A FEA e a USP respeitam os direitos autorais deste trabalho. Nós acreditamos que a melhor proteção contra o uso ilegítimo deste texto é a publicação online. Além de preservar o conteúdo motiva-nos oferecer à sociedade o conhecimento produzido no âmbito da universidade pública e dar publicidade ao esforço do pesquisador. Entretanto, caso não seja do interesse do autor manter o documento online, pedimos compreensão em relação à iniciativa e o contato pelo e-mail bibfea@usp.br para que possamos tomar as providências cabíveis (remoção da tese ou dissertação da BDTD).

UNIVERSIDADE DE SAO PAULO

FACULDADE DE ECONOMIA, ADMINISTRACAO E CONTABILIDADE

DEPARTAMENTO DE CONTABILIDADE

CONTROLE DO DESENVOLVIMENTO DE SOFTWARE SEM
ESPECIALISTA EM INFORMATICA

UMA VISAO DA CONTROLADORIA E DA QUALIDADE

JOSE CARLOS PEREIRA

1 9 9 1

T658.151 P436c e.2

T67954

260002872

Powered by RildProStar - www.totoprocess.com.br

T658.151

P436c

e.2

UNIVERSIDADE DE SAO PAULO

FACULDADE DE ECONOMIA, ADMINISTRACAO E CONTABILIDADE

DEPARTAMENTO DE CONTABILIDADE

**CONTROLE DO DESENVOLVIMENTO DE SOFTWARE SEM
ESPECIALISTA EM INFORMATICA**

UMA VISAO DA CONTROLADORIA E DA QUALIDADE

JOSE CARLOS PEREIRA

**DISSERTACAO APRESENTADA A
FACULDADE DE ECONOMIA,
ADMINISTRACAO E CONTABILIDADE
DA UNIVERSIDADE DE SAO PAULO
PARA OBTENCAO DO TITULO DE
MESTRE EM CONTABILIDADE**

67954

ORIENTADOR : ANTONIO DE LOUREIRO GIL

SAO PAULO

1991

67954

Reitor da Universidade de São Paulo

Prof. Dr. Roberto Leal Lobo e
Silva Filho

Diretor da Fac. de Economia,
Administração e Contabilidade

Prof. Dr. Eduardo Pinheiro
Gondim Vasconcellos

Chefe do Depto. de Contabilidade
e Atuária

Prof. Dr. Eliseu Martins

A Minha Esposa e Filhas
ANGELINA, BIANCA E PALOMA
Com Muito Amor

AGRADECIMENTOS

- . Ao Prof. Dr. Antonio de Loureiro Gil pela excelente e dedicada orientação, pelo companherismo, amizade e incentivo;
- . Ao Prof. Dr. Armando Catelli pelo estímulo ao livre pensamento e à busca do nosso potencial;
- . Aos colegas das empresas Tecnosan e Natron pela colaboração.
- . Aos meus familiares pelo apoio incondicional, e
- . A minha esposa e filhas pela solidariedade, e compreensão pelo tempo que as privei de atenção.

**"A liberdade é a consciência
das nossas necessidades"**

Baruch Espinosa

INDICE

Apresentação.....	PC
Capítulo 1 - IDENTIFICAÇÃO DA DISSERTAÇÃO	
1.1. INTRODUÇÃO.....	5
1.2. OBJETIVO, OBJETO DE ESTUDO E ESTRUTURAÇÃO DA DA DISSERTAÇÃO.....	8
1.3. A QUEM O TEMA PODERÁ INTERESSAR.....	12
FIGURAS.....	28
CITAÇÕES.....	30
Capítulo 2 - IDENTIFICAÇÃO DOS ASSUNTOS TRATADOS NESTA DISSERTAÇÃO	
2.1. CONCEITOS SOBRE DESENVOLVIMENTO DE SOFTWARE.....	33
2.2. CONCEITOS E CLASSIFICAÇÃO DE CONTROLE	
2.2.1. Conceituação.....	37
2.2.2. Objetivo do controle.....	37
2.2.3. Objeto do controle.....	37
2.2.4. Classificação do controle.....	37
2.2.5. Como acontece o controle.....	37
2.2.6. Eficiência e eficácia do controle.....	37
2.2.7. Limitações do controle.....	50
2.3. CONCEITOS DE CONTROLADORIA	
2.3.1. Ramo do saber.....	50
2.3.2. órgão administrativo.....	50
2.3.3. Pré-requisitos para o exercício da controladoria.....	50
2.4. CONCEITOS DE QUALIDADE DE PROGRAMAS	
2.4.1. Conceitos.....	50
2.4.2. Qualidade de especificação.....	50
2.4.3. Qualidade de projetos.....	50
2.4.4. Qualidade de programa.....	50
FIGURAS.....	50
CITAÇÕES.....	50

Capítulo 3 - CARACTERIZAÇÃO DE SOFTWARES

3.1.	O QUE É SOFTWARE.....	113
3.1.1.	Fatores que geram a necessidade de programas de computador.....	116
3.1.2.	Caracterização de um programa de computador em função da frequência de uso.....	119
3.2.	QUEM UTILIZA O SOFTWARE	
3.2.1.	Profissionais da área a que se refere o assunto tratado pelo programa.....	124
3.2.2.	Profissionais de informática.....	125
3.2.3.	Utilização conjunta.....	126
3.2.4.	Pessoas não especializadas em informática e que também não são da área a que se refere o assunto tratado pelo programa.....	127
3.3.	POSSÍVEIS CARACTERÍSTICAS DO LOCAL ONDE O PROGRAMA É DESENVOLVIDO	
3.3.1.	Estágio embrionário.....	128
3.3.2.	Despertar da razão.....	130
3.3.3.	Definição de uma identidade.....	131
3.3.4.	Estágio da convergência.....	132
3.3.5.	Estágio da consolidação.....	134
3.4.	COMO OS SOFTWARES SÃO DESENVOLVIDOS SEM ESPECIALISTAS	
3.4.1.	Quem é o desenvolvedor não especializado em informática.....	136
3.4.2.	Como o usuário final desenvolve seus programas.....	138
3.5.	QUE TIPO DE PROGRAMA É DESENVOLVIDO SEM ESPECIALISTAS.....	143
3.6.	QUALIDADE DOS PROGRAMAS DESENVOLVIDOS SEM ESPECIALISTAS.....	145
3.6.1.	Qualidade da especificação dos programas desenvolvidos sem especialistas em informática	147
3.6.2.	Qualidade de projeto dos programas desenvolvidos sem especialistas em informática.....	156
3.6.3.	Impacto da qualidade de pseudo-especificação e de pseudoprojeto na qualidade de programa....	163

	PG.
3.6.4. Comentários adicionais sobre a qualidade de programas desenvolvidos sem especialistas em informática.....	184
3.6.5. Conclusão sobre a qualidade de programas desenvolvidos sem especialistas em informática.....	185
3.7. EXEMPLO DE PROBLEMAS OCASIONADOS PELA NÃO QUALIDADE DE PROGRAMAS.....	188
 Capítulo 4 - CONTROLE ECONÔMICO DO DESENVOLVIMENTO DE SOFTWARES	
4.1. FATORES DA QUALIDADE DE SOFTWARES SÃO AGENTES ECONÔMICOS	
4.1.1. Confiabilidade conceitual de programas.....	193
4.1.2. Confiabilidade de representação de programas.....	194
4.1.3. Utilização de programas.....	195
4.2. INDICADORES DE QUALIDADE.....	198
4.3. CONTROLE DE DESENVOLVIMENTO DE PROGRAMAS	
4.3.1. Objetivo do controle no desenvolvimento de programas.....	205
4.3.2. Tipo de controle em relação à forma de gestão.....	206
4.4. COMO PODE ACONTECER O CONTROLE ECONÔMICO DO DESENVOLVIMENTO DO SOFTWARE SEM ESPECIALISTAS	
4.4.1. A participação do sistema de informações no controle do desenvolvimento de software sem especialistas.....	214
4.4.2. Modelo de tomada de decisões.....	227
4.5. A CONTROLADORIA E O DESENVOLVIMENTO DE PROGRAMAS PELO USUÁRIO FINAL.....	235
4.5.1. Forma de atuação da controladoria.....	236
4.5.2. A controladoria e o estágio da informatização.....	240
4.5.3. Relacionamento entre aqueles que exercem a controladoria e as diversas áreas que desenvolvem os seus próprios programas.....	242

	PG.
4.5.4. Produto do trabalho da controladoria.....	243
FIGURAS.....	245
CITAÇÕES.....	247
Capítulo 5 - CONCLUSÃO.....	250
5.1. INFORMAÇÕES NECESSÁRIAS PARA EXECUÇÃO DO CONTROLE.....	251
5.2. CONCEITOS QUE SUSTENTAM O CONTROLE.....	255
5.3. MODELO DE TOMADA DE DECISÕES.....	257
5.4. A CONTROLADORIA.....	258
5.5. CONSIDERAÇÕES FINAIS.....	260
GLOSSÁRIO.....	262
BIBLIOGRAFIA.....	268

ÍNDICE DE FÍGURAS

	PG.
I.1. REPRESENTAÇÃO DA ABRANGÊNCIA DA DISSERTAÇÃO.....	29
II.1. VISÃO MACRO DO DESENVOLVIMENTO DE SOFTWARE.....	99
II.2. MODELO EM CASCATA DE DESENVOLVIMENTO DE SOFTWARE.....	100
II.3. CICLO DE VIDA DE PROJETO ESTRUTURADO.....	101
II.4. CICLO DE VIDA TRADICIONAL DE SISTEMAS DE PROC. DE DADOS.....	102
II.5. CONTROLE.....	103
II.6. CICLO DE REALIZAÇÃO DO CONTROLE.....	104
II.7. GESTÃO EMPRESARIAL.....	105
II.8. VISÃO MAIS AMPLA DO PROCESSO DE CONTROLE.....	106
II.9. TRANSFORMAÇÃO DE DADOS EM INFORMAÇÕES.....	107
II.10. A CONTROLADORIA E ALGUMAS CIÊNCIAS.....	108
IV.1. VISÃO DO DESENVOLVIMENTO DE SOFTWARE E DA GARANTIA E CONTROLE DA QUALIDADE.....	246

APRESENTAÇÃO

O desenvolvimento de software sem especialistas em informática é um fato que tem ocorrido com frequência nos mais diversos tipos de entidades e a tendência é que esse fenômeno ocorra cada vez mais.

Isto nos parece ser algo muito bom, pois ao nosso ver essa forma de elaborar softwares pode ser um fator de aumento de produtividade administrativa, com excelente relação custos-benefícios, desde que seja devidamente controlada.

Nesta dissertação abordaremos alguns aspectos do controle econômico de processo de desenvolvimento de programas de computador por pessoas não especializadas em informática.

A abordagem será feita pelo ângulo da controladoria, ou seja, descreveremos os parâmetros necessários para a realização do controle econômico considerando princípios e premissas inerentes a esse ramo de saber, falaremos de procedimentos considerando a possibilidade das diversas áreas que desenvolvem os seus próprios softwares poderem contar com assessoria do órgão que executa as funções pertinentes à controladoria.

Abordaremos também os aspectos ligados à qualidade dos softwares resultantes do desenvolvimento sem especialistas em informática e efetuiremos propostas para que se trate a qualidade como agente econômico e a contemple no modelo de tomada de decisões relativas ao controle econômico.

ABSTRACT

The development of software without the aid of an expert in the computer field is occurring frequently nowadays, and the trend is to increase even more in the future.

This is a good sign, once by this approach through software production can be an increase in the productivity of the administration as a whole. Once this approach is properly controlled the cost benefit ratio can be optimized.

In this dissertation some aspects of the economic control of computer program development by non-expert in informatics are discussed.

The approach is done under the controllership point of view, that is, the requirements of the economic control are described under the basis and principles of this particular field of specialization. Also, a discussion is done on the procedures which should be adopted by the several areas of the organization which develop their own software with a consultancy by the controllership within the same organization.

Also, the aspects related to the quality of the softwares produced by non - informatics experts are discussed and some proposals are made in order to consider quality as an economic parameter and be incorporated in the decision making model related to economic control.

CAPÍTULO 1 - IDENTIFICAÇÃO DA DISSERTAÇÃO

1.1. INTRODUÇÃO

A indústria mundial de computadores tem evoluído de forma espetacular, colocando anualmente no mercado equipamentos cada vez mais avançados.

Aliados ao progresso tecnológico, os fabricantes têm oferecido preços relativos menores.

A conjugação desses dois fatores tem impulsionado o aumento do uso de recursos computacionais em todo o tipo e tamanho de entidades.

Entretanto, o preparo adequado de profissionais de informática para desenvolvimento de sistemas aplicativos não tem conseguido acompanhar essa velocidade e dinâmica.

No Brasil, essa situação também ocorre, apesar de as novidades a nível de informática serem menores que as dos países desenvolvidos.

A discrepância entre o avanço tecnológico dos equipamentos e o treinamento de pessoal tem causado nos últimos anos muitas dificuldades para a utilização do potencial dos recursos disponíveis.

É sabido que nas grandes empresas o "Backlog" (tempo de espera) para novos sistemas é superior a um ano.

As casas especializadas na elaboração de linguagem de programação têm buscado minimizar esse problema com a produção de ferramentas que apresentem maior facilidade de uso.

Como resultado, há uma participação significativa de pessoas não especializadas em informática no desenvolvimento de programas aplicativos.

Profissionais de diversas áreas têm optado por envolverem-se na elaboração de programas de computador como forma de obter, a curto prazo, ferramentas que agilizem suas atividades de rotina, mesmo que isso implique ter parte do tempo útil despendido com tarefas que não sejam pertinentes às suas funções.

Considerando-se essa situação, parece-nos irreversível o envolvimento cada vez maior de pessoas não especializadas em informática no desenvolvimento e no uso de programas de computador.

São muitas as vantagens encontradas nesse envolvimento, mas alguns problemas e prejuízos têm sido observados quando os programas de computador não são desenvolvidos por especialistas.

Admitindo-se, portanto, que o atual estágio da informática dirige-se para o usuário, participando totalmente do processo de desenvolvimento e do uso de programas para computador, e que essa participação ainda carece de discussão, orientação e controle mais refinado, julgamos indispensável a aceitação desses fatos como uma realidade que deve receber tratamento gerencial específico para convergir com os interesses institucionais, garantindo-lhes sua eficiência e eficácia.

É buscando motivar reflexões sobre essa necessidade de tratamento específico que abordaremos nesta dissertação as características que julgamos adequadas para o controle econômico desse tipo de desenvolvimento de software.

1.2. OBJETIVO, OBJETO DE ESTUDO E ESTRUTURAÇÃO DA DISSERTAÇÃO

O Objetivo Principal desta dissertação é abordar conceitos, princípios e premissas que possam ser considerados adequados para controle econômico de desenvolvimento de softwares sem especialistas em informática, o qual deve abranger aspectos inerentes à qualidade.

A abordagem é feita utilizando-se conceitos, princípios e premissas inerentes à controladoria, por isto, inclusive tentamos dizer no título que está se dando "uma visão da controladoria" sobre o controle de desenvolvimento de software sem especialistas.

Acreditamos que o controle pressupõe o conhecimento completo da situação, por isto, para atingir o objetivo principal analisaremos com detalhe às possíveis características de um software, e posteriormente abordaremos aspectos relacionados ao controle econômico do seu desenvolvimento.

Porém, antes de tudo isto exibiremos os conceitos nos quais estaremos nos apoiando no discorrer da dissertação.

O objeto de estudo será o software, ou seja, identificaremos com detalhe as possíveis características que o produto (software) pode ter para posteriormente abordar aspectos

relativos ao controle de processo de produção (desenvolvimento de software).

Não discutiremos sobre o desenvolvimento em si.

As reflexões do nosso trabalho admitem que existe, uma meta a ser atingida, no caso a construção de um programa (cuja possíveis características estudaremos), o caminho para atingir a meta já foi estabelecido, ou seja, já se escolheu a metodologia de desenvolvimento de software, e o que precisa ser discutido é como proceder (como controlar) para que o percurso pelo caminho (metodologia) escolhido seja adequadamente monitorado de forma a garantir que o limite dos sacrifícios (custos) estimados para trilhá-lo está sendo respeitado e que os benefícios futuros (receitas) que dependem desse trilhar serão alcançados (Vide Figura I.1).

A estruturação lógica da dissertação procura respeitar a linha de raciocínio exposta acima, apresentando aqui no primeiro capítulo a identificação do que se pretende abordar na dissertação como um todo e mostrando que o assunto em breve poderá ser de interesse de muitas entidades.

No segundo capítulo são colocados os conceitos básicos, adotados em todo o trabalho, relativos a controle, controladoria, qualidade e desenvolvimento de software.

O terceiro capítulo é destinado ao estudo das possíveis características de um software desenvolvido sem especialistas em informática, falamos sobre os fatos que motivaram sua existência, sobre o local e as condições em que pode ser desenvolvido, sobre a frequência de sua utilização e sobre os atributos necessários para que tenha qualidade adequada.

Sobre a qualidade discorreremos várias folhas, sendo que a maior parte desse item tenta mostrar em forma de tabela os possíveis reflexos da ausência de um atributo considerado necessário para que o software tenha qualidade adequada.

Entendemos que estas tabelas poderão servir de referência para garantia da qualidade do software (que é um importantíssimo agente econômico) por intermédio do controle econômico.

O capítulo quatro abrange o objetivo do controle, como ele pode ocorrer e como se utiliza dos princípios, premissas e conceitos inerentes à controladoria.

No capítulo cinco buscamos a conclusão sobre o controle do desenvolvimento de software sem especialistas em informática.

Logo após a conclusão encontra-se um glossário elaborado com intuito de tornar o mais transparente possível nossa forma de entendimento do assunto tratado nesta dissertação.

1.3. A QUEM O TEMA PODERÁ INTERESSAR

Com o intuito de verificar a quem o assunto desenvolvimento de software sem especialistas, poderia interessar, contatamos nove empresas, de quatro ramos diferentes, com representatividade no segmento em que atuam, para saber se esse fato ocorria em seus ambientes e se o tema de alguma forma suscitava desejo de reflexão mais profunda.

As empresas às quais nos referimos se encontram, segundo a revista "Exame", de agosto de 1990 (1), entre as quinze maiores dos seus respectivos segmentos, considerando-se faturamento ou volume de depósitos em caso de bancos.

Verificamos que todas essas empresas têm adotado e incentivado nos últimos três anos o desenvolvimento de software sem a participação de especialistas em informática.

Todas afirmaram que nos últimos anos têm melhorado o processo de tomada de decisões em função da implementação de recursos computacionais e só conseguiram maior eficiência na geração de informações quando os usuários passaram a desenvolver o seu próprio software.

Todas empresas afirmaram que apesar do sucesso e agilização que obtiveram com essa forma de se elaborar software, sentem que podem ser mais eficientes e estão seguras que em muitas

das ocasiões em que tomaram decisões equivocadas de alguma forma poderiam ter evitado a equívoco se pudessem contar com softwares de melhor qualidade.

Apesar dos benefícios (os quais não conseguiram mensurar) que têm sido gerados por softwares desenvolvidos por pessoas não especializadas em informática, são obrigadas a reconhecer que o processo de desenvolvimento de software tem acontecido sem controle econômico e isto tem permitido em certas ocasiões a existência de prejuízos (também não mensurados), por erro de software e pelo fato de alguns profissionais se entusiasmarem em demasia com o desenvolvimento, colocando em segundo plano as atividades para quais efetivamente foram contratados.

Desta forma acreditam que precisam refletir sobre o controle do processo de desenvolvimento de software sem especialistas antes que observem prejuízos não suportáveis, os quais poderiam advir principalmente de um excesso de confiança em software elaborado com erro.

Considerando a unanimidade das opiniões e a representatividade dessas entidades, poderíamos correr o risco de afirmar que nos próximos anos será bastante comum o desenvolvimento de software sem especialistas em informática.

Entretanto, para constatar se o desenvolvimento de software sem especialistas é realmente uma tendência generalizada e não apenas prática atual de algumas empresas, decidimos realizar uma verificação que mostrasse a realidade de um universo maior.

Para efetuar essa verificação, optamos por analisar as características da comercialização de computadores e softwares no mercado nacional.

1.3.1. Características da comercialização de computadores

Neste subitem mostraremos a quantidade de equipamentos comercializados nos últimos quatro anos e faremos uma breve análise sobre os números apresentados, buscando interpretar a característica do uso predominante de computadores.

Apresentamos a seguir a quantidade de equipamentos comercializados entre 1986 e 1989, segundo a Associação Brasileira da Indústria de Computadores e Periféricos

(Abicomp).

EQUIPAMENTOS COMERCIALIZADOS

TIPO DE EQUIPAMENTO	1986	%	1987	%	87/86	1988	%	88/87	1989	%	89/88
<hr/>											
16 bits	56.636	40,0	59.239	61,8	4,6	53.927	78,5	8,9	79.538	94,5	47,5
SUPERMICRO	1.240	0,9	1.055	1,0	14,9	1.362	2,0	29,1	2.592	3,1	90,3
MINICOMPUTADOR	1.370	1,0	750	0,8	45,2	403	0,5	46,3	184	0,2	54,3
SUPERMINI	86	0,1	92	0,1	6,9	169	0,2	83,7	105	0,1	37,8
SUBTOTAL	59.332	42,0	61.136	63,7	3,0	55.861	81,4	8,6	82.419	97,9	47,5
8 bits	82.075	58,0	34.705	33,3	57,7	12.753	18,6	63,2	1.784	2,1	86,0
TOTAL	141.407		95.771		32,3	68.614		28,4	84.203		22,7

Podemos observar no quadro de Equipamentos Comercializados que a quantidade de computadores de 8 bits comercializados sofreu uma queda vertiginosa.

Esse tipo de equipamento é destinado normalmente ao uso pessoal ou a processamentos setoriais de empresas. A baixa velocidade de processamento e a restrita capacidade de armazenar informações dificultam a sua utilização para processamentos corporativos até mesmo em entidades de pequeno porte.

A queda no nível de comercialização dos equipamentos de 8 bits é reflexo da sua obsolescência, que é expressa fundamentalmente pelas limitações de processamento e armazenamento de informações.

Como os equipamentos de 8 bits estão obsoletos e com tendência a desaparecerem os excluiremos de análises nesta dissertação.

O quadro Equipamentos Comercializados mostra também que, excluindo-se os computadores de 8 bits, a quantidade total comercializada em 1989 é 38,9% maior que a quantidade total comercializada em 1986, o que nos permite afirmar que cresceu o nível de comercialização entre 1986 e 1989.

Observamos que um índice muito semelhante de crescimento é encontrado na relação entre equipamentos de 16 bits comercializados em 1986 e 1989. Se verificarmos com mais cautela constataremos que o perfil da curva de comercialização total de equipamentos, exclusive os equipamentos de 8 bits, é determinada pelas quantidades dos computadores de 16 bits.

Os computadores de 16 bits também são destinados fundamentalmente a processamentos setoriais e ao uso pessoal, isso porque embora a capacidade de armazenamento de informações (em memória principal e secundária) seja bem

superior à dos equipamentos de 8 bits ainda representam restrições para processamentos corporativos de entidades de médio e grande porte.

A restrição para processamentos corporativos aumenta na medida em que esses equipamentos dificultam o multiprocessamento e a participação de multiusuários simultaneamente.

Dessa forma, ao serem analisadas as curvas de comercialização de equipamentos, concluímos que a tendência do mercado nacional tem sido de aquisição de computadores para processamentos setoriais.

Constatamos que os processamentos corporativos se dão em máquinas classificadas entre supermicros, cuja arquitetura de microprocessadores permite multiusuários, e Mainframes, que possuem elevadíssima capacidade de armazenamento de dados e possibilidade de multiusuários e multitarefas.

Nos dados sobre comercialização não aparecem os Mainframes principalmente porque nesse caso a quantidade de equipamentos comercializados não espelha analogia direta com a quantidade média de possíveis usuários, uma vez que nessas máquinas podem estar, em um mesmo instante, dezenas ou centenas de usuários.

Assim sendo, optamos por considerar a comercialização de terminais de vídeo como forma de análise do uso de Mainframes, pois a quantidade máxima de usuários num determinado instante pode ser expressa, de forma aproximada, pela quantidade de terminais ligados ao Mainframe.

A quantidade de terminais de vídeo comercializados, em 1989, pelas empresas brasileiras, às quais respondem quase à totalidade da demanda de mercado, foi de 52.023.

Nesse mesmo período a quantidade comercializada de microcomputadores de 16 bits foi de 79.538, o que significa dizer que o número de novos usuários de equipamentos de 16 bits foi 53% maior que o número de novos usuários de Mainframe.

É importante ressaltar que na realidade essa diferença deve ser ainda mais favorável aos microcomputadores de 16 bits, uma vez que parte dos terminais é destinada a supermicros e minicomputadores.

Mesmo considerando os Mainframes, podemos continuar afirmando que a tendência de mercado tem sido mais favorável à opção por máquinas de 16 bits, que são destinadas basicamente a processamentos setoriais.

1.3.2. Características da comercialização de software

Procuramos verificar aqui como tem sido a comercialização de softwares com o objetivo de esboçar o perfil da forma de uso dos computadores.

Para realização dessa verificação, utilizamos os dados divulgados pela Secretaria Especial de Informática (SEI) na publicação "Panorama do Setor de Informática" - Série Estatística V.2 Nº 1 Agosto de 1989 (2).

Segundo essa publicação, os softwares foram divididos em dezessete tipos distintos, conforme identificação abaixo:

- . Aplicativos
- . Sistema operacional
- . Gerenciador de informações
- . Desenvolvimento de sistemas
- . Ferramenta de apoio
- . Linguagem
- . Automação
- . Simulação e modelagem
- . Comunicação de dados
- . Teleinformática
- . Aplicação técnico científica
- . Segurança e proteção de dados
- . Inteligência artificial
- . Avaliação de desempenho de sistemas

- Instrumentação
- Entretenimento

Para esta dissertação o que efetivamente interessa é a análise do crescimento da comercialização de ferramentas que são utilizadas pelo usuário final e, em segundo plano, a curva de crescimento da comercialização de softwares aplicativos e dos softwares que são de uso exclusivo de profissionais de informática.

A partir desses dados poderemos verificar se está aumentando o interesse de mercado por ferramentas para usuário final e se esse crescimento acontece em detrimento da comercialização de produtos prontos e de softwares para desenvolvimento de sistemas por profissionais de informática.

Dessa forma, serão analisados exclusivamente os números de comercialização de:

- Aplicativos

Programas escritos para atender as necessidades específicas de uma determinada área.

- Linguagens

Programas especiais que transformam em linguagem de computador (representação binária) outros programas escritos de forma facilmente inteligível pelo ser humano.

- Ferramentas para o desenvolvimento de sistemas

Programas que têm como objetivo assessorar o desenvolvimento de programas aplicativos.

- Inteligência artificial

Programas que oferecem ao computador uma pseudo- capacidade de resolução de problemas, sem a intervenção do homem, mediante um conjunto específico de possibilidades previamente informadas.

Costuma-se dizer que esses sistemas possuem capacidade de aprendizagem porque elaboram hipóteses e teorias em função de dados básicos.

- Gerenciadores de informações

Softwares que viabilizam a elaboração de programas para um fácil e rápido armazenamento e recuperação de dados. A facilidade com que se pode utilizar os gerenciadores de informações faz com que eles sirvam para o desenvolvimento de aplicações básicas sem a participação de especialistas em informática.

- Ferramentas de apoio

Softwares para suporte a profissionais, gerentes e executivos, na realização de seus trabalhos e em especial no apoio à tomada de decisões. Nesta categoria encontram-se as planilhas eletrônicas, sistemas de consultas on-line, programas para geração de relatórios, geradores de gráficos. É por intermédio dessas ferramentas e gerenciadores de base de dados que o usuário final tem desenvolvido seus próprios sistemas.

Buscando ainda maior objetividade de análise, esses softwares serão agrupados em apenas duas categorias:

i) Software para processamento de dados

São aqueles desenvolvidos ou utilizados predominantemente por profissionais de informática.

São os softwares elaborados, ou que servem para elaborar programas com funções predefinidas que estão diretamente vinculados à estrutura de dados da entidade e normalmente não podem ser alterados com facilidade, exigindo a intervenção dos seus desenvolvedores para eventuais modificações nas características de dados ou de processos.

Ou seja, são os aplicativos e aqueles softwares utilizados para o seu desenvolvimento.

Assim sendo, nesta categoria ficam os aplicativos, as linguagens e as ferramentas para o desenvolvimento de softwares.

ii) Software para usuário final

São aqueles destinados a dar apoio a executivos e profissionais das mais diversas áreas na realização de seus trabalhos e na tomada de decisões.

Estão nesta categoria as planilhas eletrônicas, geradores de relatórios, softwares para consulta on-line, linguagens de fácil uso para manipulação de dados e outros similares.

Serão analisados então, como softwares para usuário final os dados referentes a gerenciadores de informações e ferramentas de apoio.

Apresentamos a seguir os valores da comercialização de softwares nos anos de 1986, 1987 e 1988.

SOFTWARES COMERCIALIZADOS

Tipo Software	1986	1987	1988
PROC.DE DADOS	62,115	64,308	62,333
USUÁRIO FINAL	15,204	14,953	15,370

VALORES EM MILHÕES DE DÓLARES

Observando-se os dados exibidos acima constatamos que a comercialização de softwares para usuário final em 1988 é 1,09% maior que o volume de 1986.

Verificamos também que os softwares para processamento de dados têm em 1988 um volume de comercialização 0,35% maior que 1986.

Entre 1986 e 1987 os softwares para usuário final apresentaram um decréscimo no volume comercializado da ordem de 1,65%.

Em 1988 os softwares para usuário final reverteram a queda observada no ano anterior e apresentaram uma tendência de crescimento com volume de comercialização 2,79% maior que 1987.

Apesar da queda observada entre 1986 e 1987, podemos afirmar que a comercialização de softwares para usuário final apresenta uma clara tendência de crescimento, visto que após sua queda voltou a crescer com fôlego, superando a marca que possuía antes desse fato.

Com base nas realizações desses três anos, podemos afirmar com segurança, que o mercado mostra que tem crescido a quantidade de usuários finais que adquirem software para desenvolvimento dos seus próprios programas.

1.3.3. Cópias ilegais de softwares para usuário final

A Associação Brasileira de Empresas de Software (ABES) afirma que a pirataria (cópia e uso de programas sem pagamento dos mesmos) de softwares para usuário final é uma realidade nacional de prática generalizada.

Algumas empresas que comercializam softwares possuem dados que comprovam que em 1988 existia no país para cada cópia legal dez cópias ilegais de softwares para usuário final.

Em 1989 essa relação diminuiu para sete cópias devido a uma agressiva campanha efetuada pelas empresas de softwares que envolveu a Polícia Federal, resultando na busca e apreensão de cópias ilegais em diversas empresas.

Com base nessas informações, podemos afirmar que o uso de softwares para usuário final é maior do que expressa o quadro de comercialização de softwares.

Se as cópias ilegais tivessem sido devidamente comercializadas em 1988, o valor dessas transações seria de 153,7 milhões de dólares, o que significaria dizer que o montante das comercializações de softwares para usuário final seria 2,5 vezes maior que o montante das comercializações de softwares para processamento de dados.

1.3.4. Conclusões sobre a relevância do assunto

Podemos constatar que a comercialização de equipamentos que normalmente são utilizados pelos usuários finais apresenta um nível significativo de crescimento.

Verificamos também que há uma tendência para os próximos anos de continuidade de aumento dos níveis de comercialização de softwares para usuário final.

Juntando-se, portanto, a realidade da comercialização de hardware e de software observada no mercado, podemos afirmar que há uma tendência de crescimento no desenvolvimento de software pelo usuário final.

Com a confirmação dessa tendência, podemos concluir que o assunto desenvolvimento de software sem especialistas em informática em breve poderá ser de interesse de um número muito grande de empresas e entidades dos mais variados segmentos de mercado.

FIGURAS DO CAPÍTULO 1

REPRESENTACAO DA ABRANGENCIA DA DISSERTACAO

FIGURA I-1

CITAÇÕES DO CAPÍTULO 1

CITAÇÕES DO CAPÍTULO 1

- (1) ABRIL Revista Exame - Edição agosto/1990
 Editora Abril - SP- 1990

- (2) SEI Panorama do Setor de Informática
 Séries Estatísticas V. 2 nº. 1 agosto/1989
 Secretaria Especial da Ciência e Tecnologia
 Secretaria Especial de Informática

**CAPÍTULO 2 - IDENTIFICAÇÃO DOS ASSUNTOS TRATADOS NESTA
DISSERTAÇÃO**

2.1. CONCEITOS SOBRE DESENVOLVIMENTO DE SOFTWARE

O assunto tratado nesta dissertação é o controle do desenvolvimento de software sem especialistas em informática.

Entendemos por desenvolvimento de software o conjunto de atividades compreendidas entre o momento em que se identifica a necessidade do uso do computador e o momento em que se consegue produzir o(s) programa(s) (conjunto de instruções logicamente ordenadas e codificadas em linguagem que pode ser interpretada pelo computador) por intermédio dos quais se fará com que o computador gere os resultados esperados (Vide Figura II.1).

Na literatura específica de análise e desenvolvimento de sistemas encontram-se diversas opções e descrições detalhadas de ciclo de desenvolvimento de software, por exemplo Yourdon (1) cita o "Modelo em Cascata de Desenvolvimento de Sistemas" (Vide Figura II.2) e "O ciclo de Vida de Desenvolvimento Estruturado" (Vide figura II.3) e Dias (2) cita um ciclo de vida de sistemas, que atualmente é referenciado pelos profissionais de processamento de dados como "Ciclo Tradicional", (Vide figura II.4).

Nesta dissertação adotaremos uma visão macro do ciclo de desenvolvimento de sistema, isto porque não discutiremos

aqui o processo de desenvolvimento de software mas sim o seu controle. O que nos permite abandonar detalhes técnicos de análise, projeto e implantação de sistemas.

Desta forma identificamos apenas três fases significativas no processo de desenvolvimento de software:

i) Concepção

Fase em que se define e se especifica exatamente O QUE se precisa e o que os programas deverão fazer;

ii) Projeto

Etapa na qual se estabelece COMO os programas devem ser feitos para que realizem o que é esperado; e

iii) Construção

Momento em que se codifica, em linguagem que pode ser interpretada pelo computador, as instruções que executarão as funções estabelecidas no projeto.

2.2. CONCEITOS E CLASSIFICAÇÃO DE CONTROLE

2.2.1. Conceituação

A gestão de negócios exige tomada de decisões cujos resultados precisam ser registrados e confrontados com os objetivos esperados.

Essa comparação entre resultados e expectativa visa a avaliação do nível de eficiência e eficácia de tais decisões, para subsidiar as ações de redirecionamento sempre que constatados desvios em relação a padrões previamente estabelecidos.

Ao conjunto de procedimentos compreendidos entre a tomada de decisões e as eventuais ações redirecionadoras que visam aproximar os fatos das expectativas chama-se controle.

Desta forma, nesta dissertação quando citarmos controle estaremos nos referindo ao conjunto de procedimentos constituído por:

- tomada de decisões;
- registro dos eventos resultantes da tomada de decisões;
- confrontação dos eventos resultantes com padrões previamente estabelecidos; e

- ações de redirecionamento de desvios

Apresentaremos na Figura II.5 a representação gráfica desse nosso entendimento e na Figura II.6 a compreensão de Ribeiro (3) onde encontramos afinidades.

Acreditamos que nossa conceituação também está coerente com Nakagawa (4) quando ele diz que "a fase de controle corresponde à implementação da decisão escolhida e o uso do feedback para que as metas sejam alcançadas".

2.2.2. Objetivo do controle

O controle visa objetivos futuros utilizando-se de informações do passado e busca fazer com que a realidade aconteça conforme planos estabelecidos previamente.

Essa preocupação com o futuro é embasada no desejo de que a entidade continue existindo futuramente.

Assim sendo, acreditamos que o objetivo maior de uma atividade de controle é garantir a continuidade da entidade.

Antony (5) define controle gerencial como "o processo de assegurar que os recursos sejam obtidos e aplicados efetiva e eficientemente na realização dos objetivos de uma organização"

Considerando esta definição podemos ampliar o nosso entendimento dizendo que o objetivo maior do controle é garantir a continuidade da entidade por meio de procedimentos que assegurem que os recursos sejam obtidos e aplicados efetiva e eficientemente na realização dos objetivos de uma organização.

2.2.3. Objetos do controle

Os objetos do controle são pessoas, departamentos, centros de resultados, áreas de responsabilidade ou quaisquer unidades administrativas às quais se delega autoridade para a busca de resultados.

2.2.4. Classificação do controle

O controle pode ser classificado em função:

- do tipo de gestão;
- do tempo de ocorrência do evento;
- da delegação da responsabilidade; e
- da abrangência.

2.2.4.1. Quanto ao tipo de gestão

Entendemos que qualquer atividade produtiva encontra-se sob três tipos de gestões (Figura II.7):

- **gestão operacioal**

administração dos recursos necessários para a produção de um bem ou serviço específico;

- **gestão econômica**

administração dos resultados (receitas e custos) provenientes da transformação de recursos em novos produtos ou serviços;

- **gestão financeira**

administração das defasagens entre desembolsos e recebimentos decorrentes do processo de produção.

Considerando essa divisão de gestões também classificamos o controle em:

- operacional;
- econômico; e
- financeiro.

2.2.4.2. Quanto ao tempo de ocorrência do evento

a) Antecedente

Quando a prática do controle se dá no momento em que se prevê o evento, diz-se que está acontecendo um controle antecedente. Isso porque a atividade de controle antecede à ocorrência do evento.

b) Concomitante

O exercício da atividade de controle no momento da ocorrência do evento caracteriza esse controle como concomitante.

c) Subseqüente

Quando se executa os procedimentos de controle de um evento após seu encerramento se está praticando um controle subseqüente.

Esse tipo de controle terá importância para entidade se os seus resultados forem utilizados para a realização de atividades futuras.

Por isso afirmamos que o controle subseqüente de um determinado evento pode ser considerado parte do controle

antecedente de outro de mesma característica e origem.

Algumas pessoas não reconhecem a existência do controle subsequente porque as ações direcionadoras, que fazem parte da atividade de controle, não afetam o evento, uma vez que este já ocorreu.

Nesta dissertação reconhecemos a existência do controle subsequente como forma de administrar as resultantes do evento ocorrido.

2.2.4.3. Quanto à delegação de responsabilidade

Considerando-se a responsabilidade do controlador, pode-se classificar o controle conforme exposto nos subítens abaixo.

a) Controle de centro de custos

Quando um gestor só tem poder para controlar os custos que em sua área incorrem, considera-se o controle como restrito ao centro de custos.

b) Controle de centro de resultados

Quando um gestor tem o seu desempenho avaliado pelos resultados que sua área propicia, pode-se afirmar que ele

tem poder para decidir não apenas sobre os custos que incorrem em sua área, mas também sobre as receitas que aufera.

Nesse caso, o controle é executado sobre resultados e por isso é chamado de controle de centro de resultados.

c) Controle de centro de investimentos

Existem casos em que é dado ao gestor o poder de decidir também sobre investimentos pertinentes à sua área, sendo avaliado pela taxa de retorno que proporciona.

Nesses casos, o controle acontece sobre o centro de investimentos

2.2.4.4. Quanto à abrangência

a) Controle lógico

Chamamos nesta dissertação de controle lógico aquele que exerce as suas atividades sobre eventos ligados a aspectos estratégicos.

b) Controle funcional

Quando os eventos controlados são inerentes a aspectos funcionais, diz-se que o controle é funcional.

2.2.5. Como Acontece o Controle Econômico

Tradicionalmente a contabilidade geral responde pela identificação mensuração, comunicação de informações econômicas, as quais subsidiam o processo de tomada de decisões.

No livro "Contabilidade Introdutória (6) seus autores (equipe de professores da USP) qualificam a contabilidade como metodologia especialmente concebida para captar registrar, acumular, e interpretar fenômenos que afetam às situações patrimoniais e econômicas de qualquer ente".

Sérgio de Indicibus (7) no seu livro "Teoria da Contabilidade" diz no item 1.3 página 27 que " o objetivo da contabilidade é fornecer informação econômica relevante para que cada usuário possa tomar suas decisões e realizar seus julgamentos com segurança".

A contabilidade de custos, que é parte da contabilidade geral, segundo John Dearden (8) tem as seguintes finalidades básicas:

" - prover valores de inventários de produtos em processamento e produtos acabados (essa informação é necessária para determinação de lucros)

- prover dados para o controle de custos e

- prover dados para decisões de receitas (por exemplo, mudança de preço ou abandono de um produto)".

A equipe de professores da USP (6) diz que a "Contabilidade de Custos tem por objetivo a compilação, determinação e análise de dados necessários para uma decisão segura e fundamentada dos problemas do tipo: preço de venda de produtos, produção máxima possível num contexto econômico etc..".

Entendemos, pelas colocações feitas pelo Prof^o Dr. Armando Catelli nas suas aulas do programa de pós-graduação, que quando se fala em Contabilidade de Custos automaticamente está se enfocando gestão econômica.

Em concordância com o entendimento que fizemos das colocações do Prof^o Catelli, da equipe de professores da USP e de John Dearden dizemos que a contabilidade de custos tem como finalidade principal apoiar a tomada de decisões inerentes à gestão econômica, apurando e informando os resultados econômicos das atividades da entidade.

Assim a contabilidade de custos pode ser vista como um sistema de informações (Langefors (9): sistema utilizado para prover informações - incluindo o seu processamento) constituído por subsistemas de identificação, mensuração e relato de informações.

Diversos autores consideram que o controle econômico de uma atividade pode acontecer exclusivamente por intermédio da contabilidade de custos.

John Dearden (10) diz no seu livro "Cost and Budget Analysis" que "quando as técnicas e procedimentos de contabilidade de custos, análise de custos e controle orçamentário são agrupados formam uma base para um sistema inteiro de controle para decisões da administração de curto prazo", esta afirmação é ainda mais ampla uma vez que no seu livro ele usa o termo "administração de curto prazo" de forma a abranger as gestões econômica, financeira e operacional.

Nesta dissertação em função do conceito que adotamos para controle econômico estamos admitindo que os instrumentos que viabilizam o controle são (Vide Figura II.8):

a - Contabilidade de Custos

i) Sistema de informações

.. Sistema de identificação e mensuração de eventos

.. Sistema de relação de informações

ii) Método de custeio

iii) Padrões e orçamentos

iv) Plano de contas

b - Modelo de tomada de decisões

2.2.5.1. Sistema de Identificação e Mensuração

Chamamos de sistema de identificação e mensuração o conjunto de procedimentos que, embasados em princípios e conceitos que estão vinculados ao modelo de gestão, coleta, identifica e atribui valor a dados referentes a eventos econômicos.

Consideramos que eventos econômicos são os acontecimentos que de alguma forma afetam a riqueza da entidade.

Entendemos que o preço de um produto/serviço é a expressão monetária que o seu usuário atribui à sua utilidade, enquanto o valor de um produto/serviço é a expressão monetária do sacrifício dispendido pela entidade para sua obtenção.

Resultado Econômico de um produto/serviço é a diferença entre o seu preço e o seu valor.

Os eventos econômicos pertinentes a um produto/serviço são aqueles que de alguma forma afetam o seu resultado econômico.

A identificação de um evento econômico consiste basicamente em vinculá-lo a algum produto/serviço.

A identificação de eventos econômicos se dá pela comparação de eventos ocorridos com a lista (Plano de Contas) de eventos pertinentes a cada produto/serviços.

É importante ressaltar que eventualmente pode-se obter resultado econômico por intermédio da gestão financeira entretanto não abordaremos esse aspecto nesta dissertação.

A mensuração de um evento econômico é a atribuição de valores aos eventos identificados.

2.2.5.2. Sistema de relato de informações econômicas

Identificamos como sistema de relato de informações o conjunto de procedimentos que transforma dados sobre eventos econômicos em resumo de acontecimentos econômicos. (vide figura II.9).

Como dissemos anteriormente a contabilidade de custos satisfaz plenamente as expectativas funcionais de um sistema de relato de informações e as realiza por intermédio de procedimentos (que chamaremos de subsistema) de acumulação e avaliação de custos e receitas.

a. Subsistema de acumulação

Subsistema de acumulação é o conjunto de procedimentos coordenados que visa a organização e registro de dados (custos e receitas) de eventos, considerando suas características, as características do processo de produção e as necessidades de informações do modelo de gestão da entidade.

É o subsistema de acumulação que transmite informações aos usuários finais da contabilidade de custos.

Existem dois métodos de acumulação de custos e receitas:

i) por ordem de produção

O acúmulo de custos e receitas é feito por produto.

ii) por processo

O acúmulo de custos e receitas é feito por Departamento ou Centro de Custos.

b. Subsistema de avaliação

É um conjunto de procedimentos que tem como finalidade a apuração de custos e receitas a produtos ou serviços gerados.

Ele recebe do subsistema de acumulação, informações sobre custos e receitas acumulados por produto ou por departamento e à partir dessas atribui valores aos produtos ou serviços.

A avaliação pode acontecer de duas formas:

i) a_valores_reais

Os custos efetivamente incorridos são apropriados a produtos/serviços.

ii) a_valores_predeterminados

Receitas e custos são estruturados previamente por intermédio de orçamentos ou padrões.

2.2.5.3. Modelo de tomada de decisões

Modelo de tomada de decisões é o conjunto de princípios e premissas que servem de base para a formação de um raciocínio e opinião e que de certa forma indicam como agir numa determinada situação.

2.2.6. Eficiência e eficácia do Controle

O controle é eficiente quando é guiado por princípios que asseguram seu bom desempenho sem interferências e influências negativas no evento controlado.

É eficaz principalmente quando:

- possibilita a projeção de tendências;
- viabiliza o redirecionamento do curso de ações de forma que essas contribuam com as metas da entidade;
- oferece subsídios para a garantia da continuidade da entidade.

2.2.7. Limitações do controle

O controle pode limitar-se por aplicações inadequadas de seus princípios, pela utilização de recursos impróprios e, principalmente, pela inexistência de sistemas de informações e mensuração compatíveis com as necessidades de controle.

Sem um sistema de informações e de mensuração, não há como efetuar a base de comparação entre eventos esperados e ocorridos, não há como elaborar projeções. Sem esses sistemas o controle não se materializa.

2.3. CONCEITOS DE CONTROLADORIA

2.3.1. Ramo do saber

Dizem Heckert e Wilson (11): "Ao controller não compete o comando do navio pois essa é a tarefa do primeiro executivo, representa entretanto o navegador que cuida dos mapas de navegação. É sua tarefa manter informado o comandante quanto à distância percorrida, ao local em que se encontra e a velocidade da embarcação, à resistência encontrada, os desvios das rotas, aos recifes perigosos e aos caminhos traçados nos mapas para que o navio chegue ao destino".

Kanitz (12) diz que "a contabilidade evoluiu para um sistema chamado controladoria cuja a função é avaliar e controlar o desempenho das diversas divisões da empresa. Ela faz parte portanto do campo mais genérico dos sistemas de Controle Gerencial".

Entendemos que controladoria enquanto ramo do saber pode ser conceituada como um conjunto organizado de conhecimentos que possibilita o exercício do controle de uma entidade, a identificação de suas metas e dos caminhos econômicos a serem seguidos para atingí-los.

A história mostra que a ciência é precedida por ações meramente instintivas que levam à absorção de conhecimentos

que promovem o surgimento da técnica, que é seleção pragmática de procedimentos adequados para a consecução de determinados objetivos. A técnica, por sua vez, inspira o surgimento da ciência, que vem ordenar os conhecimentos embutidos nos procedimentos da técnica, agrupando-os de forma a explicar e orientar as suas realizações de maneira mais adequada, ou seja, a ciência busca a explicação dos procedimentos ou relações entre seres, ao passo que a técnica é um processo que busca adaptar o meio às necessidades.

O acúmulo de conhecimentos sobre o controle de eventos econômicos determinou o surgimento da controladoria como forma de ordenar esses conhecimentos fundamentando-se em teorias de várias ciências. (vide figura II.10).

A controladoria depende de observações práticas para sedimentar os conhecimentos como verdade.

2.3.2. órgão administrativo

A controladoria, enquanto ramo do saber, pode ser praticada por qualquer entidade sem que se tenha um órgão formalmente constituído com a obrigação específica de exercitar tais conhecimentos. A realidade nacional mostra que é muito grande o número de entidades que operam sem a existência de um órgão específico para execução do controle.

Esse órgão, quando formalmente constituído, tem como missão gerar subsídios para a canalização de esforços e potencialidades da entidade como um todo e participar do processo de tomada de decisões assessorando a alta administração e as diversas áreas de forma a garantir a continuidade da entidade e assegurar a manutenção de seus propósitos.

Cabe à controladoria identificar as oportunidades e ameaças existentes no meio em que a empresa atua, bem como manter adequado o sistema de informações, suprindo a alta gestão de dados suficientes para tomada de decisões de forma ágil e segura. Também é preocupação e função da controladoria a análise da influência do comportamento humano na realização de objetivos da entidades, bem como interpretar os eventos e agir sobre os fatores que podem impactar a saúde econômica da entidade positiva ou negativamente.

Acreditamos que essa nossa visão não contraria as exposições de Heckert e Wilson (11) onde atribuem ao controller, conseqüentemente à controladoria "a responsabilidade pela medição de possibilidades de sua empresa perante as realidades externas, para fixar objetivos, estabelecer políticas básicas, elaborar organogramas com responsabilidades definidas para cada cargo dentro da organização, estabelecer padrões de controle, desenvolver

métodos de comunicação e manter sistema adequado de relatórios”.

2.3.3. Pré-requisitos para o exercício da controladoria

Para que a controladoria passe a ocorrer, seja por intermédio de um órgão específico ou não, é fundamental que exista uma sistemática de controle, uma atividade de gestão e principalmente o desejo de continuidade da entidade.

Todas as afirmações efetuadas nesta dissertação sobre controle são válidas exclusivamente para as entidades que satisfazem essas condições fundamentais.

2.4. CONCEITO DE QUALIDADE DE PROGRAMAS

2.4.1. Conceito

Entendemos por qualidade as propriedades ou atributos de uma pessoa, objeto ou serviço capaz de lhes determinar a natureza.

Assim sendo, pode se dizer que qualidade de programa é o conjunto de seus atributos ou ainda o conjunto de suas características.

Segundo Rocha (13) "qualidade de software é um conjunto de propriedades a serem satisfeitas em determinado grau de modo que o software satisfaça as necessidades de seus usuários".

Entendemos que programa de qualidade adequada é aquele que tem um conjunto de atributos que correspondem às exigências e necessidades de seus usuários e as satisfaz com eficiência e segurança.

Rocha (op.cit.) vincula a qualidade do software, enquanto produto final, à qualidade de especificação e à qualidade de projeto.

A nosso ver, essa vinculação é bastante pertinente porque a qualidade final de um produto é resultado de todos esforços

e sacrifícios ocorridos desde sua concepção até o término de sua elaboração.

Nesta dissertação utilizaremos os objetivos de qualidade (utilizabilidade, confiabilidade conceitual e confiabilidade de representação) proposto por Ana Rocha (op. cit.) como referencial de todas as colocações que fizermos sobre qualidade de especificação, de projeto e de programas, ficando entretanto por conta de nossa interpretação as descrições efetuadas sobre os respectivos assuntos.

2.4.2. Qualidade de especificação

Entende-se por especificação o conjunto de representações que orientam, explicam e determinam como os programas devem ser com descrição detalhada das características e propriedades do programa.

Para que se obtenha qualidade adequada de especificação é necessário que sejam atingidos os seguintes objetivos:

Utilizabilidade;

Confiabilidade conceitual; e

Confiabilidade de representação.

2.4.2.1. Utilizabilidade

Não se pode dizer que uma especificação tem qualidade adequada se não for utilizada. Assim sendo, a facilidade e a viabilidade do uso de uma especificação são requisitos indispensáveis para que se identifique a qualidade da especificação como adequada.

A utilização da especificação deve acontecer na elaboração do projeto do programa, onde serve de guia, e também em eventuais alterações do programa após sua implementação.

Assim sendo, para que a especificação de um programa alcance o objetivo utilizabilidade, deve possuir as seguintes características:

- facilidade de modificação;
- viabilidade econômica e financeira;
- viabilidade técnica;
- viabilidade dentro das possibilidades de recursos humanos e materiais de entidade; e
- viabilidade dentro do tempo exigido

2.4.2.2. Confiabilidade de representação de especificação

Para que a especificação de um programa tenha confiabilidade de representação, deve ser possível a qualquer pessoa acessá-la e entendê-la.

Desta forma, a especificação de um programa satisfaz adequadamente a confiabilidade da representação se fundamentalmente:

- estiver corretamente escrita respeitando as regras gramaticais da língua portuguesa;
- for concisa;
- a terminologia utilizada for uniforme;
- for modular, isto é, permita que se explore apenas determinados aspectos, sem que seja necessário a abordagem de todos aspectos relacionados ao assunto;
- possuir um encadeamento lógico de idéias.

2.4.2.3. Confiabilidade conceitual de especificação

A confiabilidade conceitual pode ser considerada como propriedade mais importante de uma especificação, sua ausência pode invalidar totalmente a especificação.

Para atingir a confiabilidade conceitual, a especificação deve possuir como atributos fidedignidade e suficiência.

Entende-se por fidedignidade a certeza de que a especificação mantém coerência absoluta com os requisitos do usuário, sem contradições e sem interpretações ambíguas.

Suficiência é o atributo da especificação que garante que a resposta aos requisitos do usuário será precisa e absolutamente explícita.

2.4.3. Qualidade de projeto

Entende-se por projeto de software o conjunto de algoritmos que detalham as funções e procedimentos necessários para que o programa satisfaça as exigências dos usuários.

A elaboração do projeto depende fundamentalmente da especificação dos programas.

Para que um projeto seja tido como de boa qualidade, também deve alcançar os objetivos:

Utilizabilidade;

Confiabilidade conceitual; e

Confiabilidade de representação.

Valem aqui abordagens semelhantes às que foram efetuadas para qualidade da especificação, como pode ser visto nos subítens a seguir.

2.4.3.1. Utilizabilidade

O projeto para ser de qualidade adequada deve:

- oferecer facilidade para eventuais modificações;
- ser viável econômica, financeira e tecnicamente;
- ser viável dentro dos recursos econômicos da entidade; e
- ser viável dentro do tempo exigido.

2.4.3.2. Confiabilidade de representação de projeto

Para que um projeto apresente como qualidade a confiabilidade de representação, deve possuir os seguintes atributos :

- uso da simbologia e linguagem de forma absolutamente correta;
- mínimas informações possíveis para um perfeito entendimento;
- encadeamento lógico e objetivo de idéias; e

- permitir que se analise e se conheça determinadas partes do projeto saindo-se de uma visão genérica para um determinado detalhe, de forma a não ser necessário o entendimento de tudo o que esteja envolvido com o assunto, mas apenas da parte que se tem interesse em conhecer ou analisar.

2.4.3.3. Confiabilidade conceitual

A confiabilidade conceitual do projeto é demonstrada através da existência de atributos que garantam a exata expressão do que foi descrito na especificação e garantam também condições de assegurar que os resultados extraídos dos programas sejam absolutamente corretos e ofereçam previsões de soluções para eventuais problemas que possam ocorrer devido ao uso inadequado ou ao fornecimento errôneo de dados.

2.4.4. Qualidade de programa

Entende-se por programa o conjunto de instruções ordenadas logicamente para transformar dados em informações que satisfaçam as necessidades predeterminadas dos usuários.

Um programa de qualidade adequada incorpora em suas propriedades a utilizabilidade, a confiabilidade conceitual e a confiabilidade de representação.

2.4.4.1. Utilizabilidade de programa

O primeiro e mais elementar atributo de um programa de qualidade adequada é a facilidade e a viabilidade de seu uso, que estão vinculadas às disponibilidades de recursos humanos e materiais da entidade.

Assim sendo, a qualidade adequada de um programa também pode ser impactada por aspectos ligados aos custos.

Desta forma, pode-se dizer que a propriedade utilizabilidade é atingida por um programa quando:

- é fácil operá-lo;
- não exige muitos recursos humanos e materiais para seu uso;
- é fácil efetuar pequenas modificações em seus roteiros e processos;
- oferece resultados em tempo adequado;
- consome o mínimo necessário de recursos de máquinas;
- é elaborado com o mínimo possível de peculiaridades computacionais, principalmente aquelas que inviabilizam seu uso em máquinas distintas;
- os benefícios de seu uso compensam o sacrifício econômico.

2.4.4.2. Confiabilidade de representação de programa

Para que um programa tenha confiabilidade de representação é indispensável que seja inteligível na sua constituição e principalmente na sua utilização.

Para que isso aconteça é necessário que ele:

- seja codificado com a maior clareza e simplicidade possível;
- esteja modularizado de acordo com a similaridade de funções, de forma que um módulo dependa o menos possível de outros; e
- esteja estruturado hierarquicamente, viabilizando o acesso independente e direto a cada um dos módulos que o constituem, mas deixando claro o elo de ligação e a forma de relacionamento dos módulos.

2.4.4.3. Confiabilidade conceitual

Para que um programa tenha confiabilidade conceitual é necessário que ele possua fidedignidade e integridade, o que significa dizer que deve possuir os seguintes atributos:

- efetuar os cálculos e procedimentos com precisão e presteza exigidas pelos usuários;
- realizar todas as funções requeridas com o nível de detalhes requerido;

- estar preparado para contornar e informar eventuais erros de uso ou de fornecimento de dados, de forma a não comprometer a veracidade dos resultados finais e, na medida do possível, sem interromper o uso;
- estar preparado para evitar equívocos que possam colocar em risco o patrimônio da entidade, a sua reputação ou ainda vidas humanas.

2.4.4.4. Quadros sintéticos dos atributos da qualidade de programas, de especificação e de projeto

Apresentamos a seguir quadros sintéticos que apresentam os atributos necessários para que cada objetivo de qualidade (utilizabilidade, confiabilidade conceitual e confiabilidade de representação) sejam alcançados.

i) Qualidade de programa

Qualidade de programa

	Fácil de operar
U	
T	Não exige muitos recursos humanos e materiais
I	para ser usado
L	
I	Fácil de efetuar pequenas modificações em
Z	seus roteiros e processos
A	
B	Oferece resultados em tempo adequado
I	
L	
I	Consome o mínimo necessário de recursos de
D	máquina
A	
D	Elaborado com o máximo possível de recursos
E	computacionais padronizados
	Uso apresenta boa relação custo-benefício

Qualidade de programa

C O N F I A B I L I D A D E D E R E P R E S E N T A Ç Ã O		Codificado com clareza e simplicidade
		Assuntos separados em módulos independentes
		Estruturado hierarquicamente viabilizando acessos independentes aos módulos
		É fácil de identificar os elos de ligações e forma de relacionamento dos módulos
		Satisfaz com precisão os requisitos do usuário
		Apresenta todas respostas aos requisitos do usuário de forma explícita

Qualidade de programa

!	!	!	!
!	!	!	!
!C	!	! Efetua cálculos com precisão e presteza	!
!O	!	!	!
!N C	!	! exigidas pelo usuário	!
!F O	!	!	!
!I N	!	!	!
!A C	!	!	!
!B E	!	! Realiza todas as funções com o nível de de-	!
!I I	!	!	!
!L T	!	! talhe requerido	!
!I U	!	!	!
!D A	!	!	!
!A L	!	!	!
!D	!	! Preparado para contornar eventuais erros de	!
!E	!	!	!
!	!	! uso ou de fornecimento de dados	!
!	!	!	!

ii) Qualidade de especificação

Qualidade de especificação

U	Fácil de modificar
T	
I	
L	Implementação viável econômica e financeira-
I	mente
Z	
A	Implementação viável tecnicamente
B	
I	
L	Implementação viável dentro dos recursos
I	disponíveis
D	
A	Implementação viável dentro do tempo exigido
D	
E	

Qualidade de especificação

C		
O		
N	Corretamente escrita	
F		
I		
A		
B		
I		
L	Concisa	
I		
D		
A		
D		
E		
	Terminologia uniforme	
R		
E		
P		
R		
E		
S	Assuntos separados em módulos independentes	
E		
N		
T		
A		
C		
Ç	Encadeamento lógico de idéias	
O		

Qualidade de especificação

C	Coerência absoluta com os requisitos do
O	usuário
N	
F	
I	
A	
B	Sem contradições
I	
L	
I	
D	
A	
D	Sem interpretações ambíguas
E	
C	
O	
N	
C	Identifica com precisão os requisitos do
E	usuário
I	
T	
U	
A	
L	
	Apresenta todas as respostas aos requisitos
	de forma explícita

iii) Qualidade de projeto

Qualidade de projeto

U	Fácil de modificar
T	
I	
L	Implementação viável economica e financeira-
I	mente
Z	
A	Implementação viável tecnicamente
B	
I	
L	Implementação viável dentro dos recursos
I	disponíveis
D	
A	Implementação viável dentro do tempo exigido
D	
E	

Qualidade de projeto

C	
O	
N	Uso correto de simbologia e de linguagem
F	
I	
A	
B	
I	
L	
I	Otimização de informações
D	
A	
D	
E	
R	
E	Encadeamento lógico e objetivo de idéias
P	
R	
E	
S	
E	
N	
T	Assuntos separados em módulos
A	
C	
Z	
O	

Qualidade de projeto

+-----+		
!C	!	!
!O	!	!-----!
!N C	!	! Coerência absoluta com as descrições de
!F O	!	! especificações
!I N	!	!-----!
!A C	!	!
!B E	!	!
!I I	!	!
!L T	!	!-----!
!I U	!	! Previsão de soluções para eventuais problemas!
!D A	!	!-----!
!A L	!	! de uso ou de fornecimento de dados
!D	!	!-----!
!E	!	!
+-----+		

2.4.4.5. Relação entre qualidade de especificação e qualidade de programa

Existem várias maneiras de se avaliar a qualidade de um programa. Uma delas, segundo Perry (14), é o levantamento e análise de atributos ausentes.

Esse método consiste em identificar os atributos necessários para que um programa seja considerado de boa qualidade e posteriormente verificar quais desses atributos o programa não possui.

Pretendemos, aqui, identificar os impactos que acreditamos que possam ser causados na qualidade dos programas pela inadequação de especificações e projetos. Para essa identificação, utilizaremos o método do critério ausente, ou seja, admitiremos que um determinado atributo não existe na especificação e analisaremos o possível impacto dessa ausência na qualidade do programa.

Listaremos na primeira coluna de uma tabela os atributos necessários para que uma especificação seja considerada de boa qualidade e na segunda coluna dessa mesma tabela identificaremos alguns atributos do programa que acreditamos que serão afetados pela sua ausência.

Impacto da não qualidade da especificação na qualidade do programa

! ATRIBUTOS DA ESPECIFICAÇÃO ! ! (AE) !	! ATRIBUTOS DO PROGRAMA AFETADOS ! ! PELA AUSÊNCIA DOS AE !
! - Fácil de modificar !	! - Satisfaz com precisão os ! ! requisitos do usuário !
	! - Apresenta todas respostas ! ! aos requisitos do usuário ! ! de forma explícita !
	! - Uso apresenta boa relação ! ! custo benefício !
	! - Fácil de efetuar pequenas ! ! modificações em seus rotei- ! ! ros e processos !

! ATRIBUTOS DA ESPECIFICAÇÃO ! ! (AE) !	! ATRIBUTOS DO PROGRAMA AFETADOS ! ! PELA AUSÊNCIA DOS AE !
! Sua implementação é viável: ! ! - economicamente ! ! - financeiramente ! ! - dentro dos recursos ! ! disponíveis ! ! - dentro do tempo exigido !	! - Fácil de operar ! ! - Não exige muitos recursos ! ! humanos e materiais para ! ! ser usado ! ! - Consome o mínimo necessário ! ! de recursos de máquina ! ! - Uso apresenta boa relação ! ! custo-benefício ! ! - Satisfaz com precisão os ! ! requisitos do usuário ! ! - Apresenta todas as respostas ! ! aos requisitos do usuário de ! ! forma explícita !

ATRIBUTOS DA ESPECIFICAÇÃO (AE)	ATRIBUTOS DO PROGRAMA AFETADOS! PELA AUSÊNCIA DOS AE
Corretamente escrita	<ul style="list-style-type: none"> - Fácil de efetuar pequenas modificações em seus roteiros e processos - Codificado com clareza - Assuntos separadas em módulos independentes - Fácil de identificar os elos de ligação e forma de relacionamento dos módulos - Satisfaz com precisão os requisitos do usuário - Apresenta todas as respostas aos requisitos do usuário de forma explícita - Realiza todas as funções com o nível de detalhe requerido

! ATRIBUTOS DA ESPECIFICAÇÃO !	! ATRIBUTOS DO PROGRAMA AFETADOS !
! (AE) !	! PELA AUSÊNCIA DOS AE !
! Concisa !	! - Fácil de operar !
!	! - Consome o mínimo necessário !
!	! de recursos de máquina !
!	! - Uso apresenta boa relação !
!	! custo-benefício !
!	! - Codificado com clareza e !
!	! simplicidade !
!	! - Assuntos separados em módu- !
!	! los independentes !
!	! - Estruturado hierarquicamente !
!	! viabilizando acessos inde- !
!	! pedentes aos módulos !

! ATRIBUTOS DA ESPECIFICAÇÃO ! (AE)	! ATRIBUTOS DO PROGRAMA AFETADOS! ! PELA AUSÊNCIA DOS AE
! Terminologia uniforme	! - Elaboração com o máximo pos- ! sível de recursos computa- ! cionais padronizados ! - Codificado com clareza e ! simplicidade ! - Assuntos separados em módu- ! los independentes ! - Fácil de identificar os elos ! de ligação e forma de rela- ! cionamento dos módulos

Atributos DA ESPECIFICAÇÃO (AE)	ATRIBUTOS DO PROGRAMA AFETADOS! PELA AUSÊNCIA DOS AE
Assuntos separados em módulos independentes	<ul style="list-style-type: none"> - Fácil de operar - Fácil de efetuar pequenas modificações em seus roteiros e processos - Consome o mínimo necessário de recursos de máquina - Uso apresenta boa relação custo benefício - Codificado com clareza - Assuntos separados em módulos independentes - Estruturado hierarquicamente viabilizando acessos independentes aos módulos - Fácil de identificar os elos ligação e a forma de relacionamento dos módulos - Realiza todas as funções com o nível de detalhe requerido

! ATRIBUTOS DA ESPECIFICAÇÃO ! (AE)	! ATRIBUTOS DO PROGRAMA AFETADOS! ! PELA AUSÊNCIA DOS AE
! Encadeamento lógico de ! idéias	! - Fácil de operar ! - Fácil de efetuar pequenas ! modificações em seus rotei- ! ros e processos ! - Oferece resultados em tempo ! adequado ! - Consome o mínimo necessário ! de recursos de máquina ! - Uso apresenta boa relação ! custo-benefício ! - Codificado com clareza e ! simplicidade ! - Assuntos separados em módu- ! los independentes ! - Estruturado hierarquicamente! ! viabilizando acessos inde- ! pendentes aos módulos ! - Efetua cálculos com precisão! ! e presteza exigidas pelo ! usuário ! - Realiza todas as funções com! ! o nível de detalhe requerido! ! - Preparado para contornar ! eventuais erros de uso ou de ! fornecimento de dados

ATRIBUTOS DA ESPECIFICAÇÃO (AE)	ATRIBUTOS DO PROGRAMA AFETADOS PELA AUSÊNCIA DOS AE
	<ul style="list-style-type: none"> - Satisfaz com precisão os requisitos do usuário - Apresenta todas respostas aos requisitos do usuário de forma explícita - Efetua cálculos com precisão e presteza exigidas pelo usuário - Realiza todas as funções com o nível de detalhe requerido

! ATRIBUTOS DA ESPECIFICAÇÃO ! ! (AE) !	! ATRIBUTOS DO PROGRAMA AFETADOS ! ! PELA AUSÊNCIA DOS AE !
! Sem contradições !	! - Fácil de efetuar pequenas ! ! modificações em seus rotei- ! ! ros e processos ! ! - Codificado com clareza e ! ! simplicidade ! ! - Assuntos separados em módu- ! ! los independentes ! ! - Estruturado hierarquicamente ! ! viabilizando acessos inde- ! ! pendentes aos módulos ! ! - Satisfaz com precisão os ! ! requisitos do usuário ! ! - Apresenta todas respostas ! ! aos requisitos do usuário ! ! de forma explícita ! ! - Efetua cálculos com precisão ! ! e presteza exigidas pelo ! ! usuário ! ! - Realiza todas as funções com ! ! o nível de detalhe requerido !

ATRIBUTOS DA ESPECIFICAÇÃO (AE)	ATRIBUTOS DO PROGRAMA AFETADOS! PELA AUSÊNCIA DOS AE
Sem interpretações ambíguas	<ul style="list-style-type: none"> - Codificado com clareza e simplicidade - Assuntos separados em módulos independentes - Fácil de identificar os elos de ligação e forma de relacionamento dos módulos - Satisfaz com precisão os requisitos do usuário - Apresenta todas respostas aos requisitos do usuário de forma explícita - Efetua cálculos com precisão e presteza exigidas pelo usuário - Realiza todas as funções com o nível de detalhe requerido!

! ATRIBUTOS DA ESPECIFICAÇÃO !	! ATRIBUTOS DO PROGRAMA AFETADOS !
! (AE) !	! PELA AUSÊNCIA DOS AE !
! Identifica com precisão os !	! - Satisfaz com precisão os !
! requisitos dos usuário !	! requisitos do usuário !
! !	! - Apresenta todas respostas !
! !	! aos requisitos do usuário !
! !	! de forma explícita !
! !	! - Efetua cálculos com precisão !
! !	! e presteza exigidas pelo !
! !	! usuário !
! !	! - Realiza todas as funções com !
! !	! o nível de detalhe requerido !

+-----+ ! ATRIBUTOS DA ESPECIFICAÇÃO ! ! (AE) ! +-----+	+-----+ ! ATRIBUTOS DO PROGRAMA AFETADOS ! ! PELA AUSÊNCIA DOS AE ! +-----+
! Apresenta todas as respos- ! ! tas aos requisitos de forma ! ! explícita ! ! ! !	! - Apresenta todas respostas ! ! aos requisitos do usuário ! ! de forma explícita ! ! ! - Realiza todas as funções com ! ! o nível de detalhe requerido ! ! +-----+

2.4.4.6. Relação entre qualidade de projeto e qualidade de programa

Para identificar a influência da não qualidade do projeto na qualidade do programa, adotaremos mesmo procedimento utilizado no subitem anterior, ou seja, colocaremos numa tabela os atributos que um projeto deve ter para ser considerado de boa qualidade e o que causa a ausência desses atributos na qualidade do programa.

! ATRIBUTOS DO PROJETO ! (AP) !	! ATRIBUTOS DO PROGRAMA AFETADOS! ! PELA AUSÊNCIA DOS AP !
! Sua implementação é viável !	! - Fácil de operar !
! - economicamente !	! - Não exige muitos recursos !
! - financeiramente !	! humanos e materiais para ser !
! - dentro dos recursos dis- !	! usado !
! poníveis !	! - Oferece resultados em tempo !
! - dentro do tempo exigido !	! adequado !
!	! - Consome o mínimo necessário !
!	! de recursos de máquina !
!	! - Uso apresenta boa relação !
!	! custo-benefício !
!	! - Satisfaz com precisão os !
!	! requisitos do usuário !
!	! - Apresenta todas respostas !
!	! aos requisitos do usuário !
!	! de forma explícita !
!	! - Realiza todas as funções com !
!	! o nível de detalhe requerido !

! ATRIBUTOS DO PROJETO ! (AP)	! ATRIBUTOS DO PROGRAMA AFETADOS! ! PELA AUSÊNCIA DOS AP
! . Uso correto de simbologia! ! e da língua portuguesa	! - Fácil de efetuar pequenas ! modificações em seus rotei- ! ros e processos ! ! - Consome o mínimo necessário ! de recursos de máquina ! ! - Elaboração com o máximo pos- ! sível de recursos computa- ! cionais padronizados ! ! - Uso apresenta boa relação ! custo-benefício ! ! - Codificado com clareza e ! simplicidade ! ! - Assuntos separados em módu- ! los independentes ! ! - Estruturado hierarquicamente ! viabilizando acessos inde- ! pendentes aos módulos ! ! - Fácil de identificar os elos ! de ligação e forma de rela- ! cionamento dos módulos ! ! - Satisfaz com precisão os ! requisitos do usuário ! ! - Apresenta todas respostas ! aos requisitos do usuário ! de forma explícita

! ATRIBUTOS DO PROJETO ! ! (AP) !	! ATRIBUTOS DO PROGRAMA AFETADOS! ! ! PELA AUSÊNCIA DOS AP !
	! - Efetua cálculos com precisão! ! e presteza exigidas pelo ! usuário ! ! - Realiza todas as funções com! ! o nível de detalhe requerido! ! ! - Preparado para contornar ! eventuais erros de uso ou de! ! fornecimento de dados !

! ATRIBUTOS DO PROJETO !	! ATRIBUTOS DO PROGRAMA AFETADOS !
! (AP) !	! PELA AUSÊNCIA DOS AP !
! Otimização de informações !	! - Fácil de operar !
!	! - Não exige muitos recursos !
!	! humanos e materiais para ser !
!	! usado !
!	! - Fácil de efetuar pequenas !
!	! modificações em seus rotei- !
!	! ros e processos !
!	! - Oferece resultados em tempo !
!	! adequado !
!	! - Consome o mínimo necessário !
!	! de recursos de máquina !
!	! - Uso apresenta boa relação !
!	! custo-benefício !

! ATRIBUTOS DO PROJETO ! (AP)	! ATRIBUTOS DO PROGRAMA AFETADOS! ! PELA AUSÊNCIA DOS AP
! Encadeamento lógico e obje- ! tivo de idéias	! - Fácil de operar ! ! - Fácil de efetuar pequenas ! modificações em seus rotei- ! ros e processos ! ! - Oferece resultados em tempo ! adequado ! ! - Consome o mínimo necessário ! de recursos de máquina ! ! - Uso apresenta boa relação ! custo-benefício ! ! - Codificado com clareza e ! simplicidade ! ! - Assuntos separados em módu- ! los independentes ! ! - Estruturado hierarquicamente ! viabilizando acessos inde- ! pendentes aos módulos ! ! - Fácil de identificar os elos ! de ligação e forma de rela- ! cionamento dos módulos ! ! - Satisfaz com precisão os ! requisitos do usuário

ATRIBUTOS DO PROJETO (AP)	ATRIBUTOS DO PROGRAMA AFETADOS! PELA AUSÊNCIA DOS AP
	<ul style="list-style-type: none"> <li data-bbox="689 357 1265 527">- Apresenta todas respostas aos requisitos do usuário de forma explícita <li data-bbox="689 538 1265 708">- Efetua cálculos com precisão e presteza exigidas pelo usuário <li data-bbox="689 719 1265 836">- Realiza todas as funções com o nível de detalhe requerido <li data-bbox="689 846 1265 1017">- Preparado para contornar eventuais erros de uso ou de fornecimento de dados

ATRIBUTOS DO PROJETO (AP)	ATRIBUTOS DO PROGRAMA AFETADOS! PELA AUSÊNCIA DOS AP
Assuntos separados em módulos	<ul style="list-style-type: none"> - Fácil de efetuar pequenas modificações em seus roteiros e processos - Codificado com clareza e simplicidade - Assuntos separados em módulos independentes - Fácil de identificar os elos de ligação e forma de relacionamento dos módulos - Preparado para contornar eventuais erros de uso ou de fornecimento de dados

FIGURAS DO CAPÍTULO 2

.

USO MACRO DO DESENVOLVIMENTO DE SOFTWARE

FIGURA II-1

MODELO EM CASCATA DE DESENVOLVIMENTO DE SOFTWARE

FIGURA II.2

CICLO DE VIDA DE PROJETO ESTRUTURADO

FIGURA I I . 3

CICLO DE VIDA TRADICIONAL DE SISTEMAS DE PROCESSAMENTO DE DADOS

FIGURA II.4

CONTROLE

FIGURA II.5

CICLO DE REALIZACAO DO CONTROLE

FIGURA II.6

FIGURA II.7

C O N T R O L E

FIGURA II-10

CITAÇÕES DO CAPÍTULO 2

CITAÇÕES DO CAPÍTULO 2

- (1) YOURDON, Edward Análise estruturada moderna
Campus - RJ - 1990

- (2) DIAS, Donaldo de S. Projetos de sistemas de proc.
& Gazzaneo, Giosafatte de dados
Livros técnicos e científicos
- RJ - 1977

- (3) RIBEIRO, Antonio M. Análise projeto de sistemas
Univ. Fed. Minas Gerais - MG -
1978

- (4) NAKAGAWA, Masayuki Aspectos comportamentais da
contabilidade gerencial
(ou a contabilidade comporta-
mental)
FEA/USP - SP - 1977

- (5) ANTONY, Robert N. Contabilidade gerencial: uma
introdução à contabilidade
Ed. Atlas - 1974

- (6) IUDICIBUS, Sergio Contabilidade introdutória
& Equipe Prof. USP Atlas - SP - 1977

- (7) IUDICIBUS, Sergio Teoria da contabilidade
Atlas - SP - 1987

- (8) DEARDEN, John Análise de custos nas empresas
Zahar - 1976

- (9) LANGEFORS, B. Theoretical analysis of infor-
mation systems
Auerbach - Philadelphia - 1973

- (10) DEARDEN, John Cost and budget analysis
Prentice Hall - 1962
- (11) HECKERT, J.B.
 & Wilson, J.D. Controllershship
The Ronald Press co. - N.York
1963
- (12) KANITZ, Stephen C. Controladoria: teoria e estu
dos de casos
Livraria Pioneira - SP - 1976
- (13) ROCHA, Ana Regina C.da Análise e projeto estruturado
de sistemas
Campus - RJ - 1987
- (14) PERRY, W.E. Effective methods of EDP
quality assurance
Prentice Hall - 1983

CAPÍTULO 3- CARACTERIZAÇÃO DE SOFTWARES

3. CARACTERIZAÇÃO DE SOFTWARES

3.1. O QUE É SOFTWARE

Chamamos de software um programa ou conjunto de programas para computador.

Entendemos como programa o conjunto de instruções logicamente ordenadas e codificadas em linguagem que pode ser interpretada pelo computador, que tem como função a transformação de dados em informações.

Dados, segundo Aurélio (1), são elementos ou quantidades que servem de base para formação de um juízo.

Informação também segundo Aurélio (1) é instrução, direção ou notícia (exposição sucinta de uma fato).

Desta forma podemos dizer também que programa é um conjunto de instruções logicamente ordenadas e codificadas em linguagem que pode ser interpretada pelo computador, que tem como objetivo a transformação de elementos ou quantidades inerentes a um determinado assunto ou coisa em instrução, ou direção ou notícia.

Os programas, com base em suas características funcionais podem ser classificados em diversas categorias.

A secretaria Especial de informática (SEI) reconhece por intermédio da publicação "Panorama do Setor de Informática" (2) a existência de dezessete tipos distintos de software no mercado nacional.

Acreditamos que a classificação da SEI é bastante abrangente e pode ser adotada como referencial adequado para a categorização de softwares.

Os dezessete tipos são os que listamos no item 1.3.2. e que voltamos a identificar a seguir:

- . Aplicativos
- . Sistema operacional
- . Gerenciador de informações
- . Desenvolvimento de sistemas
- . Ferramentas de apoio
- . Linguagem
- . Automação
- . Simulação e modelagem
- . Comunicação de dados
- . Teleinformática
- . Aplicação técnico científica
- . Segurança e proteção de dados
- . Inteligência artificial
- . Avaliação e desempenho de sistemas
- . Instrumentação

. Entreterimento

Nesta Dissertação estamos abordando exclusivamente aspectos relacionados a programas aplicativos. Desta forma toda vez que utilizarmos o termo software ou programas estaremos nos referindo exclusivamente a aplicativos.

Para nós, Programa aplicativo é o conjunto de instruções logicamente ordenadas e codificadas em linguagem que pode ser interpretada pelo computador, que tem como função a transformação de elementos ou quantidades relativas a um determinado evento da gestão empresarial em exposição sucinta sobre o que ele apresenta ou sobre os fatos que o cercaram ou que podem ser resultantes de sua ocorrência.

Os programas aplicativos também podem ser classificados de acordo com o tipo de gestão ao qual se relaciona o assunto por ele tratado.

Assim sendo podemos ter programas aplicativos destinados à gestão:

- operacional ou
- econômica ou
- financeira

No que tange à abrangência podem tratar dados estratégicos e/ou operacionais.

Trataremos nesta dissertação dos programas aplicativos pertinentes às gestões econômicas e financeira que podem possuir abrangência estratégica e/ou operacional.

Acreditamos que a totalidade dos enfoques desta dissertação se aplica também a programas ligados à gestão operacional (aqueles que auxiliam o processo de produção) entretanto por rigor técnico os excluimos do escopo deste trabalho.

3.1.1. Fatores que geram a necessidade de programas de computador

Identificamos aqui os principais fatores que motivam o desenvolvimento e o uso de programas de computador.

3.1.1.1. Nível de complexidade do serviço

O grau de dificuldade de determinada tarefa exige, muitas vezes, que o seu responsável procure novas soluções para realizá-la, e na busca de opções, podem surgir os recursos computacionais.

Ou seja, um trabalho pode necessitar do computador pelo fato de sua realização ser composta por muitas etapas ou por exigir níveis de detalhamento significativos.

3.1.1.2. Tempo exigido para a realização do serviço

Muitas vezes, independentemente do nível de complexidade do serviço, é necessário que uma determinada atividade seja feita num tempo que o seu realizador não consegue cumprir em condições normais de desempenho.

É comum, então, os recursos computacionais aparecerem como solução devido à velocidade com que realizam algumas tarefas.

3.1.1.3. Quantidade de dados a serem manipulados

Em algumas ocasiões, a quantidade de dados é um fator que leva ao uso de equipamentos eletrônicos.

Embora possa não haver complexidade na atividade, nem prazos reduzidos, opta-se por tratar os dados eletronicamente para se ter uma maior confiabilidade nos resultados, uma vez que o volume de dados pode provocar equívocos em sua manipulação.

Mesmo que não haja exigência de prazo, é conveniente, para toda e qualquer entidade, a obtenção de resultados das atividades por ela desenvolvidas no menor tempo possível.

Nesses casos, o que motiva a reflexão sobre o uso do computador é o volume de dados, mas a justificativa para a decisão é a confiabilidade e o tempo de resposta.

3.1.1.4. Repetitividade de tarefas

A frequência com que determinadas atividades rotineiras são realizadas justifica a utilização de recursos computacionais, pois estes respeitam sempre uma mesma sequência lógica.

Essas atividades podem possuir baixo nível de complexidade, tempo adequado para realização e poucos dados a serem manipulados, mas, mesmo assim, o uso de computadores pode se justificar pela otimização da mão-de-obra, uma vez que poderá liberar profissionais para atividades mais nobres na entidade.

3.1.1.5. Conjugação de fatores

Em diversas situações decide-se pelo uso de programas de computador pela soma dos fatores motivadores citados acima.

Um profissional de uma determinada área, por exemplo, pode ser motivado ao uso de computador porque a sua atividade se torna complexa em alguns períodos do ano, quando tem que repetí-la várias vezes num curto espaço de tempo.

O que motiva o uso do computador, nesse caso, é uma situação específica que congrega fatores que, se analisados isoladamente, não justificariam essa utilização.

3.1.2. Caracterização de um programa de computador em função da frequência de uso

Julgamos que nas discussões sobre os aspectos econômicos do desenvolvimento de programas de computador a análise da frequência prevista para o uso é tão importante e relevante quanto a identificação dos motivos que balizaram a decisão de elaborá-los.

Assim sendo, exibiremos neste item nossa forma de identificação de um programa de computador em função da sua frequência de uso.

3.1.2.1. Programas Permanentes

Alguns programas são desenvolvidos para serem aplicados em todas as ocasiões em que surgirem serviços relativos a um determinado assunto.

Como esses programas podem participar permanentemente dos serviços rotineiros de uma determinada área os identificamos como Programas Permanentes.

Um programa desenvolvido, por exemplo, para conversão de valores em cruzeiros para valores representados em outras moedas é um Programa Permanente, uma vez que poderá ser utilizado praticamente sem restrições todas as vezes que houver necessidade de uma conversão monetária.

A elaboração de Programas Permanentes geralmente é motivada pelo "nível de complexidade", ou pela "conjugação de fatores", ou pela "quantidade de dados a serem manipulados", ou ainda pela "repetitividade de tarefas"

3.1.2.2. Programas Transitórios

Há programas de computador cuja aplicação ocorre exclusivamente num único trabalho pelo fato de terem sido concebidos com particularidades não observadas em outros serviços.

Programas com essa característica chamamos de Transitórios.

Um programa desenvolvido para identificar as repercussões financeiras do "Plano Brasil Novo" no estoque existente no dia 16.03.90 é um Programa Transitório, pois é remota a possibilidade de voltar a ser utilizado após a obtenção de seus resultados.

Mesmo a edição de um plano econômico semelhante dificilmente justificaria o seu aproveitamento, principalmente porque uma característica marcante de um Programa Transitório é conter na sua escrita conceitos específicos, ou seja, geralmente os comandos utilizados nesses programas apresentam-se como constantes e não como variáveis, e o esforço para eventuais alterações se aproxima muito de uma nova elaboração, o que acaba acontecendo na prática.

A grande maioria dos Programas Transitórios surge motivada pelo fator "tempo exigido para a realização do serviço" ou pela "conjugação de fatores".

Encontram-se também muitos Programas Transitórios com existência justificada pela "quantidade de dados a serem manipulados".

Raramente os Programas Transitórios são justificados pelo fator "nível de complexidade".

3.1.2.3. Programas Intermitentes

Alguns programas, embora possam ser aplicados em diversos trabalhos sobre o mesmo assunto, restringem-se a circunstâncias preestabelecidas. Conseqüentemente, não têm o

mesmo tipo de uso dos Programas Permanentes nem tão pouco são Transitórios, uma vez que possuem maior abrangência de aplicação.

Esses programas identificamos como Intermitentes.

Um programa que tenha sido desenvolvido para avaliação de estoque em regime de inflação não superior a 10% a.m. é considerado Intermitente, pois poderá ser utilizado sempre que o índice inflacionário mensal estiver abaixo de 10%.

É importante salientar que esse programa não pode ser considerado Permanente pelo fato de sua abrangência encontrar restrições dentro de problemas de mesma natureza. Esse programa, por exemplo, não pode ser utilizado para regime inflacionário superior a 10% a.m.

O fator predominante na motivação para a elaboração e uso de Programas Intermitentes é o "nível de complexidade do serviço".

É exatamente o fator "nível de complexidade do serviço" que faz com que o programa se torne Intermitente.

Geralmente, a intenção inicial é de que o programa seja Permanente, mas, à medida que a sua concepção vai evoluindo, restrições de abrangência vão surgindo como solução para minimização do nível de complexidade.

A verdade também é que o conjunto de restrições desses programas deve-se muitas vezes ao fator "tempo exigido para a realização do serviço".

3.2. QUEM UTILIZA OS PROGRAMAS APLICATIVOS

Identificaremos aqui quais os possíveis usuários de programas aplicativos.

É importante saber o perfil de quem fará uso do programa, pois as abordagens sobre os requisitos para desenvolvimento devem contemplar esse aspecto.

Reconhecemos como possíveis usuários de um programa aplicativo os grupos de profissionais que identificamos abaixo.

3.2.1. Profissionais da área a que se refere o assunto tratado pelo programa

Um programa depois de elaborado poderá ser utilizado por pessoas que conhecem, profissionalmente, o assunto por ele tratado.

Geralmente esses profissionais incluem o uso de programas como parte de um trabalho verticalizado, em que eles mesmos fornecem dados ao computador e retiram informações para a continuidade de suas atividades.

Costumamos dizer que nesses casos o "dono do problema" é quem utiliza os recursos computacionais, sem a participação de intermediários.

O "dono do problema" muitas vezes não tem conhecimentos de informática, o que não deve ser restrição para uso do programa aplicativo, mas que indiscutivelmente causa impacto no tempo e na forma de obtenção dos resultados.

3.2.2. Profissionais de informática

Em diversas circunstâncias o programa aplicativo é utilizado por profissionais especializados em informática.

Essa opção muitas vezes ocorre por particularidades do programa que exigem a interação homem-máquina, requerendo um nível significativo de conhecimentos em informática.

Mas o fator que predominantemente conduz os especialistas de informática à operação de aplicativos é o "tempo exigido para a realização do serviço"

Como esses profissionais geralmente têm grande habilidade no uso de recursos computacionais, é normal que obtenham melhor performance que os usuários de outras áreas.

O fator "tempo exigido para realização" muitas vezes impede que após o desenvolvimento do programa haja um treinamento do usuário, e a necessidade de respostas imediatas faz com que a operação seja feita continuamente por profissionais de informática.

3.2.3. Utilização conjunta

Algumas circunstâncias levam o usuário "dono do problema" a efetuar a operação do programa conjuntamente com o especialista em informática.

Essa escolha geralmente é fruto do "nível de complexidade do serviço" e do "tempo exigido para sua realização".

A utilização conjunta geralmente acontece quando o problema tratado pelo programa traz complexidades que dificultam o seu uso se não houver a participação de quem tenha domínio conceitual sobre o mesmo, e o "tempo exigido para realização" sugere a participação de profissionais com grande prática no uso de recursos computacionais.

3.2.4. Pessoas não especializadas em informática e que também não são da área a que se refere o assunto tratado pelo programa

Atualmente o nível de popularização do uso de computadores tem sido muito grande. Esse fato tem propiciado motivações interessantes como, por exemplo, a presença de pessoas que não conhecem o assunto tratado por um determinado programa nem tão pouco podem ser consideradas conhecedoras de informática, mas que, apesar disso, operam esse programa.

Tal situação é observada principalmente quando o programa é dotado de "auto-explicações" e, em função dessa característica e devido ao "tempo exigido para realização do serviço" e à escassez de recursos humanos, decide-se delegar a operação a profissionais de outras áreas que tenham disponibilidade de tempo para tal tarefa.

Constata-se, freqüentemente, que as pessoas envolvidas nessas circunstâncias têm como características pessoais desejo e habilidade para o uso de recursos computacionais. A falta dessas características pessoais traz como consequência, na grande maioria das vezes, insucessos como resultado dessa forma de uso.

3.3. POSSÍVEIS CARACTERÍSTICAS DO LOCAL ONDE O PROGRAMA É DESENVOLVIDO

O desenvolvimento e o uso dos programas para computador sofrem influências significativas das características culturais da entidade.

Dessa forma, parece-nos importante que se conheça a cultura de informática da entidade antes de se discutir sobre o desenvolvimento de programas para computador.

Procuraremos identificar aqui os possíveis estágios culturais de uma entidade em relação à informática (3).

3.3.1. Estágio embrionário

Estágio embrionário é aquele em que a entidade opta pelo uso da informática sem uma definição exata do que espera obter, do valor que precisará gastar e dos procedimentos que deverá adotar.

A decisão sobre a adoção de recursos computacionais geralmente não se embasa em um raciocínio estruturado, tornando-se, principalmente por isso, muito difícil a verbalização dos fatores motivadores e dos objetivos do uso da informática.

Normalmente as entidades descobrem a necessidade da utilização da informática em suas atividades por intermédio de:

- notícias de concorrentes que estão fazendo uso desse recurso; ou

- propagandas em veículos de comunicação pertinentes à área de atuação da entidade; ou

- experiências ou contatos individuais de alguns profissionais da entidade; ou

- representantes comerciais da indústria de computação.

É comum observar que na fase embrionária nem a definição do objetivo do uso da informática está clara.

Ou seja, não está definido ainda se a meta do uso da informática será de apoio à produção, administração ou comercialização.

A entidade inicia-se na informática com os recursos que se encontram mais facilmente à sua disposição.

Assim sendo, no estágio embrionário, o critério adotado para a escolha de recursos computacionais é basicamente a facilidade na obtenção de recursos.

Nota-se comumente nessa fase expectativas inadequadas à finalidade dos recursos. Tivemos oportunidade de observar entidades esperando a melhoria de produtividade operacional com programas destinados ao apoio à comercialização.

O estágio embrionário é caracterizado fundamentalmente por:

- indefinição dos objetivos no uso de recursos computacionais;
- desconhecimento dos fatores motivadores no desenvolvimento e no uso de programas de computador; e
- indefinição do resultado econômico que se espera obter com o uso de processamento eletrônico de dados

3.3.2. Despertar da razão

A fase "Despertar da razão" é aquela em que a entidade conclui que para o sucesso de sua informatização é indispensável que defina claramente os objetivos do uso de

recursos computacionais e os hierarquize em nível de prioridade.

Nesse estágio, as entidades apesar de terem consciência de que é fundamental definir seus objetivos, e nessa ocasião realmente os definem, ainda continuam operando no estágio embrionário.

Isso acontece porque a entidade não reúne condições ou não tem interesse financeiro em efetuar a mudança de imediato.

Principalmente por esse motivo consideramos esse estágio como sendo o início do controle administrativo da informática. Do ponto de vista da ciência da computação, esse estágio não difere do anterior.

3.3.3. Definição de uma identidade

Concluimos no subitem anterior que no "Despertar da razão" a entidade inicia o controle administrativo no uso de recursos computacionais. E esse controle provoca reflexões sobre qual seria a tecnologia adequada às necessidades da entidade.

Quando a entidade dá início à busca consciente de uma tecnologia, ela está entrando na fase que identificamos como "Definição de uma identidade".

Nessa fase, a entidade já tem noção de que é fundamental alguns procedimentos que eliminem a redundância no uso de recursos computacionais.

Como, por exemplo, a utilização de um programa de contabilidade que embora utilize algumas informações geradas pelo programa de folha de pagamentos não consegue obtê-las automaticamente, obrigando os profissionais a fazerem a entrada de dados duas vezes.

Nessa fase, a entidade buscará recursos e tecnologia que viabilizem uma única entrada de dados.

No momento em que a entidade define suas alternativas tecnológicas, consideramos que ela passa a ter uma identidade computacional à qual balizará o rumo da informática.

Normalmente, as entidades só passam a tratar a informática como um agente econômico à partir da definição de sua identidade.

3.3.4. Estágio da convergência

A entidade que já possui uma identidade computacional encontra-se motivada a efetuar o controle técnico e econômico do processo de informatização.

A motivação existente no período que sucede a definição da identidade faz com que a entidade entre num período de organização de esforços e otimização de recursos de forma a viabilizar o uso da informática em função dos seus objetivos estratégicos.

Identificamos essa fase como "Estágio da convergência".

É nessa ocasião que se elabora o "verdadeiro" plano diretor de informática.

Dizemos "verdadeiro" porque em outros estágios pelo qual passa o processo de informatização são feitas tentativas para elaborar um plano de informática, mas o que se obtém são simples identificações de metas, que na maioria das vezes não estão integradas nem tão pouco consideram os objetivos da entidade de forma adequada.

Nesse estágio, aparece também o controle da qualidade do desenvolvimento e o uso de recursos computacionais.

3.3.5. Estágio da consolidação

A entidade, quando se encontra na fase de convergência, tem como meta principal a viabilização e execução do plano diretor de informática.

Consideramos que o estágio de convergência se encerra dando lugar à "consolidação" quando consegue se manter contínua e rotineiramente cumprindo o plano diretor.

A fase de consolidação é caracterizada pela dinâmica com que se utiliza do plano diretor de informática, validando-o ou retificando-o em função de mudanças estratégicas da entidade.

Nessa fase, a cultura existente na entidade é muito favorável ao desenvolvimento e ao uso de recursos computacionais.

Consideramos que o estágio de convergência se encerra dando lugar à "consolidação" quando consegue se manter contínua e rotineiramente cumprindo o plano diretor.

A fase de consolidação é caracterizada pela dinâmica com que se utiliza do plano diretor de informática, validando-o ou retificando-o em função de mudanças estratégicas da entidade.

Nessa fase, a cultura existente na entidade é muito favorável ao desenvolvimento e ao uso de recursos computacionais.

3.4. COMO OS SOFTWARES SÃO DESENVOLVIDOS SEM ESPECIALISTAS

Julgamos que é importante para o objetivo desta dissertação a identificação das formas predominantemente adotadas por pessoas não especializadas em informática para o desenvolvimento de seus programas.

Para identificação das formas de desenvolvimento de programas, observamos em diversas entidades como eram realizadas essas atividades por profissionais de diversas áreas que não possuíam especialização em informática. Em função dessas observações, caracterizamos os procedimentos adotados pela maioria.

Entretanto, nos pareceu importante primeiramente identificar as características das pessoas não especializadas em informática que desenvolvem softwares as separando em grupos, principalmente porque a forma de desenvolvimento se diferencia em função dessas características.

3.4.1. Quem é o desenvolvedor não especializado em informática

3.4.1.1. Profissionais de áreas não vinculadas à informática

Em diversas entidades encontramos profissionais de áreas não vinculadas à informática desenvolvendo programas e com

Profundos conhecimentos na ciência da computação, embora não tivessem formação acadêmica nesse campo.

A grande maioria dos desenvolvedores incluídos nesta nossa classificação, entretanto, não possui domínio completo sobre informática.

3.4.1.2. Pessoas com formação em informática

Observamos em várias entidades de grande porte (faturamento anual em torno de US\$ 230 milhões) a existência de um centro de processamento eletrônico de dados com a finalidade de atendimento exclusivo a uma determinada área, inclusive e ligado hierarquicamente a ela. Apesar de a entidade possuir um grupo específico de informática, algumas áreas contratam profissionais de computação para lhes servir com exclusividade absoluta, concorrendo dessa forma com as áreas que oficialmente prestam serviços de computação a toda entidade.

Esses profissionais de computação acabam especializando-se no desenvolvimento de programas específicos da área em que estão lotados e, com o tempo, incorporam no seu perfil características e linguagem da área em que estão atuando.

Como há um envolvimento integral desses profissionais com o problema tratado pelo programa de computador e como se

encontram lotados na área que necessita dos resultados dos processamentos, eles acabam se transformando no usuário final, pois efetuam a coleta de dados, realizam os processamentos e muitas vezes interpretam os resultados.

Com o passar do tempo estão especializados no assunto inerente a área onde se encontra lotado perdendo o perfil que poderia caracterizá-los como especialistas em informática.

3.4.2. Como o usuário final desenvolve seus programas

3.4.2.1. Profissionais de áreas não vinculadas à informática

O desenvolvedor não vinculado a informática normalmente não faz uso de ferramenta ou metodologia científica para o desenvolvimento de seus programas, pois a sua atividade final e principal não é o desenvolvimento de programas para computador.

Este fato exerce pressão negativa nas atividades envolvidas na elaboração de programas de computador, fazendo com que sejam desprezadas algumas etapas pertinentes ao processo de desenvolvimento de software, pois a preocupação, a atenção e os esforços desse desenvolvedor estão totalmente voltados para os resultados que os programas deverão produzir.

Normalmente, assim que se decide que é necessário o desenvolvimento de um programa e que ele será desenvolvido sem a participação de profissionais de informática, o próprio desenvolvedor define o que o programa deve fazer e em seguida passa a escrevê-lo numa linguagem de computação que conhece.

Tão logo consegue eliminar problemas de sintaxe do programa que escreveu, passa a executá-lo com os dados que devem gerar os resultados necessários para sua área.

Para verificação da integridade do programa, executa manualmente o processamento de dados e posteriormente o submete ao computador. Ao obter os resultados do processamento eletrônico, os confronta com aqueles obtidos manualmente.

Ao conseguir resultados coincidentes, normalmente passa a aceitar o software como correto, considerando, a partir daí, o programa totalmente operacional.

Como dissemos anteriormente nessa classe de desenvolvedores existem aqueles que embora não sejam especialistas possuem profundos conhecimentos em informática e aqueles cujos conhecimentos são superficiais, os quais adotam procedimentos idênticos para o desenvolvimento de software.

A diferença entre essas pessoas reside basicamente no tempo que consomem para a elaboração dos programas e na escolha dos recursos computacionais.

3.4.2.2. Pessoas com formação em informática

Os profissionais com formação em informática são contratados para atendimento exclusivo a uma determinada área, conforme dito anteriormente, transformando-se em conhecedores dos problemas e das necessidades da área onde se encontram lotados.

Exatamente por entenderem perfeitamente as necessidades dos usuários, esses profissionais tomam para si a responsabilidade e a obrigação de oferecer respostas totalmente compatíveis às solicitações, mesmo que isso implique na abreviação de etapas importantes no processo de desenvolvimento de programas.

Esses profissionais, após algum tempo de atuação na área do usuário final, acabam abandonando ferramentas e metodologias de desenvolvimento de software muitas vezes por determinação explícita de superiores hierárquicos que não possuem nenhum conhecimento de informática e somente estão preocupados com os resultados dos programas e para os quais o trabalho de organização e elaboração de especificações de programas e

outras atividades similares envolvidas no desenvolvimento de software é burocracia e excesso de teoria.

Tem havido uma tendência entre os profissionais com formação em informática e que ficam lotados na área que faz uso do software de abandono de teorias e técnicas, principalmente porque a ciência da computação não é o objetivo profissional da referida área.

A falta de recursos adequados na área do desenvolvedor gera também muitas dificuldades para aplicação de conhecimentos técnicos.

É importante esclarecer que por abandonarem rotineiramente procedimentos profissionalmente indispensáveis, embora possuam formação acadêmica em ciência da computação, esses profissionais não são considerados especializados em informática.

3.4.2.3. Considerações gerais sobre o desenvolvedor de software não especializado em informática

Conclusivamente podemos conceituar o desenvolvedor não especializado em informática como uma pessoa ligada hierarquicamente a uma determinada área da entidade que precisa de informações, que podem ser geradas pelo

computador, para realização de alguma atividade inerente às suas funções.

Constatamos que de maneira geral o desenvolvedor não especializado em informática não utiliza nenhuma metodologia ou ferramenta para o desenvolvimento de programas de computador, não especifica formalmente as suas necessidades nem as funções que o software deve realizar.

3.5. QUE TIPO DE PROGRAMA É DESENVOLVIDO SEM ESPECIALISTAS

Percentualmente, a quantidade maior de softwares desenvolvidos pelo desenvolvedor não especializado é de Programas Transitórios.

Normalmente a área que responde oficialmente pela informática desenvolve suas atividades mediante programação prévia.

A dinâmica das diversas áreas de uma entidade pode gerar necessidades computacionais que nem sempre podem esperar a fila imposta pelo planejamento de desenvolvimento de programas da área de informática.

Dessa forma, aqueles programas que surgem motivados pelo "tempo exigido para realização do serviço" são os que mais estimulam o desenvolvimento independente de profissionais de informática.

Por esse mesmo motivo os desenvolvedores não especializados em informática elaboram Programas Transitórios.

ou seja, o "tempo exigido para realização do serviço" faz com que o usuário procure solucionar apenas os problemas do momento, incluindo nos programas peculiaridades que inviabilizam o uso em outras ocasiões.

É importante salientar que uma grande quantidade de Programas Transitórios deveria ter as características de Permanentes e Intermitentes, mas durante a elaboração o desenvolvedor foi obrigado a buscar, por intermédio de todos seus procedimentos, resultados imediatos, atribuindo, portanto, menor importância a partes do programa que teriam funções genéricas.

Na prática das entidades observamos que também têm desenvolvidos muitos Programas, Intermitentes e Permanentes.

Desta forma podemos afirmar que o desenvolvedor não especializado em informática têm elaborado todos os tipos de softwares.

3.6. QUALIDADE DOS PROGRAMAS DESENVOLVIDOS SEM ESPECIALISTAS

Um programa de computador é elaborado para ser utilizado como ferramenta para um determinado objetivo, o qual se encontra vinculado à missão de uma certa área.

Dessa forma, pode-se afirmar que um programa por si só não representa um fim, mas um meio para consecução dos objetivos de uma determinada área de uma entidade.

Conseqüentemente aos usuários de programas de computador o que interessa efetivamente são os resultados que eles geram. Logo, a qualidade do software para o usuário é expressa quase exclusivamente pela adequação da apresentação dos resultados às necessidades.

Quando o usuário passa a desenvolver o seu próprio software tem consigo a prioridade de obter em curto prazo qualidade adequada na apresentação dos resultados.

Em função do que foi exposto no item 2.4 podemos afirmar que essa visão é muito restrita, pois a qualidade de software deve ser expressa fundamentalmente pela integridade dos conceitos do assunto por ele tratado, pela facilidade de uso e pela apresentação adequada e correta dos resultados.

Como foi exposto no subitem 3.4.2. o desenvolvedor não especializado em informática não elabora especificação formal das suas necessidades, não formaliza descrição das funções que o programa deve realizar e praticamente não elabora projeto.

Entretanto efetua anotações que em alguns momentos, embora de forma precária, podem fazer as vezes da especificação e do projeto.

Nesta dissertação chamaremos esses apontamentos de pseudoespecificação e pseudoprojeto quando fizerem às vezes respectivamente de especificação e de projeto.

Vamos apresentar neste item uma análise da qualidade dos programas sem especialistas em informática.

Como concordamos com Rocha (4) e admitimos que a qualidade de programa é fruto da qualidade da especificação e da qualidade do projeto faremos a análise da qualidade desses programas por intermédio de levantamento dos atributos ausentes nas pseudoespecificações e pseudoprojetos.

Uma vez identificados os atributos ausentes nessas anotações indicaremos quais atributos do programa podem ser comprometidos por essa ausência.

3.6.1. Qualidade da especificação dos programas desenvolvidos sem especialistas em informática.

Apresentamos em forma de tabela os atributos que uma especificação de qualidade adequada deveria ter e com base em entrevistas, em literatura e em observações prática de aproximadamente 350 softwares desenvolvidos sem especialistas em informática indicamos se esses atributos comumente existem nas suas pseudoespecificações.

Qualidade de pseudoespecificação

	! ATRIBUTOS	! Existe este atributo ! ! na pseudoespecifica- ! ção
! U	! Fácil de modificar	! Não
! T		
! I		
! L	! Implementação viável	! Não se sabe
! I	! economicamente	
! Z		
! A	! Implementação viável	! Não se sabe
! B	! financeiramente	
! I		
! L	! Implementação viável dentro	! Sim
! I	! dos recursos disponíveis	
! D		
! A	! Implementação viável dentro	! Não
! D	! do tempo exigido	
! E	!	!

!C	R!	Corretamente escrita	! Sim	!
!O	E!		!	!
!N	P!		!	!
!F	R!	Concisa	! Sim	!
!I	E!		!	!
!A	S!		!	!
!B	E!	Terminologia uniforme	! Sim	!
!I	N!		!	!
!L	T!		!	!
!I	A!	Assuntos separados em módulos	! Não	!
!D	C!	independentes	!	!
!A	ã!		!	!
!D	O!	Encadeamento lógico de idéias	! Sim	!
!E	!		!	!

Qualidade de pseudoespecificação

	!	ATRIBUTOS	! Existe este atributo !	!
	!		! na Especificação do !	!
	!		! usuário	!
!C	C!	Coerência absoluta com os	! Sim	!
!O	O!	requisitos do usuário	!	!
!N	N!		!	!
!F	C!	Sem contradições	! Sim	!
!I	E!		!	!
!A	I!		!	!
!B	T!	Sem interpretações ambíguas	! Sim	!
!I	U!		!	!
!L	A!		!	!
!I	L!	Identifica com precisão os	! Não	!
!D	!	requisitos do usuário	!	!
!A	!		!	!
!D	!	Apresenta todas as respostas	! Não	!
!E	!	aos requisitos de forma explí-	!	!
!	!	cita	!	!

3.6.1.1. Utilizabilidade de pseudoespecificação

i) facilidade de modificação

Observa-se que as pseudoespecificações não apresentam facilidades para alterações principalmente porque são simples anotações informais sobre o que se necessita num dado momento.

ii) viabilidade econômica

É muito difícil determinar se a pseudoespecificação é viável economicamente, pois a sua superficialidade impede as análises necessárias para tais conclusões.

E o que é preocupante é que essa incognita sobre a viabilidade econômica da especificação permanece até mesmo após a elaboração do programa, como veremos mais adiante.

iii) viabilidade financeira

Pela superficialidade e informalidade das anotações do desenvolvedor também é difícil a definição sobre a viabilidade financeira da implementação da especificação do programa.

O que acontece na prática até mesmo sem que o próprio desenvolvedor perceba é a mudança da especificação em função de certas dificuldades como por exemplo de tempo e de dinheiro.

Como dissemos anteriormente na maioria das vezes inicia-se o desenvolvimento de um programa que espera seja Permanente e durante a elaboração ele vai se transformando, por uma série de exigências, em Transitório, o que é uma demonstração clara de modificação de especificação.

Num caso como esse, em que se iniciou o desenvolvimento de um programa Permanente e terminou elaborando um programa Transitório, podemos dizer que a pseudoespecificação era viável financeiramente?

Se houvesse uma pseudoespecificação, fidedigna ao programa implantado, elaborada previamente então poderíamos dizer que aquela era viável financeiramente, pois transformará-se em produto.

Mas na realidade a pseudoespecificação referente ao software desenvolvido não existe. O que se tem é uma pseudoespecificação que serviu como referencial e que no momento de sua implementação sofreu alterações, e essa não sabemos se era financeiramente viável.

iv) Viabilidade dentro das disponibilidades de recursos

Os desenvolvedores que não possuem profundos conhecimentos de informática têm a tendência de planejar as atividades que utilizam recursos computacionais de forma que correspondam à disponibilidade existente.

Isto acontece principalmente por limitações de conhecimentos Técnicos que inviabilizam extrapolações sobre o uso de outros recursos para a satisfação de necessidades do momento.

Os desenvolvedores não especializados mas que possuem formação em informática, devido aos seus conhecimentos técnicos, normalmente planejam suas atividades dentro das disponibilidades de recursos computacionais existentes.

Portanto os desenvolvedores não especializados de maneira geral quando elaboram suas especificações, de forma implícita já as viabilizam para os recursos computacionais existentes.

Nas pseudoespecificações comumente está implícita também a viabilidade de sua implementação com os outros recursos materiais e com os recursos humanos disponíveis, isto porque o desenvolvedor conhece as limitações da área em que está lotado.

v) Viabilidade dentro do tempo exigido

Conforme já foi exposto anteriormente a grande maioria dos softwares surgem motivados pelo "Tempo exigido para realização dos serviços", mas durante a definição das suas necessidades os desenvolvedores não consideram a variável tempo, simplesmente anotam o que precisam e iniciam o desenvolvimento com base exclusivamente nisto. Mas durante a elaboração do programa o tempo passa a ser fator importantíssimo, afinal foi esse aspecto que motivou a existência do software.

Nesse momento o desenvolvedor opta por, diminuir a abrangência do programa para viabilizar o término no tempo exigido.

Quando faz essa opção implicitamente o usuário está alterando a sua pseudoespecificação porque como se encontrava não era passível de implementação no tempo exigido.

Assim sendo, a implementação da pseudoespecificação original normalmente não é viável dentro do tempo exigido.

3.6.1.2. Confiabilidade de representação de pseudoespecificação

i) Corretamente escrita

Normalmente as anotações do desenvolvedor não especializado apresentam-se corretamente escrita.

ii) Concisa

As anotações são bastante concisas.

iii) Terminologia uniforme

O desenvolvedor não especializado produz sua escrita com termos pertinentes à sua área de atuação mantendo uniformidade na referência dos mesmos.

iv) Assuntos separados em módulos independentes

As pseudoespecificações raramente são separadas em módulos independentes.

v) Encadeamento lógico de idéias

O desenvolvedor não especializado consegue produzir anotações que expressam idéias com bom encadeamento lógico. Talvez porque já executou sem auxílio do computador as funções que o programa deverá exercer o que lhe dá pleno domínio sobre o encaminhamento e sequência exata de procedimentos.

3.6.1.3. Confiabilidade conceitual de pseudoespecificação

i) Coerência absoluta com os requisitos do usuário

As especificações são absolutamente coerentes com os requisitos do usuário, pois é ele mesmo quem as escreve.

O que eventualmente acontece é que alguns requisitos não aparecem de forma explícita.

Isto porque é o "dono dos requisitos" quem está elaborando as pseudoespecificações e opta, pela abreviação da descrição omitindo pontos que lhe parecem obvios.

ii) Sem contradições

Devido ao domínio dos conceitos que serão embutidos na pseudoespecificação o usuário não a deixa com contradições.

iii) Sem interpretação ambíguas

As pseudoespecificações contém interpretações ambíguas pois normalmente o desenvolvedor tem bastante conhecimento sobre os conceitos e basicamente não os interpreta no momento da especificação do programa, simplesmente os transcreve conforme surgem na sua cabeça.

iv) Identifica com precisão os requisitos do usuário

O usuário tem a tendência de abreviar ao máximo descrição das suas necessidades desta forma as especificações elaboradas pelo usuário, não identificam com precisão as suas necessidades.

v) Apresenta todas as respostas aos requisitos de forma explícita

Também pelo costume de abreviar a descrição o desenvolvedor não especializado em informática não apresenta todas as respostas de forma explícita.

3.6.1.4. Comentários adicionais sobre pseudoespecificações

Analisando-se a quantidade de atributos ausentes nas pseudoespecificações nas (especificações elaboradas pelo desenvolvedor não especializado em informática) pode-se concluir que:

i) A manipulação, entendimento e utilização de pseudoespecificações é muito difícil para aqueles que não acompanharam a sua elaboração;

ii) As pseudoespecificações não servem de forma adequada às eventuais necessidades de alterações em programas;

iii) São representadas de maneira bastante confiável; e

iv) Oferecem confiabilidade conceitual.

3.6.2. Qualidade de projeto dos programas desenvolvidos sem especialistas em informática

Será apresentado também em forma de tabela, os atributos que um projeto deve ter para ser considerado de qualidade adequada e também embasado em entrevistas, em literatura e em observações práticas de aproximadamente 350 trabalhos será indicado se esses atributos aparecem com frequência nos pseudoprojetos (projetos elaborados por desenvolvedores não especializados em informática).

Qualidade de pseudoprojetos

ATRIBUTOS		Existe este atributo nos pseudoprojetos
U	Fácil de modificar	Não
T		
I		
L	Implementação viável econômica-mente	Não se sabe
I		
Z		
A	Implementação viável financeiramente	Não se sabe
B		
I		
L	Implementação viável dentro recursos disponíveis	Sim
I		
D		
A	Implementação viável dentro do tempo exigido	Não
D		
E		

C	R	Simbologia e uso da linguagem	Não
O	E	correta	
N	P		
F	R	Otimização de informações	Sim
I	E		
A	S		
B	E	Encadeamento lógico e objetivo	Sim
I	N	de idéias	
L	T		
I	A	Assuntos separados em módulos	Não
D	C		
A	ã		
D	O		
E			

!C	C!	Coerência absoluta com as	! Não
!O	O!	descrições da especificação	!
!N	N!		!
!F	C!	Previsão de solução para	! Não
!I	E!	eventuais problemas de uso ou	!
!A	I!	de fornecimento de dados	!
!B	T!		!
!I	U!		!
!L	A!		!
!I	L!		!
!D	!		!
!A	!		!
!D	!		!
!E	!		!

3.6.2.1. Utilizabilidade de pseudoprojetos

i) Facilidade de modificação

As anotações nos pseudoprojetos são apresentadas de tal forma que não é fácil modificá-las.

ii) Viabilidade econômica

A superficialidade dos pseudoprojetos, que são elaborados de maneira muito informal e com poucas anotações, impede de se fazer análise segura da viabilidade econômica.

iii) Viabilidade financeira

É difícil saber se o pseudoprojeto original é financeiramente viável principalmente porque sempre é

modificado sem registros durante o desenvolvimento de programas.

iv) Viabilidade dentro dos recursos disponíveis

O desenvolvedor não especializado em informática já se preocupa com adequação de seu software aos recursos disponíveis desde a sua pseudoespecificação, como consequência disto os seus projetos normalmente são viáveis quando se olha por esse ângulo.

v) Viabilidade dentro do tempo exigido

Comumente a implementação do pseudoprojeto original não é viável dentro do tempo exigido.

O desenvolvedor faz modificações profundas no seu projeto para completar o programa no tempo exigido.

Como o que resulta das modificações normalmente não fica ordenadamente registrado e como essas modificações são feitas costumeiramente durante a elaboração de programas considera-se que o pseudoprojeto não foi implementado.

3.6.2.2. Confiabilidade de representação de pseudoprojetos

i) Uso correto de simbologia e de linguagem

O desenvolvedor não especializado em informática não tem o costume de recorrer a simbologias para suas representações, mas utiliza corretamente a língua portuguesa.

Apesar do uso correto da linguagem Consideramos que o pseudoprojeto não possui o atributo "uso correto de simbologia e linguagem" pelo fato da ausência de simbologia comprometer o seu perfeito entendimento. Isto porque, a forma que utiliza a linguagem apesar de correta não é suficiente para superar a força de comunicação e transmissão de idéias, tão necessárias em projetos de software, que os símbolos gráficos oferecem.

ii) Otimização de informações

Nos pseudoprojetos a otimização de informações é um atributo constante.

Como conhece profundamente o tipo de informação que precisa fornecer e obter do software consegue evitar a presença daquelas desnecessárias

iii) Encadeamento lógico de idéias

Os pseudoprojetos são constituídos por um bom encadeamento lógico de idéias.

iv) Assuntos separados em módulos

Os assuntos não aparecem separados por módulos nos pseudoprojetos.

3.6.2.3. Confiabilidade conceitual de pseudoprojetos

i) Coerência absoluta com as descrições da pseudoespecificações

Como dissemos anteriormente o desenvolvedor não especializado em informática tem como fator motivador para o desenvolvimento do seu próprio software o "Tempo exigido para realização dos serviços".

Esse fator faz com que o desenvolvedor vá diminuindo a cada momento a abrangência do seu programa.

Portanto durante a elaboração do pseudoprojeto já se pode observar a redução da abrangência em relação ao que encontra-se na pseudoespecificação.

Desta forma seria incorreto dizer que o pseudoprojeto mantém coerência absoluta com as descrições da pseudoespecificação.

ii) Previsão de solução para eventuais problemas de uso ou de fornecimento de dados

O desenvolvedor não especializado em informática não faz em seu pseudoprojeto previsão de solução para eventuais problemas de uso ou de fornecimento de dados, principalmente porque não acredita que isto poderá ocorrer principalmente porque conta com a possibilidade dele mesmo fornecer dados e utilizar o programa, ou assistir de perto esse trabalho, uma vez que encontra-se lotado na área que fará uso do software.

3.6.2.4. Comentários adicionais sobre pseudoprojetos

Em função da quantidade de atributos ausentes nas anotações do desenvolvedor não especializado em informática que consideramos como projeto pode-se concluir que:

i) A manipulação, entendimento e utilização de pseudoprojetos também é muito difícil para quem não acompanhou sua elaboração.

ii) Os pseudoprojetos não são ferramentas adequadas para se entender o funcionamento dos programas e também não são úteis para apoio a eventuais necessidades de alterações em programas.

iii) As representações de pseudoprojetos são confiáveis apesar de não serem de fácil entendimento para terceiros

iv) A confiabilidade conceitual dos pseudoprojetos é questionável, uma vez que não respeita as descrições das pseudoespecificações.

Normalmente é possível afirmar-se que um projeto é conceitualmente confiável verificando apenas se a especificação é confiável e se o projeto mantém coerência absoluta com essa.

Como os pseudoprojetos de usuários raramente mantêm ligação direta com especificação é necessário que se estude detalhadamente os atributos do pseudoprojeto para saber se esse é conceitualmente confiável.

3.6.3. Impacto da qualidade de pseudoespecificação e de pseudoprojeto na qualidade de programa

Uma vez mais nos utilizaremos de tabela para indicar o impacto da qualidade de pseudoespecificação e da qualidade de pseudoprojeto na qualidade de programa.

Na vertical indicaremos os atributos necessários para que um programa seja considerado de boa qualidade, na horizontal estarão os atributos necessários para que pseudoespecificações e pseudoprojetos sejam considerados de

qualidade adequada. Na intersecção das linhas e colunas indicaremos se o atributo da pseudoespecificação ou do pseudoprojeto, que encontra-se acima está ausente e se influencia o atributo do programa que encontra-se à esquerda.

Assim sendo utilizaremos a seguinte nomenclatura:

A - I = significa que o atributo que aparece na coluna está ausente na pseudoespecificação (ou pseudoprojeto) e sua ausência influencia direta e negativamente o atributo do programa que aparece naquela linha

p - i = Significa que o atributo que aparece na coluna está presente na pseudoespecificação (ou pseudoprojeto) e sua presença influencia direta e positivamente o atributo do programa que aparece naquela linha

- = Quando aparecer o hífen na intersecção entre linha e coluna significa que o atributo que aparece na coluna está ausente na pseudoespecificação (ou pseudoprojeto) e sua ausência não influencia o atributo do programa que aparece naquela linha ou significa que o atributo que aparece na coluna está presente na pseudoespecificação e sua presença não influencia o atributo do programa que aparece naquela linha.

Nessa circunstâncias não existe interesse em diferenciar os atributos ausentes e presentes da pseudoespecificação ou pseudoprojeto porque não influenciam o atributo do programa.

Quando não soubermos se um atributo está presente na pseudoespecificação (ou pseudoprojeto) adotaremos uma posição rígida a favor da qualidade adequada e indicaremos que o atributo está ausente.

QUADRO SÍNTESE
Impactos da qualidade Pseudoespecificação qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOESPECIFICAÇÃO				
	UTILIZABILIDADE				
	Fácil de modificar	Viável economicamente	Viável financeiramente	Viável dentro dos recursos disponíveis	Viável dentro do tempo exigido
U. Fácil de Operar	-	A - I	A - I	P - I	A - I
I.					
II.					
III.					
IV. Não exige muitos recursos humanos e materiais para o uso	-	A - I	A - I	P - I	A - I
V.					
VI. Fácil de efetuar pequenas modificações em seus roteiros	A - I	-	-	-	-
VII.					
VIII. Oferece resultados em tempo adequado	-	-	-	-	-
IX.					
X. Consome mínimo necessário de recursos de máquina	A - I	A - I	A - I	P - I	A - I
XI.					
XII. Elaborado com o máximo possível de recursos computacionais padronizados	-	-	-	-	-
XIII.					
XIV. Uso apresenta boa relação custo benefício	A - I	A - I	A - I	P - I	A - I

QUADRO SÍNTESE

Impactos da qualidade de Pseudoespecificação na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOESPECIFICAÇÃO			
	CONFIABILIDADE CONCEITUAL			
	Coerência absoluta com requisitos do usuário	Sem interpretações ambíguas	Identifica com precisão requisitos do usuário	Apresenta todas respostas aos requisitos de forma explícita
UI. Fácil de Operar	P - I	-	-	-
TI				
II				
LI				
II. Não exige muitos recursos humanos e materiais para o uso	-	-	-	-
Z				
A				
B				
II. Fácil de efetuar pequenas modificações em seus roteiros	-	-	-	-
L				
II				
D				
A. Oferece resultados em tempo adequado	-	-	-	-
D				
E				
. Consome mínimo necessário de recursos de máquina	-	-	-	-
.				
. Elaborado com o máximo possível de recursos computacionais padronizados	-	-	-	-
.				
. Uso apresenta boa relação custo benefício	-	-	-	-
.				

QUADRO SÍNTESE

Impactos da qualidade de Pseudoespecificação na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOESPECIFICAÇÃO				
	CONFIABILIDADE DE REPRESENTAÇÃO				
	Corretamente escrita	Concisa	Terminologia uniforme	Assuntos separados em módulos	Encadeamento lógico de idéias
U. Fácil de Operar	-	P - I	-	A - I	P - I
I. Não exige muitos recursos humanos e materiais para o uso	-	-	-	-	-
I. Fácil de efetuar pequenas modificações em seus roteiros	P - I	-	-	A - I	P - I
A. Oferece resultados em tempo adequado	-	-	-	-	P - I
I. Consome mínimo necessário de recursos de máquina	-	P - I	-	A - I	P - I
I. Elaborado com o máximo possível de recursos computacionais padronizados	-	-	P - I	-	-
I. Uso apresenta boa relação custo benefício	-	P - I	-	A - I	P - I

QUADRO SÍNTESE

Impactos da qualidade de Pseudoespecificação na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS PSEUDOESPECIFICAÇÃO				
	UTILIZABILIDADE				
	Fácil de modificar	Viável economicamente	Viável financeiramente	Viável dentro recursos disponíveis	Viável dentro tempo disponível
IC1. Efetua cálculos com a	-	-	-	-	-
IO1. precisão e presteza					
IN1. exigida pelo usuário					
IF1.					
II1. Realiza todas as funções	-	-	-	-	-
IA1. com o nível de detalhe					
IB1. requerido					
II1.					
IL1. Preparado para contornar	-	-	-	-	-
II1. eventuais erros de uso ou					
ID1. de fornecimento de dados					
IA1.					
IO1.					
IE1.					
IC1.					
IO1.					
IN1.					
IC1.					
IE1.					
II1.					
IT1.					
IU1.					
IA1.					
IL1.	-				

QUADRO SÍNTESE

Impactos da qualidade de Pseudoespecificação na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOESPECIFICAÇÃO			
	CONFIABILIDADE CONCENTUAL			
	Coerência absoluta com requisitos do usuário	Sem interpretações ambíguas	Identifica com precisão requisitos do usuário	Apresenta todas as respostas aos requisitos de forma explícita
C1. Efetua cálculos com a precisão e presteza exigida pelo usuário	P - I	P - I	A - I	-
I1. Realiza todas as funções com o nível de detalhe requerido	P - I	P - I	A - I	A - I
L1. Preparado para contornar eventuais erros de uso ou de fornecimento de dados	-	-	-	-
D1.				
E1.				
C1.				
O1.				
N1.				
C1.				
E1.				
I1.				
T1.				
U1.				
A1.				
L1.				

Impactos da qualidade de Pseudoespecificação na qualidade do Programa

171

QUADRO SÍNTESE

Impactos qualidade de Pseudoespecificação na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOESPECIFICAÇÃO				
	UTILIZABILIDADE				
	Fácil de modificar	Viável economicamente	Viável financeiramente	Viável dentro dos recursos disponíveis	Viável dentro do tempo exigido
C1. Codificado com clareza	-	-	-	-	-
I1. Assuntos separados em módulos independentes	-	-	-	-	-
L1. Estruturado hierarquicamente viabilizando acessos indep. aos módulos	-	-	-	-	-
D1. É fácil identificar os elos de ligação e forma de relacion. dos módulos	-	-	-	-	-
E1. Satisfaz com precisão os requisitos do usuário	A - I	A - I	A - I	P - I	A - I
S1. Apresenta todas respostas aos requisitos do usuário de forma explícita	A - I	A - I	A - I	P - I	A - I

QUADRO SÍNTESE

Impactos da qualidade de Pseudoespecificação na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOESPECIFICAÇÃO			
	CONFIABILIDADE CONCEITUAL			
	Coerência absoluta com requisitos do usuário	Sem interpretações ambíguas	Identifica com precisão requisitos do usuário	Apresenta todas respostas aos requisitos de forma explícita
Ci. Codificado com clareza	P - I	P - I	-	-
Di. Assuntos separados em módulos independentes	P - I	P - I	-	-
Li. Estruturado hierarquicamente viabilizando acessos indep. aos módulos	P - I	-	-	-
Di. É fácil identificar os elos de ligação e forma de relacion. dos módulos	-	P - I	-	-
Re. Satisfaz com precisão os requisitos do usuário	P - I	P - I	A - I	-
Re. Apresenta todas respostas aos requisitos do usuário de forma explícita	-	P - I	A - I	A - I

QUADRO SÍNTESE

Impactos da qualidade de Pseudoespecificação na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOESPECIFICAÇÃO				
	CONFIABILIDADE DE REPRESENTAÇÃO				
	Corretamente escrita	Concisa	Terminologia uniforme	Assuntos separados em módulos	Encadeamento lógico de idéias
C: Codificado com clareza O: N:	P - I	P - I	P - I	A - I	P - I
F:					
I: Assuntos separados em A: módulos independentes B:	P - I	P - I	P - I	A - I	P - I
I:					
L: Estruturado hierarquica- I: mente viabilizando aces- D: sos indep. aos módulos A:	-	P - I	-	A - I	P - I
D: É facil identificar os E: elos de ligação e forma I: de relacion. dos módulos R:	P - I	-	P - I	A - I	-
E: Satisfaz com precisão os P: requisitos do usuário R:	P - I	-	-	-	P - I
E: Apresenta todas respostas S: aos requisitos do usuário E: de forma explícita N:	P - I	-	-	-	P - I
T:					
A:					
C:					
A:					
O:					

QUADRO SÍNTESE

Impactos da qualidade Pseudoprojeto na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOPROJETO				
	UTILIZABILIDADE				
	Fácil de modificar	Viável economicamente	Viável financeiramente	Viável dentro dos recursos disponíveis	Viável dentro do tempo exigido
UI. Fácil de Operar	-	A - I	A - I	P - I	A - I
II. Não exige muitos recursos humanos e materiais para o uso	-	A - I	A - I	P - I	A - I
III. Fácil de efetuar pequenas modificações em seus roteiros	A - I	-	-	-	-
IV. Oferece resultados em tempo adequado	-	A - I	A - I	P - I	A - I
V. Consome mínimo necessário de recursos de máquina	A - I	A - I	A - I	P - I	A - I
VI. Elaborado com o máximo possível de recursos computacionais padronizados	-	-	-	-	-
VII. Uso apresenta boa relação custo benefício	A - I	A - I	A - I	P - I	A - I

QUADRO SÍNTESE
Impactos da qualidade de Pseudoprojeto na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOPROJETO	
	CONFIABILIDADE CONCEITUAL	
	Coerência absoluta! Previsão de solução para eventuais problemas no uso ou fornecimento da especificações de dados	
UI. Fácil de Operar	-	-
IT!		
II!		
IL!		
II. Não exige muitos recursos humanos e materiais para o uso	-	A - I
IB!		
II. Fácil de efetuar pequenas modificações em seus roteiros	A - I	-
ID!		
IA. Oferece resultados em tempo adequado	-	-
IE!		
I. Consome mínimo necessário de recursos de máquina	-	-
I. Elaborado com o máximo possível de recursos computacionais padronizados	-	-
I. Uso apresenta boa relação custo benefício	A - I	A - I

QUADRO SÍNTESE

Impactos da qualidade de Pseudoprojeto na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOPROJETO			
	CONFIABILIDADE DE REPRESENTAÇÃO			
	Simbologia e uso da linguagem corretas	Otimização de informações	Encadeamento lógico de idéias	Assuntos separados em módulos
UI. Fácil de Operar	-	P - I	P - I	-
II. Não exige muitos recursos humanos e materiais para o uso	-	P - I	-	-
II. Fácil de efetuar pequenas modificações em seus roteiros	A - I	P - I	P - I	A - I
AI. Oferece resultados em tempo adequado	-	P - I	P - I	-
I. Consome mínimo necessário de recursos de máquina	A - I	P - I	P - I	-
I. Elaborado com o máximo possível de recursos computacionais padronizados	A - I	-	-	-
I. Uso apresenta boa relação custo benefício	A - I	P - I	P - I	-

QUADRO SÍNTESE

Impactos da qualidade de Pseudoprojeto na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS PSEUDOPROJETO				
	UTILIZABILIDADE				
	Fácil de modificar	Viável economicamente	Viável financeiramente	Viável dentro recursos disponíveis	Viável dentro tempo exigido
CI. Efetua cálculos com a	-	A - I	A - I	P - I	A - I
OI. precisão e presteza					
NI. exigida pelo usuário					
FI.					
II. Realiza todas as funções	A - I	-	-	-	-
AI. com o nível de detalhe					
BI. requerido					
LI.					
LI. Preparado para contornar	-	-	-	-	-
II. eventuais erros de uso ou					
DI. de fornecimento de dados					
AI.					
DI.					
EI.					
CI.					
OI.					
NI.					
CI.					
EI.					
II.					
TI.					
UI.					
AI.					
LI.	-				
LI.					
LI.					
LI.					

QUADRO SÍNTESE

Impactos da qualidade de Pseudoprojeto na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOPROJETO	
	CONFIABILIDADE CONCEITUAL	
	Coerência absoluta; Previsão de solução para eventuais problemas no uso ou fornecimento; da especificação de dados	
CI. Efetua cálculos com a precisão e presteza exigida pelo usuário	A - I	-
II. Realiza todas as funções com o nível de detalhe requerido	A - I	-
LI. Preparado para contornar eventuais erros de uso ou de fornecimento de dados	A - I	A - I
DI.		
EI.		
CI.		
DI.		
NI.		
CI.		
EI.		
II.		
TI.		
UI.		
AI.		
LI.		

Impactos da qualidade de Pseudoprojeto na qualidade do Programa

180

QUADRO SÍNTESE

Impactos qualidade de Pseudoprojeto na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOPROJETO				
	UTILIZABILIDADE				
	Fácil de modificar	Viável economicamente	Viável financeiramente	Viável dentro dos recursos disponíveis	Viável dentro do tempo exigido
Ci. Codificado com clareza	-	-	-	-	-
Di. Assuntos separados em módulos independentes	-	-	-	-	-
Li. Estruturado hierarquicamente viabilizando acessos indep. aos módulos	-	-	-	-	-
Di. é fácil identificar os elos de ligação e forma de relacion. dos módulos	-	-	-	-	-
E. Satisfaz com precisão os requisitos do usuário	A - I	A - I	A - I	P - I	A - I
S. Apresenta todas respostas aos requisitos do usuário de forma explícita	A - I	-	-	-	-

QUADRO SÍNTESE

Impactos da qualidade de Pseudoprojeto na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOPROJETO	
	CONFIABILIDADE CONCEITUAL	
	Coerência absolutaPrevisão de solução para eventuais com as descrições problemas no uso ou fornecimento da especificação de dados	
CI. Codificado com clareza	-	-
II. Assuntos separados em módulos independentes	-	-
LI. Estruturado hierarquica- mente viabilizando aces- sos indep. aos módulos	-	-
DI. É fácil identificar os elos de ligação e forma de relacion. dos módulos	-	-
EI. Satisfaz com precisão os requisitos do usuário	A - I	A - I
EI. Apresenta todas respostas aos requisitos do usuário de forma explícita	-	-
IT.		
IA.		
IC.		
IA.		
IO.		

QUADRO SÍNTESE

Impactos da qualidade de Pseudoprojeto na qualidade do Programa

ATRIBUTOS DO PROGRAMA	ATRIBUTOS DE PSEUDOPROJETO			
	CONFIABILIDADE DE REPRESENTAÇÃO			
	Simbologia e uso da linguagem corretas	Otimização de informações	Encadeamento lógico de idéias	Assuntos separados em módulos
Cl. Codificado com clareza	A - I	-	P - I	A - I
Il. Assuntos separados em módulos independentes	A - I	-	P - I	A - I
Li. Estruturado hierarquicamente viabilizando acessos indep. aos módulos	A - I	-	P - I	A - I
Di. É fácil identificar os elos de ligação e forma de relacion. dos módulos	A - I	-	P - I	-
Ei. Satisfaz com precisão os requisitos do usuário	A - I	-	P - I	-
Es. Apresenta todas respostas aos requisitos do usuário de forma explícita	A - I	-	P - I	-

3.6.4. Comentários adicionais sobre a qualidade de programas desenvolvidos sem especialistas em informática.

Um software desenvolvido sem especialistas em informática normalmente possui uma apresentação das informações finais bastante adequada às necessidades, isso contribui com os aspectos econômicos da entidade, na medida em que não gera desperdício de tempo no momento de seu uso por apresentar-se de forma exata para utilização.

A apresentação adequada protege também a interpretação das informações. E aqui reside uma grande vantagem de profissionais de uma determinada área desenvolverem o software que essa necessita pois informações com má apresentação podem gerar erro de interpretação e consequentemente, decisões errôneas.

No que diz respeito à qualidade do software é preocupante a não adoção, pelo desenvolvedor, não especializado em informática de uma metodologia que permita em qualquer tempo e lugar a validação do conceito embutido no programa.

Durante o uso, raramente se questiona os aspectos conceituais, pois parte-se do princípio de que o software está totalmente correto. Essa segurança torna-se maior à medida que a apresentação dos resultados se aproxima das expectativas dos que com ele trabalham.

Portanto, quanto mais tempo de existência tem um programa para computador, maior a necessidade de procedimentos estruturais e pragmáticos para a sua validação conceitual, pois o usuário é capaz de abandonar suas convicções teóricas no momento em que elas não forem ratificadas por um programa que tenha sido implantado há muito tempo, o qual pode estar errado.

3.6.5. Conclusões sobre a qualidade de programas desenvolvidos sem especialistas em informática

Segundo Rocha (4) "qualidade não pode ser definida universalmente, mas deve ser definida para o item em questão".

Acrescentamos que também deve ser definida em função de exigências e características dos usuários do item em questão.

Desta forma a variação de características dos usuários pode implicar em novo referencial de qualidade.

No item anterior indicamos parâmetros para formação de juízo sobre a qualidade de software desenvolvidos sem especialistas em informática.

Acreditamos que o juízo sobre a qualidade do software desenvolvido sem especialistas poderá ser obtido por intermédio do uso daqueles parâmetros e de reflexões sobre as características, exigências e necessidades dos possíveis usuários do software.

É importante ressaltar aqui que para o controle econômico do desenvolvimento de software sem especialistas é de fundamental importância que esse juízo sobre a qualidade do software, seja formado durante o desenvolvimento, para que possa atuar de forma preventiva.

Desta forma torna-se igualmente importante a definição no modelo de tomada de decisões sobre o controle econômico de desenvolvimento de software, de parâmetros que expressem com presteza as exigências (com respectivas relevâncias) que se faz a atributos que constituirão o software.

Como já afirmamos não é possível definir de forma genérica qual tem sido a qualidade dos softwares desenvolvidos sem especialistas mas ao considerarmos os quadros sobre impactos da qualidade de pseudoespecificações e pseudoprojetos podemos constatar que é preocupante, o nível de qualidade dos softwares resultantes dessa forma de desenvolvimento (observe a quantidade de A-I existentes nos Quadros Sínteses apresentados no item 3.6.3).

Merecendo dessa forma, a qualidade uma atenção especial da gestão econômica principalmente porque muitos softwares são utilizados por processos de tomada de decisões estratégicas e operacionais.

3.7. EXEMPLO DE PROBLEMA OCASIONADO PELA NÃO QUALIDADE DE PROGRAMA

Tivemos oportunidade de testemunhar uma equipe de engenheiros reprojeter uma rede de abastecimento de água de 200 quilômetros porque um programa que buscava a melhor opção de custos do dimensionamento apontou valores incompatíveis com aqueles que tinham sido estimados pelo grupo e adotados preliminarmente no projeto.

O que influenciou fortemente a decisão de alterar o projeto foi o fato do programa já ter sido utilizado em inúmeros projetos. Entretanto se descobriu meses após o término dos serviços que o programa utilizado possuía um erro que não havia sido detectado nos outros projetos em que fora utilizado, porque nenhum possuía uma determinada característica, que fazia o programa atuar erroneamente.

Esse programa para determinação e otimização de custos de redes de abastecimento de água foi desenvolvido por engenheiros seniores com experiência média de 12 anos em projetos hidráulicos, possuía características muito elogiáveis na comunidade, entretanto o seu desenvolvimento aconteceu sem nenhuma metodologia que viabilizasse uma avaliação conceitual com facilidade e pragmatismo.

No momento em que o programa apresentou dados divergentes, a equipe de projetos não possuía nenhum recurso para verificar

os conceitos nele embutidos. Fazer manualmente outros cálculos era impossível pelo cronograma dos trabalhos, além disso, existia a pressão natural do cliente para a entrega dos serviços.

Considerando-se apenas as correções no projeto estima-se que as retificações implicaram num custo direto adicional de 500.000 dólares.

É importante frisar que esse gasto direto representou a menor consequência econômica mensurável.

No caso mencionado a inexistência do fator utilizabilidade como propriedade de programa contribuiu para a tomada de decisão errada. Em situações como aquela é possível enxergar a qualidade como um agente econômico, isso porque a sua inexistência afeta a economia da entidade.

CITAÇÕES DO CAPÍTULO 3

CITAÇÕES DO CAPÍTULO 3

- (1) FERREIRA, Aurélio B.H. Minidicionário da Língua Portuguesa
Nova Fronteira - RJ - 1985
- (2) SEI Panorama do Setor de Informática
Séries Estatísticas V. 2 n.º 1
- agosto/1989
Secretaria Especial da Ciência e Tecnologia
Secretaria especial de Informática
- (3) - A favor do respeito intelectual, informamos que, certa ocasião, circulou pelo nosso escritório artigo cujo autor e periódico onde fora publicado, não nos lembramos.
Não fizemos anotação alguma sobre o artigo.
Definimos os estágios apresentados nesta dissertação com base em entrevistas, estudos e análises. Tudo o que está aqui escrito é por nossa conta e risco, mas é correto dizer que a motivação veio daquele artigo.
- (4) ROCHA, A.R. da Análise de Projeto Estruturado de Sistemas
Campus - RJ - 1987

CAPÍTULO 4 - CONTROLE ECONÔMICO DO DESENVOLVIMENTO DE
SOFTWARE

4. CONTROLE ECONÔMICO DO DESENVOLVIMENTO DE SOFTWARES

4.1. Fatores da qualidade de software são agentes econômico

Entendemos que a qualidade de programas é um agente econômico, e por isto pode afetar a riqueza da entidade de forma positiva ou negativa.

O Dicionário Brasileiro da Língua Portuguesa (1) diz que "agente é: tudo o que opera, causa, promotor, o que pratica a ação".

Desta forma utilizamos nesta dissertação o termo agente econômico como sendo aquilo que pratica, causa ou promove o evento econômico.

Para motivar reflexões sobre a qualidade como agente econômico faremos nos próximos subítens considerações sobre a ausência dos objetivos de qualidade (utilizabilidade, confiabilidade de representação e confiabilidade conceitual) em softwares.

4.1.1. Confiabilidade conceitual de programa

No subitem 3.7. se pode observar através de fato a importância da validação da integridade conceitual.

No caso mencionado a inexistência da integridade conceitual como propriedade do programa propiciou a tomada de decisão errada.

Em situação como aquela é fácil enxergar a qualidade como um agente econômico, isto porque a sua inexistência afeta a riqueza da entidade de forma bastante perceptível e até certo ponto mensurável.

Mas a inexistência da integridade conceitual sempre afeta a economia da entidade só que algumas vezes de maneira menos visível e com efeitos de menor relevância (é importante salientar que o somatório de fatos econômicos que isoladamente são irrelevantes pode resultar evento muito relevante).

4.1.2. Confiabilidade da representação de programas

Um atributo importante para que o programa tenha confiabilidade na sua representação é apresentação de todas as respostas aos requisitos do usuário de forma explícita.

Imaginemos um programa de faturamento que realize conversão de valores monetários expressos em dólares para cruzeiros e que não exiba o fator de conversão, se houver erro no roteiro de cálculo do programa essa omissão poderá estar

contribuindo para que o mesmo não seja facilmente detectado e assim estará ajudando à possibilidade de geração de prejuízos para a empresa.

Esse exemplo mostra uma possibilidade da não qualidade da representação afetar a riqueza da entidade.

A inexistência de confiabilidade de representação é bastante preocupante principalmente quando o Programa é Permanente, pois ao mesmo tempo que se aumenta o uso se diminui o domínio sobre a estruturação lógica do software, isto porque raramente o desenvolvedor não especializado em informática, documenta os procedimentos e conceitos adotados. Dessa forma, o desenvolvedor detém exclusivamente a estruturação do programa e, com o passar do tempo, é natural que essa vá caindo no esquecimento.

Conseqüentemente, o uso de partes ainda não utilizadas desse software torna-se cada vez mais um risco à integridade dos resultados a serem produzidos pelo software.

4.1.3. Utilizabilidade de programa

O usuário final não dá ênfase alguma à eficiência operacional de seus programas, mas sim à forma de apresentação dos resultados.

Esse fato faz com que as possibilidades de os programas se tornarem eficientes operacionalmente fiquem reduzidas, na medida em que não existe durante o processo de desenvolvimento preocupação com o consumo de recursos de máquina, que é um atributo importante do fator utilizabilidade.

Outro aspecto que contribui contra a utilizabilidade é o fato de o usuário, em geral não acompanhar a dinâmica da evolução dos recursos computacionais, permanecendo fiel às primeiras ferramentas que teve conhecimento, as quais podem, inclusive, não serem adequadas à solução dos seus novos problemas.

Por exemplo para calcular 3.000 vezes o seno de um número qualquer consomem-se 17 segundos se for utilizada a linguagem de computador chamada BASIC e 1 segundo se optar-se por outra denominada FORTRAN.

Portanto, se houver necessidade de desenvolver um programa que efetue cálculos trigonométricos, e se o BASIC for a linguagem escolhida então será produzido um programa com ineficiência operacional.

Essa ineficiência será totalmente irrelevante se a frequência de uso e a quantidade de dados forem pequenas. Entretanto, se houver uma frequência significativa de uso ou

um grande volume de dados a serem manipulados, aí sim poderá haver um fato econômico relevantemente prejudicial à entidade.

Vejamos, se fosse necessária a realização de cálculos trigonométricos que consumissem 8 horas - o que equivale a um dia útil de trabalho - se a escolha fosse BASIC, em FORTRAN seriam realizados em meia hora.

Essa relação, meia hora contra 8 horas, aguçá a sensibilidade a nível de gastos diretos. Entretanto, economicamente, a perda pode ser muito maior. Esta perda pode ser facilmente constatada se houver alguma atividade parada à espera dos resultados.

A ineficiência de um software - que é um atributo da utilizabilidade - pode comprometer a economia da entidade, a médio e longo prazo, através de um acúmulo de pequenos prejuízos observados cada vez que o programa é utilizado.

4.2. INDICADORES DE QUALIDADE

No capítulo 3 afirmamos que não é possível formar juízo absoluto sobre a qualidade de programas desenvolvidos sem especialistas exclusivamente por intermédio da ausência ou presença de atributos nas suas características, é indispensável para essa finalidade que se considere também a relevância de cada atributo.

Essa relevância varia de entidade para entidade e de um software para outro, por exemplo o atributo "oferece resultados em tempo adequado" tem relevância significativa no software de uma empresa aérea que precisa saber a cada momento, por intermédio desse software, se existe lugar no próximo voo para o passageiro que encontra-se no balcão, entretanto, este mesmo atributo não tem relevância significativa para o software, da mesma empresa, que se presta a calcular volume semanal de vendas de passagens aéreas.

A identificação da qualidade de um software passa por um alto grau de subjetividade

Principalmente por isto nos parece fundamental que se faça, dentro do possível, indicações quantitativas da qualidade de software. De tal forma que seja possível que se determine

qualidade com facilidade e objetividade por diversos pontos de vista.

Assim sendo deveríamos estabelecer alguns critérios (que são, segundo Rocha (op. cit.), atributos possíveis de serem avaliados), que permitissem determinar uma medida de qualidade.

Nesta tentativa poderíamos, por exemplo, adotar um "medidor" do tipo:

$$Q = \sum_{i=1}^n A_i \times P_i$$

onde n é a quantidade máxima de possíveis atributos do software

A é a representação numérica do grau de satisfação que o atributo oferece as expectativas do usuário, e varia entre 0 e 10 sendo que 0 (zero) significa ausência total do atributo e 10 significa que o atributo satisfaz plenamente as expectativas do usuário e os outros valores desse intervalo correspondem a níveis intermediários de satisfação.

P é a expressão numérica da relevância do atributo, quando comparado aos outros que também podem compor as características do software. O somatório dos P deve ser sempre igual a 10.

Esta variável "Q" a qual chamamos de "medidor" pode ser uma opção para se obter a medida (resultado da avaliação do produto segundo critérios - Rocha (op. cit.)) de qualidade de um software desenvolvido sem especialistas em informática.

Sendo que para diminuir ainda mais a subjetividade poderíamos definir um parâmetro de comparação de modo que ao obter-se a medida de qualidade ("Q") fosse possível saber se a qualidade está próxima do ideal.

Por exemplo, respeitando os mesmos critérios que adotamos para a determinação do "medidor Q" poderíamos definir um parâmetro de comparação.

Esse parâmetro de comparação seria o valor da medida "Q" quando todos atributos do software satisfizessem integralmente as necessidades de seus usuários.

Desta maneira o parâmetro de comparação que chamaríamos de "Qmax" seria determinado da seguinte maneira:

i) todos os atributos satisfazem integralmente as expectativas do usuário então:

$$A_1 = 10, A_2 = 10, \dots, A_n = 10 ;$$

ii) por definição da estruturação do medidor a soma dos indicadores de relevância (pêso) de cada atributo é igual a dez, logo

$$P_1 + P_2 + \dots + P_n = 10$$

iii) a fórmula de determinação do Q_{max} é a mesma que define Q portanto

$$a) \quad Q_{max} = \sum_{i=1}^n A_i \times P_i \Rightarrow$$

$$Q_{max} = A_1 \times P_1 + A_2 \times P_2 + \dots + A_n \times P_n$$

b) de i) vem que $\Rightarrow$

$$\begin{aligned} Q_{max} &= (10 \times P_1) + (10 \times P_2) + \dots + (10 \times P_n) \\ \Rightarrow Q_{max} &= 10 \times (P_1 + P_2 + \dots + P_n) \\ \Rightarrow Q_{max} &= 10 \times (P_1 + P_2 + \dots + P_n) \end{aligned}$$

c) de ii) vem que $\Rightarrow Q_{max} = 10 \times (10)$

$$\Rightarrow Q_{max} = 100$$

Logo, o parâmetro de comparação seria então $Q_{max} = 100$

A relação entre o "Q" medida de qualidade e o " Q_{max} " parâmetro de comparação que representa a qualidade desejada, poderia ser chamada de indicador de qualidade de um software.

Desta forma podemos conceituar indicador de qualidade como um parâmetro que relaciona a medida (resultado da avaliação de um software segundo atributos primitivos possíveis de serem avaliados) de qualidade de um software com a qualidade que se deseja para ele.

Como pelo nosso conceito, Indicador de Qualidade é o parâmetro que mostra a relação entre qualidade desejada e qualidade obtida, entendemos que deva ser ele um referencial para a execução do controle econômico do desenvolvimento de software sem especialistas.

Isto porque a medida da qualidade desejada (a qual não necessariamente precisa ser determinada como no nosso exemplo) deve representar as expectativas de eficiência e eficácia, que são consideradas parâmetros da gestão econômica, da área onde o software é desenvolvido.

Inclusive segundo Gil (2) os indicadores podem ser separados em função de suas características, em famílias.

Sendo que no controle econômico essas famílias (IQ eficácia, IQ eficiência e IQ segurança) poderão receber tratamentos diferenciados.

Para que aconteça a análise de IQ por família é necessário que os atributos que representam as exigências do usuário e que integram o critério de medida da qualidade sejam previamente agrupados por família.

Acreditamos que os fatores que motivam a existência de um software, os quais listamos no item 3.1.1, implicitamente identificam os principais atributos de qualidade, esperados para o programa.

Por exemplo, um software cujo surgimento tenha sido motivado pelo "Tempo exigido para a realização do serviço" deverá ter rapidez como um de seus atributos. Logo o fator motivador trouxe implicitamente a identificação de um atributo necessário.

Destacamos, no exemplo acima um atributo, mas na realidade existem vários outros. Se procedermos a busca a esses outros atributos e se agrupá-los mediante critérios predefinidos, será possível determinar os principais objetivos de qualidade.

Assim sendo podemos dizer que as expectativas relativas aos objetivos de qualidade podem ser previamente definidas em função dos fatores que motivam o desenvolvimento de software.

Entendemos que as famílias (conjunto de atributos afins) de IQ a serem consideradas no controle econômico devem corresponder aos principais objetivos de qualidade determinados pelo fator motivador (vide figura IV.1)

Desta forma os atributos que identificamos no ítem 2.4 e que estavam agrupados de acordo com o objetivo de qualidade (utilizabilidade, confiabilidade de representação e confiabilidade conceitual) podem ser rearranjados de maneira a constituírem base para determinadas famílias de IQ, as quais automaticamente estarão vinculadas a objetivos de qualidade.

4.3. CONTROLE DO DESENVOLVIMENTO DE PROGRAMAS

Procuramos abordar aqui, à luz dos conceitos exibidos no capítulo 2, as necessidades de controle no processo de desenvolvimento de programas de computador pelo usuário final

4.3.1. Objetivo do controle no desenvolvimento de programas

Nesta dissertação estamos procurando mostrar que o desenvolvimento de programas de computador por pessoas não especializadas em informática é uma realidade, presente em inúmeras entidades, que vem tornando-se um evento econômico relevante e principalmente por isso merece um controle específico.

O objetivo que temos ao sugerir o controle desse processo é o mesmo que baliza o exercício do controle em qualquer outra atividade, ou seja, garantir que os efeitos causados pelo desenvolvimento de programas por pessoas não especializadas contribuam com a missão organizacional e conseqüentemente com a continuidade da entidade.

Desta forma, a prática desse controle deve também procurar contribuir para que o desenvolvimento de programas não seja uma atividade concorrente àquelas que são de realização obrigatória da área que se utilizará dos programas.

4.3.2. Tipo de controle em relação à forma de gestão

Entendemos que o processo de desenvolvimento de software pelo usuário final é uma atividade produtiva e como tal deve receber gestão operacional, econômica e financeira .

Assim sendo o controle também deve ser dos tipos: operacional, financeiro e econômico

4.3.2.1. Controle operacional

O controle operacional do desenvolvimento de programas é de responsabilidade do gestor da unidade administrativa onde o desenvolvimento acontece.

Isto porque o desenvolvimento se dá sob seu raio de atuação e por intermédio de recursos humanos e materiais que estão sob sua responsabilidade.

Como dissemos anteriormente, nesta dissertação não será abordado esse tipo de controle.

Entretanto, vale salientar que entre as atividades do controle operacional encontra-se a tomada de decisão sobre a viabilidade técnica do desenvolvimento ou não de programas pelo pessoal da unidade administrativa.

4.3.2.2. Controle financeiro

A responsabilidade pelo controle financeiro do desenvolvimento de programas de computador sem especialistas varia em função do modelo de gestão da entidade.

Entendemos que o ideal é que os resultados do controle financeiro sejam do conhecimento do gestor da entidade onde o desenvolvimento ocorre.

É importante ressaltar que o software é instrumento para apoio à geração de produtos ou serviços de uma determinada área.

Desta forma na maioria das ocasiões o controle financeiro do desenvolvimento de software se dá de maneira indireta, uma vez que a gestão financeira enfoca desembolsos e recebimentos provenientes do produto ou serviço ao qual o software serviu como ferramenta para sua elaboração.

4.3.2.3. Controle econômico

O controle econômico do desenvolvimento de programas de computador deve ser de responsabilidade do gestor da área onde ocorre o desenvolvimento e em algumas circunstâncias deve ter a participação daquele que responde pelo controle econômico da entidade como um todo.

Isso porque acreditamos que o controle econômico do desenvolvimento de programas de computador pelo usuário final deve ser executado com base em análises e confrontos com outros eventos econômicos, e a sua execução deve visar o melhor resultado econômico da entidade como um todo e nunca apenas a área onde ocorre o desenvolvimento.

Dissemos que o controle econômico do desenvolvimento de software deve considerar outros eventos econômicos porque muitas vezes os benefícios proporcionados por um software só podem ser observados com clareza durante a execução ou após o término de outras atividades que representam outros eventos econômicos.

Por exemplo, um programa desenvolvido para apoiar decisões sobre nível adequado de estoque de matéria prima só poderá ter os seus benefícios analisados com precisão após a observação da produção de alguns bens, ou após as primeiras compras para reposição desse item.

Isto porque, imaginemos que o programa tenha feito indicações abaixo das necessidades, nesse caso a entidade pode se ver obrigada a fazer compras (que é um novo evento econômico) em condições que lhes sejam desfavoráveis ou ainda a paralizar ou diminuir o ritmo de produção (que é outro evento econômico).

Entendemos que embora a responsabilidade pelo controle econômico deva ser do gestor da área onde o desenvolvimento acontece em muitas ocasiões não é possível que ele o assuma em sua plenitude por deficiências no modelo de gestão que impede ou dificulta a sua interação com outros eventos econômicos.

Em função desse nosso entendimento julgamos que se faz necessário aqui uma breve abordagem sobre modelo de gestão que viabiliza o apropriado controle econômico do desenvolvimento de software sem especialistas.

Modelo_de_Gestão é entendido por nós como o conjunto de princípios, premissas que servem, como base para a ordenação dos fenômenos (eventos e fatores) da produção de bens/serviços e para o controle de sua eficiência e eficácia.

Desta forma para identificarmos um modelo de gestão nos parece suficiente que enunciemos os princípios e premissas que constituem a sua base para ordenação dos fenômenos da produção.

Assim sendo listaremos a seguir alguns princípios e premissas cuja existência julgamos necessária para a execução do controle econômico de desenvolvimento de software.

MODELO DE GESTÃO

ASPECTOS	PRINCÍPIOS E PREMISSAS
LIDERANÇA	<ul style="list-style-type: none"> . deve haver aceitação organizada e coordenada de manifestações de opinião de níveis de subordinação inferior; . deve ser conhecido por parte dos gestores o grau de confiança que pode ser atribuído a cada profissional bem como o perfil de cada um; . delegação de autoridade e responsabilidade, e . existe organograma preestabelecido mesmo que informalmente.

ASPECTOS	PRINCÍPIOS E PREMISSAS
FORMA DE COMUNICAÇÃO	<ul style="list-style-type: none"> . a forma de comunicação deve pressupor a necessidade de interação do receptor com o emissor; . o fluxo de comunicação (sempre que possível) deve percorrer os caminhos naturais do organograma; e . existe sistema de comunicação e informação.
FORMA DE MOTIVAÇÃO	<ul style="list-style-type: none"> . o processo de motivação deve prever esquemas de recompensas e abandonar a ameaça explícita; e . existem sistemas para avaliação e medição de performance que refletem o nível de motivação dos profissionais.
PROCESSO DE TOMADA DE DECISÕES	<ul style="list-style-type: none"> . o processo de tomada de decisões deve ser alimentado por informações tecnicamente identificadas e mensuradas; . deve prever o envolvimento de níveis hierarquicamente inferiores, no mínimo como fornecedores de informações e opiniões; e . deve considerar o efeito das decisões nos aspectos motivacionais.

ASPECTOS	PRINCÍPIOS E PREMISSAS
PROCESSO DE TOMADA DE DECISÃO	<ul style="list-style-type: none"> . deve ser ágil e objetivo; e . modelo de tomada de decisões da entidade é claramente definido e de conhecimento geral.
UTILIZAÇÃO DE RECURSOS	<ul style="list-style-type: none"> . preocupa-se com produtividade e com otimização de recursos disponíveis tanto humano quanto material, bem como com a qualidade dos serviços que oferece à entidade; e . existem parâmetros para avaliar a qualidade dos serviços prestados.
FORMA DE DEFINIR OBJETIVOS	<ul style="list-style-type: none"> . considera sempre os objetivos, crenças e valores da entidade como um todo; . considera a missão da entidade; . considera a realidade atual e perspectivas da entidade e do meio externo; . é sustentado pela análise de viabilidade.

ASPECTOS	PRINCÍPIOS E PREMISSAS
FORMA DE DEFINIR OBJETIVOS	<ul style="list-style-type: none"> . contempla formas específicas para comunicá-los aos envolvidos direta e indiretamente na tarefa de atingí-los; . envolve níveis inferiores de subordinação no estabelecimento da forma de operacioná-los; e . crenças e valores da entidade estão claramente definidos.
CONTROLE	<ul style="list-style-type: none"> . as informações de controle são utilizadas na definição de objetivos; . as ações de controle são sempre voltados para o futuro; . existe sistema de controle com modelo de decisões previamente definido; . existe preocupação com a sua aceitação; e . existe definição prévia dos parâmetros a serem indicados e mensurados para efeito de controle.

4.4. COMO PODE ACONTECER O CONTROLE ECONÔMICO DO DESENVOLVIMENTO DE SOFTWARE SEM ESPECIALISTAS.

Parece-nos que o controle econômico de desenvolvimento de software sem especialistas pode e deve acontecer da mesma maneira que ocorre o controle de outras atividades que geram produtos/serviços, ou seja, por intermédio de: um sistema de informações, método de custeio, padrões, orçamentos, plano de contas e um modelo de tomada de decisões.

4.4.1. A participação do sistema de informações no controle do desenvolvimento de software sem especialistas.

4.4.1.1. Identificação e mensuração

Como dissemos no capítulo 2 a identificação de um evento econômico consiste basicamente em comparar eventos ocorridos com um plano de contas que indique os eventos pertinentes a cada produto ou serviço.

Desta forma é necessário que haja um plano de contas que contemple a identificação dos possíveis eventos pertinentes ao desenvolvimento de software.

A mensuração dos eventos pertinentes ao desenvolvimento de software, pode ocorrer dentro dos mesmos critérios e procedimentos que são adotados para atribuir valores a eventos ligados a atividade principal da área onde o desenvolvimento ocorre.

4.4.1.2. Relato de informações

Os resultados do subsistema de relato de informações são obtidos por intermédio de dois outros subsistemas que o constituem, o de acumulação e o de avaliação de Custos e Receitas.

a) Subsistema de acumulação

O método de acumulação de custos e receitas a ser adotado para o desenvolvimento de software pode ser escolhido independentemente do método adotado pela entidade para as acumulações referentes aos seus produtos/serviços.

O software é uma ferramenta de apoio às atividades da entidade e como tal os seus custos e receitas são parte integrante de outros produtos/serviços da entidade.

Isto faz com que, na acumulação feita pela entidade como um todo, só apareçam os valores finais dos softwares, sem a preocupação de qual foi o método adotado para acumulação de receitas e custos do seu desenvolvimento.

Vejamos, se a entidade adota a acumulação por processo lhe interessa saber quais valores referentes aos softwares desenvolvidos devem ser apropriados a cada departamento ou

centro de custos; se o método de acumulação não é este, então o que se precisa são os valores referentes ao software que devem ser apropriados ao produto.

Em ambos os casos não surgiu a necessidade de saber qual o método de acumulação utilizado para se informar o valor do software.

Quando uma determinada área decide desenvolver o seu próprio software esta adotando postura semelhante à de uma empresa que resolve construir uma determinada máquina que usa na fabricação de um certo produto.

A construção dessa máquina representa um novo processo produtivo para a entidade.

Pensamos que a sistemática de contabilização de custos e receitas é determinada fundamentalmente pelas características do processo de produção.

Assim sendo, o processo de construção da máquina pode utilizar-se de sistemática de contabilização distinta daquela adotada para tratamento de custos e receitas do produto que ela (a máquina) vai ajudar a gerar.

A independência da sistemática de contabilização pode ser vista com maior clareza quando se considera que ao invés de fabricar a máquina a empresa a adquire.

Nesse caso não interessa ao sistema de acumulação de custos e receitas da entidade qual foi o método adotado para acumulação de custos e receitas da máquina, mas sim o valor total que desembolsou na sua aquisição.

O software desenvolvido numa determinada área da entidade encontra-se numa posição muito semelhante a da máquina construída pela entidade, quando o assunto é compatibilidade entre o método de acumulação utilizado para o software e o utilizado para os produtos/serviços da entidade.

Entendemos que a acumulação de custos e receitas de softwares desenvolvidos sem especialistas, portanto nas diversas áreas da empresa, deve respeitar os critérios e procedimentos do método "por ordem de produção".

Sendo que cada área abre a sua própria "OP" toda ocasião em que se inicia desenvolvimento de um software.

b) subsistema de avaliação

O subsistema de acumulação deve remeter ao subsistema de avaliação os valores gerados durante o processo de

desenvolvimento de software em cada área e esse sistema deve atribuir valores aos mesmos.

Nossa opinião é que não existe obrigatoriedade de serem idênticos os métodos de avaliação da entidade e os do desenvolvimento de software e justificamos esta posição com a mesma argumentação que utilizamos para defender a independência dos métodos de acumulação.

Entendemos que a avaliação dos softwares desenvolvidos pelas diversas áreas da organização deve ser pelo método de valores predeterminados.

Não somos partidários da adoção da avaliação a valores reais porque esse método consiste em atribuir receitas e custos dos produtos e serviços na medida que o subsistema de acumulação lhe fornece os valores, isto sem análise da eficiência, segregando apenas as ocorrências que representam grande anormalidade. Entendemos que esse método dificulta a atuação do controle em busca da eficiência do processo de desenvolvimento de software.

O método de valores predeterminados contribui fortemente com o controle antecedente na medida em que informa as receitas e os custos antes, mesmo que eles ocorram. Além de fornecer parâmetros para análise de valores incorridos, o que o torna

também um instrumento dos controles concomitante e sucedente.

Desta forma opinamos que a avaliação de custos e receitas deve ocorrer com base em padrões.

4.4.1.3. Padrões

Os padrões podem ser de três tipos - Teórico, Padrões passado e ideal atingível.

Padrão Teórico considera que as atividades acontecerão em condições perfeitas para o desenvolvimento de softwares ou seja sob as melhores condições de operação. Isto seria uma utopia para a entidade caso ela se encontrasse nos estágios preliminares de cultura em informática.

Na realidade questionamos "as condições perfeitas" até mesmo nas entidades que se encontram em estágio de "consolidação" da cultura de informática. Entretanto em função do modelo de gestão da entidade e do estágio da cultura pode ser que esse tipo de padrão seja uma opção adequada.

Padrões Passado aponta os valores que a organização espera incorrer referenciando-se na média de eventos passados ajustados com base em novas expectativas.

Considerando-se que é meta da maioria das entidades estar em constante evolução avançando dinamicamente rumo ao estágio de "consolidação", então pode se dizer que a cada momento a

organização deveria estar mais habilitada a apresentar maior eficiência no desenvolvimento de softwares sem especialistas.

Assim sendo os padrões passados podem não ser o tipo mais adequado uma vez que adotando acontecimentos do passado como referencial, a entidade esta de certa forma permitindo influências de marcas de eficiência (ou ineficiência) obtidas num dado momento em que estava menos habilitada para a elaboração de programas.

Entretanto em função do modelo de gestão da entidade esse tipo de padrão pode ser recomendado, exceto se esta encontrar-se no estágio "embrionário". Pois entendemos que padrões passados não são uteis em hipótese alguma às organizações que estão se iniciando em informática.

Padrão_ideal_atiugível indica os valores que a entidade espera incorrer considerando condições eficientes de operação.

A determinação desse tipo de Padrão não é trabalho fácil principalmente no que diz respeito à definição do que vem a ser "condições eficientes de operação".

Mas apesar dessa dificuldade entendemos que é o tipo mais adequado para o controle do desenvolvimento de softwares sem

especialistas, uma vez que representa a busca dinâmica da eficiência dentro de determinadas circunstâncias, assim sendo sugerimos que no estabelecimento de padrões, para o desenvolvimento de software se utilize os conceitos inerentes ao ideal atingível, respeitando-se o modelo de gestão e considerando também o estágio de cultura em informática que prevalece na entidade.

Deve respeitar o estágio de cultura porque este tipo de padrão deve representar as receitas e custos que deveriam ocorrer sob condições eficientes de operação. E o entendimento de "condições eficientes de operação" depende muito da cultura e conhecimento sobre o que está sendo produzido.

Desta forma acreditamos que para a eficiência do processo de desenvolvimento de software é necessário que se tenha pleno domínio da interpretação do que vem a ser "condições eficientes" em cada tempo e lugar (pensamos que as condições eficientes são mutáveis em função do tempo e do lugar) pois isto se expressa no controle por intermédio dos padrões.

Esta interdependência de domínio de interpretações sobre "condições eficientes" e uso de padrões no controle traz amadurecimento significativo ao processo de desenvolvimento de software. E isto contribui com o aumento de produtividade administrativa da entidade como um todo.

Pois como já afirmamos, acreditamos firmemente que o desenvolvimento de software sem especialistas é uma forma de se obter aumento de produtividade administrativa (o que pode acontecer por intermédio do software) sem aumento dos custos da entidade e com eficácia (isto porque o "dono do problema" reúne melhores condições para desenvolver o software na época e no prazo mais compatível com as necessidades). Entretanto esta nossa afirmação só é verdadeira se houver eficiência no processo de desenvolvimento de software.

Deve ser observado também no estabelecimento de padrões o tipo de software (permanente ou transitório ou intermitente) que será desenvolvido.

Por exemplo, tendo como base um modelo de gestão semelhante ao que identificamos no item 4.3.2.3, sugerimos que se a entidade estiver no estágio embrionário o padrão de custos do processo de desenvolvimento e uso de software transitório, seja estabelecido considerando que as funções do novo programa fosse realizada em condições eficientes de operação, pelo processo que antecedeu a sua elaboração.

Esta sugestão embasa-se nas seguintes premissas:

- a entidade em estágio embrionário terá dificuldade em identificar e/ou orçar os trabalhos de desenvolvimento de software,
- estabelecer padrões com base em preço de mercado (que seria outra opção) exigiria que se tivesse conhecimento técnico adequado para definir similaridade de software; e
- como o software será transitório não terá outras oportunidades para amortizar os seus gastos de desenvolvimento e uso, portanto o ideal é que os mesmos se equiparem aos desembolsos pelo caminho alternativo.

É importante salientar aqui, que os eventuais desvios deverão ser analisados com cuidado, pois embora os custos do desenvolvimento e uso do software possam ser superiores ao método opcional podem haver benefícios indiretos a favor do software.

Acreditamos que o tipo de padrão a ser adotado deverá variar de acordo com o modelo de gestão e o estágio cultural de informática. Assim sendo a cada nova fase da empresa na informática, será necessário a análise dos três tipos de padrão - Teórico, ideal atingível e padrões passado para verificar qual o que melhor se adapta a nova situação cultural podendo essa análise ser feita com reflexões semelhantes à que adotamos no exemplo mencionado acima.

4.4.1.4. Orçamentos

Diz Dearden (3) que "para os objetivos de controle orçamentário é necessário separar os orçamentos de despesas em dois tipos - operações e programados".

"Os orçamentos de operações cobrem aquelas despesas para as quais é possível estabelecer padrões de eficiência".

"Os orçamentos programados cobrem aquelas despesas que estão sujeitas à decisão da administração num grau razoável. Cobrem principalmente custos de pesquisa e desenvolvimento".

"O controle de custos é exercido exclusivamente pela decisão do tamanho do trabalho a ser feito".

Entendemos que para o controle do processo de desenvolvimento de softwares sem especialistas, a unidade administrativa que o elaborará pode utilizar-se de orçamentos como instrumento de controle antecipado.

Sendo que este pode (a rigor deveria) ser parte integrante do orçamento geral da entidade.

É importante salientar que o orçamento ao qual nos referimos aqui não deve ser utilizado pelo subsistema de avaliação de custos e receitas. Como já opinamos para a avaliação de

custos a valores predeterminados julgamos o padrão como o referencial mais adequado.

Os "orçamentos de operações" para o desenvolvimento de software, na medida do possível, devem ser estabelecidos com base em padrões.

Os softwares permanentes são os que oferecem maior facilidade para o uso de orçamentos de operações.

Desta forma acreditamos que as entidades deveriam periodicamente fazer orçamentos de operações para os softwares permanentes e para o mesmo período deveriam elaborar orçamentos programados para os softwares transitórios e intermitentes.

Ou seja, a cada período as unidades administrativas orçariam com precisão, considerando eficiência, o quanto gastarão com cada software permanente que desenvolverá e, como se faz com as pesquisas de uma entidade, indicariam um valor que corresponderia ao tamanho total do trabalho correspondente ao desenvolvimento de softwares intermitentes e transitórios, sendo que a aprovação desse valor seria de competência da alta administração.

Acreditamos que a existência de orçamentos para a elaboração de softwares viabiliza a otimização dos seus processos de

desenvolvimento, pois por intermédio desses torna-se possível a identificação de programas com objetivos idênticos que seriam desenvolvidos por duas ou mais áreas.

A utilização de orçamentos contribui também com o amadurecimento da entidade no uso da informática, na medida em que provoca reflexões sobre o estágio de cultura da empresa num dado momento e num período futuro. Pois ao se estabelecer orçamentos de desenvolvimento de software automaticamente está se tentando responder à questão "o que a entidade tem hoje e o que precisa e pode ter no período seguinte?".

O uso de orçamentos é possível e viável em qualquer estágio de cultura em informática. Inclusive acreditamos, que um dos primeiros passos para a elaboração de Plano Diretor de informática é a utilização de orçamentos para o desenvolvimento de softwares.

4.4.1.5. Método de custeio

Existem dois métodos de custeio: o custeio variável e o custeio por absorção.

O custeio variável apropria ao produto somente os custos empregados no processo produtivo e que podem ser a ele relacionados.

O custeio__por__absorção apropriado ao produto todos os custos de produção quer tenham sido gerados ou não por esse produto.

Consideramos o custeio variável como o mais adequado para a apropriação dos custos do desenvolvimento de software sem especialistas.

Pois para efeito de controle o custeio variável é o mais adequado.

Se, por exemplo, numa determinada área estiver ocorrendo o desenvolvimento de dois ou mais softwares com níveis de complexidade e de exigências distintas e os seus desenvolvedores também possuírem significativa diferença de conhecimento técnico, com o custeio por absorção seria difícil determinar o custo real do desenvolvimento de cada um dos softwares, isto pela adversidade de condições e características.

4.4.2. Modelo de tomada de decisões

O aspecto mais delicado do desenvolvimento de software sem especialistas prende-se ao risco da falta de qualidade proporcionar decisões ou julgamentos errôneos.

Desta forma entendemos que é de fundamental importância que o controle contemple parâmetros para o direcionamento de ações que previamente minimizem o risco da qualidade do software torna-lo um agente econômico danoso à entidade.

Por isto pensamos que o modelo de tomada de decisões que atua como ferramenta do controle deve ser constituído de tal forma que englobe explicitamente fatores de qualidade.

Além da preocupação com fatores de qualidade, enquanto agentes econômicos, o modelo de tomada de decisões deve conter princípios e premissas que estabeleçam critérios de análise de variações, de maneira que sejam ponderados os benefícios futuros expressos nas avaliações dos eventos ligados ao desenvolvimento de software.

4.4.2.1. Indicadores de Qualidade como parâmetros do modelo de tomada de decisões

Pela nossa interpretação modelo de tomada de decisões é o conjunto de princípios e premissas que servem de base para a formação de um raciocínio e opinião, e indicador de qualidade é o parâmetro que expressa a relação entre a qualidade obtida e a qualidade desejada.

Com base nesses entendimentos parece-nos coerente que indicadores de qualidade sejam considerados parâmetros de um modelo de tomada de decisões econômicas, pois contribuem

para a formação de um raciocínio e opinião sobre a qualidade (que é um agente econômico).

4.4.2.2. Uma Proposta de modelo de tomada de decisões

Para motivação de reflexões, fazemos aqui uma proposta de modelo de tomada de decisões relativas ao desenvolvimento de software sem especialistas.

Listamos a seguir os componentes do modelo de tomada de decisões e respectivos princípios e premissas.

a) Medida de qualidade

Podem ser utilizados os atributos identificados no capítulo 2 como critérios de medida da qualidade.

Entretanto antes de se efetuar a medida é importante que os atributos sejam agrupados de acordo com os IQ que se pretende obter.

b) indicadores de qualidade

Devem ser determinadas previamente as famílias de IQ que se deseja analisar.

A nossa sugestão é que se analise IQ de eficiência, eficácia e de segurança além de se analisar os objetivos de qualidade, confiabilidade conceitual, de representação e utilizabilidade.

c) Prevenção

Como o controle, dentro do possível deve ser preventivo então deverão ser medidos os atributos da especificação e do projeto.

Isto contribuirá com a garantia da qualidade do software durante o seu desenvolvimento.

d) influências ambientais

Sugerimos que sejam identificados fatores redutores de qualidade como forma de considerar influências negativas provenientes de características culturais do ambiente onde é desenvolvido.

Estes fatores devem variar de zero a um e devem ser multiplicados ao medidor de qualidade "Q".

Por exemplo, se obtivermos um $Q = 80$ e considerarmos que o ambiente onde o software foi desenvolvido tem cultura em informática que reúne apenas 70% das condições consideradas ideais, então o nosso fator redutor será 0,7.

Logo o valor a ser considerado para o medidor deve ser

$$Q = 80 \times 0,7 \Rightarrow Q = 56$$

Por intermédio de entrevistas e análise de aproximadamente 350 programas, foi possível traçar analogia que permite ilustrar essa abordagem com os seguintes fatores de redução.

	máximo	mínimo
Estágio Embrionário	0.6	0.3
Despertar da razão	0.8	0.4
Definição de identidade	0.8	0.6
Estágio da convergência	0.9	0.7
Estágio da consolidação	1.0	0.8

e) influências do desenvolvedor

Entendemos que as características do desenvolvedor influenciam significativamente a qualidade do programa, podendo inclusive compensar algumas influências negativas do ambiente, assim sendo, é conveniente identificar fatores corretivos, os quais podem reduzir ou aumentar a medida de qualidade, como maneira de considerar o domínio técnico do desenvolvedor.

A identificação desse fator dependerá do conhecimento que se tem das características técnicas do desenvolvedor.

Deve-se ter um cuidado muito grande para não efetuar julgamentos sobre a pessoa em lugar de se julgar o seu potencial em informática.

Apresentamos aqui, também, um referencial para esses índices.

Profissionais da área usuária		
	máximo	mínimo
! conhecimento rudimentares de informática	0.8	0.5
! conhecimentos razoáveis	1.1	0.7
! profundos conhecimentos	1.3	0.9
profissionais com formação em informática (que ficam lotados na área usuária)		
! já atuaram na área de informática	1.5	1.0
! nunca atuaram na área de informática	1.1	0.8

f) influências do motivador do desenvolvimento

Os motivos que geram a necessidade de um software também devem ser representados por fatores de redução ou de acréscimo na medida de qualidade.

Vejamos a nossa sugestão para esse fator:

	máximo	mínimo
. nível de complexidade de serviço	0.9	0.8
. tempo exigido para realização do serviço	0.8	0.7
. Quantidade de dados a serem manipulados	1.0	0.9
. repetitividade de tarefas	1.2	0.8
. conjugação de fatores	0.7	0.6

g) Análise de custos e receitas

A análise de desvio de custos e receitas deve ocorrer por intermédio da comparação entre os valores que ocorreram e os padrões

Os critérios de análise devem ser os mesmos que valem para interpretação dos resultados da atividade principal da área onde o software foi desenvolvido. Entretanto deve existir um cuidado especial com o reconhecimento e apropriação de benefícios indiretos gerados pela existência do novo software.

É importante ressaltar que os valores de custos de software permanentes não podem e não devem ser apropriados exclusivamente a uma atividade ou produto.

Pois não é correto que somente uma atividade/produto pague por um software que será usufruído por outras atividades.

h) Constituição de padrões

Antes de se definir, as referências que constituirão os padrões é importante que se defina explicitamente o que são "condições eficientes de operação" para que se saiba a qualquer momento e a posteriori quais são as metas de eficiência esperadas.

4.4.2.3. Considerações adicionais sobre o modelo de decisões proposto

Na proposta de modelo de decisão não tivemos o desejo de exaustar o assunto mas apenas apontar referências para maiores reflexões, as quais podem continuar por intermédio de análise mais profundas sobre as características da entidade, dos possíveis desenvolvedores e usuários do software e do próprio programa que se deseja, sem se esquecer de considerar o modelo de gestão da entidade.

É importante salientar que os fatores que sugerimos no subitem anterior tiveram origem em entrevistas e análise de aproximadamente 350 softwares durante a elaboração desta dissertação, entretanto devem ser vistos apenas como referencial, pois cada entidade tem suas peculiaridades e dinâmica que justificam reflexões particularizadas.

4.5. A CONTROLADORIA E O DESENVOLVIMENTO DE PROGRAMAS PELO USUÁRIO FINAL.

Todos os recursos de uma determinada entidade devem ser utilizados de forma a contribuir com os objetivos estabelecidos para garantir a continuidade de sua existência.

Uma das funções da controladoria enquanto órgão administrativo é gerar subsídios para a identificação dos esforços individuais e divulgá-los como forma de mantê-los harmônicos, em relação ao conjunto, e convergentes, em relação aos objetivos da entidade.

Enquanto ramo do saber, possui subsídios teóricos para viabilizar essa identificação e divulgação.

Principalmente por esse motivo acreditamos que aqueles que praticam a controladoria dentro da entidade possuem excelentes condições para assessorar, subsidiar e até mesmo orientar o controle econômico do processo de desenvolvimento de programas de computador por pessoas não especializadas em informática, de modo a garantir que cada uma das áreas da entidade, além de atingir com o desenvolvimento de seus programas os objetivos que lhe são peculiares, não se desvie daquilo que lhe é estabelecido como parcela de contribuição para a entidade e nem tão pouco perca o sincronismo com o conjunto.

4.5.1. Forma de atuação da controladoria

Kanitz (4) diz que "é função básica do controlador implantar os sistemas de informações - sistemas contábeis e financeiros, sistemas de pagamento e recebimento, Folha de pagamento, etc".

Acreditamos que o nosso pensamento sobre a controladoria e o sistema de informações esta em concordância com o autor pois julgamos que é função da controladoria projetar, implantar e manter a contabilidade de custos, que pode ser vista como um sistema de informações.

Assim sendo o sistema de informações é o instrumento que viabiliza e materializa a contribuição da controladoria aos diversos gestores para o controle do processo de desenvolvimento de software sem especialistas em informática que acontece em suas respectivas áreas.

Acreditamos que a controladoria pode ajudar na geração de informações sobre desenvolvimento de sistemas pelo usuário final de forma a garantir melhores resultados econômicos nas atividades de desenvolvimento e uso de programas para computador.

Cabe, também à controladoria manter os gestores informados sobre as consequências econômicas que podem ser ou que foram causadas à entidade como um todo pelo desenvolvimento e uso de sistemas de computador em sua área.

A controladoria deve assessorar os diversos gestores na função de análise dos benefícios econômicos que as inovações tecnológicas, a nível de informática, podem trazer às suas respectivas áreas.

Assim sendo, a controladoria deve conhecer os objetivos definidos pela entidade para cada uma das áreas e manter-se a par, por intermédio do sistema de informação dos acontecimentos ligados ao uso da informática .

A controladoria, na medida do possível, deve estimular as diversas áreas da entidade a divulgarem entre si suas realizações e forma de utilização de recursos computacionais.

Esse estímulo à divulgação e discussão deve ocorrer de forma a não se caracterizar como intervenção. Pois as áreas que estão se introduzindo na informática têm consciência de suas fragilidades e normalmente não aceitam se expor e qualquer aproximação feita sem o devido cuidado toma a aparência de intervenção e crítica.

Para garantir o não intervencionismo, a argumentação da controladoria para a divulgação e discussão sobre realizações e utilizações de recursos computacionais para cada uma das áreas deve residir exclusivamente no aspecto econômico que afinal é onde realmente devem estar as preocupações e interesses maiores da controladoria enquanto promotora de sinergia.

As discussões sobre aspectos técnicos das opções de recursos computacionais devem ficar no âmbito das próprias áreas e, quando necessário, se estender à área de informática, por intermédio dos gestores das respectivas áreas.

A controladoria só deve participar dessas discussões para esclarecer as consequências econômicas e oferecer subsídios, também econômicos, para apoio às decisões.

Deve assessorar no estabelecimento de padrões e orçamentos e se necessário na elaboração do modelo de tomada de decisões.

A controladoria deve estimular as diversas áreas da entidade a estudarem a alternativa de buscarem em mercado ferramentas de informática que possam satisfazer simultaneamente as necessidades de diversas áreas da entidade, fazendo com que a entidade como um todo se torne mais competitiva e eficiente e se mantenha dentro de sua estratégia de conduta dos seu negócios com os menores custos.

A controladoria não pode perder a visão do todo, deve sem intervenções (isto cabe exclusivamente aos executivos maiores da entidade) mostrar às diversas áreas da entidade que nem sempre o ótimo para entidade é a soma dos ótimos de cada área.

Desta forma, em algumas circunstâncias pode ser economicamente melhor para a entidade, que determinadas áreas não desenvolvam os seus próprios programas mas participem da aquisição de um software de mercado ou até mesmo de software elaborado ou a ser desenvolvido pela área de informática da própria entidade.

Para participar de discussões sobre ferramentas de informática que servem a diversas áreas a controladoria deve manter-se envolvida com o planejamento estratégico da entidade, acompanhando as suas mudanças, entendendo que espécie de infra-estrutura tecnológica esse plano necessita.

Deve, na medida do possível estar informada sobre as necessidades dos clientes, no que diz respeito ao produto que a entidade lhes oferece, observando também os recursos tecnológicos que esses mesmos clientes estão utilizando.

Nas entidades em que o uso da informática já está massificado e o desenvolvimento de softwares sem especialistas é rotina, deve ser preocupação da controladoria a divulgação dos impactos que o uso recursos

de informática tem proporcionado e que contribuição tem trazido à eficiência da gestão empresarial e aos rumos dos negócios da entidade.

4.5.2. A controladoria e o estágio da informatização

Pensamos que a informática é uma ferramenta para apoio à eficiência e eficácia e conseqüentemente à produtividade, que é combinação dos primeiros.

Se este nosso pensamento for verdadeiro então as empresas podem tornarem-se mais competitivas por intermédio da informática.

Seguindo essa linha de raciocínio concluímos que o estágio de informatização de uma entidade pode influenciar a sua competitividade.

Não estando totalmente errado o que foi expresso acima, seria interessante que a controladoria soubesse a cada momento em qual estágio cultural a entidade se encontra em relação a informática. Isto para que a sua interpretação de ameaças e oportunidades em relação ao âmbito externo à entidade não deixasse de considerar esse aspecto.

Além disso, a controladoria tem como função a assessoria e orientação às diversas áreas na elaboração de padrões (diversos autores, entre os quais se encontram Hecker & Wilson (op. cit.), incluem nas funções da controladoria o estabelecimento de padrões), e como já mencionamos no item 4.4, acreditamos que o padrão deve considerar as condições ambientais.

Desta forma, também por isto a controladoria deve conhecer o estágio de cultura da entidade em relação a informática.

Torna-se questionável a eficiência e eficácia das sugestões da controladoria para os padrões relacionados ao desenvolvimento de software se ela não tiver uma visão exata do estágio cultural da informática.

Falando-se por analogia, as atitudes da controladoria em relação ao uso de recursos computacionais sem o conhecimento do estágio em que se encontra a entidade assemelham-se às atitudes de um médico que prescreve receitas exclusivamente em função dos sintomas, sem a definição de um diagnóstico.

O médico que assim procede poderá ocasionalmente curar o paciente, mas corre o risco de aliviar os sintomas sem eliminar a causa, pois a ocultou de forma que quando se manifestar novamente pode ser fatal.

Como já dissemos anteriormente, acreditamos que o controle pressupõe o conhecimento completo da situação, assim, uma entidade não terá controle sobre o seu processo de informatização se não tiver consciência plena do estágio em que se encontra.

A controladoria, enquanto participante da definição de padrões economicamente adequados, tem inclusive condições de influenciar a velocidade pela qual a entidade passa pelos estágios de informatização. Pois padrões audaciosos porém atingíveis podem ser estimuladores de eficiência e de acultramento mais veloz.

4.5.3. Relacionamento entre aqueles que exercem a controladoria e as diversas áreas que desenvolvem os seus próprios programas

4.5.3.1. Comunicação de fatos

Toda vez que ocorrer algum desvio nas expectativas de desenvolvimento de resultados do processo de desenvolvimento de programas e a controladoria identificar tal fato deve comunicar formalmente o gestor da área.

Todas as comunicações devem ser acompanhadas de explicações sobre as conclusões e eventualmente de indicação de novos parâmetros econômicos que, se respeitados, permitirão que os resultados finais sejam atingidos.

4.5.3.2. Participação na tomada de decisões

É desejável que a controladoria conheça o modelo de tomada de decisão, de cada área, referente ao desenvolvimento de programas por pessoas não especializadas em informática, para que possa ajudar, por intermédio do sistema de informações, na geração de informações claras, objetivas e, sempre que possível, mensuráveis para a tomada de decisão de cada gestor.

4.5.3.3. Participação na definição de objetivos do desenvolvimento de programas

A controladoria deve estimular as discussões, entre as áreas, sobre o processo de desenvolvimento de programas.

Deve também motivar as diversas áreas a refletirem, analisarem e verificarem se os objetivos do desenvolvimento de seus programas convergem para resultados positivos na entidade.

4.5.4. Produto do trabalho da controladoria

Espera-se que o trabalho da controladoria em relação ao controle econômico do processo de desenvolvimento de software pelo usuário final gere concretamente os seguintes produtos:

- Organização e manutenção eficiente do sistema de informações;
- Divulgação de informações precisas e oportunas sobre o desenvolvimento de programas por intermédio de manutenção e organização eficiente e adequada do sistema de informações;
- Assessoria aos gestores para interpretação econômica das informações referentes ao desenvolvimento de software em suas respectivas áreas;
- Assessoria para aplicação de ações corretivas imediatamente após a detecção de desvios entre os planos e o que efetivamente se tenha realizado; e
- Assessoria para a definição de padrões.

FIGURAS DO CAPÍTULO 4

USO DO DESENVOLVIMENTO DE SOFTWARE E DA GARANTIA E CONTROLE DA QUALIDADE

FIGURA IV-1

CITAÇÕES DO CAPÍTULO 4

CITAÇÕES DO CAPÍTULO 4

- (1) CULTURAL BRASIL Dicionário Brasileiro da Língua Portuguesa
Cultural Brasil

- (2) GIL, A.L. A atuação da auditoria de sistemas computadorizados para obtenção de uma maior produtividade no processamento eletrônico de dados (Tese de doutorado)
FEA/USP - SP - 1984

- (3) DEARDEN, John Análise de custos nas empresas
Zahar - 1976

- (4) KANITZ, Stephen C. Controladoria: Teoria e estudos de casos
Livraria Pioneira - SP - 1976

CAPÍTULO 5 - CONCLUSÃO

5. CONCLUSÃO

Acreditamos que o desenvolvimento de software sem especialistas será em breve uma rotina em todos os tipos e tamanhos de entidades.

Interpretamos nos dados de comercialização do hardware e software que este procedimento tem sido fortalecido nos últimos quatro anos e mantém a perspectiva de crescer ainda mais.

Mas a falta de qualidade do software proveniente desta forma de desenvolvimento pode trazer prejuízos à entidade onde ele foi elaborado.

Desta maneira os gestores das áreas que optam por esta forma de desenvolvimento precisam estar atentos ao aspecto qualidade.

Entretanto esses gestores não precisam tornarem-se especialistas em informática para evitar elaboração de softwares sem qualidade. Entendemos que devem simplesmente estabelecerem parâmetros (atributos, objetivos e famílias de IQ) de qualidade a serem analisados periodicamente.

Para o estabelecimento desses parâmetros é suficiente se postarem como usuário de computação que desejam uma ferramenta adequada para apoio à realização das atividades da sua área.

Como cada gestor conhece bem as atividades desenvolvidas sob sua responsabilidade então tem condições de determinar quais atributos deve ter uma ferramenta para auxiliá-las.

Então cada um deles ao decidir que será desenvolvido software em sua área providenciará a definição das características que esse deverá possuir.

Uma vez definida as características exatas do software a ser elaborado deve dar início ao controle do processo de desenvolvimento de software.

5.1. INFORMAÇÕES NECESSÁRIAS PARA EXECUÇÃO DO CONTROLE

Para controlar o processo de desenvolvimento de software é proposta desta dissertação que o gestor da área saiba previamente:

a - Fatores que geraram a necessidade do software

- . nível de complexidade do serviço ou
- . tempo exigido para realização do serviço ou
- . quantidade de dados a serem manipulados ou
- . repetitividade de tarefas e/ou
- . conjugação de fatores.

b - Frequência de uso do software

- . Permanentes ou
- . Transitórios ou

- Intermitentes.

c - Quem utilizará o software

- Profissionais da área ou
- Profissionais de informática ou
- utilização conjunta ou
- Profissionais de outra área.

d - Características do local onde acontecerá o desenvolvimento

- estágio embrionário ou
- despertar de razão ou
- definição de uma identidade ou
- estágio da convergência ou
- estágio da consolidação.

e - Quem desenvolverá o software

- Profissionais da área e/ou
- pessoas com formação em informática.

Além das informações acima é importante que o gestor saiba qual é o plano de desenvolvimento, conhecendo as etapas previstas e os respectivos prazos e ainda as previsões de gastos com homem-hora e horas de computador, o que poderá constar num orçamento de operações.

O gestor da área que realizará o desenvolvimento, com base na determinação que fez dos atributos que o software deverá ter e conhecendo as informações que listamos acima terá condições de efetuar o controle econômico do processo de elaboração de software.

Sugerimos também que para o controle antecipado do desenvolvimento se tenha uma identificação da qualidade de outros programas desenvolvidos sem especialistas dentro da entidade.

Essa identificação pode ser como a que fizemos no item 3.6.3 ou seja, composta pela lista dos atributos que devem estar presentes nas características de um software, para que ele possa ser considerado de qualidade adequada, e para cada software já elaborado se diria quais dos atributos realmente estão incorporados no seu perfil.

A partir da análise de cada um dos softwares já elaborados é possível que se conheça os problemas mais comuns na qualidade de programas desenvolvidos na organização.

Essa análise de qualidade dos softwares já desenvolvidos é uma forma de controle antecedente.

Para viabilizar esse controle antecipado é necessário que se inclua na rotina do controle sucedente o mapeamento da

qualidade do software obtido e se integre esse mapa numa estatística geral da entidade sobre qualidade dos softwares que tem elaborado.

Desta maneira, a entidade poderia manter uma estatística dinâmica sobre a qualidade dos softwares que tem desenvolvido, o que serviria para orientar novos desenvolvimentos mostrando onde estão os pontos fortes e fracos.

Além de servir de orientação para novos desenvolvimentos, esses mapas mostrariam à entidade quais os riscos que têm corrido com a elaboração descentralizada e sem especialistas em informática.

Os gestores poderiam também se utilizar desta identificação de riscos que a entidade tem corrido como referencial de sua possível vulnerabilidade e incluí-los como IQ a ser verificado pelo seu modelo de decisão.

Independentemente de considerar esse IQ, o gestor deve também estabelecer outros, em função dos motivos que geraram o desenvolvimento do software, em função das características do ambiente e do tipo de usuário e em função do desenvolvedor.

Este trabalho sugere que aconteça o controle antecedente e concomitante, por intermédio dos estabelecimento de Medidas e Indicadores de Qualidade e Valores de Custos e receitas na fase de especificação e de projeto do software.

5.2. CONCEITOS QUE SUSTENTAM O CONTROLE

Entendemos e tentamos transmitir nesta dissertação que para acontecer o controle econômico do desenvolvimento de software sem especialistas é indispensável que exista um Sistema de Informações eficiente ágil e dinâmico sustentado por conceitos bem definidos de contabilização de custos e receitas.

Acreditamos que os conceitos a serem adotados para o sistema de informações devem ser basicamente:

a - a acumulação de custos e receitas referentes ao software deve ocorrer por produto (software) e independe do método utilizado pela entidade para acumulação de custos e receitas de seus produtos/serviços finais;

b - a avaliação de custos e receitas devem ser a valores predeterminados, e também independem do método utilizado pela entidade para avaliar os custos e receitas de seus produtos/serviços finais;

c - os padrões devem ser as referências para avaliação de custos e receitas e devem representar as receitas e custos que deveriam ocorrer sob condições eficientes de operação;

d - os padrões devem ser dinâmicos, sendo alterados em função de fatores que possam influenciar a eficiência de operações, sendo que entre os fatores que podem afetar a eficiência encontram-se os estágios de cultura de informática, as características dos desenvolvedores e usuários e o tipo de software em função da sua frequência de uso;

e - os padrões poderão ser do tipo teórica, ideal atingível e padrões passados.

Apesar da indicação do padrão mais adequado depender das características peculiares de cada entidade, corremos o risco de afirmar que o ideal atingível é o mais adequado para organizações com um modelo de gestão semelhante ao que estabelecemos no item 4.3.2.3.

Considerando essa nossa opção pelo padrão ideal atingível é importante esclarecer que os mapas estatísticos de qualidade de programas desenvolvidos que nos referimos algumas linhas acima não podem ser confundidos com padrões passados.

f - Para a entidade que se encontra em algum dos estágios preliminares, e que não tenha desenvolvido número razoável de programas, parece-nos adequado que o referencial para

determinação do padrão ideal atingível sejam os parâmetros que seriam adotados se as atividades que o software vai executar acontecessem em condições eficientes de operação, por caminhos que já tenham sido utilizados noutras ocasiões. Acreditamos que esse referencial é válido porque para a entidade deve ser interessante que o software mantenha ou melhore a performance, os custos e receitas das atividades para qual esta sendo desenvolvido.

g - o método de custeio deve ser direto para facilitar a análise e controle de eficiência do desenvolvimento de software.

h -os orçamentos de operações servem às estimativas de despesas dos programas permanentes enquanto os orçamentos programados são utilizados para previsões de gastos com programas intermitentes e transitórios.

5.3. MODELO DE TOMADA DE DECISÕES

Entendemos também que é de suma importância para a realização do controle a existência de um modelo de tomada de decisões.

Nesta dissertação elencamos os aspectos que julgamos que deveriam ser abordados por esse modelo e os respectivos princípios e premissas os quais devem dizer como fazer para:

- a - estabelecer e executar medida de qualidade;
- b - definir qualidade desejada;
- c - definir a relação de qualidade desejada e obtida, ou seja, como se define IQ;
- d - identificar que famílias de IQ deverão ser analisadas;
- e - considerar na análise de resultados influências ambientais e culturais;
- f - analisar os custos e receitas observados a cada etapa do desenvolvimento de software, considerando os benefícios futuros a serem obtidos pela sua existência.

5.4. A CONTROLADORIA

Acreditamos que o desenvolvimento de softwares sem especialistas deve ser incentivado ao máximo, de forma que as entidades possam contar rapidamente com os benefícios provenientes da geração automática de informações.

Pensamos que os conceitos de controle e qualidade devem ser juntados de maneira a garantir o sucesso dessa forma de desenvolvimento e o caminho ideal para isto acontecer nas

entidades é por intermédio da conscientização dos seus gestores.

Pois esses conceitos devem estar embasando suas atitudes enquanto participantes do processo de gestão econômica da entidade, por intermédio do controle econômico do desenvolvimento de software nas suas respectivas áreas.

Acreditamos que a controladoria tem um papel importantíssimo nesse processo de conscientização, uma vez que faz parte das suas funções a assessoria aos diversos gestores na determinação de parâmetros e conceitos a serem utilizados no controle econômico.

Essa assessoria da controladoria, que abrange desde apoio teórico no estabelecimento de padrões até instruções de como interpretar os resultados econômicos de cada área, deve ser a forma pragmática de promoção da conscientização geral, isto sem desprezar a eventual necessidade de treinamento formal dos gestores.

A Controladoria pode ser o canal de sincronismo entre os processos de desenvolvimento de software das diversas áreas motivando as discussões e divulgando os resultados econômicos das elaborações de programas.

Defendemos, nesta dissertação, a idéia de que todo o suporte teórico do controle de desenvolvimento de software sem especialistas pode ser dado pela controladoria.

À entidade a controladoria pode dar sua contribuição, no que tange ao desenvolvimento de software pelas próprias áreas, estando atenta, por intermédio do sistema geral de informações, à eficácia de cada uma delas informando sobre os resultados de suas atividades.

Isto torna-se bastante importante porque o desvio de alguns profissionais de suas tarefas para o desenvolvimento de software pode comprometer sobremaneira a eficácia de suas áreas e até mesmo da entidade.

5.5. CONSIDERAÇÕES FINAIS

Nesta dissertação não tivemos a pretensão de exaustar o assunto, mas sim oferecer referencial para reflexões sobre parâmetros necessários para execução do controle econômico do desenvolvimento de software, ressaltando a importância que deve ser dada pelo controle econômico aos atributos que definem a qualidade dos programas e buscando explicitar as contribuições que a controladoria pode oferecer para o sucesso econômico dessa forma de elaborar softwares.

Assim sendo acreditamos que o assunto tratado nesta dissertação pode ser mais amplamente explorado detalhando, por exemplo, a forma de reconhecimento de receitas geradas pelo software desenvolvido sem especialistas, a influência do modelo de gestão na qualidade dos softwares, o aumento de produtividade administrativa proporcionado pelo desenvolvimento de softwares sem especialistas, verificação mais profunda da tendência e forma de desenvolvimento de softwares, as contribuições das universidades ao desenvolvimento sem especialistas (as quais precisam diminuir a ênfase do ensino de linguagens de programação a favor do ensino da estruturação lógica dos problemas a serem tratados pelo computador) e inúmeros outros aspectos que deixamos de abordar.

G L O S S Á R I O

GLOSSÁRIO

ADMINISTRAÇÃO	- Conjunto de princípios, normas e funções que têm por fim ordenar os fatores de produção e controlar sua produtividade e eficiência para se obter determinado resultado.
ALGORITIMOS	- Conjunto de instruções ordenadas com uma finalidade específica.
ATRIBUTO (*)	- O que é próprio de um ser ou de alguma coisa.
CENTRO DE CUSTOS	- Unidade contábil onde são acumulados custos, com objetivo de custeio de produtos e controle de operações.
CENTRO DE RESULTADOS	- Unidade contábil. Onde se acumulam custos e receitas. Apurando-se um resultado.
CIÊNCIA	- Conjunto organizado de conhecimento relativo a um determinado objeto, especialmente os obtidos mediante a observação, a experiência dos fatos e um método próprio.
CONCEITOS	- Representação dum objeto pelo pensamento, por meio de suas características gerais.
CONTROLE DE QUALIDADE	- Avaliação da qualidade de software após a elaboração (avaliação do produto). Essa avaliação é acompanhada de ações que buscam a qualidade ideal e evitam repercussões danosas na eventualidade de se observar qualidade inadequada.
CUSTOS	- Sacrifícios financeiros, arcados pela entidade relativo a bens e/ou serviços consumidos na produção de outros bens ou serviços.

CUSTO VARIÁVEL	- É aquele que pode ser objetivamente identificado com o produto. Altera-se de acordo com o volume produzido.
CRENÇA	- Convicção (*) certeza adquirida.
DADO (*)	- Elemento ou quantidade conhecida que serve como base à resolução dum problema, elemento para formação dum juízo (opinião).
DECISÃO	- Julgamento(*), opção(*), opção após exame.
DEFINIÇÃO	- Enunciação dos atributos essenciais de (uma coisa) de modo que a torne inconfundível.
DEPARTAMENTO	- Unidade administrativa, que pode corresponder a um ou mais centros de resultados ou de custos.
DESPESAS	- Sacrifício financeiro, arcado pela entidade relativa a bens e/ou serviços consumidos direta ou indiretamente para obtenção de receitas.
DISSERTAÇÃO	- Exposição de matéria estudada.
EFICÁCIA	- Obtenção dos resultados esperados.
EFICIÊNCIA	- Desempenho adequado na realização de uma atividade.
GARANTIA DE QUALIDADE	- Avaliação da qualidade do software durante a elaboração (avaliação ao processo). Essa avaliação é acompanhada de ações que garantam que o produto/serviço que encontra-se em desenvolvimento terá a qualidade desejada.

GESTÃO	--- Ato de gerir, administrar, dirigir.
INFORMAÇÃO	- Dados sobre alguém ou algo, instrução, direção, notícia.
INFORMÁTICA	- Informação automática.
MAINFRAMES	- Computadores com grande capacidade de armazenamento simultâneo de dados.
MISSÃO	- Algo que deve ser feito por intermédio dos recursos humanos e materiais de uma determinada entidade
MODELO	- Aquilo que serve de exemplo, norma, aquilo que se estabelece como base ou medida para realização de alguma coisa.
MULTITAREFAS	- Computadores com capacidade para efetuar simultaneamente processamento de dados relativos a diversas atividades.
MULTIUSUÁRIOS	- Computadores que permitem o uso simultâneo de diversos usuários.
NOTÍCIA	- Resumo dum acontecimento ou dum assunto qualquer.
PREMISSAS	- Teoria que serve de base a um raciocínio. Princípios que servem de base a um raciocínio. Cada uma das proposições que serve de base à conclusão (*). O que deveria existir.
PRINCÍPIOS (*)	- Preceitos, regras, lei, norma. Formúla que indica o modo correto de falar, raciocinar e agir.

PROC. DE DADOS	- Transformação de elementos ou quantidade conhecidas em resumo dum acontecimento ou de um assunto.
PROCESSAMENTO	- Ato de Processar (Organizar Processo).
PROCESSO	- Sucessão de estados ou mudanças maneira pela qual se realiza uma operação segundo determinadas normas.
PRODUTIVIDADES	- É uma medida da relação entre a eficiência e a eficácia.
PROGRAMAS	- Conjunto organizado de instruções ordenadas e codificadas em linguagem que pode ser interpretado pelo computador, que tem como função a transformação de dados em informações.
RECEITA	- É o valor do possível benefício a ser gerado pelo software.
REGRA (*)	- Formúla que indica segundo alguns valores, o modo correto de falar, pensar, de raciocinar, agir num caso determinado.
SISTEMAS	- Um conjunto organizado ou estabelecido. - Conjunto de partes coordenadas que concorrem para a realização de um conjunto de objetivos.
SISTEMA DE INFORMAÇÃO	- Qualquer sistema usado para prover informação (incluindo o seu processamento) qualquer que seja o uso feito dessa informação.
TÉCNICA	- Parte material ou conjunto de processos de uma arte.

TEÓRIA

- Conjunto de princípios fundamentos
duma arte ou duma ciência.

VALORES

- Importância que se atribui a
determinadas coisas (*).

(*) Ferreira, Aurelio B. de Holanda Minidicionário da Língua Portuguesa.

B I B L I O G R A F I A

ABNT	Normas sobre Documentação-NB-66-ABNT-RJ-1978
ABRIL	Revista Exame Abril-SP-ago/1990
ACKOFF, R.L.	Planejamento Empresarial Livros Técnicos e Científicos-RJ-1979
ANDERSEN, Arthur	Trends Information Technology A.Andersen-Chicago-1985
ANTONY, R. e Hekinian, J.	Controle de Custos de Operações Ed. Brasiliense-SP-1974
ANTONY, Robert N.	Contabilidade Gerencial: Uma Introdução à Contabilidade Atlas-1974
AWARD, Elias M.	Processamento de Dados nas Empresas Atlas-SP-1977
BERGAMINI, Cecília W.	Avaliação do Desempenho Humano na Empresa Atlas-SP-1973
BOEHM, B.W.	Software Engineering Economics Prentice Hall - N.Jersey-1981
BUSINESS Week	ZOOM! Here Come the New Micros(artigo) Business Week - dez/1986

CARROL, John M	Segurança do Computador Agents Editores - RJ 1979
CASTRO, Claudio de Moura	Estrutura e Apresentação de Publicações Científicas McGraw Hill
CATELLI, Armando	Sistema Contabilidade de Custos Estándar (Tese Doutorado) FEA/USP-SP-1972
CHURCHMANN, C.W.	Introdução à Teoria de Sistemas Ed. Vozes-1971
CITMAN, L.A.	Princípios de Adm. Financeira Ed. Harbra-1978
CULTURAL BRASIL	Dicionário Bras. da Língua Portuguesa - Cultural Brasil
DANTAS, Marcos	O Crime de Prometeu Como o Brasil Obteve Tecnologia da Informática ABICOMP-RJ-1989
DEARDEN, John	Análise de Custos e Orçamentos nas Empresas Zahar-RJ-1976
DEARDEN, John	Cost and Budget Analysis Prentice Hall - N. Jersey-1962

DEUTSH, M.S.

Software Verification and Validation: Realistic Project Approaches
Prentice Hall - N.Jersey-1982

DIAS, Donaldo de Souza
& Gazzaneo, Giosafatte

Projeto de Sist. de Processamento Dados Livros Técnicos e Científicos-RJ-1977

DUNN, R.
& Ullman, R.

Quality Assurance for Computer Software
McGraw Hill-1982

ELTON, Samuel

Management Control
Pergamon Press-1979

FAIRLEY, R.E.

Software Engineering Concepts
McGraw Hill -N.York-1985

FEA - Mestrandos
Depto Contabilidade
Turma de 1986

Anotações sobre Discussões, Estudos e Pontos de Vista sobre Contabilidade de Custos e Outros Temas de Diversos Colegas da Turma de Mestrando de 1986

FERREIRA, Aurélio B.H.

Minidicionário da Língua Portuguesa - Ed.Nova Fronteira-RJ 1985

FLEURY, Paulo F.

A Ind. Brasileira de
Informática Uma
Análise de Potencial
Locacional Nordeste
ABICOMP-RJ-1986

FLEURY, Paulo F.

Protecionismo e
Desenvolvimento Eco-
nômico: Um Estudo
Comparativo da Ind.
Bras. Informática
ABICOMP RJ-1987

FREITAS, Afonso Celso C.T.

Software Empresarial
Qualidade ao Longo
Desenvolvimento
(Dissertação de Mes-
trado)
Inst .Militar Enga.-
RJ-1986

GANE, Chris
& Sarson, Trish

Análise Estruturada
de Sistemas
Livros Técnicos e Ci-
entíficos-RJ-1983

GIL, A.L.

A Atuação da Audi-
toria de Sistema
Computadorizados
para Obtenção de
Uma Maior Produ-
tividade Proc.
Eletrônico de Dados
(Tese Doutorado)
FEA/USP-SP-1984

GIL, A.L.

Auditoria de Compu-
tadores
Atlas - SP-1989

GIL, A.L.

Sist.de Informações
Contábeis
Atlas-SP-1978

HAMPTON, David R.	Contemporary Management Mcgraw Hill-N.York- 1977
HECKERT, J.B. & Wilson, J.D	Controllershhip The Ronald Press Co- 1963
HORNGREN, Charles T.	Introdução Contab. Gerencial Prentice Hall-RJ- 1985
IUDICIBUS, Sergio & Equipe de Prof. USP	Contabilidade Intro dutória Atlas-SP-1987
IUDICIBUS, Sergio de	Teoria da Contabili dade Atlas-SP-1977
IUDICIBUS, Sergio de & Martins, Eliseu & Gelbcke,Ernesto	Manual Contabilidade Sociedades por Ações Atlas-SP-1986
KANITZ, Stephen C.	Controladoria : Teo ria e Estudo de Casos Biblioteca Pion .Adm Negócios-SP-1976
LANGFORS, B.	Theoretical Analysis Information Systems Auerbach- Philadelph 1973
LEITE, José A.Américo	Metodologia Elabo ração de Teses McGraw -Hill-SP-1978

LEONE, George S.G.

Custos: Um Enfoque
Adminstrativo
Fund.Getúlio Vargas-
RJ-1980

LONGWORTH, G.

Padrões Programação:
Métodos e Procedi-
mentos
CAMPUS-RJ-1985

LOON, Rudy Van

Working the Informal
Channels (artigo)
Datamation-mai/1990

MARTINS, Eliseu

Contabilidade de
Custos
Atlas-1985

MATHOTRA, Ashok

Computers in Mana-
gement
McGraw Hill N.Del
1979

McCOSH, Andrew M.
& Morton, Michael S.S.

Management Decision
Support Systems
The Macmillan Press-
1978

McCUSKER, Tom

Why Business Ana-
lysts are Indispen-
sable to IS
Datamation-jan/1990

McDANIEL, Herman

Careers in Computers
and Data Processing
Petrocelli Books -
1978

MELNICK, Julio

Manual Projetos
Desenvolvimento Eco-
nômico
Entrelivros -
Cultural-RJ

MOTTA, Fernando C.Prestes

Teoria Geral da
Administração
Livraria Pioneira
Editora-SP-1976

NAKAGAWA, Masayuki

A Controladoria e os
Métodos Quantita-
tivos
FEA/USP-SP-1978

NAKAGAWA, Masayuki

Aspectos Comporta-
mentais da Conta-
bilidade Gerencial
(ou Contab. Compor-
tamentamental)
FEA/USP-SP-1978

NAKAGAWA, Masayuki

Estudo de Alguns
Aspectos de Contro-
ladoria que Contri-
buem para a Eficácia
Gerencial
(Tese Doutorado)
FEA/USP-SP-1987

PEREIRA, Rogério Costa

APX - Avaliação e
Planejamento de Sist
de Informação
Edgar Blucher - SP-
1979

PERRY, W.E.

Effective Methods of
EDP Quality Assu-
rance
Prentice Hall-1983

PINHEIRO, Nizar da Silva

TCC -Time do Coorde-
nador Chefe
Edgar Blucher - SP-
1979

PUC-RJ

Anais do VIII
Congresso Latino -
Americano e Ibérico
sobre Métodos Com-
putacionais para
Engenharia
Puc-RJ-1987

RIBEIRO, Antonio M.

Análise e Projeto de
Sistemas
Univ. Federal Minas
Gerais-MG-1978

ROCHA, A.R.C. da

Controle Qualidade
Software: Uma
Necessidade Atual
Anais do XVIII
Congresso Nacional
de Informática
SP-1985

ROCHA, Ana Regina de C.da

Análise e Projeto
Estruturado de
Sistemas
Ed. Campus-RJ-1987

ROSA Neto, Ernesto

Apostila História
das Ciências
Diretório Acadêmico
A.Moraes
Univ. Mackenzie-SP-
1980

RUBIN, Martin L.

Documentation Stan-
dards and Procedure
on Line Systems
Van Nostrand
Reinhold- N.Y.-1979

SEI

Panorama do Setor de
Informática Série
Estatísticas - V.2
nº 1 Ago/1989
Secr. Esp .Ciência e
Tecnologia
Secr.Esp.Informática

SUCESU	Anais do XXII Congresso Nacional de Informática Sucesu-SP-1989
TANNENBAUM ,Arnold S.	O Controle nas Organizações Editora Vozes Petrópolis-RJ-1975
TREUHERZ, Rolf M.	Controladoria por Objetivos - Estudo de Um Caso Bibl .Pioneira Adm. Negócios-1974
VASCONCELLOS, Augusto de	Computadores Eletr. Digitais Ltd/Datamec-RJ-1974
VASCONCELLOS, Eduardo & Marcovitch,J	Meio Ambiente e Adm. de Projetos (apostila) FEA/USP-SP
WARNIER, J.D.	Lógica Construção de Programas Campus-RJ-1985
WOILTER, Sansão	Cooperação - Adminis- trador-Analista em Sist.de Informações (apostila) FEA-SP
YÁZIGI	Yázigi Dictionary For High Schools Yázigi -SP-1973

YOSHITAKE, Mariano

02/11/83 AS
203/11/83
LAQ
Funções Controller :
Conceitos e Aplica -
ções de Controle
Gerencial
(Dissertação do
Mestrado)
FEA/USP-SP

YOURDON, Edward

Análise Estruturada
Moderna
Ed. Campus-RJ-1990

1991/JCP