Contextual:**

Apesar das constantes evoluções que sofre a área de TI, muitos problemas ainda resistem. As implementações de projetos de sistemas informatizados são constantemente questionados, no que tange aos seus custos e aos seus prazos. Isto reflete diretamente na satisfação do cliente que demanda o serviço. Poder medir, acompanhar e controlar o desenvolvimento é uma meta a ser alcançada.

## **1.7 HIPÓTESES E QUESTÕES**

Este projeto é um trabalho de investigação aplicada em informática e à administração, visando atender as necessidades de gerenciamento e gestão de tecnologia de informação, na área de desenvolvimento e manutenção de sistemas.

As métricas usadas como estimativas para construção e manutenção de software vêm se tornando um dos principais tópicos na área de TI, com a crescente exigência dos consumidores pela qualidade, rapidez, comodidade e baixo custos para implantação e manutenção. É impossível não enxergar as métricas como técnicas para o aumento da produtividade e melhoria da qualidade e com custos adequados, na área de desenvolvimento e manutenção de sistemas informatizados.

Existem, ainda, muitas barreiras que impedem os profissionais da área de TI de utilizarem tais técnicas ou de a fazerem de maneira correta, embora as literaturas disponíveis atualmente sobre engenharia de informação sejam relativamente amplas e variadas.

Isto nos leva a refletir nas seguintes questões:

- Porque as métricas e estimativas de software propostas para o desenvolvimento de sistemas não são fiéis à realidade e à dimensão dos problemas?
- Tais técnicas acompanharam a rápida evolução do setor de TI?
- Será que estamos no caminho certo?
- Devemos desenvolver uma metodologia de medição para desenvolvimento de sistemas ou utilizamos uma já existente no mercado?

- Será que escolhermos a metodologia de medição certa para controle do desenvolvimento e manutenção dos nossos sistemas?
- Como contratar e controlar o desenvolvimento e a manutenção de sistemas?

Tais questões são constantemente ouvidas e na maioria dos casos ficam sem uma resposta que ponha fim as dúvidas, as quais nos levam a deduzir que algumas métricas e estimativas de software ficaram obsoletas e outras ganharam atualmente mais vigor, quando a busca da qualidade de software vem crescendo com grande rapidez.

Existem algumas metodologias de gestão de TI para desenvolvimento e manutenção de sistemas, algumas tem como base o cálculo de homem/hora, outras em pontos de função, porém sem nenhum respaldo para o gerente de TI, ou seja há nestes casos uma eterna dúvida de como gerenciar com eficiência e efetividade os recursos financeiros disponíveis.

## **1.8 DELIMITAÇÃO DO TRABALHO**

Este trabalho foi limitado ao estudo de métricas tendo como foco a Área de TI de um órgão público, no que tange aos serviços de desenvolvimento e manutenção de sistemas informatizados.

Limitou-se a avaliar as métricas mais famosas de desenvolvimento e manutenção de sistemas informatizados, do ponto de vista de um órgão público, sem perder o foco de um gerente de TI, dos clientes internos e dos gestores de sistemas.

Não pretende este chegar a uma métrica como a ideal, mas demonstrar os pontos fortes e fracos para uma futura avaliação.

## **1.9 ESTRUTURA DO TRABALHO**

A apresentação deste estudo está organizada rigorosamente segundo a publicação da ABNT, específica para apresentação de trabalhos acadêmicos, a NBR 14724 e segundo o documento Apresentação de Trabalhos Monográficos de Conclusão de Curso – 6ª Edição Revisada e Ampliada da UFF, respeitando suas composições e elementos pré-textuais, textuais e pós-textuais.

Este trabalho está estruturado dentro das recomendações das publicações acima referenciadas, no que tange ao corpo textual, da seguinte forma:

#### Capítulo 1 – Introdução

Considerações iniciais do capítulo; contextualização do Tema; problema; objetivos; aspectos teóricos; aspectos metodológicos; justificativa; hipóteses e questões; delimitação do trabalho; estrutura do trabalho e considerações finais do capítulo.

#### Capítulo 2 – Fundamentação Teórica

Considerações iniciais do capítulo; a medição na terceirização e na gestão de contratos; métricas de construção de software; contagem de linhas de códigos; contagem de símbolos – Software Science Halstead; Construtive Cost Model – COCOMO; modelo de estimativa de Putnam; modelo Feature Points e considerações finais do capítulo.

#### Capítulo 3 – Metodologia Científica

Considerações iniciais do capítulo; método de abordagem; procedimentos e técnicas; considerações finais do capítulo.

#### Capítulo 4 – Perfil do Estudo de Caso

Considerações iniciais do capítulo; o instituto; análise de soluções; visão global; principais processos; planejamento tecnológico; disponibilização das soluções; acompanhamento das soluções e avaliação dos resultados; estrutura e funções necessárias; a área de atendimento; apoio ao atendimento; consultores de TI; ambiente de produtividade; apoio ao usuário; a área de suporte e produção; planejamento e projeto de infra-estrutura; suporte e rede; produção; metodologia para gestão de projetos de tecnologia da informação; metodologia para desenvolvimento e manutenção de sistemas; metodologia de manutenção de equipamentos e considerações finais do capítulo.

#### Capítulo 5 – Análise Comparativas das Métricas

Considerações iniciais do capítulo; dados em análise; análise comparativa e considerações finais do capítulo.

#### Capítulo 6 – Conclusão

Resultados; Conclusão e Considerações Finais e Sugestões para Novos Trabalhos.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 CONSIDERAÇÕES INICIAIS DO CAPÍTULO**

Com este capítulo pretende-se posicionar o leitor a respeito da medição voltada para terceirização e por consequência para a gestão de contratos.

Pretende-se também, apresentar as principais técnicas utilizadas na construção de uma métrica para desenvolvimento e manutenção de sistemas informatizados.

Não pretendemos com este trabalho esgotar o estudo muito menos analisar todas as métricas existentes, pois isto nos levaria a um trabalho muito extenso e enfadonho, cansando até o mais interessado leitor, além do fato de não podermos afirmar que neste exato momento não esteja sendo aplicada uma nova métrica, haja visto a dinâmica que é imposta a área de informática, principalmente depois da globalização.

Escolheu-se sob o ponto de vista da disseminação os mais conhecidos.

### **2.2 A MEDIÇÃO NA TERCEIRIZAÇÃO E NA GESTÃO DE CONTRATOS**

Para este item tivemos como fonte o documento Modelo de Maturidade da Capabilidade de Software (CMM) Versão 1.2 de 11/10/2001, publicado pelo CPqD Telecom & IT Solutions na página do MCT, conforme descrito nas referências bibliográficas.

Em novembro de 1986, nos Estados Unidos o Software Engineering Institute (SEI) iniciou o desenvolvimento de uma estrutura de um modelo de maturidade, com o propósito de auxiliar as empresas de software a melhorar seus processos. Este trabalho foi iniciado como resposta a uma requisição do governo federal americano, que precisava de uma metodologia para avaliar a capacidade de seus fornecedores de software. Após alguns anos, esta estrutura evoluiu para o CMM - Capability Maturity Model for Software, ou Modelo de Processo de Maturidade para Software, baseado no conhecimento adquirido através da avaliação dos processos de software e uma grande resposta da indústria e do governo.

O Processo de Maturidade para Software significa o quanto um determinado processo está definido, gerenciado, medido, controlado e efetivo. Maturidade implica no potencial de crescimento da capacidade de um processo, além de indicar os detalhes dos processos e a

consistência com que são aplicados dentro da organização, mostrando que a produtividade e a qualidade resultantes destes processos pode ser consistentemente melhorada ao longo do tempo.

Apesar de engenheiros e administradores conhecerem os problemas existentes de forma bem detalhada, normalmente ocorrem divergências sobre quais melhorias são mais importantes. Sem uma estratégia organizada para afrontar os problemas, é muito difícil chegar a um consenso entre a administração e a produção sobre quais ações tomar. Para que os esforços de melhoria forneçam resultados duradouros é necessário desenhar uma rota evolucionária que aumentará a maturidade da organização de forma escalonada, onde cada estágio serve de base para implementar as melhorias necessárias para atingir o estágio seguinte, criando um ciclo constante de melhoria dos processos. Este sistema não fornece soluções rápidas para projetos com problemas, mas identifica as deficiências e baliza o avanço.

O fato do desenvolvimento e manutenção de sistemas poderem ser em parte ou no todo terceirizados, não elimina a necessidade de medição, pelo contrário, isto aumenta a necessidade do controle. Com a criação do PMI e do CMM, ficou evidente que a aplicação de métricas é um ponto determinante no relacionamento entre empresas que contratam e empresas que são contratadas.

O PMI separa toda uma área de conhecimento denominada gerência das aquisições do projeto dos processos envolvidos na obtenção de bens e serviços externos à organização. O planejamento das contratações é o primeiro dos processos compreendido por essa área de conhecimento. Ele objetiva identificar as necessidades do projeto que podem ser mais bem atendidas através da contratação dos produtos ou serviços no mercado.

A escolha do modelo de contratação pode ou não simplificar a terceirização e a gestão do contrato. Basicamente temos dois modelos de contratação, o de Preço Global, também chamada de Preço Fixo ou Pacote e o por Homem-Hora.

O primeiro modelo, Preço Fixo, caracteriza-se pela adoção de um preço fixo por um produto, tal qual um pacote. Para tal é necessário que seja feita uma especificação o mais detalhada possível, nas fases iniciais da concepção do sistema, o que é bastante complicado, pois nesta fase ainda existem características ocultas que somente serão percebidas com o avançar do projeto. Sendo assim, a probabilidade de haver um erro de concepção de projeto e por conseqüência um aumento do escopo originalmente previsto é muito elevada. Levando-se em consideração que ambos, contratado e contratante não querem ficar no prejuízo, ou seja, não querem arcar com os custos do que não foi especificado corretamente, muitas vezes o projeto acaba sendo executado de forma incorreta e gera posteriormente uma lista de

manutenções infundáveis, pois é muito mais fácil para a contratada sacrificar os aspectos de maior dificuldade em detrimento dos mais fáceis.

O segundo, Homem-Hora, caracteriza-se por outro tipo de risco. A dificuldade de medição é bastante grande, pois hora não é um bem ou um serviço prestado não é uma unidade de tempo, ou seja, não é unidade de medição de custos. A hora é uma ferramenta para auxiliar a medição, ela por si só não mede nada, ou seja, não dimensiona o quanto está sendo produzido. Temos com exemplo o caso de uma empresa que quanto menos ela produz na unidade de tempo, ou seja, em horas, mais ela será remunerada, sendo assim diametralmente oposta a política da produtividade. Para ter controle da situação a contratante necessita criar pontos de controle adicionais e com isto os seus custos de gerenciamento aumentam. Este risco, ou seja, a perda de produtividade, afeta diretamente a contratante e não a contratada, que em tese deveria estar se preocupando com a produtividade. Ainda podemos incorrer em casos como o de ter a equipe ficando até mais tarde com o objetivo de aumentar o número de horas trabalhadas, sem ter com isto o aumento de produtividade. É necessário que sejam inseridos no processo ferramentas gerenciais com o objetivo de tornar tais atitudes visíveis e transparentes. O acompanhamento da produtividade da equipe exige a criação de um sistema com indicadores que evidencie os fatos de modo a que a gerência tome uma ação. Para tal é necessário que sejam introduzidas duas medidas, a saber: Esforço empreendido e Resultado obtido.

## **2.3 MÉTRICAS DE CONSTRUÇÃO DE SOFTWARE**

O termo Métrica de Software refere-se à mensuração, medição, dos indicadores quantitativos do tamanho e complexidade de um sistema. Estes indicadores são, por sua vez, utilizados na comparação com os desempenhos observados no passado a fim de estimar previsões de desempenho.

Para BOEHM (1981) uma métrica é uma forma padrão de medir um atributo do processo de desenvolvimento de software. As métricas fornecem informações quantitativas sobre o ambiente e o progresso de desenvolvimento do produto.

A utilização de uma métrica ajuda na construção de um processo de desenvolvimento de software, baseado na determinação do seu tamanho e conseqüentemente, do custo, do esforço, da complexidade e da sua qualidade. Os subsídios fornecidos nesta abordagem servem para alcançar um modelo de gestão administrativa com a efetiva influência da ação de



variáveis humanas, técnicas, ambientais e políticas, onde as estimativas de projetos de software são substituídas por uma série de passos sistemáticos oferecendo resultados com riscos aceitáveis. Para alcançar-se medições confiáveis, uma série de hipóteses são apresentadas por PRESMANN (1995).

- Atrasar as estimativas até um ponto tardio do desenvolvimento do projeto;
- Usar técnicas de decomposição relativamente simples para gerar estimativas de custos e de esforços de projeto;
- Desenvolver um modelo empírico para o custo e esforço de software;
- Adquirir uma ou mais ferramentas de estimativas automatizadas.

Observando-se as opções acima, percebe-se que a primeira, ainda que atraente, não é prática, pois as estimativas de custo e esforço devem ser oferecidas “logo de início”, embora seja reconhecido que, quanto mais se espera mais se sabe e, quanto mais se sabe, menor a probabilidade de cometer erros nas medições. Porém esta é bastante temerária, pois todo administrador deseja, o quanto antes, saber quais os custos envolvidos no projeto. Muitas vezes a decisão de adotar um projeto ou outro é tomada mediante os custos versus o benefício que este trará. As demais opções são abordagens viáveis às estimativas de projeto de software, porém agregam outras perguntas como quais técnicas, modelos e ferramentas são aplicáveis ou não ao projeto.

O ideal é que as técnicas disponíveis para cada opção sejam aplicadas em série, cada uma sendo usada como uma verificação cruzada das demais. As técnicas de decomposição para as estimativas de projetos de software assumem uma abordagem de divisão para a conquista, ou seja, fragmentar para melhor enxergar cada parte. Ao decompor um projeto em funções importantes, as estimativas de tamanho, custo e por tabela de esforços, podem ser realizadas por etapas.

### 2.3.1 CONTAGEM DE LINHAS DE CÓDIGO

A primeira forma encontrada para medir o tamanho de um sistema foi a contagem de suas linhas de código.

Esta medida é representada pelos símbolos  $S_s$  e  $S$ , onde  $S_s$  representa unidade (LOC), que significa linha de código, do inglês *Lines of Code* e  $S$  representa milhar de LOC (KLOC), ou seja, multiplicado por 1000 (mil).

Autores como S.D. Conte a consideram muito simples e afirmam que a sua principal vantagem é esta, ou seja, a sua simplicidade; outros afirmam que, embora esta métrica possa parecer simples, não o é. Esta discordância está baseada fundamentalmente no fato de que existe discordância sobre o que constitui uma linha de código.

Não há um pensamento majoritário sobre o que a contagem de linhas de códigos deve ou não contar, pois para alguns ela não deveria contar linhas de comentários e linhas em branco, uma vez que estas servem para documentação interna e sua presença ou ausência não influencia nas funções do programa. A sua inclusão na contagem pode encorajar os programadores a introduzir artificialmente diversas linhas nos projetos de desenvolvimento, criando a ilusão de grande produtividade.

No entanto existe um consenso de que os comandos de tipo condição (If End, Case End, Do Return, For Next) congregam diversos comandos em seu bloco, sendo contados apenas uma vez mas, os comandos que os compõem são contados normalmente.

Além disso, se o principal interesse é o tamanho suportado pelo programa, é razoável incluir somente comandos executáveis. Pode-se facilmente ver o potencial da principal discrepância para programas grandes com muitos comentários e programas escritos em uma linguagem que permite um grande número de declarações, mas não comandos executáveis. Assim, a vantagem de linguagem de programação que permite codificação em formato livre cria outros problemas, pois estas linguagens, muitas vezes, permitem composição com dois ou mais comandos em uma mesma linha.

Uma solução para esta situação é uma definição padrão para linha de código como sendo qualquer linha do texto do programa que não uma linha de comentário ou linha em branco, sem levar em conta o número de comandos ou fragmentos de comandos nas linhas.

Esta é a definição predominante para linhas de código usadas hoje.

### **2.3.2 CONTAGEM DE SÍMBOLOS – SOFTWARE SCIENCE HALSTEAD**

Em 1972, Maurice Halstead, na Universidade de Purdue, realizou estudos sobre algoritmos, tentando testar empiricamente a hipótese dos operadores e dos operandos em um programa se relacionarem com a quantidade de erros no algoritmo. Dado o sucesso do estudo, a pesquisa continuou em 1977, originando um sistema que consiste em registrar para cada programa desenvolvido o número de operadores e operandos utilizados, permitindo o cálculo do tamanho do programa e uma medida do esforço de programação.

Um programa de computador é considerado, dentro da métrica, como uma coleção de símbolos que podem ser classificados como operadores ou operandos.

Quaisquer medidas de Halstead são funções de contagem destes símbolos. A métrica básica é definida como:

- $n_1$  = número de operadores único;
- $n_2$  = número de operandos único;
- $N_1$  = total de ocorrências de operadores;
- $N_2$  = total de ocorrências de operandos.

Onde:

Operadores Único ( $n_1$ ) = Quantidades de operadores ao longo do programa, independente de repetição, ou seja, é contado quantos operadores aparecem no programa. No exemplo utilizado  $n_1 = 14$ .

Operandos Único ( $n_2$ ) = Quantidades de operandos ao longo do programa, independente de repetição, ou seja, é contado quantos operandos aparecem no programa. No exemplo utilizado  $n_2 = 13$ .

Total de Ocorrências de Operadores ( $N_1$ ) = Quantidade de vezes que os operadores se repetem ao longo do programa. No exemplo utilizado  $N_1 = 51$ .

Total de Ocorrências de Operandos ( $N_2$ ) = Quantidade de vezes que os operandos se repetem ao longo do programa. No exemplo utilizado  $N_2 = 42$ .

São Operandos (itens de dados):

Símbolos usados para representar dados (156) e  
Variáveis, constantes e ainda rótulos (x, y, tel).

São Operadores (comandos, palavras reservadas, símbolos ou palavras chaves em um programa que especificam uma ação):

Símbolos aritméticos (+, - e /);  
Nome de comandos (While, For, e Read);  
Símbolos especiais (:=, ()) e  
Nomes de funções.

O tamanho de um programa, em termos de total de números de símbolos usado, é calculado por:

$$N = N_1 + N_2$$

A Tabela 1 mostra um exemplo de análise de operadores e operandos de um programa qualquer. As regras originais estabelecidas por Halstead excluem a contagem de declarações de comandos e declarações de entrada e saída. Entretanto, não existe atualmente qualquer concordância sobre a qual seria a melhor forma de classificar e contar estes símbolos. A

classificação é, geralmente, determinada pela conveniência do programador ou de quem está construindo a ferramenta de contagem.

As regras usadas são, também, dependentes da linguagem. Além disso, ambigüidades na contagem de operadores e operandos ocorrem com uma certa frequência. Por exemplo, o “-” em Pascal pode designar um operador (negação) ou um operador binário (subtração).

Assim, na Tabela 1,  $N_1 = 51$  e  $N_2 = 42$ , e o tamanho, chamado comprimento por Halstead, é  $N = N_1 + N_2$ . De onde se obtém:

$$N = 51 + 42 = 93$$

O vocabulário consiste no conjunto de palavras distintas que foram usadas no desenvolvimento do programa. Ele é definido como:  $n = n_1 + n_2$ , então:

$$n = 14 + 13 = 27$$

Estas medidas levam a outras. A medida “volume”, que consiste no número de caracteres necessários para codificar o programa, que é definida por:

$$V = N * \log_2 n, \text{ sendo assim:}$$

$$V = 93 * \log_2 27 = 93 * 5 = 465$$

Outra medida que podemos extrair é o tempo em segundos para desenvolver o programa, sendo definido como:

$$T = (n_1 * N_2 * V) / (2 * n_2).$$

Onde:

$n_1$  = Número de Operadores;

$N_2$  = Número de Ocorrências dos Operandos;

$V = N * \log_2 n$ , acima descrito e

$n_2$  = Número de Operandos.

Logo:

$$T = (n_1 * N_2 * N * \log_2 n) / (2 * n_2).$$

Então:

$$T = (14 * 42 * 465) / (2 * 13) = 10.516 \text{ segundos}$$

Operadores	Ocorrências	Operandos	Ocorrências
<b>SUBROUTINE</b>	1	SORT	1
( )	10	X	8
,	8	N	4
<b>INTEGER</b>	1	100	1
<b>IF</b>	2	I	6
<b>.LT.</b>	1	J	5
<b>GO TO</b>	2	SAVE	3

<b>DO</b>	2	IM1	3
=	6	2	2
-	1	200	2
<b>.GE.</b>	1	210	2
<b>CONTINUE</b>	2	1	2
<b>RETURN</b>	1	220	3
<b>End-of-line</b>	13		
<b>TOTAIS</b>			
14	51	13	42

TABELA 1 - Análise de Símbolos da rotina de Ordenação

Fonte: HALTEAD, M.H. Elements of Software Science

A tabela 1 é apenas um simples exemplo de uma rotina de ordenação (SORT), poderia ter sido utilizado qualquer outra porém escolhemos esta, pois estava a mão.

Outra medida adicional que pode ser obtida é o tempo estimado de programação. O psicólogo John Stroud sugeriu que a mente humana é capaz de executar um número limitado de discriminações elementares por segundo. Esta capacidade está entre 5 e 20 por segundo. Haltead chegou ao valor 18, como melhor tempo estimado. Esta medida é definida na fórmula:

$$T = E / b$$

Se aplicarmos os valores na fórmula acima, teremos:

$$T = 10516 / 18 = 584 \text{ segundos} = \text{aproximadamente } 10 \text{ minutos}$$

Segundo a métrica, seriam necessários aproximadamente 10 minutos para desenvolver a Sub-rotina de ordenação exposta na Tabela 1.

### 2.3.3 CONSTRUCTIVE COST MODEL – COCOMO

O método de COCOMO foi idealizado pelo Prof. Dr. Barry Boehm durante os anos 70 e foi publicado em seu artigo em 1981. Este modelo utiliza como subsídio a aplicação dos conceitos de Linhas de Código, para estimativa de esforço, prazo e custo. Segundo BOEHM a métrica de estimativa de software COCOMO apresenta uma hierarquia de modelos da seguinte forma:

- 1) Básico - é um modelo estático de valor simples que soma o esforço (e custo) de desenvolvimento de software como uma função do tamanho de programa expresso em linhas de código estimadas;
- 2) Intermediário - soma o esforço de desenvolvimento de software como uma função do tamanho do programa e de um conjunto de direcionadores de custo e
- 3) Avançado - incorpora todas as características da versão intermediária, com uma avaliação do impacto dos direcionadores de custo.

O COCOMO pode ser aplicado em três classes de projetos de software. Estas classes são:

- 1) Modo orgânico - projetos de software simples, relativamente pequenos, nos quais pequenas equipes com boa experiência em aplicações trabalham num conjunto de requisitos não tão rígidos;
- 2) Modo semidestacado - um projeto intermediário (em tamanho e complexidade) no qual equipes com níveis de experiência mistos devem atingir uma combinação de requisitos rígidos e não tão rígidos;
- 3) Modo embutido - um projeto de software que deve ser desenvolvido dentro de um conjunto rígido de restrições operacionais, de hardware e de software.

As equações COCOMO básicas assumem a forma:

$$E = a \cdot S^b$$

$$T = c \cdot E^d$$

Onde:

$E$  é o esforço mensurado em pessoas-mês,

$T$  é o tempo de desenvolvimento em meses cronológicos e

$S$  representa o número estimado de linhas de código do projeto expresso em milhares (KLOC).

O coeficiente  $a$  significa o inverso da produtividade.

O expoente  $b$  nos leva a uma economia ou gasto conforme o caso. Se  $b < 1$  existe uma economia de escala, ou seja,  $E$  cresce em uma velocidade menor que  $S$ . Se  $b > 1$  existe um gasto em escala, ou seja,  $E$  cresce em uma velocidade maior que  $S$ .

O coeficiente  $c$  e o expoente  $d$  são tabelados de acordo com a complexidade do programa em questão, conforme tabela 3.

Podemos citar, outros pontos que nos levam a questionamentos, como por exemplo:

Porque não se pode aplicar COCOMO diretamente?

Porque o valor de  $a$ , depende da produtividade que varia de empresa para empresa.

Para solucionar estas questões, utiliza-se regressão linear e dados históricos de Burnett, 1995, para obtenção dos valores de  $a$  e  $b$ , como exemplificado na Tabela 2.

Nome do Sistema	Esforço (E) (h/m)	Tamanho (S) (Kloc)	Tempo (T) Meses
Sistema 1	11	8,5	6,2
Sistema 2	10	10,8	5,3
Sistema 3	12	11,7	7,1
Sistema 4	8	8,8	4,5
Sistema 5	7	6,5	4,0
Sistema 6	5	4,0	3,2

TABELA 2 - Histórico do Esforço de uma Empresa

Fonte: BOEHM, B.W. Software Engineering Economics.

Leitura: O Sistema 1 usou 11 homens/mês, com um tamanho total de 8,5 Kloc e durou 6,2 meses. Do mesmo modo podemos ler as demais linhas desta tabela.

São sistemas exemplos e representam realidades diferentes e, portanto são sistemas com variação de esforço, tamanho e tempo de conclusão.

Uma variável do método COCOMO é o COCOMO II. Este modelo é mais amadurecido sob a ótica das tecnologias e da engenharia de software, trazendo várias mudanças em relação ao seu predecessor. Essas mudanças não foram poucas, variando de sistemas que rodam em batch, ou seja, desenvolvidos para o processamento noturno, a outros em processamento em tempo real.

Tipo do Projeto do Programa	$c$	$d$
Orgânico	2,4	1,05
Restrito ou Semidestacado	3,0	1,12
Difuso ou Embutido	3,6	1,20

TABELA 3 – Tabela dos Valores do Coeficiente e do Expoente

Fonte: O próprio autor.

### 2.3.4 MODELO DE ESTIMATIVA PUTNAM

O modelo de Putnam é um modelo dinâmico de múltiplas variáveis que pressupõe uma distribuição de esforços específica ao longo da existência de um projeto de desenvolvimento de software. Foi construído a partir da coleta de distribuição de mão-de-obra encontrada em grandes projetos (esforço total de 30 pessoas-ano ou mais). Porém, a extrapolação para projetos de software menores é possível, segundo PUTNAM (1978).

A distribuição de esforço para grandes projetos de software pode ser caracterizada em curvas que assumiram uma forma clássica que foi descrita pela primeira vez por Lord Rayleigh. Também, foram usados dados empíricos sobre o desenvolvimento de sistemas, compilados por Norden. Daí, a distribuição de esforço indicado ser denominada curva de Rayleigh-Norden, segundo FENTON (1993).

A curva Rayleigh-Norden pode ser usada para derivar uma “equação de software” que relaciona linhas de código (L) ao tempo e esforço de desenvolvimento:

$$L = C_k K^{1/3} t_d^{4/3}$$

Onde:

$C_k$  é uma constante do estado de tecnologia e reflete as restrições de throughput que impedem o progresso do programador. A constante  $C_k$  pode ser derivada para condições locais, usando-se dados históricos compilados de esforços de desenvolvimento passados.

Valores típicos para  $C_k$  são:

2.000 para um ambiente de desenvolvimento de software “pobre” (por exemplo, sem aplicações de metodologia, documentação e revisões precárias, executando em modo batch);

8.000 para um ambiente de desenvolvimento de software de “bom” nível (por exemplo, metodologia posta em prática, documentação/revisões adequadas, modo de execução interativo);

11.000 para um ambiente de nível “excelente” (por exemplo, aplicações de ferramentas e técnicas automatizadas).

$K$  é o esforço despendido (em pessoas-ano) ao longo de todo o ciclo de vida para desenvolvimento e manutenção do software, e

$t_d$  é o tempo de desenvolvimento em anos.



Ao efetuar um novo arranjo da equação de software, pode-se chegar a uma expressão para o esforço de desenvolvimento K, ou seja, pode-se calcular K a partir de L, C<sub>k</sub> e t<sub>d</sub>:

$$K = \frac{L^3}{C_k^3 \times t_d^4}$$

A equação para o esforço de desenvolvimento pode ser relacionada com o custo de desenvolvimento mediante a inclusão de um fator de taxa de mão-de-obra (\$/pessoas-ano).

### 2.3.5 MODELO DE PONTOS DE FUNÇÃO

A Métrica Orientada a Função, ou simplesmente, por Pontos de Função, consiste em um método para medição de software do ponto de vista do usuário, que determina de forma consistente o tamanho e a complexidade de um software, sob a perspectiva do usuário, como já dito. Ele dimensiona o software, quantificando a funcionalidade proporcionada ao usuário a partir do seu desenho lógico, ou seja, é uma medida indireta do software e do processo por meio do qual ele é desenvolvido.

No lugar de contar linhas de código, a métrica por pontos de função concentra-se na funcionalidade ou utilidade do programa.

Os pontos de função são determinados usando-se uma relação baseada em medidas de informações e complexidade do software. Um dos pontos básicos da métrica por pontos de função fundamenta-se no fato de como o usuário enxerga os resultados que o sistema produz.

No início da década de 70, funcionários do Serviço de Processamento de Dados da IBM, a pedido de um grupo de usuários, começaram a analisar centenas de programas para isolar as variáveis críticas que supostamente poderiam determinar a produtividade da programação. Descobriram que poderiam basear a avaliação de um software em uma simples descrição em formato livre, que a especificação do software não precisava estar totalmente completa e que poderiam medir o valor das funções executadas pelos programas, em vez de utilizar como base o volume ou a complexidade do código dos programas.

Allan J. Albrecht, em 1979, foi encarregado de medir a produtividade de vários projetos de software desenvolvidos pela IBM. Esses projetos tinham sido desenvolvidos em uma grande variedade de linguagens, tornando inviável uma análise conjunta de produtividade. Isto motivou a busca por uma medida que fosse independente da linguagem de programação utilizada. Sendo assim, ele, prosseguindo com suas pesquisas, introduziu uma

técnica de avaliação conhecida como Pontos por Função. A métrica está baseada na visão externa do usuário, permitindo calcular o esforço de programação e auxiliando o usuário final a melhorar o exame e avaliação de projetos.

Após a apresentação da técnica à comunidade, no final da década de 70, e dos sucessivos trabalhos efetuados por Capers Jones, destacando sua importância, houve um crescimento acelerado do número de usuários de Pontos de Função, culminando, em 1986, na fundação do IFPUG. O IFPUG é uma entidade sem fins lucrativos, composta por pessoas e empresas de diversos países, cuja finalidade é promover um melhor gerenciamento dos processos de desenvolvimento e manutenção de software, com o uso de pontos de função e outras técnicas de medição. Atrelado a este crescimento surgiram também diversas variações da proposta apresentada por Allan Albrecht. Em 1990, foi lançada a primeira versão do CPM, com o objetivo de padronização técnica, hoje este manual está na versão 4.0 IFPUG, 1984. Desde então este é o padrão reconhecido pela indústria para pontos de função.

O uso da técnica de análise de pontos de função no Brasil começou no início da década de 90, com um forte apoio de uma grande empresa a Unisys. Entre 1991 e 1994 foram realizados seis Encontros Nacionais de Usuários de Pontos de Função, denominados ENUPF, contudo o interesse do mercado somente se consolidou quando grandes contratantes públicos passaram a se basear em pontos de função em suas contratações. Também contribuiu para a expansão da Análise de Ponto de Função o crescente interesse das organizações de desenvolvimento de software em modelos de qualidade e maturidade como o ISO e CMM, assim como a criação do BFUG – Brazilian Function Point Users Group,, que é o braço do IFPUG no Brasil, foi também um marco para a consolidação da Análise por Pontos de Função.

Os principais objetivos do modelo de análise por pontos de função são:

- Medir o que foi requisitado e recebido pelo usuário;
- Medir independentemente da tecnologia utilizada para a implementação;
- Prover uma métrica de medição para apoiar a análise de produtividade e qualidade;
- Prover uma forma de estimar o tamanho do software;
- Prover um fator de normalização para comparação de software.

Determinam-se os Pontos por Função de uma aplicação em três etapas de avaliação, conforme o IFPUG.

◆ A primeira resulta na contagem de pontos por funções não ajustados, que refletem as funções específicas e mensuráveis do negócio, provida ao usuário pela aplicação. Uma aplicação vista sob a ótica do usuário, é um conjunto de funções ou atividades do negócio que o beneficiam na realização de suas tarefas. Essas funções podem ser divididas nos seguintes grupos:

- Arquivo lógico interno: grupo lógico de dados do ponto de vista do usuário cuja manutenção é feita internamente pela aplicação;
- Arquivo de interface externa: grupo lógico de dados utilizados na aplicação cuja manutenção pertence a outra aplicação;
- Entrada externa: transações vindas diretamente do usuário que referenciam arquivos internos;
- Saída externa: compreende os dados extraídos da aplicação, tais como relatórios e mensagens do terminal de vídeo;
- Consulta externa: combinação de uma entrada e uma saída de dados, isto é, uma requisição de dados que gera uma aquisição e exibição imediata de dados.

Cada tipo de função deve ser classificada de acordo com sua complexidade funcional relativa, como:

- Simples;
- Média ou
- Complexa.

O valor dos pontos varia de 3 a 15, dependendo do seu tipo e grau de complexidade.

◆ A segunda etapa da avaliação gera o fator de ajuste, que representa a funcionalidade geral provida ao usuário pela aplicação. O fator de ajuste é calculado a partir de 14 características gerais dos sistemas, que permitem uma avaliação geral da funcionalidade do mesmo.

As características gerais de um sistema são:

- Comunicação de dados;
- Processamento distribuído;
- Desempenho;
- Utilização de equipamento;
- Volume de transações;

- Entrada de dados “*on-line*”;
- Eficiência do usuário final;
- Atualização “*on-line*”;
- Processamento complexo;
- Reutilização de código;
- Facilidade de implantação;
- Facilidade operacional;
- Múltiplos locais e
- Facilidade de mudanças.

Atribui-se valores de 0 a 5 para cada uma das 14 características acima, de acordo com o seu nível de influência na aplicação, onde:

- 0 - nenhuma influência;
- 1 - influência mínima;
- 2 - influência moderada;
- 3 - influência média;
- 4 - influência significativa e
- 5 - grande influência.

♦ A terceira etapa resulta na contagem de pontos por função ajustados, que refletem o fator de ajuste, apurado na segunda etapa, aplicado ao resultado da primeira etapa. Aí sim chegando ao cálculo dos pontos por função ajustados.

O total de pontos por função da aplicação pode ser encontrado multiplicando-se o fator de ajuste pela quantidade de pontos por função não ajustados, conforme demonstrado abaixo:

$$PF-D = PF-NA * FA$$

Onde:

PF-D → Pontos de Função do Sistema que se pretende desenvolver;

PF-NA → Quantidade de Pontos de Função Não Ajustados obtido na primeira fase e

FA → Fator de Ajuste. Valor obtido na segunda fase, que se deve multiplicar o valor encontrado na primeira fase para chegar a quantidades de Pontos de Função Ajustados.

### 2.3.6 MODELO FEATURE POINTS

O modelo Feature Points foi proposto no final dos anos 80 por Carpes Jones (1996) com o objetivo de analisar algoritmos em complemento às funções básicas do modelo de Análise de Pontos por Função. Este modelo propõe resolver as dificuldades que a análise de pontos por função encontra, quando mede sistemas em tempo real e sistemas embutidos.

A justificativa desta proposta é para cobrir a deficiência da análise de pontos por função quando é aplicada para estes tipos de sistemas que não levam em consideração o fator complexidade de algoritmos.

A técnica consiste em adicionar um novo parâmetro (algoritmo) as cinco funções, a saber:

- Arquivo lógico interno,
- Arquivo de interface externa,
- Entrada externa,
- Saída externa e
- Consulta externa.

Este parâmetro é assinalado com o valor padrão três. O método também reduz os pesos empíricos conforme demonstrado na Tabela 4.

Desde que os Feature Points são direcionados para complexidade de algoritmos a definição de algoritmo é apropriada. Algoritmo é definido como o conjunto de regras que devem ser expressas completamente, a fim de resolver um problema computacional.

Segundo Jones, algoritmos variam em dificuldade e complexidade, tornando difícil estabelecer a sua magnitude. Isto faz com que o enquadramento de um algoritmo possa assumir limites de um (1) até dez (10). Ou seja, algoritmos que requerem operações aritméticas simples podem ter um valor mínimo de um. Algoritmos que requerem equações complexas, operações de matriz, processamento lógico e matemático difícil podem ter um valor de dez. E, o peso padrão para um algoritmo normal, usando matemática simples poderá ser três.

Parâmetros	Quantidade	Valores Empíricos	Total de Pontos
Número de algoritmos		3	
Número de entradas		4	

Número de saídas		5	
Número de consultas		4	
Número de arquivo de dados		7	
Número de arquivos de interface		7	
Total de Pontos não ajustados			
Total de ajuste de complexidade			
Total de Feature Points			

TABELA 4 - Método Feature Points

Fonte: JONES, Carpers. Applied Software Measurement: Assuring Productivity and Quality

A base para os valores provisórios para algoritmo baseia-se:

- Primeiro, no número de passos calculados ou regras requeridas pelo algoritmo,
- e
- Segundo no número de fatores ou elementos de dados requeridos pelo algoritmo.

Existem algumas regras suplementares para determinar quais algoritmos são contáveis e significativos: algoritmo deve tratar de um problema resolvido; delimitado; definido; valores iniciais; produzir um resultado e capaz de representar estruturas padrões.

Exemplos de algoritmos típicos são: sorting; searching, step-rate calculation functions e feedback loops.

Jones nos mostra dois exemplos para elucidar o assunto.

O primeiro exemplo analisa um programa de computador para calcular pontos de função, usando a metodologia original de análise de pontos por função de 1979. A seqüência de cálculo constitui em multiplicar os dados para entradas, saídas e arquivos mestres, pelos pesos empíricos de Albrecht, derivando os pontos por função não ajustados. Depois, multiplica-se os pontos não ajustados por um fator de complexidade especificado pelo usuário, gerando-se, os pontos por função ajustados. Este cálculo da seqüência inteira incluiria só um algoritmo, o "algoritmo de cálculo de pontos por função". Dada a sua pouca complexidade este algoritmo deve ser enquadrado em uma magnitude de peso um.

No segundo, supondo-se que está sendo escrito um programa de computador, para calcular pontos por função, usando a metodologia de análise de pontos por função versão 4.0: a seqüência de cálculo seria identificar a quantidade de registro lógico e os itens de dados para determinar complexidade alta, baixa, ou média dos cinco parâmetros (arquivo lógico interno, arquivo de interface externa, entrada externa, saída externa e consulta externa), para

quantificar os pontos por função não ajustados. Então, aplicar-se-ia as 14 características para identificar os níveis de influência e determinar o fator de ajuste. Finalmente, os pontos por função não ajustados seriam multiplicados pelo peso de influência, para gerar o total de ponto por função ajustado.

Assim, a métrica de pontos por função original de 1979, poderia ser contada com o peso de um. Mas a métrica de pontos por função versão 4.0 requer um peso mais elevado, ou seja, de três.

Como pode ser visto, a complexidade de algoritmo promove um aumento do peso de complexidade. Para ajudar na identificação dos pesos das complexidades, Jones, 1996 criou uma tabela de exemplos de algoritmos em programas de estimativas de software.

## **2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO**

Neste capítulo foi enfatizada a medição na terceirização e por conseqüência a gestão dos contratos de terceirização, assim como cuidados que devem ser tomados e alguns pontos chaves na administração.

Foram levantadas preocupações com a gerência e controle do desenvolvimento e manutenção de sistemas, aliadas as técnicas e métricas de medição do trabalho desempenhado, sempre tendo como fundamento básico o controle de todas as etapas do desenvolvimento e manutenção de sistemas informatizados.

Também foram apresentadas as principais técnicas utilizadas na construção de métricas para desenvolvimento e manutenção de sistemas informatizados.

No próximo capítulo serão abordados aspectos a respeito do estudo de caso com a metodologia aplicada.

### 3 METODOLOGIA CIENTÍFICA

#### 3.1 CONSIDERAÇÕES INICIAIS DO CAPÍTULO

Os critérios de classificação dos tipos de pesquisa variam de acordo com o enfoque dado pelo autor. Conforme a classificação proposta por MARCONI e LAKATOS (1996), este trabalho de pesquisa pode ser classificado quanto aos fins, como pesquisa aplicada e descritiva.

- Aplicada - porque se caracteriza por seu interesse prático, em que os resultados são utilizados na solução de problemas que ocorram na realidade.
- Descritiva - porque aborda quatro aspectos: descrição, registro, análises e interpretação do problema, objetivando seu funcionamento no presente.

#### 3.2 ESCOLHA DO MÉTODO DE PESQUISA

Levando-se em consideração o exposto a seguir, enquadramos o método de abordagem utilizado no presente trabalho como o método hipotético-dedutivo.

A pesquisa foi assim enquadrada, uma vez que o método hipotético-dedutivo, segundo LAKATOS e MARCONI (1991), dois dos maiores autores neste ramo da ciência, se inicia pela percepção de uma lacuna no conhecimento, a cerca da qual se formulam as hipóteses e, pelo processo de inferência dedutiva, resta a predição da ocorrência de fenômenos abrangidos pela hipótese. Isto também é afirmado por Popper, que consiste na adoção da seguinte linha de raciocínio: *“quando os conhecimentos disponíveis sobre um determinado assunto são insuficientes para a explicação de um fenômeno, surge o problema”* o que é perfeitamente aderente ao nosso trabalho, e uma vez que os conhecimentos disponíveis não são suficientes para explicação qual métrica seria a mais adequada de aplicação em uma determinada situação.

Sendo assim, como não foram encontradas documentações que tratam do assunto, permitindo assim afirmar que há uma lacuna nesta parte do conhecimento.



### **3.3 DELINEAMENTO DA PESQUISA**

A presente pesquisa tem como contorno básico o desenvolvimento e manutenção de sistemas informatizados, tendo como foco determinar qual o procedimento mais adequado ao seu controle. Para tal relacionamos as características de um Instituto, o Inmetro, visando dar um contorno mais claro e bastante definido.

Não é nossa pretensão determinar qual a melhor ou a pior técnica a ser utilizada por empresas sejam elas públicas ou privadas e sim dentro do nosso campo de visão, ou seja, dentro do Inmetro, no que tange aos seus relacionamentos com empresas terceirizadas, determinar com a qual, ou as quais, queremos trabalhar.

Com este enfoque buscamos no mercado quais as principais técnicas e as esclarecemos no capítulo 2. Estas foram o objeto de estudo. É bem verdade que cada uma poderia ser melhor explorada, até poderiam gerar cada uma um capítulo ou até mesmo um livro, porém isto seria bastante cansativo para o leitor e não traria grandes contribuições para o trabalho, tendo em vista o escopo já delimitado.

### **3.4 DEFINIÇÃO DO INSTRUMENTO DE PESQUISA**

No presente trabalho de pesquisa foram utilizados os métodos monográfico e comparativo, segundo LAKATOS e MARCONI (1991).

- Método Monográfico – Consiste no estudo de determinados indivíduos, profissões, condições e instituições, grupos ou comunidades com a finalidade de obter generalizações.
- Método Comparativo – Realiza comparações com a finalidade de verificar similaridades e explicar divergências.

Foram feitas comparações entre as principais métricas adotadas no mercado e relatados os pontos considerados relevantes, e óbvio que existe muito mais a ser dito, caso se queira estudar com profundidade, porém como já dito anteriormente o objetivo é uma breve apresentação de cada uma, enfatizando as informações que foram utilizadas nos demais capítulos, como base para o presente trabalho.

Tendo em vista que o objetivo não é determinar qual a melhor ou qual a pior técnica e sim qual a que melhor atende as necessidades da delimitação do estudo, não será aprofundado o estudo, como dito anteriormente.

Visando melhor entendimento dos leitores apresentamos um passo-a-passo de como concebemos o nosso estudo.

- 1) Identificamos o problema a ser levantado ou estudado, para tal nos remetemos ao nosso dia-a-dia. (Como medir?).
- 2) Pesquisamos na literatura existente e acabamos identificando uma lacuna a respeito da questão a ser desenvolvida.
- 3) Identificamos quais os principais métodos existentes no mercado para se medir.
- 4) Uma vez identificados os principais, identificamos quais as divergências em cada um, em relação ao outro.
- 5) Por último traçamos um paralelo entre o que pretendemos e o que necessitamos no nosso dia-a-dia, para determinar, na medida do possível qual o que melhor atende as nossas necessidades.

### **3.5 DEFINIÇÃO DA AMOSTRA E FORMA DE COLETA DE DADOS**

Na realidade não há neste trabalho a coleta de dados por não se tratar de um trabalho estatístico, o mesmo foi fundamentado em literaturas já publicadas e em alguns trabalhos divulgados em páginas na Internet, como pode ser visto ao longo do mesmo.

Também é fato que não há neste ramo uma unanimidade quanto ao melhor critério que deveriam nortear a melhor métrica, ou seja, existem algumas tendências, sendo assim fica impossível determinar estatisticamente algum resultado.

Não foi definida como pesquisa aplicada uma vez que o universo da pesquisa foi delimitado quando do delineamento da pesquisa, a saber, o Inmetro, enquanto um órgão da Administração Pública Federal, vinculado ao MDIC. A amostragem é do tipo não-probabilística, do tipo Intencional, escolhidas no mercado de TI em publicações diversas. Já na coleta de dados, como dito anteriormente, citamos os autores com os principais fundamentos das métricas para uma análise, tais como BAHIA, BOEHM, HAZAN, PUTNAM, FENTOM, entre outros.

### **3.6 LIMITAÇÕES DA PESQUISA**

Várias limitações podem ser expostas neste trabalho, porém nos detivemos a enfatizar que a principal limitação encontrada é a de que não haver unanimidade quanto a escolha de um procedimento de métrica que seja amplamente aceita e aplicada. Existem algumas tendências, conforme já descritas, no capítulo anterior. Todas possuem vantagens e desvantagens, aliás, esta é a lacuna que encontramos no conhecimento, a qual nos dispomos a estudar, ou seja, a partir destas vantagens e desvantagens escolher a que melhor corresponda a nossa expectativa.

Também é fato que não pretendemos finalizar o estudo com este trabalho, há ainda muito a ser estudado e o que serve para a delimitação imposta neste trabalho não, necessariamente, vale para outro escopo, ainda, mas quando se fala em termos de tempo, ou seja, hoje a métrica “X” pode ser a que melhor atenda, porém numa avaliação futura a “Y” pode atender melhor.

### **3.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO**

Neste capítulo foram apresentados aspectos relacionados ao método, tipo de pesquisa, tipo de instrumento, delimitação e delineação do estudo assim como os procedimentos e as técnicas, sob o ponto de vista acadêmico.

No próximo capítulo será apresentado o perfil do estudo de caso, com sua inferência ao referencial adotado.

## **4 PERFIL DO ESTUDO DE CASO**

### **4.1 CONSIDERAÇÕES INICIAIS DO CAPÍTULO**

Neste capítulo pretende-se posicionar o leitor a respeito do ambiente que teremos como foco no presente trabalho.

Serão evidenciados a estrutura, os aspectos técnicos operacionais, o planejamento e as questões metodológicas.

Os conceitos apresentados neste capítulo estão fundamentados no Plano Estratégico do Inmetro para 2002 – 2010, versão de março de 2003, publicado na Internet e na publicação Qualidade Brasil - Relatório de Atividades 2003, ambos do Inmetro.

### **4.2 O INSTITUTO**

O Inmetro - Instituto Nacional de Metrologia, Normalização e Qualidade Industrial, criado pela lei n.º 5966, de 11 de dezembro de 1973, é uma autarquia federal, vinculada ao MDIC. É o órgão oficial responsável pelas atividades de Normalização, Certificação da Qualidade e Metrologia no Brasil.

Estão entre suas competências:

- Desenvolver atividades de pesquisa básica e aplicada em áreas críticas de metrologia;
- Gerenciar o Sistema Brasileiro de Certificação da Qualidade;
- Fomentar na indústria nacional a utilização de técnicas de gestão da qualidade;
- Coordenar a Rede Brasileira de Laboratórios de Calibração e Rede Brasileira de Laboratórios de ensaios e Rede Brasileira de Metrologia Legal e Qualidade;
- Regulamentar, fiscalizar e verificar os instrumentos de medir empregados na indústria e no comércio;
- Coordenar a participação brasileira em organizações internacionais;
- Secretariar o CONMETRO e seus comitês técnicos;
- Regulamentar e fiscalizar os produtos pré-medidos

- Difundir informações tecnológicas, notadamente sobre normas, regulamentos técnicos e qualidade;
- Promover e supervisionar o Sistema de Normalização Técnica consensual;
- Prover o país de padrões metrológicos primários;
- Promover o reconhecimento internacional do Sistema de Metrologia, Normalização e Qualidade Industrial e da Avaliação da Conformidade.
- Disseminar padrões de medidas.

O Inmetro tem como Missão:

***“Promover a qualidade de vida do cidadão e a competitividade da economia através da metrologia e da qualidade.***

***Indicadores:***

- *Credibilidade do Inmetro junto à sociedade*
- *Credibilidade do Inmetro junto ao setor empresarial.”*

O Inmetro, na concretização de seus objetivos de se transformar na Primeira Agência Executiva do País, desenvolveu o Plano de Modernização, que incluiu, dentre os seus dezoito projetos, o de Modernização da Gestão. No que tange à informática, este projeto contemplou o desenvolvimento do PGSI e a adequação da infra-estrutura tecnológica de hardware e software para permitir o atingimento das metas estabelecidas.

### **4.3 ANÁLISE DE SOLUÇÕES**

O estudo preliminar a respeito da capacidade de atendimento, pela área de informática, às demandas na época existentes, originário da constatação da crescente insatisfação dos clientes, pelo represamento de suas solicitações e pela falta de perspectiva de atendimento em um horizonte próximo, evidenciou a necessidade premente de adotar solução que vise atender, a médio prazo, às demandas reprimidas.

Na avaliação dos diversos órgãos, ficou clara a impossibilidade de atendimento a estas demandas, com os recursos disponíveis, tanto em termos quantitativos quanto qualitativos.

Surgiram como alternativas:

- o aumento do quadro de pessoal alocado à área de informática;

- a contratação de serviços de gerência da área de Informática, para desenvolvimento e manutenção de sistemas e para Manutenção de Equipamentos.

A impossibilidade de adoção da primeira alternativa, uma vez que isto depende da política governamental, nos remeteu à segunda alternativa. Além disso, há uma diretriz governamental que incentiva a contratação através de licitação de empresas terceirizadas nas áreas meio. Somente são autorizadas contratações por meio de concurso público, ou seja, para integrar o quadro das instituições federais, para as áreas fins e no Inmetro a área de informática não é área fim, pois não é um dos fins da instituição.

Ficou evidenciada a necessidade do estabelecimento de alguns critérios, visando a garantia de contratação de prestador de serviço com reais condições de atendimento aos requisitos técnicos, operacionais e econômico-financeiros, imprescindíveis à alta complexidade e especialização dos trabalhos a serem executados, tais como:

- Capital social e porte da empresa;
- Quantidade e qualificação técnica dos empregados contratados pela empresa na época da licitação;
- Capacidade e compatibilidade da estrutura de processamento de dados da empresa, aí compreendidos, os equipamentos, os Softwares básicos utilizados, os recursos de comunicação de dados existentes;
- Experiência na prestação de serviços de gerência e desenvolvimento e manutenção de sistemas;
- Qualidade da metodologia e das ferramentas de gestão, desenvolvimento, manutenção de sistemas e manutenção do parque de equipamentos utilizadas pela empresa;

Também foram feitas as seguintes determinações:

- Garantir que a empresa Contratada providencie a atualização tecnológica dos seus empregados alocados aos projetos do Inmetro;
- Evitar a fadiga e a conseqüente queda de produtividade dos empregados alocados aos projetos do Inmetro, limitando a utilização de horas extraordinárias e restringindo o acesso aos recursos computacionais. A partir das 22 horas, exceto nos casos justificáveis e previamente autorizados.

#### **4.4 VISÃO GLOBAL**

Procurando tornar claras as etapas envolvidas na gestão de uma área de TI, identificamos os principais processos e objetivos que necessitavam serem desenvolvidos pela estrutura da área. A clareza destes processos permitiria a identificação das funções necessárias na estrutura e daquelas ainda não desempenhadas na estrutura atual.

As funções que precisavam ser realizadas em cada um dos processos deveriam estar profundamente alinhadas com a missão do Inmetro e com as estratégias traçadas para alcance dos objetivos de TI no órgão.

#### **4.5 PRINCIPAIS PROCESSOS**

Traçando um paralelo com modelos de qualidade, pudemos generalizar que a estrutura de gestão de uma área deve estabelecer três processos básicos:

- O de planejamento,
- O de execução e
- O de avaliação.

Levando em consideração as peculiaridades de uma área de TI de um órgão técnico governamental, podemos prever os seguintes processos e seus objetivos:

##### **4.5.1 PLANEJAMENTO TECNOLÓGICO**

Entendendo a área de TI como apoio às atividades do órgão, por isso considerada área meio, o processo de planejamento tecnológico deve se preocupar com a equalização das estratégias do órgão com as da área de TI, devendo:

- Facilitar o Inmetro no cumprimento de sua missão;
- Apoiar o atingimento das metas estabelecidas para o órgão;
- Estreitar a relação com as áreas atendidas pela estrutura de TI a fim de permitir melhor entendimento das necessidades das diversas áreas;

- Apoiar a identificação e programar o atendimento das necessidades das áreas no que concerne às soluções de tecnologia da informação atuando em:
  - Atividades operacionais corporativas;
  - Suporte à tomada de decisão;
  - Aplicações departamentais;
  - Comunicação com o público e outros órgãos da administração ou afins, e;
  - Disseminação de informações técnicas.
- Acompanhar a evolução tecnológica e propor a adequação das soluções, analisando a relação custo/benefício;
- Rever periodicamente os métodos utilizados para o planejamento e as metas estabelecidas.

#### **4.5.2 DISPONIBILIZAÇÃO DAS SOLUÇÕES**

O processo de disponibilização das soluções é entendido como aquele que realiza as atividades de execução do planejado até a fase de implementação das soluções e, para tanto, deve:

- Fortalecer a cooperação e o comprometimento das áreas atendidas pela estrutura a fim de permitir a execução das atividades dos projetos em todos os níveis;
- Garantir a qualidade das soluções de tecnologia disponibilizadas no que diz respeito não só a técnicas utilizadas, mas ao atendimento das expectativas das áreas;
- Garantir a integridade dos dados e a consistência das informações disponibilizadas através das soluções;
- Garantir o cumprimento dos prazos acordados com as áreas envolvidas nas soluções;



- Implementar as soluções projetadas, identificando procedimentos rotineiros e pontos de controle a serem avaliados;
- Capacitar o cliente no uso das soluções adotadas;
- Estar voltado para atender padrões de excelência para atender no prazo e com a qualidade as expectativas dos clientes;
- Definir a infra-estrutura tecnológica para a operacionalização das soluções adotadas;
- Suportar os clientes no uso das tecnologias;
- Manter disponível a infra-estrutura tecnológica;
- Manter operacional as soluções implementadas;

#### 4.5.3 ACOMPANHAMENTO DAS SOLUÇÕES E AVALIAÇÃO DOS RESULTADOS

O que chamamos de acompanhamento das soluções e avaliação dos resultados é o processo de garantia de funcionamento das soluções tecnológicas implementadas, evitando discontinuidades e avaliando os resultados obtidos para alimentar o processo de planejamento tecnológico, e deve:

- Estabelecer critérios e métricas para avaliação dos resultados, revendo-os periodicamente;
- Avaliar o aumento da produtividade e da agilidade na realização de suas atividades;
- Identificar a necessidade de evolução das soluções acompanhando o dinamismo das áreas;
- Avaliar a qualidade das soluções disponibilizadas e a satisfação dos clientes quanto ao atendimento das expectativas.

Os processos descritos acima podem ser decompostos em vinte e cinco funções elementares, agrupados em treze áreas na estrutura organizacional.

O quadro a seguir apresenta as funções que deverão ser executadas pela área de TI e que devem ser utilizadas para a operacionalização dos processos, agrupadas por área na estrutura organizacional.

Área na Estrutura Organizacional	Funções
Serviço de Informática	<ul style="list-style-type: none"> <li>• Executar a gestão do contrato.</li> </ul>

<b>Área de TI</b>	<ul style="list-style-type: none"> <li>• Propor diretrizes e estratégias para o ambiente de informações.</li> <li>• Propor prioridades de desenvolvimento de soluções e revisões no PGSI a um Plano Diretor de Sistemas.</li> <li>• Executar a Gerência de conflitos e oportunidades identificadas durante a execução de projetos de TI.</li> </ul>
<b>Apoio Tecnológico</b>	<ul style="list-style-type: none"> <li>• Propor diretrizes, estratégias e padrões de tecnologia.</li> <li>• Propor consolidações à metodologia para desenvolvimento de sistemas.</li> <li>• Implantar padrões que assegurem a qualidade dos serviços de desenvolvimento.</li> <li>• Apoiar as equipes de TI no desempenho de suas funções.</li> </ul>
<b>Atendimento</b>	<ul style="list-style-type: none"> <li>• Propor prioridades de Atendimento de soluções e propor revisões no PGSI a um Plano Diretor de Sistemas.</li> <li>• Executar a gerência de projetos de desenvolvimento de sistemas.</li> <li>• Executar a Gerência de conflitos e oportunidades identificadas durante a execução de projetos de TI.</li> </ul>
<b>Suporte e Produção</b>	<ul style="list-style-type: none"> <li>• Executar a Gerência de conflitos e oportunidades identificadas durante a execução de projetos de TI.</li> <li>• Planejar a capacidade do ambiente de TI.</li> <li>• Avaliar o ambiente de TI.</li> <li>• Planejar o suporte ao ambiente de TI.</li> <li>• Avaliar os resultados de suporte.</li> </ul>
<b>Apoio ao Atendimento</b>	<ul style="list-style-type: none"> <li>• Garantir a consistência, evitar a duplicidade dos dados e garantir o acoplamento entre dados e processos.</li> <li>• Dar suporte a Banco de Dados.</li> </ul>
<b>Planejamento e Projeto de Infra-estrutura</b>	<ul style="list-style-type: none"> <li>• Planejar a capacidade do ambiente de TI.</li> <li>• Avaliar o ambiente de TI.</li> </ul>
<b>Consultorias de TI</b>	<ul style="list-style-type: none"> <li>• Proporcionar soluções de TI para as áreas usuárias do Inmetro.</li> <li>• Coordenar as equipes de Atendimento.</li> <li>• Avaliar os resultados obtidos com as soluções adotadas.</li> </ul>
<b>Apoio Administrativo</b>	<ul style="list-style-type: none"> <li>• Apoiar administrativamente a estrutura de TI.</li> </ul>
<b>Suporte e Rede</b>	<ul style="list-style-type: none"> <li>• Administrar os recursos computacionais da rede do Inmetro.</li> <li>• Avaliar ambiente de TI.</li> <li>• Executar o suporte ao ambiente de TI.</li> </ul>
<b>Ambiente de Produtividade</b>	<ul style="list-style-type: none"> <li>• Promover a produtividade dos serviços da área de TI.</li> <li>• Desenvolver, implantar soluções e treinar os usuários.</li> </ul>
<b>Produção</b>	<ul style="list-style-type: none"> <li>• Manter as soluções operacionais.</li> </ul>

**Apoio ao Usuário**

- Manter as soluções da área de TI nos Clientes.

**QUADRO 1 - Estrutura e Funções**

Fonte: Projeto Básico de Terceirização de TI do Inmetro

**4.6 ESTRUTURA E FUNÇÕES NECESSÁRIAS**

Para implementar os processos e as funções descritas anteriormente, foi desenvolvida uma proposta de estrutura organizacional. No diagrama abaixo a estrutura proposta está povoada com um efetivo considerado como suficiente para se iniciar a prestação de serviços. A medida em que o Plano de Modernização evolua, poderá haver a necessidade de adequação do número de profissionais ou até mesmo a revisão da estrutura.

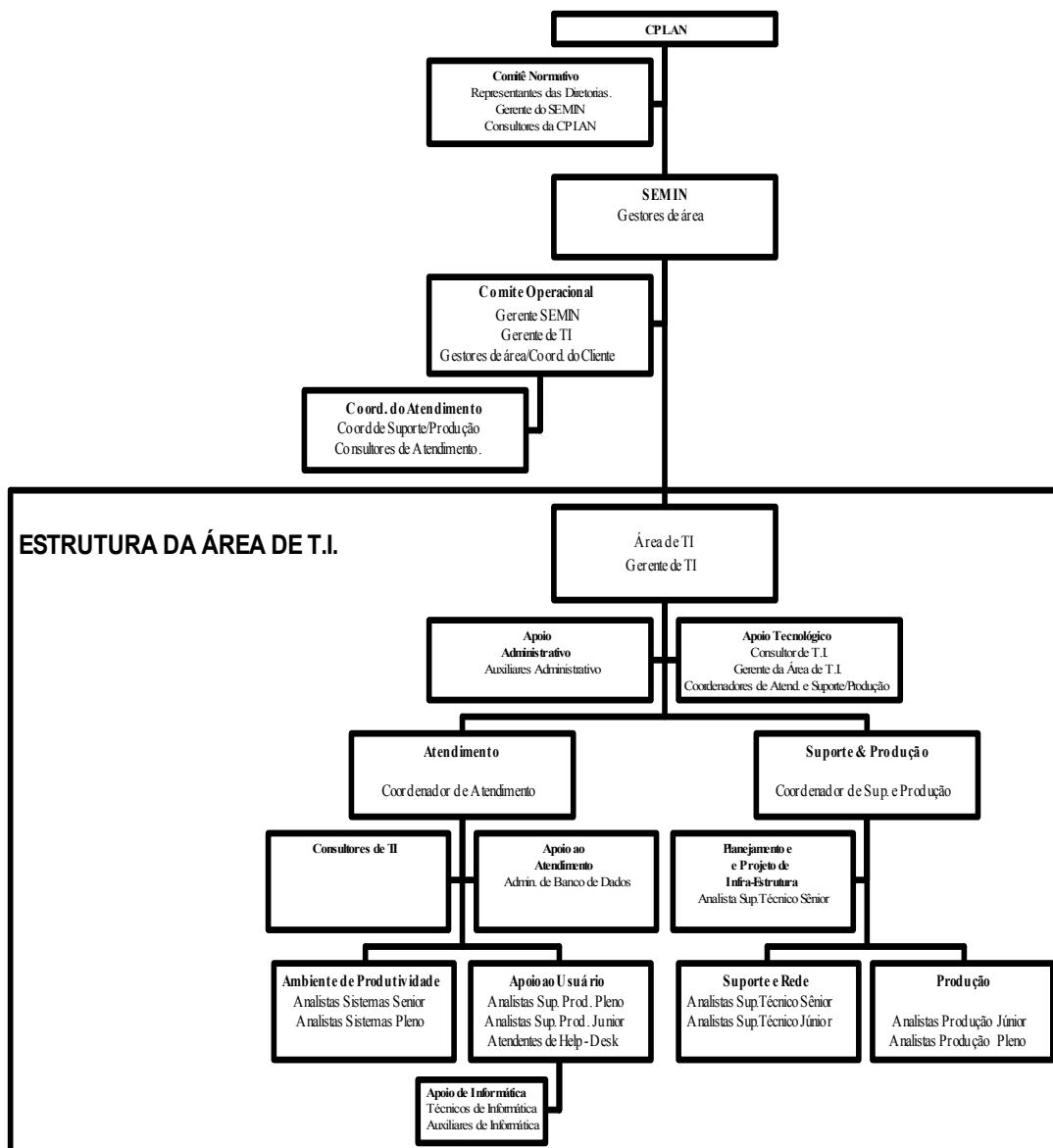


FIGURA 1 – Organograma para Gerenciamento

As descrições das funções atribuídas a cada elemento da estrutura organizacional estão relacionadas abaixo. A numeração das funções refere-se àquelas descritas na tabela de funções necessárias apresentada anteriormente.

Função	Atribuições
<b>1. Propor diretrizes e estratégias para o ambiente de informações</b>	<ul style="list-style-type: none"> <li>• Elaborar um plano estratégico anual de TI juntamente com o comitê operacional de informática definindo estratégias, orçamentos e estudos de viabilidade dos projetos;</li> <li>• Propor e quando aprovado, implementar processos para gestão da qualidade total;</li> <li>• Elaborar, submeter à aprovação e acompanhar os orçamentos dos projetos;</li> <li>• Divulgar benefícios e resultados alcançados;</li> <li>• Manter sólido relacionamento com as diversas diretorias do Inmetro;</li> <li>• Promover o alinhamento das soluções de TI às diretrizes e estratégias estabelecidas pelo Inmetro;</li> </ul>
<b>2. Propor diretrizes, estratégias e padrões de tecnologia</b>	<ul style="list-style-type: none"> <li>• Propor arquitetura tecnológica para suportar o ambiente de informações do Inmetro;</li> <li>• Manter as plataformas de hardware, software e comunicação do Inmetro Identificadas, homologadas e padronizadas conforme processo aprovado.</li> </ul>
<b>3. Propor prioridades de Atendimento de soluções e propor revisões no PGSI</b>	<ul style="list-style-type: none"> <li>• Executar reuniões trimestrais com um comitê operacional de informática para gestão do PGSI;</li> <li>• Rever trimestralmente as prioridades para Atendimento de sistemas, propondo alterações como incluir, alterar o escopo ou excluir projetos do plano;</li> <li>• Manter documentação das prioridades identificadas e revisões efetuadas;</li> <li>• Elaborar e rever periodicamente a arquitetura de sistemas do Inmetro;</li> </ul>

<p><b>4. Executar a gestão do contrato</b></p>	<ul style="list-style-type: none"> <li>• Manter relacionamento com a empresa Contratada para gerenciar os recursos técnicos através de plano de carreira, treinamento, avaliação de desempenho e todas as atribuições pecuniárias daí decorrentes;</li> <li>• Aprovar as equipes sugeridas pela contratada;</li> <li>• Acompanhar cronogramas físicos e financeiros, bem como da qualidade dos serviços prestados;</li> <li>• Garantir as condições técnico-operacionais para a prestação de serviços pela contratada;</li> <li>• Garantir a fidelidade das operações Terceirizadas aos processos do Inmetro e da administração pública, garantindo a prática das normas, dos procedimentos e dos métodos de gestão;</li> <li>• Disponibilizar recursos necessários para o bom desempenho das atividades da contratada;</li> </ul>
<p><b>5. Executar a gestão dos projetos de desenvolvimento de sistemas</b></p>	<ul style="list-style-type: none"> <li>• Acompanhar prazos, equipes e cronogramas físicos e financeiros, bem como da qualidade dos serviços prestados;</li> <li>• Promover reuniões semanais com consultores de TI para acompanhamento do progresso dos trabalhos;</li> <li>• Documentar e divulgar as decisões tomadas nas reuniões;</li> <li>• Garantir a execução da metodologia de desenvolvimento de sistemas;</li> <li>• Aprovar e manter documentação dos subprodutos da metodologia;</li> <li>• Divulgar os resultados alcançados;</li> </ul>
<p><b>6. Executar a gerência de conflitos e oportunidades identificadas durante a execução de projetos de TI</b></p>	<ul style="list-style-type: none"> <li>• Promover melhoria no sistema de comunicação entre as áreas usuárias e de TI;</li> <li>• Convidar os usuários para as reuniões de acompanhamento;</li> <li>• Construir um ambiente cooperativo entre as equipes prestadoras de serviços e de usuários e, conseqüentemente reduzir o conflito;</li> <li>• Propor processo participativo para gestão dos projetos de TI, buscando o comprometimento das áreas envolvidas;</li> </ul>
<p><b>7. Consolidar uso da metodologia para desenvolvimento de sistemas</b></p>	<ul style="list-style-type: none"> <li>• Divulgar e capacitar as equipes de TI no uso da metodologia;</li> <li>• Estabelecer mecanismos de controle que assegurem o cumprimento da metodologia estabelecida;</li> </ul>
<p><b>8. Implantar padrões que assegurem a qualidade dos serviços de desenvolvimento</b></p>	<ul style="list-style-type: none"> <li>• Estabelecer métrica que possibilite o acompanhamento de resultados;</li> <li>• Estabelecer mecanismos para garantir a aderência entre os requerimentos funcionais do sistema especificado e o desenvolvido;</li> </ul>

<p><b>9. Apoiar as equipes de T.I. no desempenho de suas funções</b></p>	<ul style="list-style-type: none"> <li>• Avaliar permanentemente o nível tecnológico em uso e pesquisar a viabilidade de uso de novas ferramentas;</li> <li>• Apoiar a coordenação das equipes de TI na proposição e justificativa de adoção de novas tecnologias a serem utilizadas;</li> <li>• Disseminar novas tecnologias para as equipes de TI;</li> <li>• Identificar e promover programa de capacitação e desenvolvimento das equipes de TI;</li> </ul>
<p><b>10. Garantir a consistência, evitar a duplicidade dos dados e garantir o acoplamento entre dados e processos.</b></p>	<ul style="list-style-type: none"> <li>• Manter o dicionário de dados corporativo;</li> <li>• Produzir e manter o modelo de entidades e relacionamentos corporativo e de sistemas;</li> <li>• Aprovar modelos lógicos de projetos;</li> <li>• Elaborar e manter documentação das bases de dados;</li> <li>• Apoiar as equipes de desenvolvimento na utilização de repositório;</li> <li>• Identificar, avaliar, selecionar e garantir a utilização de ferramenta case;</li> <li>• Definir processos para gestão das tabelas corporativas.</li> </ul>
<p><b>11. Proporcionar soluções de TI para as áreas usuárias do Inmetro</b></p>	<ul style="list-style-type: none"> <li>• Atuar como consultor junto aos usuários do Inmetro para apoiar a identificação e implementação de soluções de TI;</li> <li>• Buscar continuamente o aumento da produtividade e a melhoria da qualidade dos processos das áreas do Inmetro, adotando soluções adequadas aos prazos e custos do instituto;</li> <li>• Estabelecer e implementar processos para identificação de necessidades de treinamento;</li> <li>• Efetuar o planejamento para atendimento às necessidades dos usuários;</li> </ul>
<p><b>12. Coordenar as equipes de Atendimento</b></p>	<ul style="list-style-type: none"> <li>• Quantificar e disponibilizar recursos tecnológicos e humanos para implementação das soluções de TI;</li> <li>• Planejar a execução dos serviços estabelecendo cronogramas;</li> <li>• Acompanhar modelagem e detalhamento de sistemas;</li> <li>• Garantir a adoção pelas equipes de Atendimento da metodologia adotada;</li> <li>• Garantir aderência das especificações aos produtos desenvolvidos;</li> <li>• Quantificar e documentar os requerimentos para implantação;</li> <li>• Exercer a liderança sobre a equipe de Atendimento/manutenção de sua responsabilidade, com características pessoais de motivação, busca de sinergia, comprometimento e credibilidade;</li> </ul>

<p><b>13. Desenvolver, implantar soluções e treinar os usuários</b></p>	<ul style="list-style-type: none"> <li>• Participar no desenvolvimento de projetos de sistemas atuando nas fases de detalhamento, testes e implantação;</li> <li>• Utilizar metodologias para testes e implantação de sistemas;</li> <li>• Interagir com os usuários nas especificações de requerimentos;</li> <li>• Criar e manter documentação técnica e do usuário;</li> <li>• Comprometer-se com a qualidade dos serviços prestados;</li> <li>• Avaliar e recomendar alternativas de soluções baseadas nas necessidades do Inmetro e nas oportunidades tecnológicas;</li> <li>• Informar ao líder de projeto sobre o status dos trabalhos e necessidades operacionais de projeto;</li> <li>• Desenvolver e implementar soluções departamentais para o atendimento de solicitações não previstas nos projetos corporativos;</li> <li>• Cumprir as exigências para implantação em produção dos sistemas desenvolvidos;</li> <li>• Treinar os usuários do sistema;</li> </ul>
<p><b>14. Avaliar os resultados obtidos com as soluções adotadas</b></p>	<ul style="list-style-type: none"> <li>• Elaborar e executar pesquisa de satisfação com os usuários;</li> <li>• Promover reunião mensal com os usuários para atender as demandas presentes e futuras decorrentes de erros, modificações na legislação vigente, necessidades de gestão, necessidades de processos ou melhorias de performance;</li> <li>• Conduzir auditorias de sistemas;</li> <li>• Identificar necessidades de treinamento para novos usuários e de reciclagem para usuários com dificuldade de utilização dos sistemas implantados;</li> <li>• Avaliar o resultado das soluções departamentais e, quando possível propor a integração ou substituição por soluções corporativas;</li> </ul>
<p><b>15. Planejar a capacidade do ambiente de TI</b></p>	<ul style="list-style-type: none"> <li>• Estabelecer procedimentos para implantação das soluções de TI;</li> <li>• Garantir a infra-estrutura necessária à implementação das soluções de TI;</li> <li>• Participar da aprovação e quantificação de requerimentos;</li> <li>• Identificar, avaliar e implantar ferramentas para automação da produção;</li> <li>• Identificar mecanismos e estabelecer metas para desempenho do ambiente computacional de TI;</li> <li>• Planejar procedimentos de segurança e contingências;</li> <li>• Criar mecanismos de exceção para processos críticos;</li> </ul>

<b>16. Manter as soluções operacionais</b>	<ul style="list-style-type: none"> <li>• Executar os procedimentos descritos no manual de produção de cada sistema;</li> <li>• Executar procedimentos de segurança e contingência;</li> <li>• Monitorar o ambiente de produção com o objetivo de executar a revisão do planejamento;</li> </ul>
<b>17. Avaliar ambiente de TI</b>	<ul style="list-style-type: none"> <li>• Rever periodicamente a capacidade dos recursos computacionais para atendimento às soluções implementadas;</li> <li>• Elaborar e divulgar relatório mensal com informações que contemplem: tempo entre falhas, tempo e quantidade de paradas de processos, sistemas ou rede e tempo de recuperação de falhas;</li> <li>• Recomendar as ações corretivas e preventivas para o alcance das metas estabelecidas;</li> </ul>
<b>18. Planejar o suporte ao ambiente de TI</b>	<ul style="list-style-type: none"> <li>• Propor padrões de desempenho e segurança do ambiente de rede;</li> <li>• Propor normas e procedimentos técnicos para operação do ambiente de rede envolvendo os serviços disponibilizados pelos servidores e os processos das estações de trabalho;</li> <li>• Propor a aquisição de software, alocação de equipamentos e disponibilização de pontos de rede;</li> <li>• Identificar e propor metas para o atendimento aos usuários do ambiente de produtividade;</li> <li>• Identificar e propor critérios para contratação de parcerias de treinamento;</li> <li>• Identificar e propor ações que permitam aos usuários o desenvolvimento e a capacitação no uso das tecnologias do ambiente de produtividade;</li> </ul>
<b>19. Manter as soluções da área de TI nos clientes</b>	<ul style="list-style-type: none"> <li>• Instalar e configurar software nas estações de trabalho;</li> <li>• Atender aos chamados direcionados ao helpdesk, buscando prioritariamente solucionar por telefone;</li> <li>• Direcionar para outras áreas solucionadoras, quando necessário, os chamados pendentes e manter controle do atendimento do cliente;</li> <li>• Fornecer retorno aos usuários sobre o andamento de seus chamados;</li> <li>• Criar e disponibilizar o banco de dados de questões mais frequentes – FAQ aos usuários;</li> </ul>



<b>20 Administrar os recursos computacionais da rede do Inmetro</b>	<ul style="list-style-type: none"> <li>• Garantir a operação dos servidores e da rede de computadores dentro das normas e procedimentos técnicos estabelecidos;</li> <li>• Realizar o gerenciamento das contas;</li> <li>• Controlar mudanças na rede minimizando impactos para os usuários;</li> <li>• Efetuar rotinas de cópia e restauração dos dados dos servidores;</li> <li>• Instalar software e configurar hardware dos servidores;</li> </ul>
<b>21 Avaliar os resultados de suporte</b>	<ul style="list-style-type: none"> <li>• Medir e acompanhar padrões de desempenho e segurança do ambiente de rede;</li> <li>• Avaliar as mudanças no ambiente de rede;</li> <li>• Medir e divulgar os resultados obtidos com a capacitação dos usuários;</li> <li>• Elaborar e aplicar pesquisa de satisfação;</li> <li>• Elaborar e divulgar relatório estatístico mensal com tipo e frequência das chamadas ao suporte;</li> </ul>
<b>22 Dar suporte a banco de dados</b>	<ul style="list-style-type: none"> <li>• Avaliar tecnicamente produtos relacionados a área de administração de banco de dados;</li> <li>• Propor e acompanhar as rotinas de recuperação, reinicialização e de cópia das bases de dados;</li> <li>• Propor e acompanhar as rotinas de replicação das bases de dados;</li> <li>• Propor, implementar e analisar os procedimentos de atualização dos bancos de dados;</li> <li>• Propor e acompanhar os procedimentos de Segurança do banco de dados;</li> <li>• Suportar as equipes de desenvolvimento na execução de trabalhos de manutenção, projetos de sistemas e resolução de problemas;</li> <li>• Gerenciar as autorizações de acesso aos dados conforme política definida pelo Inmetro;</li> <li>• Propor, divulgar e monitorar toda a normatização relativa a administração, gerenciamento, auditoria e uso tanto dos dados como do dicionário de dados;</li> <li>• Propor a instalação de software de banco de dados;</li> <li>• Participar da aprovação do modelo físico;</li> </ul>

<p><b>23 Apoiar administrativamente a estrutura de TI</b></p>	<ul style="list-style-type: none"> <li>• Controlar a utilização de licenças de software, bem como a localização dos aplicativos instalados;</li> <li>• Manter inventário do parque de hardware e software instalado;</li> <li>• Fazer requisições para aquisição de licenças de software sempre que diagnosticada esta necessidade;</li> <li>• Estabelecer relacionamento com a biblioteca gerenciada pelo CIDIT;</li> <li>• Manter controle de fornecimento dos suprimentos para a área de informática;</li> <li>• Secretariar os níveis de gerência e coordenação da estrutura de TI;</li> </ul>
<p><b>24 Executar o suporte ao ambiente de TI</b></p>	<ul style="list-style-type: none"> <li>• Monitorar padrões de desempenho e segurança do ambiente de TI;</li> <li>• Cumprir as normas e procedimentos técnicos para operação do ambiente de TI envolvendo os serviços disponibilizados pelo software básico nos servidores e estações de trabalho;</li> <li>• Propor a aquisição de software básico, equipamentos e ferramentas de software básico;</li> <li>• Cumprir com as metas de satisfação dos usuários do atendimento;</li> <li>• Propor treinamento;</li> <li>• Implementar ações que permitam aos usuários o desenvolvimento e a capacitação no uso das tecnologias do ambiente de TI;</li> </ul>
<p><b>25 Promover a produtividade dos serviços da área de TI no Cliente</b></p>	<ul style="list-style-type: none"> <li>• Atender as necessidades eventuais ou emergenciais identificadas pelos consultores de TI para atendimento aos clientes.</li> <li>• Executar relatórios, planilhas e bancos de dados para atendimento de necessidades locais, emergenciais, propostas de correções e melhorias nas soluções implementadas.</li> </ul>

QUADRO 2 – Funções e Atribuições

Fonte: Projeto Básico de Terceirização de TI do Inmetro

Com o objetivo de solucionar as inadequações identificadas, a estrutura proposta contempla fundamentalmente as seguintes inovações:

- Um Comitê Normativo composto por representantes das diversas Diretorias do Inmetro passa a realizar a definição de políticas, prioridades e estratégias da Área de TI. O mesmo comitê deverá reunir-se periodicamente para avaliar os resultados alcançados e agilizar a comunicação dos mesmos;

- Um Comitê Operacional formado pela Gerência da Área de Informática, Gerência de T.I., Gestores de área , Coordenadores do Cliente, Coordenador de Atendimento, Coordenador de Suporte e Produção e Consultores de Atendimento. Esse comitê será responsável pela coordenação da operacionalização das decisões oriundas do Comitê Normativo.
- Uma Área de Apoio Tecnológico que deverá preocupar-se com padronização, normatização e metodologia deixando as áreas de Atendimento e Produção e Rede com foco de atuação nos clientes ;
- Suporte & Produção coordenação responsável pela operacionalização das soluções implantadas e manutenção do ambiente operacional (hardware, software e rede);
- A Área de Planejamento e Projeto de Infra-estrutura, que será responsável pelo planejamento do crescimento e da manutenção do nível de confiabilidade da infra-estrutura de TI e constitui uma equipe de apoio da coordenação de Suporte e Produção;
- Área de Suporte e Rede que dará assistência específica à implantação, administração e suporte à rede corporativa do Inmetro, acrescida das atividades de produção, necessárias à operação das soluções que estão sendo implementadas no Inmetro a partir de 1997;
- A Área de Produção será composta por uma equipe responsável pela operação dos processos dos sistemas corporativos implantados e estará subordinada à Área de Suporte e Produção;
- Atendimento coordenação responsável pela integração dos recursos de hardware e software visando prover e otimizar o ambiente e a solução desenvolvidos para o cliente/usuário focados na área de negócio dos clientes;
- A Área de Apoio ao Atendimento, uma equipe voltada para a integração das soluções corporativas, foi acrescida à estrutura de Atendimento para atuar fundamentalmente na necessidade de garantir consistência e integridade nas implementações de bases de dados e processos;
- Consultores de TI assistirão as Diretorias do Inmetro, criando o elo de ligação entre a Diretoria e a estrutura proporcionada pela área de TI. Este profissional deverá desenvolver o relacionamento com as Diretorias para

facilitar o entendimento das necessidades deste cliente e propor as soluções de escopo, prazo e custos adequados às expectativas;

- Apoio ao Usuário, é uma área específica para prestar os serviços de suporte aos usuários das estações de trabalho da rede. Esta estrutura deverá ser composta por um serviço de helpdesk que centralizará os chamados de problemas relacionados com o ambiente de informação do Inmetro e os solucionará ou encaminhará a outras áreas solucionadoras;
- Ambiente de Produtividade, área encarregada de acompanhar os processos de desenvolvimento e implantação de soluções para os clientes, buscando continuamente o aumento da produtividade e a melhoria da qualidade dos processos e provendo soluções de contorno ou emergenciais.

Através da descentralização planejada e da criação de novas funções, bem como, da reorganização e definição de responsabilidades, a Gerência da Área de TI ganharia agilidade e flexibilidade para:

- Traçar as diretrizes e estratégias de informática do Inmetro;
- Manter estreito relacionamento com os clientes, facilitando a definição das prioridades e o atendimento das necessidades;
- Indicar as prioridades para disponibilização das soluções;
- Administrar a execução de contratos por prestadores de serviços;
- Manter e Otimizar as soluções de tecnologia da informação, bem como a infra-estrutura para suportá-las;
- Atender as demandas presentes e futuras decorrentes de erros, modificações na legislação vigente, necessidades de gestão, necessidades de processos ou melhorias de tempo de resposta;
- Executar as tarefas constantes nos planos de TI com eficiência e eficácia.

#### **4.7 A ÁREA DE ATENDIMENTO**

A estrutura da área de Atendimento deveria ser composta por um coordenador que desempenhará suas funções de acordo com a metodologia e ferramentas para gestão de projetos de TI e terá a seguinte estrutura:

Assessoramento – Composta de Apoio ao Atendimento e os Consultores de TI.

Área de Serviços – Composta de Apoio ao Ambiente de Produtividade e Apoio ao Usuário.

#### **4.8 APOIO AO ATENDIMENTO**

Deveriam existir Administradores de Dados e de Banco de Dados, voltados para a integração das soluções corporativas, ligados à estrutura de Atendimento para atuar fundamentalmente na necessidade de garantir consistência e integridade nas implementações de bases de dados e processos;

#### **4.9 CONSULTORES DE TI**

Deveriam ainda existir consultores de TI, também chamados de analistas de negócios, subordinados ao Coordenador de Atendimento, sendo estes responsáveis pela interface entre o cliente e a área de TI. Seriam profissionais voltados para o negócio de seus clientes. Eventualmente estes profissionais farão a gestão dos serviços de desenvolvimento e manutenção dos sistemas das diretorias do Inmetro.

#### **4.10 AMBIENTE DE PRODUTIVIDADE**

As necessidades eventuais ou emergenciais identificadas pelos consultores de TI deveriam ser atendidas pelos profissionais alocados na equipe de Ambiente de Produtividade. Enquadram-se nestes casos aqueles relatórios, planilhas e bancos de dados para atendimento de necessidades locais, emergenciais, propostas de correções e melhorias nas soluções implementadas.

A construção de páginas para Internet, bem como o desenvolvimento e manutenção de sistemas que se utilizem da tecnologia de Intranet ou Extranet seriam de responsabilidade da Área de Ambiente de Produtividade e portanto tratadas como um dos projetos da Área de

Atendimento e geridos de acordo com a metodologia para gestão de projetos de TI. Existiriam diferenças somente no perfil da equipe que poderá contar com um designer além dos analistas de sistemas alocados ao projeto.

#### **4.11 APOIO AO USUÁRIO**

Todos os usuários das estações da rede do Inmetro deveriam ser atendidos por profissionais desta área para solução de qualquer problema relativo aos serviços prestados pela área de TI.

Com as facilidades proporcionadas pela rede do Inmetro, a centralização dos chamados para solução de problemas de TI deveria estar em uma área denominada Helpdesk.

Cabe ressaltar que a abertura de projetos do Serviço de Desenvolvimento e Manutenção de Sistemas, e as respectivas alocações de líderes de projeto e das equipes ocorreriam a partir da priorização das soluções, das definições de escopo de cada um dos projetos a serem desenvolvidos e da demanda dos usuários do Inmetro por correções e melhorias nas soluções implementadas.

Para a abertura de um destes projetos dever-se-ia propor uma metodologia e ferramentas que contemplem:

- Detecção das necessidades,
- Viabilidade,
- Prioridades e
- Definições estratégicas, subordinando a decisão de realização ou não aos Comitês Normativo, Operacional.

#### **4.12 A ÁREA DE SUPORTE E PRODUÇÃO**

A estrutura da área de suporte deveria ser composta por um coordenador de suporte, a quem estarão subordinados os seguintes serviços:

- Assessoramento no Planejamento e no Projeto de Infra-estrutura e
- Serviços de Suporte e de Rede e Produção.

#### **4.12.1 PLANEJAMENTO E PROJETO DE INFRA-ESTRUTURA**

O desempenho das funções da área deveria ser apoiado pelo profissional do Planejamento e Projeto de Infra-estrutura, que de maneira análoga a área de apoio ao Atendimento, seria responsável pelo planejamento e acompanhamento do crescimento da rede e garantia da qualidade dos serviços prestados.

#### **4.12.2 SUPORTE E REDE**

A equipe de administração e suporte a rede deveria ser constituída por administradores de rede e software básico, lotados nas unidades do prédio do Rio Comprido e no campus de Xerém.

#### **4.12.3 PRODUÇÃO**

A equipe de produção deveria ser constituída por analistas de produção permanentemente alocados nas unidades do Inmetro, tanto no campus de Xerém como no prédio do Rio Comprido.

O ponto central do relacionamento com as demais unidades do Inmetro é a Área de Informática, que é o órgão responsável pela gestão dos diversos serviços a serem executados.

Para a gestão da terceirização dos serviços de T.I. é fundamental que se utilize metodologia e ferramentas que subsidiem a realização de:

- Acompanhamento da operacionalização das funções de T.I. soluções e resultados;
- Garantia de funcionamento das soluções tecnológicas implementadas;
- Controle da propriedade do conhecimento dos processos e dos sistemas;
- Controle da execução da política de liberação de informações;
- Controle dos procedimentos emergenciais e de contingência;
- Monitoração da satisfação dos clientes em relação aos serviços prestados;

- Relacionamento administrativo.

#### **4.13 METODOLOGIA PARA GESTÃO DE PROJETOS DE TECNOLOGIA DA INFORMAÇÃO**

A gestão de qualquer projeto envolve as etapas de planejamento, acompanhamento e a revisão dos resultados atingidos pela equipe, em comparação àqueles planejados ou desejados e os eventuais ajustes no planejamento original.

Para a adequada gestão de projetos de TI, portanto, torna-se imprescindível utilizar uma metodologia que contemple:

- Definição de papéis dos membros da equipe e sua participação no planejamento e acompanhamento dos projetos;
- Definição dos tópicos a serem definidos formalmente quando do planejamento do projeto;
- Instrumentos de acompanhamento do progresso das atividades do Projeto;
- Instrumentos de acompanhamento dos custos do projeto;
- Instrumentos de avaliação dos resultados alcançados em relação aos objetivos do projeto;
- Instrumentos de identificação e gerenciamento dos riscos associados ao projeto;
- Instrumento de avaliação do desempenho da equipe;
- Instrumento de avaliação do grau de atendimento e satisfação do cliente.

#### **4.14 METODOLOGIA PARA DESENVOLVIMENTO E MANUTENÇÃO DE SISTEMAS**

A fim de padronizar o processo de desenvolvimento e manutenção de sistemas, de forma a facilitar o planejamento e acompanhamento de atividades, bem como dar garantia de qualidade aos produtos gerados, torna-se necessário que uma metodologia de desenvolvimento e manutenção e suas ferramentas sejam praticadas pelas equipes.



Esta metodologia deverá estabelecer padrões para as etapas e atividades, bem como para os produtos do projeto que contemplem no mínimo:

- Estimativa de esforço da especificação do sistema/manutenção,
- Estimativa do esforço da Implementação da solução especificada,
- Ferramentas de gerações de código,
- Uso de prototipação nas soluções propostas,
- Uso de estruturação com ferramentas Case das soluções propostas,
- Especificação de níveis de certificação de qualidade de software a serem obtidos pela solução que contemple no mínimo os itens relacionados abaixo:
- Documentação;
- Operacionalização;
- Desempenho Esperado;
- Integridade e Segurança;
- Estabilidade;
- Compatibilidade entre as diversas partes e
- Satisfação do cliente.

#### **4.15 METODOLOGIA DE MANUTENÇÃO DE EQUIPAMENTOS**

A perfeita execução dos serviços de manutenção de equipamentos pressupõe, além da qualificação do corpo técnico, a utilização de métodos de trabalho eficazes, e o uso de instrumentos e ferramentas adequadas em cada etapa dos trabalhos. Assim uma Metodologia de Manutenção de Equipamentos e de Softwares deve descrever as diversas etapas do trabalho de manutenção com a indicação das técnicas, ferramentas e instrumentos utilizados.

Para a adequada prestação do serviço, é imprescindível utilizar uma metodologia que contemple tanto para manutenção preventiva, como para a corretiva, assim como:

- Os procedimentos para chamada do técnico;
- A sistemática de acompanhamento dos atendimentos;

- A sistemática de controle de estoque das peças de reposição e dos equipamentos sobressalentes;
- As atividades de campo e de laboratório;
- As ferramentas e instrumentos utilizados em cada atividade;
- As atividades do centro de suporte;
- Os procedimentos de diagnóstico com indicação dos Softwares de autodiagnóstico utilizados;
- Os procedimentos de detecção de vírus com indicação dos Softwares utilizados;
- Os materiais de limpeza utilizados;
- Os procedimentos de avaliação do nível de satisfação do Cliente.

#### **4.16 CONSIDERAÇÕES FINAIS DO CAPÍTULO**

Neste capítulo foi apresentada a estrutura do perfil do estudo, tendo como foco o Inmetro, seus processos, planejamentos, sua estrutura, abrangendo ainda as metodologias de desenvolvimento, manutenção e gestão destes, evidenciando as técnicas de apoio, atendimento e suporte, com isto delimita-se o escopo do presente trabalho.

No próximo capítulo serão feitas considerações a respeito das métricas apresentadas no segundo capítulo, objetivando realizarmos uma análise comparativa entre elas.

## **5 ANÁLISE COMPARATIVA DAS MÉTRICAS**

### **5.1 CONSIDERAÇÕES INICIAIS DO CAPÍTULO**

Neste capítulo, pretende-se fazer uma análise comparativa das métricas apresentadas, para isto foi desenvolvido um trabalho com o objetivo de evidenciar as vantagens e desvantagens de cada uma delas.

São colocados os dados em análise delimitando o escopo de atuação, assim como será apresentado um quadro comparativo das principais métricas, que foram objeto deste estudo.

Após isto trataremos das justificativas e hipóteses e questões e fechando o capítulo, delimitaremos o presente estudo e apresentaremos sua estrutura.

### **5.2 DADOS EM ANÁLISE**

O objetivo desta comparação é determinar quais métricas podem ser utilizadas em estimativas de tamanho de projetos, aplicando-as no início de desenvolvimento e, também, durante todo o processo de construção de software.

Algumas observações a respeito do assunto podem ser relacionadas, conforme segue abaixo.

#### **5.2.1 LINHA DE CÓDIGO**

É a mais intuitiva, pois até mesmo os mais jovens iniciantes em programação começam sua jornada comparando o número de linhas de códigos do seu programa com o do seu companheiro de turma. A aparente facilidade na utilização de linhas de código pode levar os menos avisados a incorrer em grandes erros. Pode-se cair na falha de usar duas avaliações ou medidas diferentes, se alguns dos pontos não forem bem esclarecidos ao exprimir uma determinada quantidade de Linhas de Códigos, como segue:

- A inclusão ou não de linhas com comentários na contagem;

- A inclusão ou não de linhas em branco;
- A inclusão ou não de comandos nulos;
- A inclusão ou não na contagem de diretrizes de compilação;
- A contagem ou não de múltiplos comandos ou declarações em uma única linha como várias linhas;
- A contagem ou não de uma única linha nos casos em que um único comando ou declaração é expresso em múltiplas linhas;
- A inclusão ou não de delimitadores de blocos de comandos, nos casos em que haja mais de um comando;
- A desconsideração ou não de delimitadores de blocos de comandos nos casos em que sua participação seja opcional.

Sendo assim a contagem de linhas de código pode não ser o que parece. Não é tão intuitiva assim, pois pode envolver uma série de regras que devem ser bem definidas antes de sua aplicação, como visto.

Como vantagem, tem-se a utilização em estimativas e contabilização automática. Algumas organizações, as quais usam, principalmente, linguagens de terceira geração homogêneas como COBOL, podem usar a contagem de Linhas de Código com um pouco mais de sucesso se a organização têm padrões de codificação rígidos e procedimentos de contagem rigorosos. Porém, com o advento de ambientes de linguagens múltiplas, tecnologia orientada a objetos e a documentos, a contagem de linhas de código tem sido crescentemente irrelevante.

Outro aspecto que influencia na aplicação da Contagem de Linhas de Código é a falta de significado para o cliente, ou seja, para um cliente, saber que o seu sistema possui 10.000 linhas de código não é muito significativo.

### **5.2.2 COCOMO**

A desvantagem da métrica COCOMO é que os coeficientes da métrica (a, b, c, d) não são aplicáveis a tamanho, ou seja, a produtividade é diferente, ficando difícil a sua comparação. Na realidade o tamanho é o parâmetro de entrada da métrica. Como vantagens, pode-se utilizar em estimativas e contabilização automáticas.

### **5.2.3 PUTNAM**

A desvantagem da métrica PUTNAM é estar vinculada à linguagem utilizada e exige tempo para obterem-se valores reais para os parâmetros da fórmula. Como vantagem, pode ser utilizada em estimativas e contabilização automáticas.

### **5.2.4 FEATURE POINTS**

A desvantagem da métrica Feature Points é não poder ser aplicada por leigos, pois exige o conhecimento de complexidade de algoritmo. Como vantagens, pode-se utilizar em estimativas e contabilização automáticas e ser utilizada para gerar pontos por função de software de tempo real e softwares embutidos.

### **5.2.5 CONTAGEM DE SÍMBOLOS – SOFTWARE SCIENCE HALSTEAD**

A métrica de Contagem de Símbolos está restrita a programas e não a projetos de sistemas, baseando-se, na sintaxe e não no contexto, o que a torna difícil de ser aplicada. Além disto, não é utilizada em estimativa.

### **5.2.6 PONTOS DE FUNÇÃO**

A desvantagem da métrica de pontos de função, segundo KITCHENHAM (1997), é ter falhas em sua construção. Isto se deve à utilização de pesos para definir a classificação das funções. Este fato pode nos levar a diferentes abordagens quando da contagem, ou seja, pode haver problemas se a classificação se comportar de modo inesperado, levando-se a afirmar, erroneamente, que uma aplicação é maior que outra, quando não o é. Isto alerta para o fato de se usar a métrica com cuidado sobre a versão utilizada. Desde a sua criação em 1979, surgiram várias versões da métrica e, quando se compara tamanho de software, é

preciso saber qual a versão da métrica que está sendo usada. Mesmo utilizando-se a mesma versão pode haver uma diferença de mais ou menos 12%, estudo realizado por Kitchenham na contagem para o mesmo produto realizado por pessoas da mesma organização.

FUREY (1997) incentiva o uso de pontos de função por quantificar o tamanho e a complexidade da aplicação de software. Sendo consistente e repetitiva, ajuda a normalizar os dados, possibilitando comparações em nível do âmbito do projeto e expectativas de cliente.

A Análise de Pontos de Função leva as organizações a normalizar seus dados como custo, duração, defeitos, previsão de pessoal, e assim por diante. Também pode ajudar, efetivamente, na comparação de tecnologias discrepantes. Ela estabelece um modo melhor para comparar as organizações e a produtividade de projeto, especialmente por tecnologias diferentes, não sendo uma medida de produtividade individual. Quando usada com dados de esforço, pode servir como uma medida de macro-nível de produtividade de processo.

### 5.3 ANÁLISE COMPARATIVA

Com base nos dados podemos desenvolver uma análise excluindo os métodos e tentando assim chegar a um único método.

<b>Características</b>	<b>Linha de Código</b>	<b>Software Science</b>	<b>Cocomo</b>	<b>Feature Points</b>	<b>Putnam</b>	<b>Pontos de Função</b>
<b>1. Produção de resultados aceitáveis</b>	Sim	Sim	Sim	Sim	Sim	Sim
<b>2. Significado para o Usuário final, sem conhecimento da linguagem de programação</b>	Não	Não	Não	Não	Não	Sim
<b>3. Utilizado em estimativas</b>	Sim	Não	Sim	Sim	Sim	Sim
<b>4. Contabilização automática</b>	Sim	Sim	Sim	Sim	Sim	Sim

QUADRO 3 - Comparativo de Métricas

Fonte: O próprio autor

Analisando o quadro acima pode afirmar que do ponto de vista da produção de resultados todas as métricas são aceitáveis, o que de um modo grosseiro não poderia ser

balizador na hora da contratação, ou seja, do ponto de vista dos resultados práticos esta é uma característica que não serve como qualificador ou diferenciador das métricas, assim como a Contabilização Automática que não gerou nenhum ponto crítico na avaliação, ou seja, todos passaram por este critério, assim como no anterior. Já o critério que determina que o método estudado pode ou não ser utilizado em estimativa somente elimina o método Software Science. Com isto e analisando o critério que sobrou, a saber, Significado para o Usuário final, sem conhecimento da linguagem de programação, chegamos ao método de Pontos de Função como o mais apropriado sob a ótica dos critérios apresentados.

#### **5.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO**

Neste capítulo foi apresentada uma análise comparativa das métricas envolvidas no presente trabalho, apontando os pontos fortes e fracos das mesmas.

Para conhecimento do leitor, foram apresentadas, no capítulo anterior, seis métricas, conforme demonstrado no quadro 3, enfocando a capacidade de estimativa com o objetivo de melhoria do processo de desenvolvimento e manutenção de software. Tomando-se como base as características adotadas pelo IFPUG, elas foram analisadas e comparadas, apontando-se as suas vantagens e desvantagens. Portanto, das seis, cinco são métricas utilizadas em estimativas, uma vez que a Contagem de Símbolos - Software Science não pode ser utilizada em estimativa:

- CONTAGEM DE LINHAS DE CÓDIGO;
- COCOMO;
- PUTNAM;
- FEATURE POINTS;
- PONTOS DE FUNÇÃO.

No próximo capítulo serão feitas considerações finais e conclusivas a respeito do estudo a respeito das métricas apresentadas.

## **6 CONCLUSÃO**

### **6.1 RESULTADOS**

A indefinição por parte do Governo federal de uma política única de contratação de empresas prestadoras de serviços para desenvolvimento e manutenção de sistemas informatizados, e como conseqüência a falta de uma política de gerenciamento, a necessidade de podermos controlar com mais eficiência nossos contratos com os prestadores de serviços, um domínio total da situação no que diz respeito aos questionamentos feitos pelos diversos órgãos de controle internos e externos, nos motivou a realizar o presente estudo.

As métricas de contagem de linhas de código, COCOMO e PUTMAN, exigem o prévio desenvolvimento dos programas para a medição, mas podem ser utilizadas em estimativa, porém não nos atendem, pois precisamos estimar o desenvolvimento antes de tê-lo desenvolvido.

Baseados nas hipóteses e questões apresentadas no primeiro capítulo, podemos concluir que as métricas e estimativas de tamanhos e por conseqüência de custos de softwares propostas para o desenvolvimento somente são fiéis à realidade da situação e momentos particulares, ou seja, em cada tempo e condições deve ser avaliada a ferramenta apropriada, uma vez que as técnicas acompanham a rápida evolução do setor de TI, o que não garante que uma técnica adotada hoje seja boa amanhã. Porém é fato que precisamos medir e que estamos no caminho correto, ou seja, temos que ter como comprovar todos os investimentos e justificar tecnicamente os passos tomados e atrelada a isto vem à questão da escolha de qual metodologia adotar, mais ainda, se adotamos uma já existente ou criamos uma nossa. Do nosso ponto de vista, e entendedores que não somos um órgão de TI, ou seja, somos prestadores de serviços de informação, optamos por adotar uma já existente, pois a criação de uma própria demandaria muitos recursos que ora devem ser aplicados na solução dos problemas dos nossos usuários. A metodologia escolhida deve ser constantemente reavaliada, pois como já dito anteriormente a questão temporal é um ponto balizador e sustentador para uma boa metodologia.

### **6.2 CONCLUSÃO E CONSIDERAÇÕES FINAIS**



Conforme pôde ser evidenciado em todo estudo, tendo como base a revisão bibliográfica realizada, bem como a vivência profissional, optamos pela métrica de Pontos por Função como a mais adequada ao escopo objeto deste trabalho.

A métrica que mais atende aos objetivos de comparação mostrada no quadro 3 é a métrica de pontos por função, pois ela faz estimativa antes e no decorrer do desenvolvimento de um projeto de software, além de produzir resultados mais exatos, não dependendo da interpretação do usuário.

Cabe ressaltar que esta é uma decisão estratégica e que foi delimitada ao escopo deste trabalho, no tempo e condições presentes, e que tal decisão deve ser reavaliada a cada tempo e condições, como já dito anteriormente.

Como dito anteriormente, não pretendemos calar todas as perguntas a respeito deste extenso assunto que é a métrica de desenvolvimento de sistemas informatizados, porém queremos mostrar que este assunto não é considerado pela área de TI do Inmetro como um assunto banal e que temos um longo caminho para chegarmos a um equilíbrio e que este pode ser rapidamente desequilibrado por ações externas advindas do Governo Federal ou até mesmo de políticas de mercado.

### **6.3 SUGESTÕES PARA NOVOS TRABALHOS**

Vários trabalhos podem ser desenvolvidos a partir da presente dissertação, desde temas e relacionados à gestão, métricas, entre outros, que certamente são de interesse de vários gestores públicos e privados.

Como primeira sugestão, há a oportunidade de desenvolver estudo relacionado a um levantamento da evolução da metodologia de desenvolvimento e manutenção de sistemas informatizados sob a ótica do Governo Federal como um todo, quem sabe criando uma metodologia comum aos órgãos da administração direta e autarquias, este poderia ser apresentado ao Ministério de Planejamento visando uma unificação de procedimentos e a criação de uma base de dados única, criando assim uma tabela de pontos de sistemas, quem sabe classificados por complexidade, e outros aspectos.

Outra investigação relevante seria quanto aos aspectos comportamentais envolvidos nos programas de terceirização das áreas de TI do Governo Federal. Hoje estas informações são bastante incipientes e requerem um esforço muito grande para sua organização. Neste trabalho poderia ser levantada uma estatística sobre quais as métricas mais usadas no Governo

e quais as experiências bem sucedidas, ajudando assim os gerentes de TI na tomada de decisão a respeito deste assunto.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

A EMPRESA. Disponível em <http://www.dataprev.gov.br/aprese/aempresa.shtm>. Acessado em novembro de 2002.

ALVES-MAZZOTTI, Alda Judith; GEWANDSZNAJDER, Fernando. **O Método das Ciências Naturais e Sociais – Pesquisa Quantitativa e Qualitativa**. 2 Ed. São Paulo: Pioneira, 1998.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Informação e Documentação – Referências – Elaboração: NBR 6023**. Rio de Janeiro: ABNT, 2002.

\_\_\_\_\_. **Preparação de Índice de Publicações: NBR 6034**. Rio de Janeiro: ABNT, 1989.

\_\_\_\_\_. **Numeração Progressiva das Seções de um Documento: NBR 6024**. Rio de Janeiro: ABNT, 1989.

\_\_\_\_\_. **Informação e Documentação - Trabalhos Acadêmicos - Apresentação: NBR 14724**. Rio de Janeiro: ABNT, 2001.

BAHIA, Álvaro de Sá. **O Uso de Métricas de Complexidade para Controle da Qualidade de Software Científico**. Porto Alegre. GPGCC da UFRGS, 1991.

BOEHM, B.W. **Software Engineering Economics**. Englewood Cliffs, NJ: Prentice-Hall, 1981.

BRAGA, Antônio. **Análise de pontos de função**. Rio de Janeiro: Infobook, 1996

CELEPAR. MDS: **Metodologia de Desenvolvimento de Serviços**. Curitiba: CELEPAR, 1995.

CENTRO NACIONAL DE PESQUISA EM INFORMÁTICA (Brasil). Disponível em <http://www.cnpi.org.br/news.asp>. Acessado em fevereiro de 2004.

CONTE, S.D.; SHEN V.Y. **Software Engineering Metrics and Models**. Menlo Park, California: Benjamin/Cummings Publishing, 1985.

DEMARCO, T.. **Controle de Projetos de Software**. local: Campos, 1991.

**DIRETRIZES TECNOLÓGICAS**. Disponível em [http://www-comite/default\\_ie.htm](http://www-comite/default_ie.htm). Acessado em junho de 2003.

DFJUG. **BOLETIM DFJUG 121**. Disponível em [http://www.dfjug.org/DFJUG/boletins/bolet\\_121.htm](http://www.dfjug.org/DFJUG/boletins/bolet_121.htm). Acessado em Dezembro de 2004.

**FOLHA ON LINE**. Disponível em <http://www.folha.uol.com.br> . Acessado em março de 2003.

FUREY, Sean. **Why We Should Use Function Points**. IEEE Software, Los Alamitos, Mar./Apr. 1997

**GERENCIAMENTO DE PROJETOS – PMI.** Disponível em [http://www.pmisp.org.br/exe/pmi/ger\\_projetos.asp](http://www.pmisp.org.br/exe/pmi/ger_projetos.asp). Acessado em Junho de 2004.

HALTEAD, M. H.. **Elements of Software Science.** New York. Elsevier Computer Science Library, 1977.

HAZAN, Cláudia. **Metodologia para o uso de indicadores na gerência de projetos de desenvolvimento de software.** 213 f. Dissertação (Mestrado em Sistema e Computação) - Instituto Militar de Engenharia, Rio de Janeiro, 1999.

HAZAN, Cláudia; SILVA, Paulo Afonso Lopes e. **Implantando a Melhoria Contínua no Processo de Desenvolvimento de Software – Artigo.** Instituto Militar de Engenharia – Departamento de Engenharia de Sistemas.

INTERNATIONAL FUNCTION POINT USER GROUP (IFPUG). **Análise de Pontos de Função.** IFPUG, 1991, Release 3.1.

\_\_\_\_\_. **Análise de Pontos por Função.** [S.l.]: IFPUG, 1991, Release 3.1 do IFPUG.

\_\_\_\_\_. **Manual de Contagem Práticas de Pontos de Função.** Versão 4.0. Atlanta, Chairperson: IFPUG, 1994.

JONES, Carpers. **Applied Software Measurement: assuring productivity and quality.** New York: McGraw-Hill, 1996.

KITCHENHAM, Barbara. **The Problem With Function Points.** *IEEE Software*, Los Alamitos, v.14,n.2. Mar./Apr. 1997

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Metodologia Científica**. 2 Ed. São Paulo: Atlas, 1991.

LOTUS NOTES. Disponível em <http://www.lotus.com>. Acessado em janeiro de 1998.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Técnicas de pesquisa: Planejamento e execução de pesquisas amostragens e técnica de pesquisa, elaboração, análise e interpretação de dados**. 3 ed. São Paulo: Atlas, 1996.

MICROSOFT. Disponível em <http://www.microsoft>. Acessado em Janeiro de 1998.

BRASIL. Ministério da Ciência e Tecnologia. **Tecnologia da Informação. Política Nacional de Informática**. Disponível em <http://www.mct.gov.br/Temas/info/pni/pni.htm>. Acessado em julho de 2002.

\_\_\_\_\_. **Tecnologia da Informação. Indicadores do Setor. Glossário de termos da qualidade**. Disponível em <http://www.mct.gov.br/Temas/info/Dsi/quali99/99anexo2.htm>. Acessado em fevereiro de 2002.

**MODELO DE MATURIDADE DE CAPABILIDADE DE SOFTWARE** – Versão 1.2 de 11/10/2001. Disponível em [http://www.mct.gov.br/Temas/info/Dsi/PBQP/Reuniao20BSB/CMM-TR24\\_V1.2.pdf](http://www.mct.gov.br/Temas/info/Dsi/PBQP/Reuniao20BSB/CMM-TR24_V1.2.pdf). Acessado em Junho de 2004.

**O PMBOK – A GUIDE TO THE PROJECT MANAGEMENT BODY OF KNOWLEDGE**. Disponível em <http://www.pmirs.org>. e [http://www.pmirs.org/PMI21\\_PMBOK.htm](http://www.pmirs.org/PMI21_PMBOK.htm). Acessados em novembro de 2002.

**O PMI – PROJECT MANAGEMENT INSTITUTE.** Disponível em [http://www.pmir.org/PMI21\\_PMI.htm](http://www.pmir.org/PMI21_PMI.htm). Acessado em novembro de 2002.

**PLANO ESTRATÉGICO DO INMETRO PARA 2002 – 2010 VERSÃO DE MARÇO DE 2003.** Disponível em <http://www.inmetro.gov.br>. Acessado em julho de 2004.

POPPER, Karl, **La Logica de la Investigación Científica**, Tradução de V. Sanchez de Zavala, Madrid: Tecnos, 1973.

PRESMANN, Roger. **Engenharia de Software**. São Paulo: Makron Books, 1995.

**QUALIDADE BRASIL – RELATÓRIO DE ATIVIDADES 2003 – Relatório de Atividades**, Serviço de Comunicação. Rio de Janeiro, 2003. 27 p.

**SISTEMA DE QUALIDADE** – Disponível em <http://www.abes.org.br/gruptrab/QUALIDADE/cmm.htm>. Acessado em Junho de 2004.

SOFTEX, **Clipping Pesquisa MIT/SOFTEX** por Ana Lucia do Nascimento – Disponível em <http://www.softex.br/cgi/cgilua.exe/sys/start.htm?infoid=2152&sid=176>. Acessado em Dezembro de 2004.

UNIVERSIDADE FEDERAL FLUMINENSE, **Apresentação de Trabalhos Monográficos de Conclusão de Curso**, Pró-Reitoria de Pesquisa e Pós-Graduação. 6. Ed. Niterói: EdUFF, 2003. 86 p.

\_\_\_\_\_. **Guia de Formatação do LATEC/UFF baseado nas Normas da ABNT para Monografias e Dissertações**. Niterói: Laboratório de Tecnologia, Gestão de Negócios e Meio

Ambiente Centro de Documentação Miguel de Simoni – Organizadoras: Maria Emília Peluso Teixeira e Andréa Brasil, 2004. 68 p.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. **Análise de Pontos de Função – Medição, Estimativas e Gerenciamento de Projetos de Software**. local: Érica, 2003.



## GLOSSÁRIO

**ALGORITMO** – É o conjunto de regras que devem ser expressas completamente, a fim de resolver um problema computacional.

**CAPABILITY MATURITY MODEL (CMM)** - É o fundamento para se construir sistematicamente um conjunto de ferramentas, inclusive o questionário da maturidade, que são úteis na melhoria do processo de software. O ponto essencial a ser lembrado é que o modelo, não o questionário, é a base para a melhoria do processo de software.

**DFJUG** – Brasília Java Users Group. O Grupo de usuários na linguagem de programação Java iniciou suas atividades em Fevereiro de 1998 com o intuito de difundir a linguagem de programação Java em seus mais diversos aspectos, com cursos, palestras entre outros meios de divulgação.

**FERRAMENTAS CASE** – Ferramenta ou conjunto de ferramentas que automatizam tarefas que compõem o processo de desenvolvimento de software. Sistema Computacional composto de ferramentas que suportam a automação do ciclo de vida do software e permite o uso efetivo dos princípios e práticas gerais de engenharia de software.

**INTERNATIONAL FUNCTION POINT USERS GROUP (IFPUG)** - É uma entidade sem fins lucrativos, composta por pessoas e empresas de diversos países, cuja finalidade é promover um melhor gerenciamento dos processos de desenvolvimento e manutenção de software com o uso de pontos de função e outras técnicas de medição.

**PONTOS DE FUNÇÃO** - A Métrica Orientada à Função, ou simplesmente, por Pontos de Função, consiste em um método para medição de software do ponto de vista do usuário, que determina de forma consistente o tamanho e a complexidade de um software, sob a perspectiva do usuário.

PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK®) - É um termo que abrange o universo do conhecimento da profissão de Gerenciamento de Projetos. Este universo de conhecimento vem dos praticantes e acadêmicos que utilizam e desenvolvem tanto as práticas amplamente testadas e aprovadas quanto aquelas modernas e inovadoras, com aplicação mais restrita.

SOFTEX – Sociedade para promoção da Excelência do Software Brasileiro. Criada em 1996 é uma Organização da Sociedade Civil de Interesse Público (OSCIP), que atua de diversas formas, e em diversas frentes, para o conhecimento e difusão da indústria brasileira de software.