

**COMO INVESTIR EM TECNOLOGIA COM SEGURANÇA:  
CRITÉRIOS IMPORTANTES PARA SE ADQUIRIR E DESENVOLVER  
SOFTWARE**

**Paulo Roberto Rodrigues de Souza**

Dissertação submetida à Universidade Federal de  
Santa Catarina para a obtenção do título de Mestre  
em Engenharia de Produção

Florianópolis – SC  
dezembro de 2000  
Paulo Roberto Rodrigues de Souza

**COMO INVESTIR EM TECNOLOGIA COM SEGURANÇA:  
CRITÉRIOS IMPORTANTES PARA SE ADQUIRIR E DESENVOLVER  
SOFTWARE**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia, especialidade em Engenharia de Produção, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina, em dezembro de 2000.

Florianópolis, 12 de dezembro de 2000

---

Prof. Ricardo Miranda Barcia, Phd.D.  
Coordenador do Curso de Pós-Graduação  
em Engenharia de Produção

Banca Examinadora:

---

Prof. Francisco Antônio Pereira Fialho, Dr.  
Orientador

---

Prof. Alejandro Martins, Dr.

---

Profa. Ana Maria B. Franzoni, Dra.

### DEDICATÓRIA

Dedico este trabalho a todos aqueles que buscam através do uso das tecnologias a melhoria da qualidade de vida das pessoas e do mundo em que vivemos.

## AGRADECIMENTOS

**A Deus e à minha protetora Nsa. Sra. Aparecida** pela Sua  
proteção, sabedoria e Bondade Infinita ...

Este “marco” em minha vida não poderia se concretizar sem o  
apoio e a colaboração das seguintes pessoas as quais merecem meu  
sincero agradecimento:

Ao **Prof. Francisco Antonio Pereira Fialho, Dr.**, pelo apoio e  
carinho que me acolheu e, sobretudo pela amizade e atenção que  
sempre nos recebeu.

**Aos nossos Professores e Professoras** que através de muito  
empenho, repartiram seus conhecimentos.

Aos amigos **Edson Lourenço da Aparecida e Fernando César  
Moraes** que foram os primeiros mestres de minha carreira profissional e  
incentivadores incansáveis que sem este apoio nada disso teria se  
tomado realidade.

Aos **Meus pais**, que me deram apoio e carinho, incentivo e  
solidariedade, permitindo prosseguir meus estudos mesmo nas horas  
difíceis. E por terem me ensinado a gostar de aprender!

Aos grandes amigos **Hélio Lemes Costa Júnior e Lilian M.  
Ribeiro Conde** que foram sempre companheiros e solidários nesta  
caminhada e em muito colaboraram para minha formação neste curso.

Aos amigos **Fabício J. Pontes, Juliano Vitor de Miranda e  
Ronã Rinston A. Mendes** que foram o apoio necessário e providencial  
em momentos importantes desta caminhada.

À **todos colegas de mestrado** pelos momentos inesquecíveis  
e pela partilha dos conhecimentos durante este curso.

À **Diretoria e algumas pessoas da Unimed Varginha** e à  
**Diretoria e Colaboradores da Unimed Federação Sul de Minas** que  
viabilizaram minha presença neste curso.

À **UNINCOR, FEPEMIG** e ao **Colégio Catanduvás** que  
acreditaram em meu trabalho neste importante momento de minha vida  
profissional.

Ao meu irmão **Rafael** e a **todos meus grandes amigos** que de  
alguma forma colaboraram para a realização deste feito.

De maneira muito especial e carinhosa, à **Kelly**, minha noiva  
e incentivadora, pelos diversos momentos que não pudemos estar  
juntos devido à esta árdua e dedicada caminhada.

## SUMÁRIO

<b>1. Introdução</b>	<b>1</b>
<b>1.2 Justificativa</b>	<b>3</b>
<b>1.3 Estabelecimento do Problema</b>	<b>5</b>
<b>1.4 Objetivo Geral e Específico</b>	<b>6</b>
<b>1.5 Hipóteses Gerais e Específicas</b>	<b>7</b>
<b>1.6 Metodologia</b>	<b>8</b>
<b>1.7 Limitações</b>	<b>10</b>
<b>1.8 Descrição dos Capítulos</b>	<b>11</b>
<b>2. Revisão da Literatura</b>	<b>13</b>
<b>3. Critérios de Avaliação</b>	<b>51</b>
<b>4. Conclusões</b>	<b>88</b>
<b>Referências Bibliográficas</b>	<b>91</b>

## RESUMO

SOUZA, Paulo Roberto Rodrigues. **Como investir em tecnologia com segurança: critérios importantes para se adquirir e desenvolver software**. Florianópolis, 2000. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção – UFSC, 2000.

Esta pesquisa tem por objetivo determinar ou buscar “construir” um roteiro para compra de Software determinando critérios para que usuários de Informática ou pessoas que estejam buscando adquirir ou desenvolver um Software possam se orientar. Tendo conhecimento da extrema dificuldade encontrada pelas pessoas em adquirir Softwares de qualidade, e aproveitando o fato de atuar no ramo a algum tempo, a escolha pela tarefa de elaborar um texto de fácil entendimento buscando facilitar a árdua tarefa que é a de escolher um produto de Software para as empresas foi uma atividade quase natural, concedendo prazer em sua construção. Busca-se também com este trabalho, aprimorar conhecimentos e fornecer subsídios para as pessoas que pretendem investir em um produto de Software. Seja a compra ou o desenvolvimento mas que não querem ficar a mercê de interesses comerciais e mercantilistas das empresas fornecedoras de Software. Acredita-se que neste trabalho estas pessoas encontrarão muitas respostas para suas dúvidas e receios e possam assim facilitar seu trabalho.

## ABSTRACT

This research has for objective to determine or to look for "to build" a route for purchase of Software determining criteria so that users of Computer science or people that are looking for to acquire or to develop a Software can be guided. Tends knowledge of the extreme difficulty found by the people in acquiring quality Softwares, and taking advantage of the fact of acting in the branch the some time, the choice for the task of elaborating a text of easy understanding and intending looking for to facilitate the arduous task that is the one of choosing a product of Software for the companies was an almost natural activity, granting pleasure in your construction. It is also looked for with this work, to perfect knowledge and to supply subsidies for the people that intend to invest in a product of Software is the purchase or the development but that don't want to be the thanks to commercial interests and markets of the supplying companies of Software. It is believed that in this work these people will find a lot of answers for your doubts and fears and so they can facilitate like this your work.

# CAPÍTULO I

## INTRODUÇÃO

### 1.1 Considerações Iniciais

A necessidade das empresas de obterem informações confiáveis e rápidas provocou uma verdadeira corrida atrás de toda “parafernália” computacional necessária para colocá-la em sintonia com a modernidade e com a quebra das fronteiras, fenômeno hoje conhecido simplesmente pelo termo globalização.

De um lado as empresas que precisam das informações, à cada dia procuram recursos para que possam obtê-las e se manterem competitivas. É uma procura desenfreada por computadores modernos, recursos de comunicação de dados, interligação entre os equipamentos e junto com tudo isso a procura de Softwares que possam solucionar seus problemas e fornecer-lhes subsídios para que possam a tomada de ações rápidas e acertadas. De outro lado, temos as empresas fornecedoras de soluções cada vez mais sofisticadas, modernas e, muitas vezes caras. Algumas destas empresas visam somente o lucro e oferecem produtos de qualidade e resultado duvidosos, o que não pode de maneira alguma proliferar-se, razão que despertou o interesse pela construção desta pesquisa.

Marcelo Bauer em "A revolução dos Bytes" relatou bem o que está ocorrendo e colocou com muita propriedade sua posição com relação ao confronto Homem X Máquina e sua relação com o Brasil e as empresas. Disse assim:

*"A resolução dos problemas tecnológicos não garante ao Brasil, nem as empresas um lugar no grupo de países e empresas de sucesso. Tudo isso não significará quase nada se nosso país e nossas empresas não dispuserem de material humano qualificado. Em outras palavras, garantindo uma eficiente rede de telecomunicações e computadores modernos e baratos, teremos um bom campo de futebol e uma bola novinha. Mas é preciso jogadores treinados, entrosados, conhecedores das regras, sádios". (BAUER, 1997).*

Ou seja, não basta que tenhamos tecnologia, computadores, Softwares, equipamentos de última geração se não tivermos pessoas envolvidas e comprometidas com estas mudanças, pessoas que estejam treinadas e capacitadas para colocar as mudanças em prática.

Aliás, tanto o mundo dos negócios como o da educação estão descobrindo que o maior entrave para a utilização eficaz de computadores não é o conhecimento insuficiente sobre o funcionamento dos computadores, mas a necessidade de um maior conhecimento do papel da tecnologia da informação nas organizações, e de como ela pode ser aplicada aos problemas empresariais.

Igualmente importante é o desenvolvimento do pensamento crítico e de habilidades na resolução de problemas, uma das razões que nos levaram a desenvolver o texto constante desta pesquisa. O que importa é que as empresas, finalmente, entendam que para permanecerem competitivas, produtivas e prósperas, precisamos educar as pessoas com o conhecimento necessário e mais abrangente para solucionar novos problemas que venham a encontrar no futuro. E uma vez que elegemos a mudança como fator preponderante de interferência nas comunicações empresariais devemos não apenas denunciar a

necessidade de buscar alternativas que supram as lacunas de conhecimento tecnológico, de tecnologia dentro das organizações, mas que apresentemos, ao mesmo tempo, soluções viáveis para a superação de crises desta natureza. Assim surgiram as bases para a construção desta pesquisa, que procura destacar-se através de uma perspectiva explícita de apresentação de solução de problemas genéricos na obtenção de Softwares e no desenvolvimento do pensamento crítico a respeito do assunto, fazendo com que, por ocasião da obtenção destes recursos para as empresas, os empresários ou responsáveis por sua aquisição tenham subsídios para escolher com parcimônia o material que realmente possa suprir as necessidades da empresa em termos tecnológicos e de informação.

## **1.2 Justificativa**

- ✓ Promover o debate a respeito da informatização nas empresas buscando sempre que o investimento em tecnologia seja preservado e que se obtenha resultados satisfatórios. Investimento este tão necessário em nosso país que muitas vezes é desperdiçado ou feito de forma incorreta gerando inclusive um descrédito no uso da tecnologia e nos benefícios que ela poderia estar trazendo para as pessoas e empresas.

- ✓ Pesquisar sobre os maiores problemas enfrentados pelos dirigentes ou pessoas que desejam investir em Softwares e para pessoas que querem desenvolver ou implantar um Software em uma empresa, suas dificuldades e problemas mais freqüentes.
  
- ✓ Estabelecer critérios mais claros na avaliação de Software sob o ponto de vista de qualidade, ponto este tão abstrato e difícil de se avaliar.
  
- ✓ Pesquisar e esclarecer alguns pontos de forma mais clara sobre usabilidade e ergonomia de Software, pontos tão importantes nesta área atualmente porém, com material escasso no Brasil e, muitas vezes de difícil compreensão.
  
- ✓ Aperfeiçoar meus conhecimentos nesta área de forma a contribuir e poder estar disseminando estes conhecimentos em minhas atividades profissionais empresariais e pedagógicas.
  
- ✓ Proporcionar, através deste trabalho, uma discussão mais ampla nos meios acadêmicos das formas de investimento em tecnologia com segurança de modo que os investimentos em tecnologia sejam incentivados proporcionando melhor qualidade de vida para pessoas e conseqüentemente aumentando o mercado de trabalho na área.

### 1.3 Estabelecimento do Problema

A proposta apresentada por este trabalho passa pelo levantamento conceitual (aberto) de toda uma terminologia comumente empregada pela nova configuração de mercado de trabalho voltada para a tecnologia de informação, com o intuito de colocar o leitor a par do discurso empregado em boa parte dos tópicos do trabalho, atinge seu momento de maior destaque ao elaborar propostas que auxiliarão o leitor na tarefa de reconhecimento de Softwares que realmente atendam às suas necessidades operacionais, e finaliza com comentários que podem servir de impulso para o levantamento de novos estudos afins, que virão a contribuir para a divulgação e aceitabilidade desta nova configuração operacional do mercado mundial, onde os computadores e Softwares ocupam lugar de destaque.

Neste sentido, pretende-se oferecer ao leitor subsídios suficientes tanto para que a leitura se torne prazerosa diante da identidade com termos e conceitos apresentados, quanto para que classifique com criticidade as propostas apresentadas de maneira que possa desenvolver sozinho novas soluções para os problemas levantados como apoio à construção do hipertexto que rege os parâmetros direcionais da construção literária que nos propomos a apresentar neste momento.

O estudo que ora apresentamos ao leitor debruça-se sobre um objeto reconhecível – a aquisição ou desenvolvimento de Softwares – e procurar fornecer elementos para a verificação e a contestação das hipóteses apresentadas no **tópico 1.5**, encontrando-se aberto para uma continuidade crítica contestatória ou de continuidade das análises aqui expostas.

## **1.4 Objetivo Geral e Específico**

Especificadas as diretrizes e justificativas da apresentação deste trabalho, elegeu-se como objetivo geral desta pesquisa a apresentação de ferramentas que permitam ao leitor, a partir da leitura e compreensão dos conceitos apresentados, a emissão de julgamentos criteriosos acerca de programas, ERPs (Enterprise Resource Planning, Softwares de planejamento dos recursos empresariais, que integram as diferentes funções da empresa para criar operações mais eficientes em áreas como montagem ou entrega de produtos) que vem sendo amplamente difundidos no mercado, ou mesmo qualquer outro tipo de Software vendido pelos negociadores como sendo a ferramenta gerencial da qual a empresa necessita.

O objetivo específico é a apresentação de idéias e técnicas pessoais a respeito de aquisição de Softwares de forma a que a cada nova ferramenta adquirida se transforme realmente em veículo de

mudança que venha a fornecer agilidade aos processos e divulgação de informações já existentes na empresa.

## 1.5 Hipóteses Gerais e Específicas

Segundo levantamentos iniciais, pode-se eleger duas hipóteses gerais acerca dos motivos que levam a aquisição de Softwares, Hardwares e acessórios necessários para o funcionamento de um sistema básico de informações dentro das empresas a ser uma atividade problemática e geradora de despesas (no lugar de desejável geração de valores reais para a empresa):

- a) A empresa não determina critérios de qualidade dos Softwares a serem implantados de forma a alcançar produtividade e resultados satisfatórios;
- b) O processo de implementação por vezes é complexo e sem controle e as empresas ou pessoas envolvidas no processo nem sempre estão preparados para tal tarefa;

Como é possível perceber, as hipóteses levam a confabular acerca de especificidades inerentes a seus objetos de estudo. Se o problema é a implantação de sistemas que promovam uma mudança qualitativa das informações da empresa, eles têm, necessariamente que suprir necessidades com integração de dados-chave para a empresa,

fazendo com que todas áreas envolvidas estejam trabalhando de forma integrada tais como: compras, finanças, faturamento, contabilidade entre outras atuem conjuntamente e de maneira a manter as maiores margens de segurança possíveis e produzirem os resultados desejados.

O programa implantado deve estar apto a englobar as vendas, marketing, recursos humanos e outras funções, razão pela qual uma das hipóteses específicas pode ser: o programa adquirido, ou a ser adquirido atende a este tipo de necessidade?

Outra hipótese específica que nos propomos a conceder comprovação ou aceitar sua inviabilidade é: o sistema que se pretende adquirir é capaz de fornecer informações detalhadas sobre as operações da empresa? Como afirmar ou desconsiderar este tipo de reconhecimento?

A cada uma destas hipóteses procuraremos responder, comprovando ou desconsiderando suas realidades enquanto compatíveis com as necessidades das empresas no atual momento econômico vivido por todos neste final de século.

## **1.6 Metodologia**

A elaboração de uma construção literária conhecida nos círculos acadêmicos como pesquisa ou dissertação é, em primeira mão, uma descoberta da arquitetura reflexiva presente em toda investigação.

A presente pesquisa teve como preocupação primordial a pesquisa científica, elaborada frente a bases sólidas de autores e métodos que vem se destacando no ramo tecnológico na atualidade. Mas, tratando-se de tecnologia, seria compreensível se o primeiro caminho levasse a estabelecer caminhos panorâmicos acerca do assunto escolhido. Entretanto, compreendendo a impossibilidade de apenas um trabalho abarcar todas as variáveis necessárias para o estudo de Softwares que concedam apoio mediano aos empresários para controle de seus processos internos, optou-se por apresentar um trabalho que concedesse subsídios suficientes aos empresários a fim de que pudessem, eles mesmos, sem a participação excessiva de argumentação externa, optar pela melhor solução.

Objetivando atingir os propósitos aqui especificados procurou-se amparo em Eco (1991: 77) ao afirmar que mesmo no gênero das teses que empregam o estudo de caso a experiência que se pretende divulgar “*deve ser enquadrada numa discussão da literatura científica precedente, de forma que também aí os livros entram em jogo*”, o que serve para justificar o método empregado, qual seja o de pesquisa bibliográfica acrescida de teorias próprias ao autor.

A literatura selecionada procura conceder a compreensão genérica acerca das afirmativas que se pretende discutir, assim como as fontes (primárias ou secundárias) empregadas procuram envolver-se com o texto elaborado de maneira a que se transformem em um único corpo conciso e esclarecedor para o público-alvo.

## 1.7 Limitações

As limitações deste trabalho encontram-se associadas às dificuldades de levantamentos de dados junto a empresas com grande expressividade no mercado. Sendo assim, nossas hipóteses ficam restritas à realidade enfrentada por empresas de pequeno e médio porte localizadas dentro do mercado varginhense e sul mineiro. Muito embora tenhamos de restringir nosso campo de atuação, é importante verificar que o sul de Minas Gerais vem se transformando em importante foco industrial e tecnológico graças ao reconhecimento político acerca das verdadeiras necessidades de seu povo no tocante ao uso de tecnologias e informatização de suas empresas. Sendo assim, aquele que poderia parecer um estudo sem grande significação global, cresce em importância, demonstrando as atuais tendências mercadológicas desta que é uma das maiores regiões em franco desenvolvimento na atualidade em nosso país. Esta pesquisa se limita a analisar o conteúdo do CMM (Capability Maturity Model) e especificamente na norma ISO9126 e elaborar critérios para que sejam considerados na aquisição ou desenvolvimento de Software para que se possa ter uma probabilidade de acerto maior ou seja, para que com base nesta pesquisa, os dirigentes de uma empresa ou até mesmo profissionais liberais tenham um embasamento e critérios definidos que devam ser considerados na compra ou desenvolvimento de um Software. Que ela

lhes traga segurança e maiores possibilidades de êxito e que, conseqüentemente, seu investimento traga benefícios claros quanto ao Software a ser adquirido ou desenvolvido.

## 1.8 Descrição dos Capítulos

O **Capítulo I** é a **Introdução**, onde encontram-se delimitadas as idéias que se pretende discutir. No **Capítulo II – Revisão da Literatura**, constam alguns aspectos e embasamento teórico para a pesquisa. Envolve ainda este capítulo a descrição dos métodos e fontes de pesquisa utilizadas para que desenvolva a pesquisa com subsídios que a encaminhem para um resultado consistente e com argumentos. Procurarei me balizar nas regras de usabilidade descritas por alguns autores, pela norma ISO9126 e outras normas ISO que também tratam de qualidade de Software e do CMM (Capability Maturity Model) que é um método bastante utilizado para se avaliar os desenvolvedores de Software sobre que nível estaria seu processo de desenvolvimento.

No **Capítulo III – Critérios de Avaliação**, que encontra-se dividido em duas partes, tem-se: Parte I – Relato de forma heurística, baseado nos estudos realizados no capítulo II e em experiência profissional do autor, de alguns critérios que devem ser adotados ou avaliados para que se possa investir em Software com segurança. Para facilitar o entendimento destes critérios esta parte encontra-se dividida em: avaliação do fornecedor, da empresa onde será implantado o

Software e aspectos da implantação. Na parte II, também de forma heurística, e embasando nos estudos do capítulo II, encontram-se os principais critérios que devem ser adotados ou devem existir em um Software sob o ponto de vista de usabilidade.

O **Capítulo IV – Conclusão** ficou reservado para a conclusão da pesquisa com apresentação dos resultados e comprovações obtidas e sugestões para futuros trabalhos.

## Capítulo II REVISÃO DA LITERATURA

Através de pesquisas diversas na área e da busca de ferramentas acadêmicas e criteriosas que pudessem balizar a presente pesquisa, deparou-se com vários métodos de avaliação do Software assim como de inúmeros critérios para sua avaliação. Outro fator importante para citar nesta etapa da pesquisa é a da polêmica e difícil decisão de quem precisa de um Software porém, não sabe ainda se irá desenvolver este Software em sua própria empresa ou se irá contratar uma empresa externa para realizar tal tarefa. Dentre os inúmeros métodos e linhas de pensamento sobre avaliação de Software, preferiu-se ater em dois principais: Na ISO9126 que trata da avaliação do Software pronto, suas características, problemas e soluções e também no método CMM (Capability Maturity Model) que se enquadra bastante em minha pesquisa e avalia o Software de forma inversa ou seja, estuda a maturidade ou o quão é desenvolvido o processo de desenvolvimento de Software ou seja, não se preocupa somente com o produto final, Software, mas sim, com o processo, a forma como ele foi desenvolvido, que critérios foram adotados e, conseqüentemente, a organização do processo de desenvolvimento do Software que, segundo entendimento próprio, irá influenciar o produto final. Estes dois métodos foram escolhidos porque têm sido amplamente utilizados pelas empresas avaliadoras de Software e bastante divulgados nos meios acadêmicos

como ferramentas maduras e bastante abrangentes para que se possa estabelecer critérios para avaliação de Software.

## **2.1 Desenvolver Internamente ou Externamente? Eis a questão**

Sempre que alguém deseja investir em Software e tem em sua empresa uma estrutura mínima de Tecnologia de Informação, um analista ou programador talvez, constantemente é levantada a questão: será que não se poderia fazer este trabalho internamente ou realmente é necessário contratar um fornecedor externo? São várias considerações a serem feitas para quem tem um problema como este a ser resolvido. Normalmente, quando se tem recursos humanos para desenvolver o projeto, ou Software, as empresas preferem desenvolvê-lo internamente, mas tem-se que lembrar que não basta ter pessoas somente, tem-se que ter estrutura física, de equipamentos entre outras considerações que serão feitas a seguir sobre desenvolvimento interno. Já quando estes recursos não são disponíveis, se terceiriza o processo e então é contratada uma empresa para que o Software seja desenvolvido. Uma outra forma bastante e talvez mais utilizada nos dias atuais é o desenvolvimento híbrido. A seguir será possível visualizar segundo Strauss (1997), como é que se realiza este formato:

*“O Cliente fornece o conteúdo e as entradas para o projeto conceitual e o fornecedor fornece o projeto técnico e a produção. Então, por exemplo, o projeto pode ser feito internamente, enquanto a programação é feita externamente. A preparação das informações pode ser contratada externamente. De uma maneira geral, quanto mais trabalho de projeto, arquitetura de Software e desenvolvimento seja feito externamente, mais se considera que o esforço de desenvolvimento seja externo.”*

Como é possível visualizar através dos dizeres acima, poderia-se estar escolhendo uma das três formas para desenvolver um projeto. A escolha mais acertada vai depender dos recursos humanos e também tecnológicos disponíveis na empresa para que o projeto possa ser realizado. Para que se possa definir como será feito o projeto, vários itens devem ser observados e pensados, razão pela qual o estudo encontra-se dividido entre desenvolvimento externo ou híbrido e desenvolvimento interno

### 2.1.1 Critérios sobre desenvolvimento Externo ou Híbrido

Quando se pensa em desenvolver externamente ou de forma híbrida, deve-se pensar que a relação entre fornecedor e cliente não é uma relação que se iniciará e terminará quando o projeto for entregue. Sempre que se pensar em desenvolver um Software tem-se que ter em mente que este projeto poderá e com certeza sofrerá modificações, tanto durante o desenvolvimento e principalmente após sua entrega e, que a relação continua, portanto, a escolha de um fornecedor deve ser encarada como a escolha de um parceiro e nunca como um simples acordo comercial. A empatia entre as partes e principalmente o bom

relacionamento entre os funcionários da empresa contratante dos serviços com os do fornecedor é vital para o bom andamento dos trabalhos e para o êxito do projeto. A relação entre contratante e fornecedor muitas vezes extrapola, como disse as relações comerciais e financeiras. Um bom entrosamento e a lealdade na definição dos objetivos e do papel de cada um no projeto é bastante importante. Os interesses devem ficar bem claros logo na primeira reunião, cada responsável deve expor suas expectativas e objetivos ao participar do projeto. Esta fidelidade e clareza no relacionamento trará benefícios para o projeto e para ambas as partes envolvidas.

Strauss (1997) apresenta a seguinte opinião:

*“Algumas vezes os clientes sentem que estão em uma situação em que não podem abandonar um fornecedor e que este está levando vantagem. Por outro lado, o fornecedor pode sentir que um cliente que esteja pouco reconhecendo o trabalho não o esteja recompensando justamente por seus esforços criativos”.*

Ou seja, tudo deve estar muito claro pois em um relacionamento onde há dinheiro envolvido sempre há jogo de interesses e se algo não estiver muito bem definido pode-se ter problemas. Uma boa prática que tem evitado muitos aborrecimentos quando se tem uma relação comercial desta natureza é que tudo esteja documentado e que cada um tenha consigo uma cópia atualizada do projeto, suas características, cronograma de trabalho, custos e tarefas desenvolvidas e próximas etapas. Um ponto bastante importante no estudo sobre CMM (Capability

Maturity Model) é de que as alterações no projeto também devem estar contidas neste material documental. Isso auxilia e evita que se tenha precedentes e dúvidas quanto a remuneração a ao cumprimento do papel de cada um.

A escolha de um fornecedor externo é a tarefa mais árdua e difícil pois a empresa estará entregando nas mãos deste fornecedor um trabalho que pode ser vital para a organização, e isso implica em analisar custos, prazos, localização, equipe de desenvolvimento entre outros fatores que detalhados no capítulo III, alguns dos quais é necessário tratar ainda nestes momentos iniciais:

- 1) O primeiro é o acesso junto ao fornecedor e sua equipe, ou seja, a proximidade física do fornecedor, a facilidade em se dirigir ao coordenador do projeto naquela empresa ou até mesmo ao diretor da empresa para possíveis reuniões de alinhamento do projeto. É muito bom quando se tem esta facilidade porque mesmo que se esteja entregando o projeto ao fornecedor externo porque julga-se conveniente esta modalidade de desenvolvimento, a administração e acompanhamento dele, deve ficar sempre com a empresa contratante. O fácil acesso às pessoas responsáveis pelo seu projeto na empresa contratada é princípio básico para sua contratação. Empresas desenvolvedoras onde o acesso é dificultado, onde seja difícil agendar reuniões de revisão do projeto ou que se tenha acesso apenas a pessoas

“operacionais” ou seja, aos operários do projeto, devem ser descartadas da lista de possíveis contratadas. Normalmente esta dificuldade de acesso se dá quando a empresa fornecedora é grande e o projeto para ela não representa uma importância significativa. Isso muitas vezes traz problemas diversos no desenvolvimento do Software como o não cumprimento dos prazos, a perda de qualidade e o não cumprimento dos requisitos estabelecidos na reunião inicial. O ideal é se escolher um fornecedor que tenha estrutura capaz de desenvolver o projeto e que ele tenha importância significativa para esta empresa.

- 2) Outro ponto que é de certa forma ligado ao anterior, é a proximidade do fornecedor a ser escolhido com a empresa. Um fornecedor que fica muito distante da sede torna difícil o acompanhamento do projeto, aumenta os custos e pode trazer problemas como o desenvolvimento de um Software que não estaria atendendo as expectativas. Quando cita-se fornecedor distante, isso é muito relativo em dias onde a comunicação entre as pessoas e empresa está tão modificada diante dos novos recursos de comunicação como a vídeo conferência e a Internet. Mas a questão deve ser citada porque o acompanhamento do projeto de perto é bastante importante. Quando se contrata um fornecedor distante, ou seja, que torne difícil a ida freqüente na sede deste fornecedor, isso dificulta

reuniões de revisão do Software, testes de módulos, demonstração de como está o Software entre outras atividades de que se necessite estar lado a lado para que as coisas andem como seria desejável.

Uma desvantagem em se contratar o desenvolvimento externamente é que nem sempre os fornecedores estão familiarizados com a rotina de trabalho da empresa, suas particularidades, seu processo, seu fluxo de documentos e até mesmo seu ramo de negócio. Isso pode trazer alguns problemas e o fornecedor pode necessitar de algum tempo para se adaptar à rotina de trabalho e conhecimento do negócio antes de iniciar o processo de desenvolvimento do Software. Desenvolver Software em áreas hostis para o novo fornecedor pode resultar em um produto que funcione, realize tarefas para o qual foi designado mas, não esteja em conformidade com o que se deveria estar recebendo. Para a solução ou para se amenizar este problema, é preciso voltar a citar o desenvolvimento híbrido, ou seja, o acompanhamento de um colaborador da empresa experiente e conhecedor de todo processo para que se possa evitar estes desvios.

A escolha de um fornecedor e a manutenção deste por todo projeto e para futuras implementações e versões é produtivo e trará benefícios visíveis. A mudança de fornecedor no meio do projeto ou entre versões dificultará a continuidade do processo de desenvolvimento ou de adaptações e poderá incorrer em atrasos e não cumprimento de

prazos. Isso sem falar no aumento dos custos pois, o novo fornecedor não familiarizado com o Software e nem com o projeto no todo, demorará horas para que possa se estabilizar e apresentar produtividade e qualidade razoável.

Se a empresa tiver característica de contratar desenvolvimento externo, deve-se desenvolver, sempre que possível um pseudo segundo fornecedor. Embora isso não seja muito fácil, pode ser muito importante quando se necessitar, por algum motivo de um outro fornecedor. Uma prática bastante razoável é a de se contratar trabalhos menores ou secundários com um outro fornecedor que não seja considerado seu principal, isso gerará uma situação de concorrência que, com certeza alertará o fornecedor principal e o fará realizar os trabalhos com a melhor qualidade possível para que possa continuar oferecendo trabalho. É razoável deixar que os dois fornecedores saibam da mesma situação, mas sem gerar conflitos ou problemas entre eles. Deve-se isolar sempre que necessário e possível as atividades de um e outro fornecedor.

Uma vantagem de se realizar o trabalho externamente ou de forma híbrida, desde que a escolha do fornecedor seja criteriosa e com acompanhamento, é o cumprimento dos prazos, ou seja, quando se contrata o trabalho externamente e isso é colocado em contrato, amarrando as tarefas realizadas condicionando-as com o pagamento, os prazos normalmente são cumpridos. Diferentemente do desenvolvimento interno, quando se tem pouco controle sobre os prazos devido a interferência de outros trabalhos que possam surgir dentro da empresa e

forcem o seu a ficar para depois. Isso é de extrema importância se o cumprimento dos prazos é fator de muita relevância para a empresa.

Strauss (1997) auxilia, neste aspecto, a compreensão sobre preços de desenvolvimento:

*“Embora fornecedores inescrupulosos façam cotação extremamente baixa apenas para ganhar a concorrência, acreditando que quando o aplicativo estiver em andamento, poderão pedir mais dinheiro para terminarem o trabalho. Outros fornecedores podem fazer cotações com preços mais baixos, sabendo que entregarão um produto de baixa qualidade. Alguns fornecedores também são conhecidos por cotações baixas para “fisgar” um novo cliente, isto é, intencionalmente perdem dinheiro em um aplicativo de tal maneira que podem “abrir as portas”, tendo a esperança de conseguirem outros projetos no futuro, a um preço bem maior.”*

Torna-se por conseguinte, complicada a escolha de um bom fornecedor, como explica o capítulo III que trata a questão de forma mais profunda, tem-se que averiguar para quem este fornecedor já desenvolveu algo, sua reputação e capacidade para cumprir prazos e manutenção dos custos iniciais. Para que isso não ocorra, a escolha de um fornecedor deve ser algo realizado com muito cuidado, devendo ir até o fornecedor, conhecer sua equipe, conversar com seus clientes, verificar produtos já desenvolvidos, e, analisar se ele trabalha com foco no produto final ou no processo ou seja, qual sua real capacidade de gerenciar seu processo de desenvolvimento de Software, qual sua organização em torno do processo de desenvolvimento. Nem sempre o menor custo ou o maior será a melhor escolha, tem-se que analisar

vários pontos antes de decidir e, decidir simplesmente pelo menor ou maior custo certamente trará dor de cabeça.

O que se diz a respeito de se escolher entre o desenvolvimento interno ou não é que têm-se que avaliar a situação antes de decisões serem tomadas. Desenvolver Software externamente não é simplesmente “pagar” para que o trabalho seja feito e pronto, a realização e desenvolvimento de um projeto externamente também exigem controle sobre o que se deseja fazer, quem irá fazer o que, com que prazo, a que custo ou seja, o desenvolvimento externo exige que se tenha na empresa contratante pessoas administrando e gerenciando o processo. Não é simplesmente entregar modelos de formulários e pilhas de documentos ao fornecedor e ponto final. O projeto deve ter sempre a participação de colaboradores da empresa contratante, é claro que se o fornecedor será pago para projetar, desenvolver e implantar o Software, o esforço operacional será muitas vezes menor mas não se pode perder a noção administrativa e o controle da situação pois, se não houver acompanhamento, o fornecedor poderá realizar o trabalho da forma que achar conveniente e nem sempre estas são as necessidades da empresa contratante.

Se a empresa realmente pensa em desenvolver seu projeto externamente, Strauss (1997) pode auxiliar ao explicar sobre o acompanhamento do fornecedor:

*“A melhor maneira de garantir a capacidade de um fornecedor é ver alguns de seus trabalhos prévios. Um fornecedor de qualidade deveria Ter pelo menos alguns exemplos de projetos para mostrar, esperançosamente algum do mesmo tipo que o*

*seu. Se o trabalho for um aplicativo diferente ou não for de alta qualidade, há pouca necessidade de continuar a negociação. Se o trabalho parece bom, tente descobrir quais partes dele foram feitas pelo fornecedor.”*

A verificação da experiência do fornecedor em trabalhos do mesmo tipo, porte e características parecidas com o que se pretende desenvolver é fator importante a ser considerado se a empresa imagina desenvolver externamente o Software.

Ter cuidado com os prazos é bastante importante para desenvolver o Software externamente ou de forma híbrida. Imagine que o fornecedor pode ter vários projetos para serem contratados e ele não pode ficar esperando por um específico já que até a contratação efetiva não há garantias de que o trabalho será feito por ele. Esta é a prática comum, principalmente quando se está em uma empresa grande onde as decisões nem sempre são rápidas e o que ocorre é a rotina do “Apreste e espere”, ou seja, pode-se apressar o fornecedor para apressar o projeto e depois aguardar infinitamente pela aprovação. Isso tende a sufocar o fornecedor, principalmente se for uma empresa pequena ou média onde as pessoas que deverão desenvolver seu trabalho dependem dele e de outros ou seja, precisam se planejar para atender bem os clientes. Imagine se todos aprovassem seus projetos ao mesmo tempo, não terão como dar vazão à infinita carga de trabalho e contratar mais funcionários para este pico de trabalho seria, talvez uma atitude inviável. Ou seja, o que se deve cuidar é de um planejamento rápido acompanhado de uma aprovação também rápida. Uma prática

bastante razoável seria de se planejar e obter um preço base para o projeto e conseguir aprovação rapidamente ou até mesmo de uma pré-aprovação para que se possa realizar o projeto sem “sufocar” o fornecedor.

Planejar todo conteúdo e definições do projeto juntamente com pessoas das diversas áreas da empresa é bem prudente, sobretudo porque a adição de outros recursos durante o desenvolvimento pode desgastar a relação fornecedor-cliente e provocar problemas quanto aos custos do desenvolvimento. Esgotar todas possibilidades e recursos que estarão disponíveis no Software é um ato interessante para se evitar problemas durante e após o desenvolvimento do Software. É claro que isso não é muito fácil mas o acompanhamento do desenvolvimento e do projeto do Software deve ser feito de perto e bastante detalhado para que se evite problemas como estes.

Envolver o departamento financeiro na definição dos prazos e condições de pagamento do desenvolvimento externo é uma prática recomendável, pois acredita-se não caber à área técnica/administrativa tais definições porém, a diretoria da empresa deve participar destas definições já que, provavelmente irá conferir e autorizar o pagamento das parcelas referentes ao desenvolvimento.

## 2.1.2 Critérios sobre desenvolvimento interno

Segundo Strauss (1997):

*“Fazendo o desenvolvimento em casa, você tem o máximo controle sobre como o projeto está sendo executado. O pessoal de desenvolvimento trabalha diretamente para a empresa e pode ser monitorado bem de perto, se necessário. Você terá grande flexibilidade sobre como os recursos são aplicados. Em adição, se o desenvolvimento for em casa, presume-se que a manutenção possa ser feita também em casa. Em projetos onde se tenha importância estratégica, a manutenção em casa é o método mais seguro e também menos oneroso.”*

A facilidade que se tem quando o processo de desenvolvimento está dentro da empresa é visível, podendo-se dizer que são dois lados da moeda, se de um lado, com o processo internamente o controle é facilitado, o acompanhamento do desenvolvimento, a integração entre a equipe de desenvolvimento e os outros departamentos, aumenta a motivação de todos na empresa pela participação no projeto, quando todos participam, com certeza há um comprometimento maior com o projeto e o interesse de que ele seja bem sucedido. Por outro lado, o desenvolvimento interno também traz problemas, normalmente os profissionais envolvidos no projeto têm outras atribuições na empresa e atribuições que já fazem parte do dia a dia da empresa e não podem parar. O projeto novo pode ficar para segundo plano se não houver uma definição muito clara e o comprometimento de todos níveis da empresa quanto a importância e tempo de dedicação desta equipe a este novo projeto. Os prazos podem não ser cumpridos e isso pode ser um problema a ser considerado para quem quer desenvolver Software internamente.

Uma questão que sempre está em jogo quando se deseja realizar um trabalho é o custo deste trabalho. Muitos dirigentes têm a falsa impressão de que se o desenvolvimento for realizado dentro da própria empresa ele sairá bem mais barato que se fosse contratada uma empresa ou profissionais externos para realizar tal tarefa. Realmente, como se afirmou anteriormente, um novo projeto em andamento na empresa traz motivação e comprometimento de toda equipe mas isso pode ser complicado se estas pessoas não tiverem tempo hábil para dedicar a este trabalho.

Se o pessoal interno está capacitado para realizar tal trabalho os trâmites iniciais correrão mais rápido. Caso contrário, eles necessitarão de treinamento e capacitação antes de iniciar os trabalhos. Isso pode ser muito bom pois existirão pessoas treinadas “dentro de casa” que poderão alterar ou realizar manutenções no Software futuramente mas se a empresa acredita que o cumprimento dos prazos é um ponto importante, esqueça, a utilização de qualquer ferramenta nova requer amadurecimento no seu uso para que se tenha produtividade aceitável e isso normalmente leva tempo. Muitos profissionais, quando desenvolvem um novo projeto em uma ferramenta que acabaram de aprender, sempre o refazem depois de algum tempo pois, quase sempre o desenvolvimento não fica como gostariam ou com qualidade e performance aceitável.

Para se desenvolver um novo Software, quase sempre são necessárias ferramentas de desenvolvimento que normalmente são

ferramentas de custo elevado, presente em empresas que desenvolvem Softwares. Se a empresa imagina estar desenvolvendo um projeto internamente, é preciso analisar esta questão e verificar se haverá necessidade de investimento nestas ferramentas e se este investimento realmente é justificável.

Segundo Strauss (1997):

*“Sobre a execução do trabalho em casa é o acréscimo de experiência e especialização que será ganho pelo pessoal (embora isto não seja feito sem um custo real). Os talentos e capacidades de seus empregados são um patrimônio real e valioso para a empresa.”*

É importante, diante destas considerações, observar que o investimento realizado em pessoal para desenvolvimento de um projeto específico, além da especialização e conhecimento do pessoal interno em ferramentas e técnicas de desenvolvimento podem ser aproveitadas em outros projetos ou seja, o investimento realizado para se fazer um Software pode perfeitamente ser aproveitado para futuros projetos e trabalhos internos. Talvez se tenha um custo relativamente alto no primeiro trabalho mas que será compensado em outros trabalhos que forem realizados internamente. Ainda segundo a linha de pensamento de Strauss (1997), a satisfação do pessoal interno em realizar tais projetos aumentará a produtividade e gerará dentro da empresa e das pessoas envolvidas com estes trabalhos uma motivação muito especial na equipe.

Agora, se desejarmos que o desenvolvimento for realizado internamente temos que questionar: a empresa terá pessoas disponíveis para realizar tal projeto?

É importante, se ela está pensando em realizar o desenvolvimento internamente perceber se a equipe de profissionais já existente conta com pessoas disponíveis para realizar tal tarefa. Aumentar a equipe somente para um projeto pode ser dispendioso e pode trazer problemas futuros pois, após a realização do projeto, seus colaboradores mais antigos farão uma forma de pressão para que os novos colaboradores continuem na empresa e a demissão destes poderá gerar insegurança e insatisfação dentro da empresa. Compensa montar uma área de desenvolvimento? Compensa deslocar pessoas somente para este fim ou contratá-las? A empresa terá mais projetos para serem desenvolvidos que justificam a contratação de pessoas? Terá espaço e equipamentos necessários para estas pessoas trabalharem?

Tais perguntas têm que estar respondidas antes que se tome a decisão de contratar ou alocar pessoas dentro da empresa para tal fim. Não basta contratar ou alocar profissionais, temos que ter estrutura de espaço físico e equipamentos disponíveis.

Se a empresa realmente resolver contratar novos profissionais para realizar tais projetos, as condições de trabalho dos novos profissionais têm que estar bem claras para os novos colaboradores e para os colaboradores mais antigos na empresa para que não se tenha problemas com este importante item.

Se a empresa pretende investir no desenvolvimento interno, é importante que conheça a potencialidade de sua equipe, atribuições e pessoas que estarão envolvidas no projeto. Deve ser verificado também com o pessoal envolvido no processo de desenvolvimento, as funções na empresa e outras atribuições que possam atrapalhar o desenvolvimento. Não devem ser feitas promessas ou planejamento de custos e prazos sem antes considerar estas variáveis. É importante que não seja prometido algo que não pode cumprir, pois a empresa estará colocando em risco a reputação própria, da equipe e de outras pessoas envolvidas neste projeto. Lembre-se que do êxito do primeiro projeto é imprescindível para que se possa estar conseguindo novos recursos e melhor remuneração para a equipe. Se o primeiro projeto for bem sucedido, outros virão e com certeza o profissional responsável terá respaldo para realizar os investimentos necessários e trazer mais benefícios para si próprio e para toda equipe envolvida nele.

Como vimos, a escolha entre o desenvolvimento interno ou externo ou ainda híbrido, é uma opção que deve ser analisada caso a caso ou seja, não existe uma regra clara e fixa para determinarmos como deve ser realizado este desenvolvimento. O que se deve considerar são os fatores abordados anteriormente, de acordo com os recursos da empresa e a capacidade de investimento que se poderá determinar como será o desenvolvimento.

## 2.2 O Método CMM como ferramenta no desenvolvimento de Software

*Em 1988, Watts Humphrey, do Software Engineering Institute (SEI), da Carnegie Mellon University (CMU), nos Estados Unidos, propôs um modelo com cinco níveis de maturidade, conhecido como Capability Maturity Model (CMM), para a melhoria dos processos de Software. (WEBER, 1997)*

O CMM se revelou como importante instrumento de avaliação do processo de desenvolvimento de Software, com um foco voltado inicialmente para o produto final em seus primeiros níveis até que se atinja o nível 5, os níveis de avaliação do processo de desenvolvimento de Software CMM avaliam com muita propriedade o fornecedor ou o desenvolvedor de Software, pessoalmente o autor encontra-se neste momento considerando este método já que acredita ser importante que se avalie o processo de desenvolvimento tendo em vista que, quanto mais avançado o processo de desenvolvimento do Software, mais chances de se atingir maior qualidade o produto final terá. Especificamente o método foi escolhido por se tornar quase que um padrão de avaliação do processo de desenvolvimento de Software e ser de bastante consideração e credibilidade entre desenvolvedores. Também é importante ressaltar que somente o uso de metodologias no processo de desenvolvimento não valem muita coisa. Tem que se ter metodologias e estar sempre utilizando realmente e atualizando-as, uma vez que ter metodologias sem atualização é pior que se não as tivesse. O Capability Maturity Model proporciona às organizações diretrizes de

como se conseguir controle de seus processos para desenvolver e manter Software e como evoluir em direção a uma cultura de excelência em engenharia e manutenção de Software. O CMM foi projetado para orientar organizações em seleção de estratégias de melhorias, determinando a atual maturidade e identificando as poucas questões mais críticas para a qualidade do produto e a melhoria do processo de desenvolvimento. Enfocando um conjunto limitado de atividades e trabalhando agressivamente para alcançá-las, uma organização de desenvolvimento pode decididamente melhorar seu processo de desenvolvimento para ganhos contínuos e duradouros em seu processo e conseqüentemente conseguir desenvolver Softwares com mais qualidade e eficácia.

Como está contido no **Relatório Técnico CMU/SEI-93-TR-024 de fevereiro de 1993 da Carnegie Mellon University**, o CMM não diz os caminhos para a melhoria entre os níveis, ele se limita a descrever os níveis e quais critérios devem ser enfocados e implementados para se mover de um nível para outro. É preciso que se tenha um grupo de gerência de qualidade para que se possa alcançar os níveis mais altos do CMM embora para que isso ocorra tem que se ter esforço considerável de toda equipe, desde os profissionais menos experientes até a alta direção da empresa desenvolvedora.

### 2.1.1 Avaliando os níveis

Nível 1: Conhecido como Inicial, o desenvolvimento é caótico. Não existem procedimentos padronizados, estimativas de custos e planos de projeto. Cada qual desenvolve como quer, não existe documentação e não há mecanismos de controle que permitam ao gerente saber o que está acontecendo, identificar problemas e riscos e agir de acordo. Como consequência, os desvios não são corrigidos e ocorrem os problemas que todos conhecemos muito bem: prazos não cumpridos, orçamentos estourados, Software sem qualidade e usuários insatisfeitos. Na verdade, raramente existe um cronograma ou um orçamento.

Tabela 1 – Nível 1 do CMM

Nível CMM	Áreas-chave de processo	Objetivos
1) Inicial		

Fonte: CMM (SOFTWARE ENGINEERING INSTITUTE. The Capability Maturity Model: guides for improving the Software process. Reading : Addison Wesley, 1997.

Nível 2: Conhecido como repetível: A Empresa deve possuir controles básicos de projeto, incluindo o Gerenciamento de Projetos (técnicas para planejar e estimar o esforço em projetos, e controlar o progresso), deve-se possuir um controle gerencial com verificações pela gerência do projeto em alguns momentos determinados, incluindo verificando-se a qualidade dos produtos. Deve-se também instituir um Grupo de Controle de Qualidade e de procedimentos básicos de

Controle de Mudanças (para garantir que mudanças no projeto e manutenções solicitadas não destruam o que já foi feito) garantindo um mínimo de estabilidade no desenvolvimento. Quando a empresa estiver neste nível, terá maiores condições de ter mais controles sobre o projeto e terá estimativas e estatísticas mais precisas sobre o processo. Enfim, a partir de então, espera-se que a qualidade do Software seja melhor.

Tabela 2– Nível 2 do CMM

Nível CMM	Áreas-chave de processo	Objetivos
2) Repetível	Gerenciamento de requisitos	<p>Os requisitos do sistema definidos para o Software são controlados de forma a estabelecer um perfil mínimo a ser utilizado pela engenharia de Software e pela administração</p> <p>Os planos, produtos e atividades do Software são sempre consistentes com os requisitos de sistema definidos para o Software</p>
	Planejamento do projeto	<p>Estimativas relativas ao Software são documentadas para uso no planejamento e acompanhamento do projeto do Software.</p> <p>As atividades de projeto de Software e compromissos assumidos são planejados e documentados.</p> <p>Grupos e pessoas afetadas concordam com seus compromissos relacionados ao projeto do Software.</p>

Nível CMM	Áreas-chave de processo	Objetivos
2) Repetível	CONTINUAÇÃO	
	Visão geral e acompanhamento do projeto	<p>Resultados reais são acompanhados de acordo do com o planejamento do Software</p> <p>Quando os resultados apresentam um significativo desvio do planejamento do Software, são tomadas ações corretivas que são acompanhadas até o final do projeto</p> <p>Mudanças nos compromissos assumidos são feitas em comum acordo com os grupos e indivíduos afetados</p>
	Gerenciamento de sub-contratados	<p>O contratante seleciona sub-contratos qualificados</p> <p>O contratante e os sub-contratatos estão de acordo no que diz respeito aos compromissos assumidos um com o outro.</p> <p>O contratante e os sub-contratados mantém uma comunicação constante</p> <p>O contratante acompanha os resultados reais do sub-contratado de acordo com os compromissos assumidos</p>
	Garantia da qualidade do Software	<p>As atividades de garantia de qualidade de Software são planejadas</p> <p>A conformidade dos produtos de Software e atividades com os padrões, procedimentos e requisitos é verificada objetivamente.</p> <p>Os grupos e indivíduos afetados são informados das atividades de garantia de qualidade de Software e de seus resultados.</p> <p>Questões relacionadas à não conformidade que não são resolvidas dentro do projeto de Software são encaminhadas à gerência geral</p>

Nível CMM	Áreas-chave de processo	Objetivos
2) Repetível	CONTINUAÇÃO	
	Gerenciamento de configuração	As atividades de gerenciamento de configuração são planejadas. Os produtos de trabalho de Software são identificados, controlados e estão disponíveis. Mudanças nos produtos de trabalho identificados são controladas. Os grupos e pessoas afetadas são informados da situação atual e projetada dos produtos de trabalho de Software.

Fonte: CMM (SOFTWARE ENGINEERING INSTITUTE. The Capability Maturity Model : guides for improving the Software process. Reading : Addison Wesley, 1997.

Nível 3: Conhecido como Definido: É necessário introduzir uma Metodologia de Desenvolvimento formal, com um ciclo de vida definido (entenda-se como ciclo de vida como a forma de desenvolvimento: Têm-se os ciclos de vida clássico, protótipo, espiral ou 4ª geração que não descreveremos cada um por achar desnecessário pelo objetivo principal desta pesquisa) acompanhada de métodos, técnicas e ferramentas apropriadas, como inspeções e técnicas abrangentes de teste. É o momento também de estabelecer o SEPG, isto é, o time encarregado exclusivamente da melhoria contínua do processo de Software. Ao chegar a este nível, a empresa terá um fundamento claro para desenvolver sistemas e também para melhorar o próprio processo, especialmente quando surgirem crises. No nível 3, entretanto, os controles ainda são basicamente qualitativos, não havendo meios de quantificar a qualidade dos produtos e a eficiência do processo. Assim, a

empresa deve estabelecer métricas de forma a medir características específicas dos produtos. A forma de coletar, armazenar e analisar estes dados é definida e, com base nesta informação, pode-se sugerir melhorias específicas nos produtos.

Tabela 3– Nível 3 do CMM

Nível CMM	Áreas-chave de processo	Objetivos
3) Definido	Foco do processo organizacional	<p>São coordenadas atividades de desenvolvimento e melhoramento do processo de Software em toda a organização</p> <p>Os pontos fortes e fracos do processo de desenvolvimento de Software utilizado são identificados, de acordo com um padrão de processo.</p> <p>São planejadas atividades de desenvolvimento e melhoramento do processo, a nível de organização.</p>
	Definição do processo organizacional	<p>O processo padrão de desenvolvimento de Software da organização é desenvolvido e mantido.</p> <p>A informação relacionada ao uso do processo padrão de desenvolvimento de Software é coletada, revisada e disponibilizada.</p>
	Programa de treinamento	<p>As atividades de treinamento são planejadas</p> <p>É fornecido treinamento para o desenvolvimento de habilidades e conhecimentos necessários para realizar o gerenciamento do Software e as funções técnicas.</p> <p>As pessoas no grupo de engenharia de Software e outros grupos relacionados a Software recebem o treinamento necessário para realizar as suas funções</p>

Nível CMM	Áreas-chave de processo	Objetivos
3) Definido	CONTINUAÇÃO	
	Gerenciamento de Software integrado	O processo de Software definido para o projeto é uma versão adaptada do processo padrão de desenvolvimento de Software da organização O projeto é planejado e gerenciado de acordo com o processo de desenvolvimento de Software definido para o projeto
	Engenharia de produto de Software	As atividades de engenharia de Software são definidas, integradas e consistentemente realizadas para produzir o Software. Os produtos de trabalho do Software são mantidos consistentes entre si.
	Coordenação intergrupos	Todos os grupos de trabalho afetados concordam com os requisitos dos cliente. Todos os grupos de trabalho afetados concordam com os acordos entre os grupos de engenharia Os grupos de engenharia identificam, acompanham e resolvem todas as questões intergrupos.
	Revisão conjunta	Atividades de revisão conjunta são planejadas Defeitos nos produtos de trabalho são identificados e removidos.

Fonte: CMM (SOFTWARE ENGINEERING INSTITUTE. The Capability Maturity Model : guides for improving the Software process. Reading : Addison Wesley, 1997.

Nível 4, ou Gerenciado: Deve-se estabelecer meios para a coleta automática de métricas e para a utilização da informação coletada de forma a prevenir problemas. Enquanto no nível 4 as métricas têm seu foco no produto (o Software sendo desenvolvido)

Tabela 4– Nível 4 do CMM

Nível CMM	Áreas-chave de processo	Objetivos
4) Gerenciado	Gerenciamento quantitativo dos processos	As atividades de gerenciamento quantitativo dos processos são planejadas A performance do processo de desenvolvimento de Software definido para o projeto é controlada quantitativamente A capacidade do processo desenvolvimento de Software padrão da organização é conhecida em termos quantitativos.
	Gerenciamento da qualidade de Software	As atividades de gerenciamento da qualidade de Software do projeto são planejadas. Objetivos mensuráveis da qualidade do produto de Software e suas prioridades são definidos. O progresso real em direção à realização dos objetivos de qualidade para os produtos de Software é quantificado e gerenciado.

Fonte: CMM (SOFTWARE ENGINEERING INSTITUTE. The Capability Maturity Model : guides for improving the Software process. Reading : Addison Wesley, 1997.

Nível 5 ou Otimizado: São coletadas e utilizadas informações sobre o processo em si. A idéia é analisar as causas do problemas e atacá-las para evitar que voltem a ocorrer. Enquanto os dados coletados no nível 4 podem informar, por exemplo, quantos erros existem em um programa, a preocupação no nível 5 é melhorar o processo para evitar que tais erros aconteçam no próximo projeto.

O modelo CMM é extremamente útil para uma organização de desenvolvimento poder avaliar a si mesma e estabelecer prioridades

para melhorar seu processo de desenvolvimento e, conseqüentemente, aumentar a qualidade do Software que desenvolve.

Tabela 5– Nível 5 do CMM

<b>Nível CMM</b>	<b>Áreas-chave de processo</b>	<b>Objetivos</b>
5) Otimizado	Prevenção de defeitos	As atividades de prevenção de defeitos são planejadas As causas comuns de defeitos são procuradas e identificadas As causas comuns de defeitos são priorizadas e sistematicamente eliminadas.
	Gerenciamento de mudanças tecnológicas	A incorporação de mudanças tecnológicas é planejada Novas tecnologias são avaliadas para determinar seu efeito na qualidade e na produtividade Novas tecnologias adequadas são incorporadas na prática normal de toda a organização.
	Gerenciamento de mudanças no processo	O melhoramento contínuo do processo é planejado Toda a organização participa das atividades de melhoramento do processo de Software O padrão de processo de Software da organização e os processos de Software de cada projeto definido são melhorados continuamente.

Fonte: CMM (SOFTWARE ENGINEERING INSTITUTE. The Capability Maturity Model : guides for improving the Software process. Reading : Addison Wesley, 1997.

Os benefícios do uso de uma Metodologia de Desenvolvimento são maximizados quando a empresa tem sua MDS "encaixada" no contexto mais amplo do Processo de Software. O esforço por aumentar sua Capacitação Tecnológica, através do treinamento e incentivo aos desenvolvedores, fará com que a transição entre os vários níveis se dê mais rápida e suavemente. Como vimos o CMM não trata de algumas

questões para que existam projetos bem sucedidos. o CMM não considera por exemplo a experiência em áreas específicas, não trata do uso de tecnologias específicas no desenvolvimento do Software, nem sugere como selecionar, contratar, conseguir motivação ou manter pessoal que seja competente no desenvolvimento dos Softwares. Embora estas questões sejam muito importantes para o sucesso de um projeto, o CMM foi desenvolvido para se prover uma estrutura ordenada e disciplinada onde estarão direcionados assuntos relativos a gerência do projeto do Software e os processos da engenharia do Software. Alguns itens que citei como importantes e não são contemplados pelo CMM serão descritos nesta pesquisa no capítulo III.

## **2.2 ISO 9126: 1991**

A ISO (International Organization for Standardization) é uma federação mundial, integrada por Organismos Nacionais de Normalização, contando com um representante por país. É uma organização não governamental, estabelecida em 1947, da qual a ABNT (Associação Brasileira de Normas Técnicas) é membro fundador, contando atualmente com 132 membros, sendo 90 participantes, 33 correspondentes e 9 subscritos.

Outra norma bastante conhecida e de respaldo entre os desenvolvedores é a norma ISO9126 :1991. Esta norma equivale a

norma brasileira NBR 13596, define seis características de qualidade para produtos de Software.

### 2.2.1 Funcionalidade

Este item se refere à existência de um conjunto de funções que satisfaçam necessidades estabelecidas ou implícitas e suas propriedades específicas. Propõe como sub-características: Adequação, acurácia, interoperabilidade, conformidade e segurança.

Vejamos as sub-características:

Tabela 6– Sub-características de funcionabilidade

<b>Adequação</b>	Se o Software propõe-se a fazer o que é apropriado.
<b>Acurácia</b>	O Software faz o que foi proposto de forma correta?
<b>Interoperabilidade</b>	O Software interage com os Sistemas especificados?
<b>Conformidade</b>	O Software está de acordo com as normas e legislação necessárias.
<b>Segurança</b>	O Software evita acesso não autorizado aos dados?

### 2.2.2 Confiabilidade

Refere-se à capacidade do Software manter seu nível de desempenho, sob condições estabelecidas, por um determinado período de tempo. Propõe como sub-características: Maturidade, tolerância a falhas e recuperabilidade.

Vamos às sub-características:

Tabela 7– Sub-características de confiabilidade

<b>Maturidade</b>	Com que frequência o Software apresenta falhas?
<b>Tolerância a falhas</b>	Se apresenta falhas, como ele reage a estas falhas?
<b>Recuperabilidade</b>	O Software é capaz de recuperar dados em caso de falhas?

### 2.2.3 Usabilidade

Trata do esforço necessário ao uso e homologação individual de tal uso por um conjunto de usuários estabelecidos ou subentendidos. Propõe como subcaracterísticas: Inteligibilidade, apreensibilidade e operacionalidade.

Tabela 8– Sub-características de usabilidade

<b>Inteligibilidade</b>	É fácil de entender o conceito e a aplicação do Software?
<b>Apreensibilidade</b>	É fácil aprender a usar o Software?
<b>Operacionalidade</b>	É fácil de operar e controlar o Software?

### 2.2.4 Eficiência

Refere-se ao relacionamento entre o nível de desempenho do Software e a quantidade de recursos utilizada, sob condições estabelecidas. Propõe as seguintes subcaracterísticas: Comportamento no tempo e comportamento dos recursos.

Tabela 9– Sub-características de eficiência

<b>Tempo</b>	Qual o tempo de resposta, a velocidade de execução, é aceitável?
<b>Recursos</b>	Quanto recurso o Software utiliza, quanto tempo?

## 2.2.5 Manutenibilidade

Refere-se ao esforço necessário para fazer modificações específicas no Software. Propõe as seguintes subcaracterísticas: Analisabilidade, modificabilidade, estabilidade e testabilidade.

Tabela 10– Sub-características de manutenibilidade

<b>Analisabilidade</b>	É fácil encontrar a falha no Software quando ocorre?
<b>Modificabilidade</b>	É fácil modificar e adaptar o Software?
<b>Estabilidade</b>	Há grande risco quando se faz alterações?
<b>Testabilidade</b>	É fácil testar quando se faz alterações?

## 2.2.6 Portabilidade

Trata da habilidade do Software ser transferido de um ambiente operacional para outro. Propõe as seguintes sub-características: Adaptabilidade, instalabilidade, conformidade e padrões de portabilidade e substituíbilidade.

O que trata cada sub-característica?

Tabela 11 – As sub-características de portabilidade

<b>Adaptabilidade</b>	É fácil adaptar o Software a outros ambientes?
<b>Capacidade de ser instalado</b>	É fácil instalar em outros ambientes?
<b>Conformidade</b>	Está de acordo com padrões de portabilidade?
<b>Capacidade para substituir</b>	O Software pode ser utilizado para substituir outro?

Como deve ter sido fácil perceber, as normas ISO9126 são extremamente subjetivas quando estipulam características e sub-características para avaliação de Software, são conceitos muito vagos e imprecisos para se pontuar um Software quanto à qualidade. Por este motivo, procurou-se enriquecer esta pesquisa com outros tópicos de outras normas ISO e métodos de avaliação de Software. Para pessoas que já tem experiência em desenvolvimento e conhecem Software a mais tempo, fica mais fácil utilizar estas normas propostas acima, porém, mesmo conhecendo Software elas ainda são muito abstratas quanto à avaliação. Algumas características não são claras quanto ao espaço e tempo de teste, com que parâmetros em que ambiente e condições serão testados os Softwares. Uma boa prática é o teste de Softwares utilizando usuários de mesmo nível e mesmos equipamentos sob as mesmas condições. Ainda assim não se deve considerar como claras estas formas de teste. Estas normas servem de orientação, como lista de checagem para que não se esqueça de alguns tópicos. Não somente o autor percebe esta deficiência da falta de subsídios e padrões para testes. As normas ISO também e estão desenvolvendo outras guias e itens para que os testes de Software sejam cada vez mais eficazes. Até o momento desta pesquisa, algumas destas normas ainda não tinham sido publicadas no Brasil mas acredita-se que ajudariam bastante para validarem os critérios de avaliação propostos no capítulo III.

Veja-se, a título de complemento, um exemplo da ISO14598, onde deve-se ainda lembrar que, até o momento, estas normas ainda

não tinham sido traduzidas e publicadas no Brasil, e que esta nova norma trará muitos recursos interessantes aos avaliadores, já que trata o processo de avaliação em grande detalhe. Ela leva em conta a existência de três grupos interessados em avaliar um Software, o que define os três tipos básicos de certificação:

Tabela 12– Definição dos tipos básicos de certificação

<b>Certificação</b>	<b>Quem realiza</b>	<b>Finalidade</b>
de 1a. parte	Empresas que desenvolvem Software	Melhorar a qualidade de seu próprio produto
de 2a. parte	Empresas que adquirem Software	Determinar a qualidade do produto que irão adquirir
de 3a. parte	Empresas que fazem certificação	Emitir documento oficial sobre a qualidade de um Software

Esta norma se constituirá, na verdade, de seis documentos distintos, relacionados entre si. Veja:

Tabela 13– Documentos básicos da certificação

<b>Norma</b>	<b>Nome</b>	<b>Finalidade</b>
14598-1	Visão Geral	Ensina a utilizar as outras normas do grupo
14598-2	Planejamento e Gerenciamento	Sobre como fazer uma avaliação, de forma geral
14598-3	Guia para Desenvolvedores	Como avaliar sob o ponto de vista de quem desenvolve
14598-4	Guia para Aquisição	Como avaliar sob o ponto de vista de quem vai adquirir
14598-5	Guia para Avaliação	Como avaliar sob o ponto de vista de quem certifica
14598-6	Módulos de Avaliação	Detalhes sobre como avaliar cada característica

Em resumo, esta nova norma complementar a ISO/IEC 9126 e permitirá uma avaliação padronizada das características de qualidade de

um Software. É importante notar que, ao contrário da 9126, a 14598 vai a detalhes mínimos; incluindo modelos para relatórios de avaliação técnica para medição das características, documentos necessários para avaliação e fases da avaliação. Isso é bastante importante pois, como disse anteriormente a norma ISO9126 não trazia detalhes sobre a avaliação e esta norma 14598 irá corrigir esta deficiência.

Outra norma ISO igualmente importante, sobretudo para quem está interessado em comprar “pacotes” de Software, ou seja, adquirir um Software pronto é a ISO12119, ela foi publicada em 1994 e além de estabelecer alguns requisitos de qualidade para Softwares deste tipo, ela descreve algumas instruções para teste destes Softwares, considerando estes requisitos de qualidade. Ela se divide em itens da seguinte forma:

Tabela 14– Requisitos de qualidade

Item	Descrição
1. Escopo	
2. Definições	
3. Requisitos de qualidade	
3.1. Descrição do Produto	Descreve o produto, de forma a ajudar o comprador em potencial, servindo como base para testes. Cada declaração deve ser correta e testável. Deve incluir declarações sobre funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.
3.2. Documentação do usuário	Deve ser completa, correta, consistente, fácil de entender e capaz de dar uma visão geral do produto.
3.3. Programas e dados	Descreve em detalhes cada uma das funções do Software, incluindo declarações sobre funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.
4. Instruções para teste	
4.1. Pré-requisitos de teste	Lista de itens necessários ao teste, incluindo documentos incluídos no pacote, componentes do sistema e material de treinamento.
4.2. Atividades de teste	Instruções detalhadas sobre os procedimentos de teste,

	inclusive instalação e execução de cada uma das funções descritas.
4.3. Registro de teste	Informações sobre como os testes foram realizados, de tal forma a permitir uma reprodução destes testes. Deve incluir parâmetros utilizados, resultados associados, falhas ocorridas e até a identidade do pessoal envolvido.
4.4. Relatório de teste	Relatório incluindo: identificação do produto, hardware e Software utilizado, documentos utilizados, resultados dos testes, lista de não conformidade com os requisitos, lista de não conformidade com as recomendações, datas, etc.

Um das grandes vantagens desta norma está na profundidade com que são descritas cada uma das características e sub-características mencionadas na norma 9126. A norma inclui detalhes que devem estar presentes no produto, tais como:

- Documentação do usuário de fácil compreensão
- Um sumário e um índice remissivo na documentação do usuário
- Presença de um Manual de instalação com instruções detalhadas
- Possibilidade de verificar se uma instalação foi bem sucedida
- Especificação de valores limites para todos os dados de entrada, que deverão ser testados
- Operação normal mesmo quando os dados informados estão fora dos limites especificados
- Consistência de vocabulário entre as mensagens e a documentação
- Função de auxílio (help) com recursos de hipertexto

- Mensagens de erro com informações necessárias para a solução da situação de erro
- Diferenciação dos tipos de mensagem: confirmação, consulta, advertência e erro
- Clareza nos formatos das telas de entrada e relatórios
- Capacidade de reverter funções de efeito drástico
- Alertas claros para as conseqüências de uma determinada confirmação
- Identificação dos arquivos utilizados pelo programa
- Identificação da função do programa que está sendo executada no momento
- Capacidade de interromper um processamento demorado

Outras características importantes são a ênfase nos testes e os modelos de relatórios incluídos. Tudo isso facilita bastante o trabalho de quem estará avaliando o Software. Até o momento desta pesquisa não foi possível ter acesso ou conhecimento de uma versão traduzida desta norma pela ABNT (Associação Brasileira de Normas Técnicas).

Acredita-se que os critérios e modelos de avaliação descritos neste capítulo auxiliarão a validar os critérios descritos no capítulo III assim como servem de referencial e de observação para verificarmos que alguns organismos internacionais e nacionais estão preocupados com a aquisição e desenvolvimento dos Softwares. Esta preocupação tem se tornado comum, mesmo em empresas privadas estas normas já

tem sido utilizadas e discutidas, isso mostra que o processo de desenvolvimento de Software tem evoluído e tem sido amplamente discutido entre desenvolvedores e clientes. Estes fatores são bastante positivos pois demonstram preocupação e objetivos claros de se buscar a qualidade e produtividade em Software.

A medida e condições de se definir critérios e avaliá-los mereceu destaque em PRESMAN, 1995 apud Lorde Kelvin:

*“Quando se pode medir aquilo sobre o qual se está falando e expressá-lo em números, sabe-se alguma coisa sobre o mesmo; mas quando não se pode medi-lo, quando não se pode expressá-lo em números, o conhecimento que se tem é de um tipo inadequado e insatisfatório; este pode ser o começo do conhecimento, mas dificilmente alguém terá avançado em suas idéias para o estágio da ciência.”*

Acredita-se que os critérios estabelecidos acima nos auxiliarão a avaliar com mais certeza e critérios os Softwares que porventura tivermos que desenvolver ou adquirir.

A norma ISO9000-3 trata do desenvolvimento, fornecimento e manutenção de Software e julguei importante estar citando-a aqui. Ela é utilizada para garantir que o fornecedor de Software atenda a requisitos especificados nas etapas de desenvolvimento do Software. Tais etapas como: Projeto, desenvolvimento, codificação, implantação e suporte.

A norma ISO 9000-3 traz alguns roteiros para aplicar a ISO 9001 especificamente na área de desenvolvimento, fornecimento e manutenção de Software. Todas as orientações giram em torno do contrato onde teremos o cliente ou seja, a empresa que está contratando os serviços e a empresa contratada para desenvolver o Software em

questão. Acompanhe na tabela abaixo os processos definidos na ISO

9000-3:

Tabela 15– Processos ISO 9000

<b>Grupo</b>	<b>Atividade</b>
Estrutura do Sistema de Qualidade	Responsabilidade do fornecedor; Responsabilidade do comprador e Análise crítica conjunta
Atividades do Ciclo de Vida	Análise crítica do contrato; Especificação dos requisitos do comprador; Planejamento do desenvolvimento; Projeto e implementação; Testes e validação; Aceitação; Cópia, entrega e instalação; Manutenção
Atividades de Apoio	Gerenciamento de configuração; Controle de documentos; Registros da qualidade Medição; Regras, convenções; Aquisição; Produto de Software incluído Treinamento

Após a implantação da ISO9000-3, a empresa desenvolvedora necessitará de seguir vários passos para alcançar a certificação, passos estes que não serão citados aqui por julgar dispensáveis para o objetivo principal desta pesquisa.

## Capítulo III

# CRITÉRIOS DE AVALIAÇÃO

### 3.1 Objetivo

Estabelecer regras básicas para avaliação de Software tendo como referência as normas ISO e alguns critérios do CMM. A utilização destes critérios devem ser acompanhadas das definições sobre quais empresas os Softwares serão implantados e qual o público alvo usuário dos Softwares.

Escolher um Software é tarefa difícil e um processo delicado e cheio de riscos, tanto para a empresa quanto para quem escolhe. Afinal, a partir desta decisão serão investidos tempo, esforços e dinheiro.

Vários fatores identificam e influenciam na escolha de um bom Software.

Entre vários outros critérios de avaliação, podem-se considerar alguns que serão demonstrados a partir deste momento.

### 3.2 Alguns Critérios de Avaliação do Software

#### 3.2.1 Avaliação baseado em heurísticas

Podemos avaliar um Software de forma informal ou seja, simplesmente testando-o e verificando se atende nossas necessidades. Porém, este método é bastante ineficaz e arriscado e só deve ser

utilizado quando outros critérios mais “científicos” não puderem ser utilizados.

### 3.2.2 Observação do usuário utilizando sistemas espões

Uma das formas das formas de se avaliar um sistema quanto à sua facilidade de utilização ou seja, quanto à sua ergonomia e usabilidade é a utilização de sistemas espões. Sistemas espões são Softwares que “ligamos” em um computador, sem que o usuários saiba, por um período e depois que o usuário utilizar o sistema que desejamos avaliar por um tempo, “desligamos” o sistema espião. Captamos a execução do Software que desejamos avaliar e verificamos a facilidade que o usuário teve ao realizar suas tarefas, sua desenvoltura ao trabalhar com o Software e verificamos também vários aspectos ergonômicos e de usabilidade. Estes Softwares são eficazes quando não queremos que o usuário sofra influência de nossa presença na realização do teste de usabilidade e ergonomia. Este critério é bastante difícil de ser utilizado devido à sua complexidade para que possa realmente funcionar. Não é comum, principalmente para usuários de Software, conhecer estas ferramentas, porém, estas ferramentas podem ser utilizadas por analistas ou programadores para identificar o grau de facilidade do usuário com o Software.

### 3.2.3 Avaliação da utilização do sistema mensurando a carga de trabalho com e sem o sistema, se a implantação do sistema aumentou ou diminuiu esta carga.

A implantação de um Software, seja ele qual for, deve mudar a realidade e o dia a dia das pessoas, seja para melhor ou para pior, o que se deseja é que ocorra sempre a primeira situação. Eu digo sempre que a informatização de uma empresa, departamento ou de qualquer trabalho que seja, deve primeiramente focar o ser humano. A implantação de qualquer forma informatizada de trabalho deve se preocupar com a qualidade de vida das pessoas envolvidas no processo. Se, ao implantar Softwares ouvirmos coisas do tipo: “Desde que este sistema foi implantado eu tenho feito horas extras”, “Ao invés de reduzir meu trabalho, ele aumentou”, “Eu alimento este sistema com tantas informações e no entanto, tenho poucos recursos em troca”. Se esta situação perdurar por muito tempo, algo está errado!! Algumas variáveis devem ser analisadas. Se a empresa está tendo que se adaptar ao Software implantado; Se, realmente o fluxo de informações da empresa deveria ser modificado e não foi; se as pessoas envolvidas no processo estão nas funções onde deveriam estar ou seja, se posições de chefia ou coordenação estão tendo que fornecer informações demais e isso deveria ser feito por pessoas mais operacionais; se realmente o Software implantado fornece, ao final do fluxo mais informações; ou ainda se realmente o Software solicita informações em demasia, sem necessidade e isso acaba por penalizar quem está utilizando. Esta

avaliação deve ser cuidadosa, no tempo e momento certo porque, um aumento da carga de trabalho momentânea, principalmente no início da implantação ou no início da utilização do Sistemas é comum, devido a falta de afinidade entre usuário e Sistema, a demora na execução das tarefas, mas isso não pode perdurar por muito tempo senão, algo realmente estará errado, deve-se verificar, como detalharemos a seguir se o treinamento foi eficaz.

### 3.2.4 Qualidade do sistema quanto à qualidade do treinamento empregada aos usuários do sistema

Boa parte do êxito da implantação de um Software está no treinamento dos usuários do Sistema, implantar um Software simplesmente “instalando” os programas em um computador e fornecendo senhas para os usuários é sinônimo de fracasso na utilização do Software. Ter os usuários mal treinados é dar margem para fracassar a implantação, e se a implantação iniciar mal ficará difícil e exigirá um grande esforço para reverter este processo. Treinamento de má qualidade irá provocar, entre outros problemas, na sub-utilização do Sistema, problemas de perda de tempo na execução das tarefas, demora ou mesmo não obtenção dos resultados do Software, execução de rotinas de forma errada, desgaste dos usuários e do Software. Um

treinamento mal feito ou não realizado pode até mesmo paralisar a implantação de um Sistema podendo, por fim tendo até que se reiniciar todo processo.

### 3.2.5 Avaliação de lista de clientes que utilizam o sistema realizando pesquisa de satisfação

Muitas das vezes, quando vamos pesquisar sobre um determinado Software, o que nos angustia é a falta de referências sobre o produto, mesmo com a explanação da empresa fornecedora demonstrando todos os benefícios que seu produto nos trará, com os *folders* bonitos e explicativos, com uma demonstração que muitas vezes impressiona, o que queremos mesmo é verificar o funcionamento do produto na prática, ou seja, queremos verificar o produto funcionando no dia a dia, suas vantagens e desvantagens, seus pontos positivos e negativos, e, para obtermos estas informações, temos que recorrer a lista de clientes da empresa fornecedora, o tempo que utiliza o Sistema e sua satisfação com relação ao produto ofertado. Talvez, esta seja uma das formas mais eficazes de verificarmos se o produto realmente atende nossas expectativas ou não. Com base em uma relação de clientes em mãos e uma pesquisa com estes clientes, teremos um índice de acerto muito maior e tranquilo. É de acordo com estes contatos que poderemos portanto, estar negociando com o fornecedor uma possível adequação do Software às nossas necessidades. Neste item, também é relevante avaliar o porte dos clientes deste fornecedor, se são de porte comparável com o da empresa que pretende adquirir o Software.

### 3.2.6 Realização de pesquisa para verificar quantas cópias do sistema já foram comercializadas

Um número razoável de cópias vendidas de determinado produto de Software é garantia de continuidade do produto. Esta pesquisa deve ter critérios bem definidos e deve ser feita com muito cuidado pois, um produto de Software pode ser novo e, então pode não ter muitas cópias vendidas, neste caso então pode não ser sinônimo de produto de má qualidade. Se isso ocorrer, devemos partir para a avaliação do fornecedor, seus produtos anteriores, clientes e outros fatores que estaremos avaliando sobre fornecedores. Mas se tivermos um produto com um número razoável de cópias vendidas, isso nos trará uma segurança maior na aquisição do Software, isso nos assegurará a continuidade do produto e conseqüentemente a garantia do atendimento.

### 3.2.7 Avaliação da portabilidade do sistema sob o ponto de vista de equipamentos necessários para a implantação do sistema (pré-requisitos)

Muitas vezes, vimos usuários reclamando que um Software não funciona, que demora na execução das tarefas, que tarefas antes realizadas por outro sistema em determinado tempo, agora demorava mais do dobro para ser executada. Esta queixa poderia ser evitada se, no momento da implantação ficasse bem claro quais seriam os equipamentos necessários para a implantação do Software, muitas

empresas, para não investirem em equipamentos, somente mudam o sistema sem se preocupar com a estrutura de *hardware* necessária para se implantar uma nova versão ou um novo produto de Software. Vemos empresas que utilizavam *Softwares* baseados em modo caracter e passa-se a utilizar *Softwares for windows*, ou seja, que não exigia muito da máquina para que os usuários pudessem trabalhar, funcionavam bem em equipamentos básicos e, em função de economizar dinheiro simplesmente mudam o Software e não se preocupam com o Hardware na qual o Sistema irá funcionar. Economia errada e sem fundamentos, a economia realizada com Hardware é, paga em mão de obra, falta de recursos, demora na apresentação dos resultados e outros problemas.

A definição clara de quais equipamentos básicos o Sistema deve funcionar é premissa básica para uma implantação segura e um perfeito funcionamento do Software.

### 3.2.8 Avaliação da portabilidade sob ponto de sistema de sistema operacional necessário para a implantação do sistema

Assim como a questão "*hardware*" é muito importante para o bom funcionamento de um Sistema, o sistema operacional a ser utilizado também o é. Temos dois pontos principais a serem considerados: O primeiro é que um Software funcionando sobre um Sistema Operacional para o qual não foi homologado, ou seja, não foi testado pode ter problemas e não funcionar direito. Por isso é importante que o fabricante

de Software deixe bem claro quais Sistemas Operacionais são suportados por seu produto. O segundo ponto, não menos importante, é da evolução que o Software pode sofrer. Por exemplo: temos uma empresa que deseja implantar determinado Software, esta empresa é uma empresa pequena mas com perspectivas de crescimento. Se o Software a ser comercializado ou adquirido somente funcionar em Windows95 por exemplo, quando for necessário portá-lo para uma plataforma mais evoluída, para um Sistema Operacional de rede melhor, e ele não permitir esta evolução, isso trará um problema para a empresa que adquiriu o Software que pode perder todo investimento realizado, treinamento e nas informações que ali estão armazenadas.

### 3.2.9 Avaliação do banco de dados que o sistema foi desenvolvido principalmente para verificar se o banco de dados oferece perspectivas de atualização

Quando pensamos em investir em tecnologia, deve estar em nossa mente que a Informática assim como toda tecnologia de Informação evolui de forma avassaladora, se iremos investir, temos que investir em algo que nos permita estar atualizados por um tempo considerável. Se fossemos investir em um banco de dados hoje por exemplo, viria sempre em nossas cabeças algo como Oracle, Informix, Sybase, SQL Server que são bancos de dados modernos, eficazes e que estão sempre inovando e buscando novas implementações, bancos de

dados que tem se mantido atualizados com as novas necessidades e novos recursos da Informática.

Então, adquirirmos um Software com banco de dados ultrapassado é correr risco de se ter realizado investimento sem retorno desejado.

### 3.2.10 Avaliação da linguagem no qual o sistema foi desenvolvido para o mesmo fim

Assim como no caso do Banco de dados, a linguagem na qual o Software foi desenvolvido é de vital importância para que possa sofrer implementações e atualizações sem problemas. É importante para quem está adquirindo o Software, estar observando os perfis de atualização e de Interface do Banco de Dados com a Linguagem. Não se tem uma regra para definição destes parâmetros, o que se deve observar é da própria condição de renovação e das diretrizes que estão sendo seguidas pelo fabricante da linguagem na qual o Software foi desenvolvido. Por exemplo: Em tempos onde a utilização da Web, comunicação de dados via Internet ou Linhas privadas, a condição de se estar utilizando este Software via Internet ou por outros meios de comunicação que possam ser criados ou sua adaptabilidade nestes meios é importantíssimo. É claro que não temos pretensão nenhuma, e não temos a mínima condição de estarmos prevendo algo, o que seria

um ato insano em dias onde as tecnologias surgem e desaparecem num piscar de olhos. O que digo é que, assim como no caso do Banco de Dados, adotar linguagens de programação que não acenam para atualizações e busca de se adequarem aos novos tempos é arriscar o investimento realizado.

Este parâmetro pode e deverá ser discutido com o Fornecedor ou com a equipe de desenvolvimento do Software.

### 3.2.11 Avaliação do Software quanto à sua capacidade de parametrização ou seja, se o Software permite que seja parametrizado ou se podemos parametrizar o Software de acordo com nossas regras de negócio

Imaginemos Bill Gates, o grande mago da indústria de Software quando planejou produzir um Editor de textos que pudesse ser utilizado por todo o mundo. Se pensarmos quantas sugestões ele teve dos recursos que deveriam ser disponibilizados em seu Software. Imaginemos então se ele tivesse que criar várias versões o quanto eram diferentes as necessidades e sugestões realizadas para que pudessem ser colocadas em seu editor. Podemos ir além, imaginemos nós, criando um Software que seria utilizado por pessoas tão diferentes, tanto culturalmente como em termos de conhecimento, experiência no uso da Informática e seus recursos, gostos, necessidades, anseios, utilizações diferentes, ou seja, como seria possível desenvolver um Software que

tivesse a pretensão de atender a tamanha diversidade? Mesmo com as diferentes configurações disponíveis no Word ®, sempre nos deparamos com uma situação ou necessidade que poderia estar disponível para personalização. Ou seja, quando formos adquirir um Software, temos que pensar que podemos ter necessidades e anseios diferentes dos desenvolvedores ou possuímos necessidades bem diferentes das pessoas que auxiliaram no desenvolvimento do Software. Para podermos amenizar este problema, se podemos considerá-lo assim, somente com um Software que possua várias das suas configurações parametrizáveis ou seja, com capacidade de se adequar à pessoas, necessidades e situações diferenciadas. Nem sempre isso é tarefa fácil, mas dizemos que Software perfeito não existe, a menos que ele seja desenvolvido para você, mesmo assim, ao término deste desenvolvimento sempre encontramos problemas ou situações indesejáveis. O que podemos verificar e estar escolhendo Softwares que nos atendam na maioria de nossas necessidades e situações, ou seja, que possua parâmetros configuráveis e opções que possam ser personalizadas. Assim poderemos ter menos problemas com a adaptabilidade do Software com nossa rotina de trabalho.

3.2.12 Verificação do Software quanto à sua capacidade de integração com outros Softwares de sua empresa, deve-se verificar se o Software é amigável neste sentido.

A integração entre dois sistemas diferentes pode se dar de duas formas principais: De forma *on-line*, que não é a forma mais comum ou *off-line* ou seja, gerando-se arquivos do Software origem para que o Software destino possa importar estes dados para sua base. A forma *on-line* ocorre normalmente quando se tem sub-sistemas ou módulos da mesma empresa fornecedora ou quando se têm Softwares de empresas diferentes mas que tem acordo operacional. A forma *on-line*, em linhas gerais pode ser descrita como a integração total ou seja, quando um lançamento é realizado em um módulo, imediatamente é realizada a gravação das informações necessárias no outro módulo. Podemos exemplificar isso melhor quando é realizado o lançamento de uma Nota Fiscal de entrada de produtos por exemplo e automaticamente é gravado um registro no módulo de Contas a Pagar. Isso é bastante útil em termos operacionais porque evita-se uma série de erros, as informações serão únicas tanto em um módulo quanto no outro e evita-se o retrabalho ou a digitação da mesma informação diversas vezes para que se possa alimentar os módulos diferentes do Sistema. A outra forma que citamos acima, é a integração por geração de arquivos ou por meio magnético. A operação funciona da seguinte maneira:

O produtor do Software que receberá as informações ou o que importará os dados define um layout ou o posicionamento das informações e seu posicionamento neste arquivo. O produtor do Software origem então, desenvolve um programa que possa gerar estas informações da forma e posicionamento exigido pelo Software destino,

estas informações são geradas, geralmente em um arquivo do tipo texto que é o tipo mais simples de se gerar e interagir e estas informações são importadas pelo Software destino, também sem a necessidade da redigitação das informações. A diferença básica entre estas duas formas é o momento ou seja, as informações somente estarão disponíveis no Software destino a partir do momento que são importadas e não instantaneamente como na forma de importação *on-line*.

A capacidade do Software possibilita, entre várias outros detalhes, facilitar a geração de diversas informações gerenciais e evita erros e divergências nas informações.

### 3.2.13 Facilidade de se gerar novos relatórios e consultas ou seja, a capacidade do Software permitir que se extraia dele dados diversos sobre sua empresa, criação de relatórios pelos usuários que não estejam disponíveis no Software padrão do fornecedor

Este fator de avaliação não é vital para o bom funcionamento normal do Software mas é um ponto que pode ser considerado como diferencial perante outros a serem avaliados. Não são raras as vezes que os usuários de Software se frustram perante uma necessidade não atendida. A capacidade dos Software de se gerar relatórios e consultas específicas é um ponto a favor daqueles que permitem este recurso. A informação precisa do dicionário de dados e da localização facilitada das informações é bastante importante para que se possa realizar consultas

e relatórios específicos. Em vários Softwares este recurso está presente porém, o grau de dificuldade para se gerar novas consultas e relatórios é um impecilho para usuários novatos e inexperientes. Facilitar o acesso às informações e possibilitar que se extraia delas relatórios gerenciais estatísticos e gerenciais específicos é um ponto bastante favorável. Em dias onde o acesso a informação de forma rápida e precisa é crucial, as alternativas para que se customize este acesso sem dúvida deve ser considerado e ponderado na escolha do Software. Principalmente porque se tivermos este recurso o custo para futuras customizações será reduzido por não necessitar de auxílio da empresa desenvolvedora para estes novos recursos.

### 3.2.14 Verificação da capacidade do Software trabalhar em rede ou em ambiente multiusuário

Nos primórdios tempos do surgimento dos PC's (Personal Computer), na era pós *Mainframes*, quem tinha suas informações disponíveis em seu próprio PC eram considerados como pessoas importantes já que em um momento anterior da história da Informática o que víamos eram os "todo poderosos" dos CPDs (Centro de Processamento de Dados) que detinham todo poder nas mãos e então, os usuários dos PC's decretaram sua independência momentânea. Esta independência durou até que estes mesmos usuários perceberam que de nada valia a informação isolada e que era muito difícil para os dirigentes quando necessitavam de agrupar dados dos diferentes setores

das empresas e que estas informações teriam maior valor agregado quando estivessem em um mesmo local ou interligadas. Foi neste momento que os projetistas e engenheiros de Software e de comunicação de dados teriam uma tarefa importante a ser cumprida, interligar os computadores de forma rápida e segura de modo que todos pudessem ter acesso às informações que necessitassem. Neste momento perceberam a importância das redes e dos Sistemas Multiusuários. Um Software que não tem a capacidade de funcionar de modo multi-usuário ou em rede deve ser retirado de seu elenco para escolha. Mesmo que imediatamente você não irá utilizar este Software em rede, a disponibilidade deste recurso é importante porque as empresas evoluem e após a implantação do Software surge uma demanda cada vez maior de usuários e de mais pessoas querendo realizar seus controles de forma informatizada e esta condição é muito importante. Considero este critério de vital importância na escolha do Software.

### 3.2.15 Verificar integridade e segurança do Software, pelo banco de dados utilizado ou pela própria concepção do produto

Imagine você, tendo suas informações acessadas por qualquer pessoa, todos tendo acesso a informações estratégicas da empresa. A segurança de um Software e a integridade das informações é fator primário em qualquer avaliação. Pelos administradores, o acesso aos

dados disponíveis no Software deve ser restrito e também deve-se ter níveis de acesso passíveis de serem alterados. O acesso ao banco de dados também deve ser restrito aos administradores do Sistema. Sobre a integridade dos dados, que se entende como a condição de que os dados constantes no Banco de Dados estiverem corretos e confiáveis, deve-se ter condições ou maneiras de se corrigir possíveis corrupções do Banco de Dados assim como a capacidade de que o Banco de Dados possa, de alguma forma, acusar o corrompimento de sua base.

### 3.2.16 Avaliação do processo de desenvolvimento do Software, ou seja questionar se eles tem um processo de desenvolvimento definido

Verificar a existência na empresa desenvolvedora de documentação atualizada sobre o Software, como funcionam as revisões, se são documentadas, como são realizadas alterações no Software, se existem controles destas alterações ou seja, se analisarmos um pouco os níveis do CMM poderemos estar questionando uma série de itens e garantindo que se esteja adquirindo um produto de Software de boa qualidade. É importante verificar a organização da equipe de desenvolvimento, qual o papel de cada um e quem serão os contatos durante o desenvolvimento e implantação.

## 3.3 Alguns critérios quanto à implantação

### 3.3.1 TurnOver, verificar a rotatividade de pessoal antes, durante e depois da implantação do sistema

Em termos de implantação, é importante verificar em outras empresas ou clientes que implantaram o Software, em se tratando de fornecedor externo, se houve um *turnover* maior durante ou após a implantação do Sistema. Entenda-se por *turnover*, a rotatividade do pessoal na empresa. Mas porque considerarmos este ponto como critério de avaliação? Toda implantação de sistema gera mudanças nas empresas e é importante verificar se estas mudanças foram favoráveis ou não. Um alto índice de rotatividade de pessoal durante e após a implantação do Software deve ser avaliada e investigada. Uma alta rotatividade pode indicar falhas na implantação ou radicalismo excessivo ou ainda falta de adaptabilidade do Software às rotinas empresariais. Pode-se também identificar possíveis problemas na equipe que estará implantando o Software, mudanças radicais nas rotinas empresariais pode causar problemas para as empresas que desejam implantar tal Software. Este critério, se considerado deve ser avaliado com bastante cuidado e investigado a fundo nas empresas que tiveram este ponto como um problema. Nem sempre os problemas estão nos Softwares mas sim, podem estar nas próprias empresas onde o Software está sendo implantado ou até mesmo nos fornecedores do Software.

### 3.3.2 Equipamentos utilizados, verificação de clareza nos pré-requisitos para implantação do sistema

A utilização de equipamentos inadequados pode comprometer a implantação e o funcionamento do sistema. Esta definição deve estar muito clara e de acordo com o porte e capacidade de investimento da empresa. Uma definição mal feita ou indefinida pode comprometer toda implantação. Quando não se deixa claro as necessidades e pré-requisitos para o bom funcionamento do Software tanto de computadores como de recursos de comunicação de dados, a implantação pode sofrer problemas durante seu andamento. Se no início dos trabalhos equipamentos necessários não se tornarem disponíveis poderemos ter problemas durante a implantação que pode inviabilizar a implantação ou trazer problemas e atrasos no cronograma das atividades. Este tópico, quase sempre é deixado de lado porque envolve investimentos e as empresas normalmente acreditam que investir no Sistema e nos Softwares de base (Bancos de dados e Linguagem) já é o suficiente para que se possa concluir a implantação. Este item não é necessariamente uma obrigação do fornecedor do Software mas sim, da equipe pertencente a empresa que deseja implantar o Software e deve ser analisado com muito critério juntamente com a empresa, fornecedores e com quem está comprando o Software para que não se tenha problemas. Embora este tópico já foi citado na parte sobre o Software,

resolvi citá-lo aqui também por ser igualmente importante para a implantação.

### 3.3.3 Observação de horários de trabalho antes da implantação do sistema e após, verificar se a implantação do sistema reduziu a carga de trabalho ou provocou mais horas extras

Assim como no tópico tratado sobre *turnover*, normalmente quem informatiza uma empresa deve ter como objetivo principal a melhoria da qualidade de vida no trabalho de seus colaboradores, da melhoria da qualidade do trabalho e a redução de custos. Quando se implanta um sistema, o que observa é que durante o que se chama de “período de adaptação”, o número de horas trabalhadas normalmente aumenta em função das pessoas ainda não estarem habituadas e adaptadas na utilização do Software, isso gera portanto uma necessidade de mais horas de trabalho. O que se deve observar, é se esse acréscimo das horas não se tornou uma rotina, então, se isso realmente ocorrer, tem se que se avaliar de forma isenta onde poderia estar o problema. Se é realmente problemas com o Software, com as pessoas envolvidas no processo, se foram treinadas e se são realmente capacitadas para as tarefas que estão desempenhando, se não existem tarefas que antes do Software ser implantado não se tornam mais necessárias e continuam sendo feitas. Este ponto, a eficácia da implantação, normalmente é verificado com outros clientes da empresa

através de pesquisa ou contatos informais e também deve ser avaliado durante a implantação. O responsável pela implantação do Software deve estar atento para estas questões pois senão, poderemos estar atingindo somente a melhoria da qualidade do serviço prestado e não alcançando a melhoria de qualidade de vida dos colaboradores e redução de custos ou, até mesmo, nem mesmo a melhoria da qualidade do serviço.

#### 3.3.4 Planejamento da Implantação

A implantação não deve ser iniciada enquanto não se tiver um planejamento bem definido em mãos. Mesmo durante a implantação é comum ocorrer problemas que nos levam a alterar o fluxo da implantação ou dos trabalhos, mesmo estas correções de rumos devem ser discutidas com todos dirigentes ou pessoas “chave” envolvidas no trabalho para um planejamento de emergência. Nada deve ser feito sem planejamento e organização. Devem participar deste planejamento uma equipa da empresa desenvolvedora e outra da empresa onde será implantado o Software.

#### 3.3.5 Coordenação da Implantação

Determinar um colaborador da empresa para coordenar a implantação do Software é interessante, qualquer implantação deve ter um “pivô” central que possa atuar como regulador das ações entre os desenvolvedores, sejam eles internos ou externos e as pessoas envolvidas no processo de implantação, as pessoas operacionais. Deve estar atento para quaisquer problemas que possam ocorrer, deverá intervir para que tudo corra bem, agir prontamente quando algo ameaçar a implantação e comunicar ou discutir com os responsáveis para que tudo que foi planejado seja cumprido. Esta pessoa deve ser, normalmente de nível médio ou alto dentro da empresa e deve, de preferência conhecer bem os processos que estão sendo informatizados para que possa, com sua experiência “somar” forças em prol da implantação do Software.

### 3.3.6 Realizar a revisão dos processos da empresa e do fluxo de informações antes da implantação

Informatização não combina com desorganização, antes de qualquer tentativa de informatização de uma empresa ou departamento, deve-se, antes do início do processo de implantação propriamente dito, embora considere esta uma etapa do todo, realizar a revisão dos processos da empresa, sua continuidade e necessidade, verificar se o fluxo de informações é eficaz, se algo pode ser alterado antes mesmo da instalação dos programas. Ou seja, tem que se realizar um levantamento

da situação atual e revisar todos processos e fluxo de informações para que durante a implantação não surjam questionamentos de como funcionarão estes processos com o novo Software. Analisar este item antes da implantação só trará benefícios e redução de custos na implantação já que irá racionalizar as horas gastas na implantação além de trazer benefícios visíveis para a organização. Bill Gates, o grande mago da Microsoft já 1995 havia dito em seu livro “A estrada da informação” (GATES, 1995):

*“Se ouço um dono de empresa dizer: “estou perdendo dinheiro, é melhor comprar um computador”, digo-lhe para repensar sua estratégia antes de investir. A tecnologia, na melhor das hipóteses, irá adiar a necessidade de mudanças mais fundamentais. A primeira regra de qualquer tecnologia utilizada nos negócios é que a automação aplicada a uma operação eficiente aumenta a eficiência. A Segunda é que a automação aplicada a uma operação ineficiente aumenta a ineficiência.”*

Ou seja, antes de qualquer tentativa de informatização temos que organizar o que está funcionando atualmente para que a informatização não venha pôr abaixo a empresa que está atualmente funcionando. Em meados da década de 90, não me lembro exatamente o ano, ventilava-se que o Banco Bamerindus iria trocar seus computadores por outros novos, seus computadores estavam funcionando e o Banco estava realizando suas operações normalmente. Porém, com a troca dos computadores e meios de comunicação de dados, com os altos investimentos realizados e os resultados esperados não surgiram, o Banco foi vendido para o HSBC. Com certeza este não foi o único motivo dos problemas financeiros do Banco mas pode em

muito ter colaborado para que seus problemas se agravassem. Ou seja, antes de informatizar, organizar!

### 3.3.7 Comprometimento da Alta Administração ou Direção

Em qualquer implantação de Sistema Informatizado, se não houver o comprometimento de todos os níveis hierárquicos da empresa pode-se ter problemas. Sobretudo se a alta administração ou direção não estiver comprometida e disposta a colocar em funcionamento este Sistema. Este respaldo, perante a direção é importante porque qualquer desvio na implantação, mudança organizacional ou de processo deve passar pela aprovação destas pessoas e se não houver este comprometimento pode-se enfrentar resistência interna e ter problemas com a implantação. É normal durante a implantação de um Software seja ele para substituir um outro ou mesmo para se informatizar tarefas realizadas manualmente uma resistência pelas pessoas envolvidas no processo. As pessoas sempre buscam realizar as tarefas como faziam antes sem perceber que a informatização ou a nova forma de trabalho invariavelmente lhe trará benefícios.

### 3.3.8 Atentar durante a implantação para os custos e cronograma estipulado

Deve-se ter descrito claramente um cronograma para implantação do Software e o custo da implantação. Os custos devem estar claros, inclusive sobre questões como alimentação, locomoção, estadias dos técnicos e horas extras eventuais. O cronograma deve estar descrito de forma bem fácil de ser entendido com prazos definidos para que se evite problemas e que a implantação seja bem sucedida e no prazo esperado.

### 3.3.9 Escolha dos participantes da implantação

Normalmente, toda empresa ou todo o departamento onde o Software será implantado irá se envolver com a implantação porém, escolher pessoas competentes e motivadas para serem pessoas-chaves na implantação provoca resultados bem interessantes, é bastante interessante que se convide para tarefas relevantes, pessoas da empresa que estejam desmotivadas mas que possuam habilidades e competências a serem exploradas, estas pessoas, podem recuperar a auto-estima e criar uma motivação muito especial nestas pessoas além de ser muito produtivo e bem vindo para a implantação do Software

### 3.3.10 Análise dos resultados obtidos na implantação do Software

Após a implantação finalizada, é muito importante e motivador verificar os resultados obtidos, a situação anterior à implantação e a situação atual ocorrida, esta avaliação deve ser feita após um período do término da implantação ou seja, após a adaptação das pessoas à nova realidade, não se tem um prazo estipulado, estima-se que dentro de três meses após a conclusão dos trabalhos de implantação possa-se avaliar os resultados colhidos com a implantação. Deve-se elaborar um relatório confrontando as informações colhidas antes da implantação e depois e apresentá-lo aos colaboradores e à Direção.

### **3.4 Alguns critérios quanto aos custos de manutenção dos sistemas**

#### **3.4.1 Avaliação do custo mensal de manutenção do sistema**

A empresa fornecedora do Software deve ter política definida de preços e condições comerciais, desde o princípio das negociações. A compra de um Software deve considerar condições comerciais claras, devemos verificar também o custo/hora de manutenção e implementação no Software. Este ponto é bastante importante, sobretudo se estamos trabalhando em um empresa que possui muitos de seus processos específicos ou se o Software em questão é específico. Isso se deve à

necessidade de mudanças no produto e conseqüentemente à compra de horas para implementações e customizações no sistema. É importante lembrar que quando se compra um Software, a manutenção será paga por tempo indeterminado e uma negociação bem feita trará grandes benefícios financeiros.

### 3.4.2 Análise do tempo de resposta do fornecedor, para alterações no Software ou em caso de problemas. Se isso fará parte do contrato de manutenção ou não

O tempo de resposta do fornecedor do Software é muito importante para a continuidade da rotina empresarial. Eu considero este critério, também um dos mais importantes a serem considerados quando se deseja adquirir Software. Imagine que você tenha em sua empresa os mais modernos computadores, a mais nova rede de comunicação e o Software mais avançado. Porém, quando o Software apresenta problemas você tem que aguardar horas ou até mesmo dias para que o problema seja resolvido. Se pudéssemos pontuar cada item dentro do elenco de critérios para a escolha de um Software, boa parte dos pontos eu determinaria para este critério. O tempo de atendimento e solução dos problemas ocorridos no Software é muito importante, melhor que não os tenha, mas como sabemos todo recurso tecnológico algum dia poderá apresentar pequenas falhas, sejam elas por sua própria concepção ou por alguma operação irregular. É muito comum estar descrito no contrato de manutenção de Software o tempo de

atendimento máximo e o tempo de resolução do problema que o Software apresentar.

### **3.5 Alguns critérios quanto ao fornecedor do software**

#### **3.5.1 Avaliação do corpo técnico que dará suporte ao sistema**

O atendimento e o suporte técnico devem estar bem estruturados e transmitir segurança para o grupo envolvido com a implantação, incluindo aí o pessoal de informática. O corpo técnico do fornecedor de Software deve ter formação e experiência tal que possa resolver prontamente todos problemas técnicos e possa prover alterações solicitadas no Software. Comprar Software de empresas que não possuam suporte técnico adequado pode comprometer o bom funcionamento e a obtenção de resultados através do Software. Devemos avaliar se a equipe que apoiará na implantação terá conhecimento profundo do Software e se terá disponibilidade para tal. Uma implantação de Software “quebrada” pode por fim a todo processo de implantação. Uma equipe de implantação mal treinada ou não capacitada pode onerar a implantação com consumo de horas além do previsto no início dos trabalhos.

#### **3.5.2 Avaliação da distância da empresa que prestará suporte ao sistema da sede do cliente para efeito de cobrança de deslocamento, alimentação e eventual estadia**

Este item de nossa pesquisa não considero como de muita relevância para quem está interessado em investir em Software porém, coloco-o nesta pesquisa porque não deixa de ser um ponto a ser considerado na compra do Software. Softwares, por melhores e experimentados que são podem ter problemas na sua utilização, são problemas inesperados com arquivos, novas implementações, treinamento de novos funcionários e isso tudo normalmente tem custo para o cliente. Por estes motivos que resolvi inserir nesta pesquisa este item. O cliente, quando for investir o Software deve negociar com o fornecedor o menor custo a hora técnica cobrada, inclusive as horas de deslocamento que, dependendo da distância do fornecedor podem sair mais caras que o trabalho realmente realizado.

### 3.5.3 Devemos avaliar as instalações da empresa fornecedora do Software, quanto à sua estrutura para treinamento e profissionais

Se os treinamentos para implantação do Software forem realizados em sede do fornecedor, devemos avaliar as condições de suas instalações: mesas, cadeiras, iluminação e ventilação adequadas, banheiros, localização da sala com relação à sua empresa para verificarmos facilidades de locomoção, limpeza, equipamentos a serem utilizados, devemos verificar também se os equipamentos a serem utilizados são do mesmo porte ou compatíveis com os que você irá utilizar em minha empresa. Verificar também se os profissionais que

ministrarão os treinamentos são capazes e possuem experiência e conhecimento suficiente do Software na qual estarão treinando. Esta infraestrutura, irá influenciar no rendimento e aproveitamento do treinamento a ser ministrado.

#### 3.5.4 Avaliação da solidez da empresa fornecedora

Um dos fatores importantíssimos na escolha do Software é da solidez e da “certeza” da continuidade da empresa fornecedora e produtora do Software. Uma boa empresa produtora neste sentido, é a empresa que irá permanecer no mercado pelo menos o mesmo tempo que a sua. Escolhermos um Software de uma empresa sem solidez e tradição, podemos estar correndo riscos de ficar sem manutenção e assistência no Software. Um indicador da continuidade da empresa é a solicitação de seu balanço se assim julgar necessário ou até mesmo analisar, como citei em outro tópico do número de clientes da empresa, empresa com muitos clientes, normalmente tem menores chances de “desaparecer” de uma hora para outra.

#### 3.5.5 Estrutura suficiente para acompanhar e dar todo apoio durante a implantação do sistema

Boa parte do sucesso do funcionamento do Sistema depende da implantação. Uma implantação mal feita ou ineficiente pode comprometer todo o funcionamento do Software. Podemos ter Softwares

de boa qualidade, equipamentos bons e pessoas motivadas. Se a empresa ou pessoa que for implantar o sistema não tiver estrutura suficiente para implantar e acompanhar o processo de Implantação, pode pôr tudo por água a baixo. O processo de implantação não é tarefa das mais fáceis, como vimos neste estudo, são inúmeros fatores envolvidos para que se tenha Softwares de boa qualidade e funcionando. São vários pontos a serem considerados, na implantação, as pessoas envolvidas tem que possuir experiência suficiente para coordenar todo processo e agir no momento necessário. Por exemplo: pessoas envolvidas no processo que podem pedir demissão, outras que podem faltar ao trabalho em um momento crucial, equipamentos que podem falhar entre outros fatores que devem ser atentados e observados no momento da implantação. Portanto, uma empresa ou pessoa que for implantar o Software que não tiver experiência suficiente pode por fim e pode comprometer todo processo.

### 3.5.6 Verificar quanto ao fornecedor, referente à sua estrutura e preocupação com a atualização legal dos sistemas, se possui profissionais especializados nesta área

A atualização de Softwares de acordo com a legislação, tem que ser uma preocupação constante em várias áreas tais como: Saúde, Fiscal, Contábil, entre outras... Os fornecedores ou desenvolvedores de Software tem que ter esta preocupação, as empresas ou desenvolvedores tem que ter estrutura e conhecimento necessário para

que se possa estar com os Softwares sempre atualizados de acordo com a legislação vigente e exigida. Em vários casos, Softwares desatualizados podem gerar problemas graves, inclusive com perda monetária considerável. Na área de saúde por exemplo: atualmente, a ANS (Agência Nacional de Saúde) tem liberado várias normas e regulamentos para que se possa estar fornecendo informações estatísticas para seu controle. Softwares que estiverem desatualizados com relação à estas regras podem comprometer ou até mesmo provocar problemas para as empresas que os tiverem utilizando e assim ocorre também em outras áreas. Este aspecto do Software é um dos pontos importantes a serem considerados, e dos mais difíceis e onerosos de serem cumpridos. Para que os Softwares possam estar em conformidade com a legislação exigida, é necessário que se tenha pessoas envolvidas na pesquisa e acompanhamento destas normas e leis para que possam ser interpretadas e implementadas nos Sistemas.

### **3.6 Avaliação de Usabilidade**

Um dos grandes desafios dos desenvolvedores de Software é de se conviver com tamanha inovação tecnológica, dispor estas inovações para os usuários de sistema e ao mesmo tempo manterem uma interface amigável, fácil e agradável de se trabalhar. Porém, mantendo um nível tal que não seja maçante e cansativa para outros usuários mais experientes. Outro ponto bastante importante e desafiador

para os desenvolvedores é também discutido em estudos de IHC (Interação Humano Computador) é a dificuldade que se tem para desenvolver Softwares que sejam aplicáveis e possuam boa usabilidade para usuários diversificados ou seja, quando está se desenvolvendo Softwares que possam ser utilizados por pessoas de diferentes formações, culturas, localidades, costumes. Podemos perceber então, o quanto é realmente difícil se desenvolver Softwares que sejam agradáveis e eficazes na sua utilização e que possam ser utilizados por pessoas diferentes sem problema algum. Quando estava pesquisando para escrever este trabalho fiquei pensando, o quão foi trabalhoso para a Microsoft desenvolver o Word, um Software que é utilizado no mundo todo por pessoas jamais vistas pelos desenvolvedores, de diferentes costumes e experiências profissionais e com o manejo do computador. Nós, que já possuímos alguma experiência no uso deste Software, sabemos que os recursos de que ele dispõe são inúmeros e podemos configurá-los de acordo com nossa necessidade ou anseio. Porém, percebemos também que várias pessoas tem dificuldades em configurar estes diversos recursos. Portanto, o que realmente é complicado é disponibilizar recursos para os usuários e ao mesmo tempo tornar estes recursos acessíveis para diversos níveis de pessoas usuárias dos Softwares.

### 3.6.1 Alguns critérios importantes a serem considerados sobre Usabilidade

Esta parte deste capítulo não tem nenhuma pretensão de ser um guia para avaliar a usabilidade de Softwares por se tratar de assunto muito vasto que por si só já serviria para se elaborar uma dissertação. Tento através destes critérios difundir alguns conceitos de usabilidade e citar alguns importantes que devem ser considerados na aquisição e desenvolvimento de Software. Alguns critérios já foram citados acima nos tópicos de avaliação do Software, portanto, vejamos alguns destes itens de usabilidade:

### 3.6.2 Customização de acordo com necessidades individuais

Segundo Baranaukas e Rocha (Rocha & Baranauskas, 2000), “... *Um modo de tratar essa diversidade é projetar sistemas flexíveis que possam ser “Customizados” de forma a se adequar às necessidades individuais*”, este trecho se refere à dificuldade que se tem em projetar e desenvolver sistemas que possam ser adaptados a diferentes pessoas sob o ponto de vista intelectual e também no que se refere a sua experiência na utilização de computadores e operação de Softwares. Portanto, Softwares não devem somente serem parametrizáveis somente com relação à suas funcionalidades mas também de acordo com suas necessidade e desejos individuais como: Cores de tela, fonte e fundo, teclas de atalho entre outras.

### 3.6.3 Utilização de termos ou palavras que não sejam unânimes em seu significado devem ser evitados

As vezes nos deparamos com Softwares que utilizam termos técnicos ou palavras que possuam significados diferentes dependendo do usuário que estará a frente do Software por exemplo: CRM (Customer Relationship Management) que são Softwares voltados para colher informações sobre as preferências e informações sobre os clientes pode ser confundido por Conselho Regional de Medicina, portanto, tais termos quando utilizados devem vir acompanhados de funções de ajuda para que evite confusões.

### 3.6.4 Presença de objetos e recursos desnecessários na tela

Na tela de trabalho do usuário, só devem estar visíveis os objetos e recursos que realmente puderem ser utilizados pelo usuário ou se estiverem desabilitados, deve-se utilizar cores e contrastes que facilitem esta identificação.

### 3.6.5 Padronização nas telas

O Software como um todo, deve seguir um padrão em suas telas, cores e padrões de operação devem ser padronizadas, por exemplo: Se em uma tela a tecla F2 serve para consultas ela deve servir para a mesma função em todo o Sistema. Se em uma tela as

mensagens de apoio ou advertência aparecem embaixo, elas devem estar presentes em todo Software embaixo. Estas padronizações facilitam o uso do Sistema e proporcionam mais produtividade.

### 3.6.6 Feedback, ou seja, o Software tem que dar retorno

De acordo com BARANAUKAS & ROCHA (Rocha & Baranauskas, 2000), *“Retornar ao usuário informação sobre as ações que foram feitas, quais os resultados obtidos, é um conceito conhecido da teoria da informação e controle”*

Infelizmente ainda encontramos Softwares que deixam o usuário sem saber o que fazer, simplesmente paralisam as atividades sem dar nenhum retorno do que está acontecendo ou que tarefa está realizando, isso é bastante ruim pois deixa o usuário inseguro com relação ao que fazer, principalmente se este usuário for inexperiente.

### 3.6.7 Número de funções e recursos presentes na tela

BARANAUKAS & ROCHA (Rocha & Baranauskas, 2000) dizem o seguinte sobre este item: *“Sempre que o número de funções excede o número de controles, o design torna-se arbitrário e não natural, e complicado.”* Ou seja, o excesso de recursos em uma só tela pode deixar o usuário confuso e atrapalhar a utilização dos inúmeros recursos disponíveis.

### 3.6.8 Contexto geral agradável

O Software deve ser de utilização fácil e o contexto geral deve ser agradável, Softwares com telas muito “carregadas” no que se refere a cores ou excesso de informações para o usuário devem ser evitados. Excesso de avisos sonoros ou assistentes visuais em situações normais irritam o usuário. A operação deve tranquila e natural e a “navegação” entre os campos da tela e entre as telas deve ser uma operação natural.

### 3.6.9 O Software deve ter Presteza

Segundo Cybis (CYBIS, 1999) *“A presteza diz respeito as informações que permitem ao usuário identificar o estado ou contexto no qual se encontra”*.

Ou seja, o Software deve ser bastante claro para o usuário quanto o estágio que se encontra, qual tarefa está realizando, que função do Sistema está executando, digo o usuário e não o Software e principalmente orientá-lo de como retornar ao programa anterior ou aos menus.

Como exposto anteriormente, se fossemos dissertar sobre todos itens de usabilidade teríamos que fazer uma pesquisa somente sobre este assunto. São inúmeros os critérios e extremamente vasto e rico a questão de usabilidade de Software, citei alguns para que se tenha

conhecimento da importância destes tópicos quando se pretende investir em Software.

### 3.7 Sintetização dos critérios de Avaliação

Para Facilitar o entendimento e proporcionar uma melhor visualização dos critérios de avaliação, segue uma tabela contendo todos critérios citados nesta dissertação.

Tabela 16 – Síntese dos critérios de avaliação

<b>Crítérios de Avaliação do Software</b>
1) Avaliação baseado em heurísticas
2) Observação do usuário utilizando sistemas espiões.
3) Avaliação do sistema mensurando a carga de trabalho com e sem o sistema, se a implantação do sistema aumentou ou diminuiu esta carga.
4) Qualidade do sistema quanto à qualidade do treinamento empregada aos usuários.
5) Avaliação de lista de clientes que utilizam o sistema realizando pesquisa de satisfação.
6) Realização de pesquisa para verificar quantas cópias do sistema já foram comercializadas.
7) Avaliação da portabilidade do sistema sob o ponto de vista de equipamentos necessários para a implantação (pré-requisitos).
8) Avaliação da portabilidade sob o ponto de vista do sistema operacional necessário para a implantação do sistema.
9) Avaliação do banco de dados que o sistema foi desenvolvido principalmente para verificar se o banco de dados oferece perspectivas de atualização.
10) Avaliação da linguagem no qual o sistema foi desenvolvido também para verificar perspectivas de atualização.
11) Avaliação do software quanto à sua capacidade de parametrização ou seja, se o software permite que seja parametrizado ou se podemos parametrizar o software de acordo com nossas regras de negócio.
12) Verificação do Software quanto à sua capacidade de integração com outros Softwares de sua empresa, deve-se verificar se o Software é amigável neste sentido.
13) Facilidade de se gerar novos relatórios e consultas ou seja, a capacidade do Software permitir que se extraia dele dados diversos sobre sua empresa, criação de relatórios pelos usuários que não estejam disponíveis no software padrão do fornecedor.
14) Verificar capacidade do Software trabalhar em rede ou ambiente multiusuário.
15) Verificar integridade e segurança do Software, pelo banco de dados utilizado ou pela própria concepção do produto.
16) Avaliação do processo de desenvolvimento do Software, ou seja questionar se eles tem um processo de desenvolvimento definido.

**Cont. Critérios de Avaliação do Software**

- 17) Avaliação baseado em heurísticas
- 18) Observação do usuário utilizando sistemas espíões.
- 19) Avaliação do sistema mensurando a carga de trabalho com e sem o sistema, se a implantação do sistema aumentou ou diminuiu esta carga.
- 20) Qualidade do sistema quanto à qualidade do treinamento empregada aos usuários.
- 21) Avaliação de lista de clientes que utilizam o sistema realizando pesquisa de satisfação.
- 22) Realização de pesquisa para verificar quantas cópias do sistema já foram comercializadas.
- 23) Avaliação da portabilidade do sistema sob o ponto de vista de equipamentos necessários para a implantação (pré-requisitos).
- 24) Avaliação da portabilidade sob o ponto de vista do sistema operacional necessário para a implantação do sistema.
- 25) Avaliação do banco de dados que o sistema foi desenvolvido principalmente para verificar se o banco de dados oferece perspectivas de atualização.
- 26) Avaliação da linguagem no qual o sistema foi desenvolvido também para verificar perspectivas de atualização.
- 27) Avaliação do software quanto à sua capacidade de parametrização ou seja, se o software permite que seja parametrizado ou se podemos parametrizar o software de acordo com nossas regras de negócio.
- 28) Verificação do Software quanto à sua capacidade de integração com outros Softwares de sua empresa, deve-se verificar se o Software é amigável neste sentido.
- 29) Facilidade de se gerar novos relatórios e consultas ou seja, a capacidade do Software permitir que se extraia dele dados diversos sobre sua empresa, criação

<b>Cr�terios Quanto � Implanta�o</b>
<ol style="list-style-type: none"> <li>1) TurnOver, verificar a rotatividade de pessoal antes, durante e depois da implanta�o do sistema.</li> <li>2) Equipamentos utilizados, verifica�o de clareza nos pr�-requisitos para implanta�o do sistema, em todos fatores: Equipamentos, pessoas e software b�sico.</li> <li>3) Observa�o de hor�rios de trabalho antes da implanta�o do sistema e ap�s, verificar se a implanta�o do sistema reduziu a carga de trabalho ou provocou mais horas extras.</li> <li>4) Planejamento da implanta�o.</li> <li>5) Coordena�o da implanta�o.</li> <li>6) Realizar a revis�o dos processos da empresa e do fluxo de informa�es antes da implanta�o.</li> <li>7) Comprometimento da alta administra�o ou dire�o.</li> <li>8) Atentar durante a implanta�o para os custos e cronograma estipulado.</li> <li>9) Escolha dos participantes da implanta�o.</li> <li>10) An�lise dos resultados obtidos na implanta�o do software.</li> </ol>
<b>Cr�terios quanto aos custos de manuten�o do Software</b>
<ol style="list-style-type: none"> <li>1) Avalia�o do custo mensal de manuten�o do sistema.</li> <li>2) An�lise do tempo de resposta do fornecedor, para altera�es no software ou em caso de problemas. Se isso far� parte do contrato de manuten�o ou n�o.</li> </ol>
<b>Cr�terios quanto ao Fornecedor do Software</b>
<ol style="list-style-type: none"> <li>1) Avalia�o do corpo t�cnico que dar� suporte ao sistema.</li> <li>2) Avalia�o da dist�ncia da empresa que prestar� suporte ao sistema da sede do cliente para efeito de cobran�a de deslocamento, alimenta�o e eventual estadia.</li> <li>3) Devemos avaliar as instala�es da empresa fornecedora do Software, quanto � sua estrutura para treinamento e profissionais.</li> <li>4) Avalia�o da solidez da empresa fornecedora.</li> <li>5) Estrutura suficiente para acompanhar e dar todo apoio durante a implanta�o do sistema.</li> <li>6) Verificar quanto ao fornecedor, referente � sua estrutura e preocupa�o com atualiza�o legal dos sistemas, se possui profissionais especializados na �rea.</li> </ol>
<b>Alguns Cr�terios de Usabilidade</b>
<ol style="list-style-type: none"> <li>1) Alguns cr�terios importantes sobre usabilidade.</li> <li>2) Customiza�o de acordo com necessidades individuais.</li> <li>3) Utiliza�o de termos ou palavras que n�o sejam unanimes em seu significado devem ser evitados.</li> <li>4) Presen�a de objetos e recursos desnecess�rios na tela.</li> <li>5) Padroniza�o nas telas.</li> <li>6) Feedback, ou seja, o Software tem que dar retorno.</li> <li>7) N�mero de fun�es e recursos presentes na tela.</li> <li>8) Contexto geral agrad�vel.</li> <li>9) O Software deve ter presteza. (identifica�o do est�gio que se encontra).</li> </ol>

## CAPÍTULO IV CONCLUSÃO

Quando iniciou-se esta pesquisa, confabulou-se em torno de como seria seu desenrolar, que desafios seriam encontrados ao escolher este assunto tão importante e desafiador. As pesquisas para elaboração desta dissertação foram iniciadas e restou a surpresa com a infinidade de critérios e métodos que possibilitariam avaliar Software. A motivação para a elaboração desta pesquisa foi a percepção da dificuldade das pessoas de avaliarem e investirem em Software, captada através de entrevistas com muitos dirigentes verificando seus anseios e dificuldades. Nesta ocasião, vários deles indagavam quanto a segurança que teriam com relação ao investimento realizado, se realmente obteriam resultados positivos, como poderiam investir com segurança, que certeza teriam ao investirem em Software. Outros questionavam com relação a outra parte desta dissertação, se deveriam comprar ou desenvolver Software em suas próprias empresas já que tinham pessoal de Tecnologia de Informação em suas empresas porém, não tinham segurança se este pessoal estaria capacitado e se tinham equipamentos e Softwares adequados para tal desenvolvimento. A acertabilidade da elaboração desta pesquisa adveio da possibilidade de acompanhar e coordenar a implantação de Softwares de gestão em cinco cooperativas de saúde do sul de Minas, onde foi possível comprovar e aferir várias das colocações inseridas nesta pesquisa, o que transmitiu segurança

quanto à estas colocações. De outro modo, ao buscar material acadêmico e científico para poder argumentar as colocações constantes neste trabalho, pode-se conhecer mais profundamente vários métodos de avaliação de Software, de seu processo de desenvolvimento e dos fornecedores. Cremos que os objetivos foram alcançados, pode-se elaborar esta pesquisa em torno do assunto proposto e cremos que esta pesquisa poderá, realmente, dentro de suas limitações orientar dirigentes e/ou pessoas que desejam investir em Software, seja desenvolvendo interna ou externamente assim como comprando estes Softwares prontos. Foi possível concluir também que o assunto, “Critérios de Avaliação” é um assunto em ascensão ou seja, atualmente existe uma maior preocupação com o acerto dos investimentos e com a aquisição de Softwares com maior qualidade. Outra questão observada é que principalmente as instituições regulamentadoras como a ISO e a ABNT tem se preocupado em estabelecer meios e métricas para se medir a qualidade de Software fornecendo subsídios para quem deseja investir em tecnologia.

### **Propostas para trabalhos futuros**

Com a elaboração desta pesquisa, verificamos dentro de suas limitações, várias outras idéias que poderiam servir para trabalhos futuros, tais como:

- Aprofundamento nas normas ISO com elaboração de um manual do desenvolvedor fornecendo-lhe caminhos para a certificação de Software;
- Estudo mais detalhado do CMM, com elaboração de material que possa orientar desenvolvedores a passar de um nível de desenvolvimento para outro com formas de se alcançar o nível 5.
- Detalhamento e aperfeiçoamento desta pesquisa no que se refere a ergonomia e usabilidade;
- Realização de pesquisas de satisfação com Softwares diversos verificando grau de satisfação de usuários de Software.
- Pesquisa à nível regional com empresas desenvolvedoras de Software verificando qual o nível de desenvolvimento do CMM estarão estas empresas.

**BIBLIOGRAFIA**

- ABNT, Normas ISO9000-3, 1993
- ÁTILA Belloquim DEVELOPER'S MAGAZINE 1997
- BAUER, Marcelo, Informática, A revolução dos bytes, 1997, São Paulo, Editora Ática.
- ECO, Umberto. Como se faz uma Tese. São Paulo: Perspectiva, 1991.
- GATES, Bill, A estrada do futuro, São Paulo, Companhia das Letras, 1995
- HOLMAN, H. R. *Qualitative inquiry in medical research*. In: Journal of Clinic and Epidemiology, 46 (1): 29-36,1993.
- STRAUSS, Roy Managing Multimedia Projects, 1997. Focal Press
- LAUDON, Kenneth C.; LAUDON, Jane Price. Sistemas de Informação. Rio de Janeiro: LTC: Livros Técnicos e Científicos S/A, 1999.
- MINAYO, M. C. S. & SANCHES, O. Quantitativo-qualitativo: *oposição ou complementaridade?* In: Cadernos de Saúde Pública. Rio de Janeiro : 9 (3): 239-262, 1993.
- MINAYO, M. C. S. O desafio do conhecimento: pesquisa qualitativa em saúde. São Paulo/Rio de Janeiro: HUCITEC/ABRASCO, 1994.
- Normas ISO9000, (International Organization for Standardization), 1997
- PRESMAN, Roger S. Engenharia de Software, 1992: Traduzido por José Carlos Barbosa dos Santos, 1995; do original "Software Engineering: A Practitioner's Approach, 3ª Ed. Makron Books do Brasil Ltda.
- ROCHA, Heloísa Vieira da & BARANAUSKAS, Maria Cecília Calani, Design e avaliação de interfaces humano-computador. São Paulo, IME-USP, 2000.
- SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. Metodologia da Pesquisa e Elaboração de Dissertação. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2000 118p.
- NEGROPONTE, Nicholas. A Vida Digital. São Paulo. Companhia das Letras, 1995.
- SHMERING, A. et al. Qualitative research in medical practice. In: The Medical Journal of Australia, 158: 619-625, 1993.

SOFTWARE ENGINEERING INSTITUTE. The Capability Maturity Model: guides for improving the Software process. Reading : Addison Wesley, 1997.

WEBER, Kival Chaves. Qualidade e Produtividade em Software, 1997, Makron Books.