

JOSÉ CARLOS BEKER

ASPECTOS DA QUALIDADE DE SOFTWARE NO BRASIL SEGUNDO A NORMA ISO  
9126. UM ESTUDO DE CASO

**Dissertação apresentada ao Curso de Mestrado em  
Sistemas de Gestão da Universidade Federal  
Fluminense como requisito parcial para obtenção do  
Grau de Mestre em Sistemas de Gestão. Área de  
Concentração: Organizações e Estratégia. Linha de  
Pesquisa: Sistema de Gestão pela Qualidade Total.**

: Prof. Ruben Huamanchumo Gutierrez, D.Sc.  
ORIENTADOR

Niterói

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

2008  
JOSÉ CARLOS BEKER

ASPECTOS DA QUALIDADE DE SOFTWARE NO BRASIL SEGUNDO A NORMA ISO  
9126. UM ESTUDO DE CASO

**Dissertação apresentada ao Curso de Mestrado em  
Sistemas de Gestão da Universidade Federal  
Fluminense como requisito parcial para obtenção do  
Grau de Mestre em Sistemas de Gestão. Área de  
Concentração: Organizações e Estratégia. Linha de  
Pesquisa: Sistema de Gestão pela Qualidade Total.**

Aprovado em 11 de julho de 2008

---

Prof. Ruben Huamanchumo Gutierrez, D.Sc.  
Universidade Federal Fluminense – UFF

---

Prof. Helder Gomes Costa, D.Sc. (UFF)  
Universidade Federal Fluminense – UFF

---

Prof<sup>a</sup> Cristina Gomes de Souza, D.Sc. (CEFET-RJ)  
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

## Epígrafe

Eu tenho uma espécie de dever, dever de sonhar, de sonhar sempre, pois sendo mais do que um expectador de mim mesmo, em tenho que ter o melhor espetáculo que posso. E assim me construo a ouro e sedas, em salas supostas, invento palco, cenário para viver o meu sonho entre luzes brandas e músicas invisíveis.

Fernando Pessoa

## DEDICATÓRIA

de

A todos aqueles que me ajudaram ao longo

minha vida e contribuíram direta ou indiretamente  
na elaboração desse trabalho, destacando-se minha  
mãe, meu pai, (in memoriam),  
minha esposa, minhas filhas,  
José Beker.  
Amo vocês.

## RESUMO

Este estudo busca identificar os impactos na qualidade dos produtos de *software* com a adoção das práticas baseadas nas áreas de processo de Verificação e Validação do modelo CMMI. Neste contexto, esta pesquisa procurará identificar o nível de aderência a estas práticas com o cliente, usando questionamentos do modelo de avaliação de qualidade de *software* ISO/IEC 9126 para avaliar seus respectivos impactos. Os referenciais teóricos utilizados serão os Testes de *Software*, ISO/IEC 9126 e o CMMI do SEI (*Software Engineering Institute*), da Carnegie Mellon University.

Palavras-chave: avaliação, gerência de qualidade, produtos de software, testes de *Software*, CMMI, ISO/IEC 9126

## **ABSTRACT**

This research intends to identify the impacts of product quality with the use of practices based in process areas of Verification and Validation of the model CMMI in the. Inside this context, such research has searched for identify the level of tack to such practices with the client using questionnaires of the evaluation models of software quality ISO/IEC 9126 and its respectively impacts in quality. The theoretical references used are Software Testing, ISO/IEC 9126, the CMMI of SEI (Software Engineering Institute), of the Carnegie Mellon University.

Key-words: evaluation, quality management, software testing, CMMI, ISO/IEC

## SUMÁRIO

|  |     |
|--|-----|
| <b>1. INTRODUÇÃO</b>   | 013 |
| 1.1 - CONTEXTUALIZAÇÃO DO PROBLEMA:  | 013 |
| 1.2 - SITUAÇÃO PROBLEMA VINCULADA À PESQUISA:  | 16  |
| 1.3 - QUESTÕES DA PESQUISA:  | 16  |
| 1.4 - OBJETIVO GERAL:  | 17  |
| 1.5 - OBJETIVOS ESPECÍFICOS:   | 17  |
| 1.6 - JUSTIFICATIVA E RELEVÂNCIA DA PESQUISA:  | 18  |
| 1.7 - METODOLOGIA DA PESQUISA:   | 19  |
| 1.8 - ESTRUTURA DO TRABALHO:   | 23  |
| <b>2. REVISÃO DA LITERATURA</b>  | 25  |
| 2.1 - A QUALIDADE DE SOFTWARE:   | 25  |
| 2.2 - O PROCESSO E O DESENVOLVIMENTO DE SOFTWARE:  | 27  |
| <b>2.2.1 - ISO/IEC: 9126:</b>  | 31  |
| <b>2.2.2 - Do Capability Maturity Model (CMM) ao Capability Maturity Model Integration (SW-CMMI)</b> | 33  |
| 2.2.2.1 - O CMM  | 33  |
| 2.2.2.2 - Estrutura do Modelo CMMI   | 36  |
| 2.3 - TESTES DE VERIFICAÇÃO E VALIDAÇÃO DE SOFTWARE:   | 54  |
| <b>2.3.1 - Objetivos dos Testes:</b>   | 56  |
| <b>2.3.2 - Níveis de Teste:</b>  | 57  |
| <b>2.3.3 - Tipos de Testes</b>   | 58  |
| 2.3.3.1 - Teste Estático Versus Teste Dinâmico:  | 58  |
| 2.3.3.2 - Técnicas de testes dinâmicos   | 59  |
| 2.3.3.2.1 - Teste estrutural ou teste de caixa-branca  | 59  |
| 2.3.3.2.2 - Teste funcional ou teste de caixa-preta (black box)                                      | 60  |
| 2.3.3.3 - Testes Unitários:  | 61  |
| <b>3. FUNDAMENTOS DA PESQUISA</b>  | 64  |
| 3.1 - MEDE-PROS®   | 64  |
| <b>3.1.1 - Base Teórica do MEDE-PROS®</b>  | 65  |
| <b>3.1.2 - Estrutura do MEDE-PROS®</b>   | 74  |
| 3.2 - PERFIL DO AMBIENTE DA PESQUISA   | 77  |
| <b>3.2.1 - População e Amostra</b>   | 78  |
| 3.3 - ANÁLISE E DISCUSSÃO DOS RESULTADOS:  | 79  |
| <b>3.3.1 - Análise Descritiva</b>  | 80  |
| 3.3.1.1 - Parte I – Atributos Funcionais:  | 80  |
| 3.3.1.2 - Parte 2 – Questões Gerais  | 93  |
| 3.3.1.3 - Parte 3 – Análise de Classes Latentes  | 107 |
| <b>4. CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES</b>   | 115 |
| <b>ANEXOS</b>  | 123 |

# CAPÍTULO I

## 1. INTRODUÇÃO

### 1.1 - CONTEXTUALIZAÇÃO DO PROBLEMA:

Atualmente, os *softwares* vêm assumindo papel vital na condução e execução dos negócios nas organizações, tornando-se, em alguns casos, parte essencial dos seus processos de trabalho, refletindo na crescente demanda por novas tecnologias pelas empresas na sociedade em geral, sugerindo, assim, que a importância dos *softwares* tende a se multiplicar nos próximos anos. Nesse contexto, a confiabilidade dos *softwares* passa a ser crucial para os fabricantes junto aos seus clientes, que buscam fornecedores capazes de produzir *softwares* de alta qualidade em prazos cada vez menores.

Ao mesmo tempo, cresce a preocupação das empresas de *softwares* com a satisfação dos clientes, com a produtividade de suas equipes e com os custos de seus projetos. O relatório *The Economic Impact of Inadequate Infrastructure for Software Testing* (NIST, 2002) aponta que os defeitos em *softwares* custam às empresas americanas aproximadamente 1% do PIB dos Estados Unidos.

No Brasil muitos gestores de empresas de software ainda julgam elevados os custos de ferramentas, metodologias e acesso a recursos de capacitação. Mas não consideram fatos como o de uma desenvolvedora de aplicativos típica gastar entre 40% e 50% de seus recursos produtivos em retrabalho de conserto de um defeito no *software* liberado para o mercado, chegando assim a custar de 50 a 200 vezes mais que o necessário para identificá-lo e corrigi-lo internamente. Conquistar um novo cliente custa em média 10 vezes mais que os esforços extras para se manter um existente. Pior, não observam que a globalização traz novos concorrentes internacionais para o mercado e que seus próprios clientes também estão crescendo, demandando presença mais atuante e forte de seus parceiros, com fôlego e visão.

Dessa forma, o processo de teste almeja encontrar erros na implementação de um *software* e validar os requisitos definidos pelos clientes-usuários. A qualidade para a aquisição e manutenção do produto de *software* possui, assim, relevância especial. Em muitos casos, no entanto, o desenvolvimento e execução dos testes não seguem uma metodologia estruturada, talvez, porque a seleção dos testes seja aleatória, e os casos sejam desenvolvidos de forma não-estruturada e não-sistemática.

Em 2001, menos de 30% das empresas brasileiras desenvolvedoras de *softwares* possuíam processos estruturados de planejamento e documentação de testes, o que indicava baixo grau de maturidade das organizações (MINISTÉRIO DA CIÊNCIA E TECNOLOGIA, 2001).

Entretanto, recentemente, a indústria brasileira de *software* foi incluída na política industrial do governo brasileiro para exportações, com metas previstas para alcançar US\$ 2 bilhões dólares (COELHO, 2008).

Um estudo do MIT (*Massachusetts Institute of Technology*, 2003) apresenta China e Brasil com condições para se tornarem novas potências do mercado de software mundial. A Índia levou mais de 20 anos para chegar onde está com apoio do governo e esforço da iniciativa privada. O Brasil, por sua vez, conta com o acesso a conhecimento e aos recursos internacionais que podem rapidamente ser replicadas com criatividade, ilhas de expertise já existem como, potencial humano e conhecimento do mundo dos negócios. Alie-se a isto o carisma que vem desenvolvendo no cenário mundial que, com conscientização, disciplina e ação, o Brasil será uma das potências mundiais do software.

Tendo em vista que para ter *softwares* com qualidade, desenvolvidos nos prazos acordados com custos que atendam às exigências dos clientes, têm-se exigido dos fabricantes a melhoria dos processos de engenharia de *software*.

Vale ressaltar, que a qualidade do produto de *software* depende da capacitação dos processos de sua elaboração. Como existe pouco investimento das empresas brasileiras em certificações que comprovem a qualidade e a maturidade dos seus processos de fabricação de *software*, ainda não ocorre a venda desse produto no mercado internacional.

Nesse sentido, Institutos Internacionais como, por exemplo: *International Organization for Standardization (ISO)* e *International Electrotechnical Commission (IEC)* definiram normas de qualidade mundial para elaboração de produtos de *software*. Dentre estas, pode-se citar as normas ISO/IEC: 9126, aplicada a produtos de *software*, ISO/IEC: 12119 para pacotes de *software* e a norma ISO/IEC: 14598 padrão para processo de avaliação de produtos de *software* (ROCHA, A.R.C, MALDONADO, J. C., WEBER, K.C. *et al*, 2001, p. 41).

Com base nas normas citadas acima, o Centro de Pesquisa Renato Archer (CenPRA) vem desenvolvendo projetos com o intuito de transferir conhecimento e tecnologia para a avaliação de produtos de *software*. Como resultado destes projetos, pode-se destacar o Método de Avaliação da Qualidade de Produto de *Software* (MEDE-PROS®), que define um modelo de qualidade aplicado na avaliação de produtos de *software*.

Atualmente, pequenas empresas brasileiras ainda não possuem capital para as certificações de *software*. No entanto, os investimentos de entidades privadas, centros de estudos e do governo, têm tornado possível a melhoria dos processos de *software* no Brasil. Nesse sentido, a Melhoria de Processo do *Software* Brasileiro (MPS BR), projeto coordenado pela Softex é apontado como solução para tornar o *software* brasileiro um produto de exportação competitivo (COELHO, 2008).

Nessa linha é de vital importância a integração de atividades de teste ao processo de desenvolvimento de *software*, segundo Herbert (1999, p. 17):

Tem sido considerada cada vez mais como fator essencial para a obtenção de sistemas de *softwares* de qualidade, seja pelas normas ISO 9000, pelo CMM ou por outros modelos difundidos na literatura. Além disso, o Modelo de Referência (MR-MPS) define uma estrutura que combina os conceitos de maturidade de uma organização, definidos pelo CMMI, com o de capacidade de processo estabelecido pela ISO/IEC 15504. No MR-MPS, níveis de maturidade são uma combinação entre processos e capacidade de processos (SOFTEX. 2005).

De acordo com MR-MPS, a definição dos processos reproduz a forma utilizada pela norma ISO/IEC: 12207, onde a descrição inclui a declaração de seu propósito e dos resultados esperados a partir da efetiva implementação do processo em questão. No modelo, o termo propósito descreve o principal objetivo a ser atingido durante a execução do processo de confecção do *software*.

Além disso, a importância da incorporação de atividades de teste a cada etapa do processo de desenvolvimento de *software* é fortalecida, também, pelo fato de quanto mais próximos de sua origem os erros forem detectados, menor será o custo e a dificuldade das correções (ROCHA *et al.*, 2001). Convém ressaltar, porém, mais uma vez, que o teste de *software* ainda é encarado como uma atividade complementar ao processo de desenvolvimento e não como parte do próprio processo, sendo, em muitas empresas, mal-estruturado e realizado de maneira individualizada (BEIZER, 1990).

Os processos de testes de *software* referenciados no CMMI refletem, atualmente, no comportamento das empresas que buscam implantar, ou mesmo, melhorar o processo de teste utilizado.

No entanto, embora as técnicas de teste de *software* tenham sido criadas por volta dos anos 80, as empresas ainda têm grande dificuldade com a atividade de teste. Diante do exposto acima, chega-se a seguinte situação do problema:

## 1.2 - SITUAÇÃO PROBLEMA VINCULADA À PESQUISA:

O presente estudo pretende responder às seguintes questões: (a) como identificar os aspectos da qualidade de software no Brasil conforme a Norma ISO 9126. (b) quais os impactos na qualidade do produto de *software*, sob a ótica do cliente, com a adoção dos processos de testes baseados no modelo CMMI e MEDE-PROS®, baseado no modelo de qualidade da NBR: 9126?

## 1.3 - QUESTÕES DA PESQUISA:

Segue abaixo, as seguintes questões norteadoras da atual pesquisa:

- A avaliação da qualificação de produtos de *software*, conforme o previsto no método de avaliação MEDE-PROS®, pode constatar a presença de erros e a capacidade de eliminá-los de forma global?
- A obtenção de indicadores para o cliente pode prever a qualidade, a aceitação do produto e sua manutenção durante o período da operação, e propiciar a estimativa de gastos para o desenvolvedor e para prever as necessidades para manter e ajustar o produto em desenvolvimento?
- Os resultados das validações foram analisados e ações de correção foram identificadas?

#### 1.4 - OBJETIVO GERAL:

O principal objetivo do presente estudo, conduzido por meio de uma pesquisa bibliográfica, seguida de pesquisa de campo consiste em: analisar os aspectos da qualidade de software no Brasil segundo a norma ISO: 9126; compreender e analisar a percepção dos impactos na qualidade dos produtos de *software* com adoção das práticas baseadas nas áreas de processo de verificação e validação do modelo CMMI.

#### 1.5 - OBJETIVOS ESPECÍFICOS:

Para se alcançar o objetivo geral da presente pesquisa, foram traçados os seguintes objetivos específicos:

- Verificar as contribuições sobre a proposta da realização de testes de produtos de *softwares* no Brasil;
- Identificar mecanismos para avaliar a qualidade dos *softwares* na visão do cliente.

- Apontar as práticas de testes estabelecidas pela norma ISO 9126 no desenvolvimento de produtos de *software*.
- Descrever de que forma a adoção das práticas de testes estabelecidas pelo modelo CMMI no desenvolvimento de produtos de *software* impacta na qualidade, dos mesmos, sob a visão do cliente.

#### 1.6 - JUSTIFICATIVA E RELEVÂNCIA DA PESQUISA:

Faz-se necessário destacar que o método visa avaliar o produto de *software* no Brasil, independentemente da tecnologia e das ferramentas adotadas no seu desenvolvimento ou da plataforma necessária para sua operação, haja vista ser esse procedimento já realizado internacionalmente. Essa avaliação é realizada tendo em vista o usuário final, com base no material disponibilizado ao usuário. Dessa forma, o elemento do produto final de *software* a ser avaliado é a qualidade do *software* executável, não se aplicando qualquer análise sobre o código do programa - fonte ou sobre a documentação gerada durante seu desenvolvimento, como “avaliação de custo” ou “documentos técnicos”. Portanto, a avaliação deve mostrar o grau de atendimento do produto de *software* em relação aos requisitos funcionais e de qualidade especificados por um adquirente.

É preciso ressaltar que o uso do presente método intenciona verificar a presença das características e subcaracterísticas de qualidade descritas na norma ISO:9126, como: confiabilidade, eficiência, funcionalidade, usabilidade, manutenibilidade e portabilidade. E detalhadas sob a ótica do usuário. O presente método de avaliação é considerado um teste de “caixa preta”, isto é, são testes realizados em um programa para testar a saída dos dados usando entradas de vários tipos e não escolhidas, conforme a estrutura do programa, percorrendo a lógica de verificação do atendimento aos requisitos funcionais.

Nessa perspectiva, o estudo se justifica pela necessidade de ocorrer no Brasil procedimentos de teste dos produtos de *software*. Tem-se observado a relevância desses testes e se pretende comprovar este fato, a partir da percepção do cliente.

Dessa forma, o estudo é relevante, a partir do pressuposto utilizado no MEDE-PROS® – Método de Avaliação de Qualidade de Produtos de *Software* das práticas de utilizadas para o desenvolvimento de elaboração de *software*.

A partir da padronização do uso do modelo acima descrito novos conceitos e metodologias poderão surgir em sistemas de gestão brasileiro, possibilitando o aprimoramento do uso de sistemas de informação.

Da mesma forma, ao demandar a opinião do cliente externo, o estudo passa a estabelecer uma gestão compartilhada com o desenvolvedor. Ressalte-se, ainda, que a proposição do processo de teste busca depurar os erros no desenvolvimento do *software* e validar os requisitos definidos pelos usuários. O mundo do cliente, em que todos os processos de gestão estão voltados para o cliente interno e externo é relevante ao estudo, pois buscará a avaliação do cliente externo, fundamental para melhorar a qualidade do produto de *software* disponível no mercado.

Nesse sentido, acredita-se que este estudo possui relevância especial na avaliação da qualidade do produto de *software* no âmbito da gestão, pois integra todos os atores na busca da qualidade do serviço, embora em muitos casos, o desenvolvimento e execução dos testes não sigam uma metodologia estruturada, foco dessa pesquisa, devido, talvez, à seleção aleatória dos testes e ao desenvolvimento de forma não-estruturada e não-sistemática.

## 1.7 - METODOLOGIA DA PESQUISA:

Pesquisa é o instrumento usado pela ciência para estudar a realidade, procurando respostas para as indagações propostas dentro de uma determinada área. Segundo Demo (1987) pesquisar é a atividade científica pela qual descobrimos a

realidade, o que é uma tarefa que exige dedicação, pois de acordo com Demo (1987, p. 23) “devemos partir do pressuposto de que a realidade não se desvenda na superfície. Não é o que aparenta à primeira vista”.

O que é realidade? Para muitos parece evidente a realidade. Nada mais enganoso. É precisamente o que mais ignoramos. Por isso pesquisamos, já que nunca dominamos a realidade. Quem imagina conhecer adequadamente a realidade, já não tem o que pesquisar, ou melhor, tornou-se dogmático e deixou o espaço da ciência (DEMO, 1987.p. 27).

A pesquisa tem um caráter pragmático, para Gil (1999, p. 42), é um “processo formal e sistemático de desenvolvimento do método científico. O objetivo fundamental da pesquisa é descobrir respostas para problemas mediante o emprego de procedimentos científicos”.

Nessa linha de pensamento, a pesquisa torna-se instrumento fundamental para a transformação da realidade na busca por melhorias e aprimoramentos das práticas e produção de condições mais favoráveis de existência humana, conforme afirmado por Demo (1987) é possível sempre descobrir novos horizontes do conhecimento e da prática.

De acordo com Cervo (2002, p.24): “o método científico é um conjunto ordenado de procedimentos que se mostram eficientes, ao longo da história, na busca do saber”, que ao serem aplicados permitem a análise da realidade e a apresentação de opções de melhoria da qualidade da vida humana. De acordo com Nagel (1969) pode-se sintetizar metodologia científica como a lógica geral, tácita ou explicitamente empregada para apreciar os méritos de uma pesquisa.

Segundo Cervo (2002 p. 63): “a pesquisa é uma atividade voltada para a solução de problemas teóricos ou práticos com o emprego de processos científicos”.

Do ponto de vista de Silva (2001) de acordo com a forma de abordagem do problema as pesquisas são conceituadas da seguinte forma: 1º) pesquisa qualitativa: quando há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzida em números. A interpretação dos fenômenos e a atribuição de significados são básicas no processo de pesquisa qualitativa. Não requer o uso de métodos e técnicas estatísticas. O ambiente natural é a fonte direta para a coleta de dados e o pesquisador é o instrumento chave. É descritiva. Os pesquisadores tendem a analisar

seus dados indutivamente. O processo e o seu significado são os focos principais de abordagem e; 2º) pesquisa quantitativa: quando tudo pode ser quantificável, o que significa traduzir em números opiniões e informações para classificá-las e analisá-las. Requer o uso de recursos e de técnicas estatísticas, com por exemplo: porcentagem, média, coeficiente de correlação de correlação.

Por fim, Goldenberg (2000) aponta que a pesquisa quanti-qualitativa é o conjunto de diferentes pontos de vista e diferentes maneiras de coletar e analisar os dados, o qual permite uma idéia mais ampla e inteligível da complexidade de um problema, permitindo ao pesquisador fazer um cruzamento de suas conclusões de modo a ter maior confiança que seus dados não são produto de um procedimento específico ou de alguma situação particular.

Do ponto de vista de Gil (2002) a pesquisa bibliográfica é aquela desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos. Embora a pesquisa documental se assemelhe à pesquisa bibliográfica a diferença essencial está na natureza das fontes.

Enquanto a pesquisa bibliográfica se utiliza fundamentalmente das contribuições dos diversos autores sobre determinado assunto, a pesquisa documental vale-se de materiais que não receberam ainda um tratamento analítico, ou que ainda podem ser reelaborados de acordo com os objetos da pesquisa (GIL, 2002, p. 45).

Segundo Gil (1999) as pesquisas sociais podem ser classificadas em: estudos exploratórios, descritivos e estudos que verificam hipóteses causais. Assim:

As pesquisas exploratórias têm como principal finalidade desenvolver, esclarecer e modificar conceitos e idéias, tendo em vista a formulação de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores, envolvendo levantamento bibliográfico e documental, entrevistas não-padroneizadas e estudos de caso (GIL, 1999, p.71).

O presente estudo utiliza levantamento bibliográfico, com base na literatura especializada (normas, livros e artigos que tratam do tema engenharia de *software* e qualidade de *software*) com ênfase na avaliação de produto de *software*.

Convém enfatizar que este estudo exploratório tem a função de aumentar o conhecimento do pesquisador acerca do fenômeno investigado e o estabelecimento de prioridades para futuras pesquisas (SELLTIZ, *apud* CANEDO, 1998, p. 62).

Segundo Goode e Hatt (1975, p.171) as regras definidas pela metodologia apontam qual, dentre os vários tipos de pesquisa que se apresentam, o mais propício a direcionar a realização de um processo investigativo junto ao objeto de estudo escolhido. Dentre os tipos propostos por estes autores, o estudo de caso é um meio para se coletar dados, mantendo o caráter único do objeto de estudo, visto que ao não se dispor de meios concretos para definir precisamente os limites da totalidade de qualquer objeto de estudo, quer físico, biológico ou social por serem construções intelectuais, o estudo de caso "é uma tentativa de abranger as características mais importantes do tema que se está pesquisando".

Para Vergara (2000), o estudo de caso possui um caráter de profundidade e detalhamento que permite entender a relação entre os elementos analisados no estudo que se pretende, ou seja, o empreendedorismo desenvolvido na rede da BFH como uma forma de conquistar e manter mercados.

Ainda, Gil (1999, p. 72) diz que: "o estudo de caso é caracterizado pelo estudo profundo e exaustivo de um ou de poucos objetos, de maneira a permitir o seu conhecimento amplo e detalhado, tarefa praticamente impossível mediante os outros tipos de delineamentos possíveis".

Gil (1999) apresenta uma relação de vantagens e limitações do estudo de caso, da qual se destaca: o estímulo a novas descobertas. Em virtude da flexibilidade do planejamento do estudo de caso, o pesquisador, ao longo do seu processo, mantém-se atento a novas descobertas (...) porque é altamente recomendado para a realização de estudos exploratórios; a ênfase na totalidade. É claro que o estudo de caso também apresenta limitações. A mais grave refere-se à dificuldade de generalização dos resultados obtidos.

Além disso, o estudo de caso satisfaz adequadamente as condições propostas por Yin (1994), onde o tipo de questão básica a ser respondida corresponde a uma pergunta; não existe controle do pesquisador sobre os eventos, e o foco é sobre eventos reais contemporâneos.

Foram elaboradas as seguintes proposições para o presente estudo de caso:

- A metodologia proposta atuará como uma seqüência de passos, norteando o processo de desenvolvimento e implementação do *software*;

- A metodologia proposta estará em conformidade com a ISO 9126 (NBR: 13596), trazendo confiabilidade aos seus usuários. O estudo do caso aborda o processo de desenvolvimento do método e execução da avaliação de um produto de *software*;
- A proposição do processo de teste busca depurar os erros no desenvolvimento de um *software* e validar os requisitos definidos pelos usuários.

Segundo Yin (1994) a entrevista é uma das seis fontes de evidência e suas conclusões podem ser surpreendentes, devido à sua associação com o método *survey*, usualmente associado com questionários e entrevistas (BRYMAN, 1989).

Partindo deste pressuposto elaborou-se um questionário de pesquisa, dividido em duas partes: uma em relação à aplicação dos requisitos da ISO 9126 (NBR 13596) e outra relativa à aplicação dos requisitos do CMMI. Esse questionário foi enviado aos usuários do *software*, e após o retorno foi pontuado, e servindo, então, para construção de um parecer sobre a aceitação do perfil do produto.

Vale destacar que, neste estudo foi utilizada e referenciada a norma brasileira NBR 13596, porque o MEDE-PROS®, base da pesquisa, foi elaborado a partir dessa versão da Norma.

A metodologia proposta, também, deve possibilitar aos adquirentes de *software* e, em especial, aos pesquisadores e desenvolvedores da área, fundamentar os aspectos da qualidade dos produtos de *software* acabados.

## 1.8 - ESTRUTURA DO TRABALHO:

No Capítulo I encontra-se a contextualização do tema, sendo desenvolvida a situação-problema vinculada à pesquisa. Também, são apresentados os objetivos, a suposição-chave e as questões que norteiam o trabalho, além da justificativa e da relevância da pesquisa.

No Capítulo II encontra-se o desenvolvimento e a revisão da literatura relacionada à qualidade de *software*, onde se analisam as obras que tratam do processo e do desenvolvimento de *software*, define-se processo de *software*, enfatizando a ISO/IEC: 9126, descrevendo-se a trajetória do CMM ao CMMI, apresenta-se o processo de *software* brasileiro MPS.BR; os principais testes de verificação e validade de *software*, seus objetivos, seus níveis, tipos e técnicas.

No Capítulo III apresenta-se a metodologia da pesquisa, descrevendo-se o caminho que se tenciona percorrer para chegar aos fins a que se pretende. Para isso, são descritos o tipo de pesquisa, seus métodos e a estratégia metodológica.

No Capítulo IV são analisados e discutidos os resultados da pesquisa.

No capítulo V encontram-se as considerações finais e sugestões do estudo.

E, por fim, no Capítulo VI, encontram-se as referências e os anexos do estudo.

## CAPÍTULO II

### 2. REVISÃO DA LITERATURA

#### 2.1 - A QUALIDADE DE SOFTWARE:

Melhorar a qualidade do produto é, certamente, o principal objetivo da engenharia de *software*. O *software*, atualmente, é fundamental, estando presente no comércio, nos transportes, na medicina, no direito, na educação e em outros segmentos da sociedade contemporânea. Os produtos de *software* apresentam-se cada vez maiores e mais complexos, demandando, para seu desenvolvimento, recursos, pessoas, capital e tempo (ROCHA, *et al.*, 2001).

Entender o significado de qualidade de *software* não é uma tarefa fácil, pois há diversos pontos de vista diferentes sobre o conceito de qualidade. A norma ISO 9000 (2000) define qualidade como a totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas.

ROCHA (1987) apontou qualidade de *software* como sendo um conjunto de propriedades a serem satisfeitas em determinado grau, de modo que o software satisfaça às necessidades de seus usuários.

CROSBY (1988), ainda, na década de 80, dizia que é mais barato construir um produto correto, sugerindo que “a qualidade é gratuita” (*Quality is free*). Spehar (2005) considera que os custos da qualidade são importantes, pois o valor de cada hora de trabalho não gasta para corrigir problemas (retrabalho) pode servir para desenvolver melhores produtos em menor tempo, ou para melhorar a qualidade dos produtos e processos existentes.

PRESSMAN (1995) definiu qualidade de *software*, como a conformidade com requisitos funcionais e de desempenho explicitamente enunciados, padrões de

desenvolvimento documentados e características implícitas que se espera de qualquer *software* desenvolvido de forma profissional.

PFLEEGER (2004, p.22) considerou a qualidade a partir de perspectivas diferentes, baseado nas diferentes percepções de qualidade:

A visão transcendental, onde a qualidade é algo que podemos reconhecer, mas não podemos definir; a visão do usuário, onde a qualidade é a adequação ao propósito pretendido; a visão do fabricante, onde a qualidade é a conformidade com a especificação;

A visão do produto, onde a qualidade está relacionada às características inerentes ao produto; e a visão do mercado, onde a qualidade depende do valor que os consumidores estão dispostos a pagar pelo produto.

PFLEEGER (2004) enfatiza que a falta de qualidade também pode custar caro. Isto porque, quanto mais tempo um defeito permanece sem ser detectado, mais cara será a sua correção.

WAGNER E SEIFERT (2005) chamam a atenção de que, embora muito importante, garantir a qualidade é uma tarefa dispendiosa para o desenvolvimento de *software*. De acordo com esses autores, 50% dos custos do desenvolvimento são gastos com testes - uma atividade específica de garantia da qualidade.

Dessa forma, a qualidade pode se tornar gratuita, pois a construção de produtos de má qualidade pode custar o mesmo valor, ou até mais que o valor relativo aos custos da qualidade para se corrigirem os defeitos inseridos no produto, além do prejuízo oriundo da entrega de um produto sem qualidade. Além disso, se a qualidade do produto traduz preocupação e investimento é feito para que o produto tenha qualidade satisfatória, se gasta corretamente com a qualidade, o custo da correção de defeitos é menor, se o produto for desenvolvido com essas características.

PLEEGER (2004) e GALIN E AVRHAMI (2005) apontam que o investimento na qualidade traz um ganho de produtividade, além de redução dos custos e do tempo de entrega do produto, ou seja, quanto melhor for a qualidade do produto no sentido de atender as necessidades dos clientes, mais se ganha em maior grau de satisfação dos clientes, aumento das vendas, condições favoráveis para enfrentar a concorrência e aumento da participação da empresa no mercado.

Dessa forma, D'OLIVEIRA (2003) afirma que a qualidade de um produto tem influência direta em sua venda, pois produtos com menos deficiências permitem que se

reduzam os índices de erro, o re-trabalho e o desperdício, as falhas e aumento dos custos no uso e na garantia deste produto, e, aumenta satisfação dos clientes.

Segundo FUGETTA (2000), a qualidade de produtos de software está relacionada à qualidade do processo de *software*. ROCHA *et al.* (2001) postulam que a qualidade do processo de *software* é tão importante quanto a qualidade do produto.

## 2.2 - O PROCESSO E O DESENVOLVIMENTO DE SOFTWARE:

A definição de um processo é feita através da construção de um modelo de processo. Nesse sentido, OSTERWEIL (1987) entende que um modelo de processo de *software* é uma representação prescritiva das atividades de desenvolvimento em termos da sua ordem de execução e gerenciamento dos recursos; consiste na descrição de um processo de *software*, permitindo que seja analisado, compreendido e executado.

De acordo com o *Institute of Electrical and Electronics Engineers - IEEE* (1990), processo de *software* pode ser definido como “uma seqüência de passos executadas para um determinado propósito”.

PAULK *et al* (1993) consideram o processo de *software* como “um conjunto de atividades, métodos, práticas e transformações que pessoas utilizam para desenvolver e manter *software* e produtos associados (planos de projeto, código, casos de teste, manuais do usuário e outros)”.

A *International Organization for Standardization – ISO*, ao desenvolver uma padronização internacional para o universo da qualidade, identificou determinadas peculiaridades em determinados processos, que os diferenciava dos processos industriais tradicionais. O fato foi evidenciado na Norma NBR 9001:1994 – Sistemas da qualidade-modelo para garantia da qualidade em projeto, desenvolvimento, produção, instalação e serviços associados (ABNT, 1994) que, ao estabelecer os requisitos para controle de processo afirma que: “os processos devem requerer monitorização contínua e controle dos parâmetros de processo para assegurar que os requisitos especificados sejam atendidos”.

SOMMERVILLE (1996) entende como processo de *software* o conjunto de atividades e informações associadas necessárias para o desenvolvimento de um produto de *software*.

FUGUETTA (2000, p. 37) ainda afirma que: “um processo de *software* pode ser definido como um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos que é necessário para conceber, desenvolver, implantar e manter um produto de *software*”.

O mesmo autor diz que “aplicações de *software* são produtos complexos, difíceis de desenvolver e testar”, acrescentando que “muito freqüentemente, o *software* apresenta comportamentos inesperados e indesejados que podem mesmo acarretar problemas e danos” (FUGUETTA, 2000, p.38).

A descrição de um processo é uma expressão documentada de um conjunto de atividades executadas para alcançar um dado propósito, que provê uma definição operacional dos principais componentes de um processo (CHRISSIS *et al.*, 2003).

De acordo com NOGUEIRA (2006 p.16):

Esta conceituação é reiterada na mais recente versão dessa série de normas que, através da Norma NBR/ISO 9000:2000- Sistemas de Qualidade – Fundamentos e Vocabulário (ABNT, 2000), apresenta a seguinte observação: “Um processo em que a conformidade do produto resultante não pode ser prontamente ou economicamente verificada é freqüentemente chamado de processo especial”.

Nesse sentido, para PFLEEGER (2004), a sistematização da execução das atividades, através de processos de *software* viabiliza a repetição, independente de quem as execute.

As ISO série 9000 foram concebidas tendo, como pressuposto sua aplicabilidade direta a todo e qualquer ramo da indústria. No entanto, logo após sua publicação, ficou evidente que, para o universo da indústria de *software*, isso não seria possível. Essa constatação levou à necessidade de elaboração de um novo padrão normativo para esse universo: a Norma NBR ISO 9000-3:1993 – Norma de gestão da qualidade e garantia de qualidade. Parte 3: Diretrizes para a aplicação da NBR ISO 9001 ao desenvolvimento, fornecimento e manutenção de *software* (ABNT, 1993). A indústria de *software* passou a ser, então, o único setor da economia a merecer essa particularização. Esse documento se justifica na explicação dessa especificidade, enfatizando que:

[...] o processo de desenvolvimento e manutenção de *software* é diferente da maioria dos demais tipos de produtos industriais, tornando necessário prover, nesse campo da tecnologia, de desenvolvimento tão rápido, orientações

adicionais para o estabelecimento de sistemas de qualidade onde estejam envolvidos os produtos de *software* [...]

Tal distinção foi mantida no que se refere a um padrão normativo específico na nova edição das Normas ISO Série 9000 com a publicação da Norma ISO/IEC 9003 / 2004 –*Software Engineering –Guidelines for the Application of ISO 9.001 /2000 to Computer Software* (ISO, 2004).

Convém ressaltar que desenvolvimento de *software* é um processo complexo, realizado através de um esforço coletivo de criação. Nesse contexto, seus resultados dependem diretamente das pessoas, das organizações e procedimentos utilizados em sua construção. Segundo ROCHA *et al* (2001), a condição para a qualidade do *software* é de que o processo se caracterize como “uma atividade sistemática e passível de repetição, independentemente de quem o execute”.

SILVA FILHO (2006) enfatiza que, na indústria de *software*, o desenvolvimento e a manutenção de produtos são atividades complexas, influenciadas por fatores técnicos, culturais e ambientais que têm impacto direto no custo, prazo, esforço e na qualidade do produto. BARRETO (2006) ressalta ainda que, enquanto a qualidade do processo se preocupa em melhorar a forma como um produto é desenvolvido, a qualidade do produto pretende acompanhar o produto em desenvolvimento, avaliando a qualidade dos produtos intermediários gerados desde o início do desenvolvimento até a conclusão do produto final.

A qualidade desejável em um produto de *software* está relacionada aos requisitos identificados pelo cliente que podem ser tanto os requisitos funcionais quanto os requisitos não funcionais. Os requisitos funcionais são aqueles relacionados às funcionalidades que o produto deve oferecer, enquanto os não-funcionais dizem respeito ao comportamento que o produto deve apresentar em relação a um conjunto de características estabelecidas, como, facilidade de uso, desempenho, adequação a normas e padrões, interação com outros produtos, dentre outros. Portanto, a qualidade necessária a um produto de *software* pode ser especificada através de requisitos, os requisitos de qualidade.

MADU *et al.* (1996) afirmam que, ao se desenvolver um produto com a qualidade desejada pelo usuário, é de se esperar que a satisfação do usuário seja obtida. Tal

satisfação freqüentemente é considerada um resultado fundamental da busca pela qualidade e tem um impacto positivo nos custos, lucros e crescimento de vendas na organização.

Nesse sentido, JUNG *et al.* (2004) consideram que, em algumas organizações, a qualidade de um produto de *software* tem sido usada como base para muitas decisões importantes, incluindo a melhoria da qualidade do produto, a realização de grandes aquisições e a monitoração de contratos.

Entretanto, identificar os requisitos de qualidade de um produto é uma tarefa árdua e, para auxiliar nessa empreitada, pode-se descrever a qualidade de um produto através de um conjunto de características que devem ser alcançadas em um determinado grau para que o produto atenda às necessidades de seus usuários. Cada uma das características de qualidade pode ser detalhada em vários níveis de subcaracterísticas, atingindo-se um amplo conjunto de atributos que descrevem a qualidade de um produto de *software*.

Com o objetivo de facilitar a definição e avaliação da qualidade de um produto, através de características de qualidade, vários modelos de qualidade foram definidos e diferenciam-se entre si com relação às características modeladas, ao relacionamento e à dependência entre elas. Segundo BARRETO (2006,p. 43): “Os primeiros trabalhos nessa área são os de BOEHM (1978) e McCALL (1977), cujos modelos apresentam uma hierarquia de características, em que cada uma delas contribui para a qualidade como um todo”.

Vale citar, ainda, outros modelos, definidos ao longo dos anos, cada um com suas particularidades, tais como os modelos de ROCHA (1983); ROCHA, PALERMO (1989); BELCHIOR (1992); BAHIA (1992); CAMPOS (1994); BLASCHEK (1995); DROMEY (1996); STRONG *et al.* (1997).

Esse conjunto de singularidades confere ao processo de *software* as dificuldades observadas no gerenciamento de suas atividades e impõe ao gerente de desenvolvimento percalços para a consecução dos objetivos de eficiência e eficácia. Nessa linha, nos últimos anos, desenvolveram-se modelos específicos de Sistemas de Gestão de Qualidade com o objetivo de desenvolver, implantar e controlar processos de *software*. Dentre eles, destacam-se a Norma ISO/9126, o *Capability Maturity Model*

*Integration–CMMI do Software Engineering Institute (SEI) da Carnegie Mellon University* e o modelo de referência para Melhoria do Processo do *Software Brasileiro (MPS.BR)*.

### **2.2.1 - ISO/IEC: 9126:**

A série de normas ISO/ IEC: 91R26 (NBR: 13596) descreve um modelo de qualidade para produtos de *software* categorizando a qualidade hierarquicamente em um conjunto de características e subcaracterísticas que devem ser atendidas para que o produto seja dito de qualidade. Esta série, também, propõe métricas que podem ser utilizadas durante a avaliação dos produtos de *software* (medição, pontuação e julgamento dos produtos de *software*).

A série de normas ISO/IEC: 9126 podem ser aplicada nas seguintes ocasiões:

- .Definição dos requisitos de qualidade de um produto de *software*;
- Avaliação das especificações do *software* durante o desenvolvimento para verificar se os requisitos de qualidade estão sendo atendidos;
- Descrição das características e atributos do *software* implementado, por exemplo, nos manuais de usuário;
- Avaliação do *software* desenvolvido antes da entrega ao cliente.

O modelo de qualidade externa e interna categoriza os atributos de qualidade de *software* em seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade) as quais são, por sua vez, subdivididas em subcaracterísticas. As subcaracterísticas podem ser medidas por meio de métricas externas e internas. Eis as definições das características de qualidade interna e externa segundo a ISO/IEC: 9126 - 1:

Funcionalidade: capacidade do produto de *software* de prover funções que atendam às necessidades explícitas e implícitas, quando o *software* estiver sendo utilizado sob condições específicas.

Subcaracterísticas: adequação, acurácia, interoperabilidade, conformidade e segurança de acesso.

Confiabilidade: capacidade do produto de *software* de manter um nível de desempenho especificado, quando usado em condições específicas.

Subcaracterísticas: maturidade, tolerância à falhas e recuperabilidade.

Usabilidade: capacidade do produto de *software* de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas.

Subcaracterísticas: inteligibilidade, apreensibilidade e operacionalidade.

Eficiência: capacidade do produto de *software* de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições específicas.

Subcaracterísticas: Tempo de Resposta e Recursos Utilizados.

Manutenibilidade: capacidade do produto de *software* de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do *software* devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais.

Subcaracterísticas: analisabilidade, modificabilidade, estabilidade e testabilidade.

Portabilidade: capacidade do produto de *software* de ser transferido de um ambiente para outro.

Subcaracterísticas: adaptabilidade, capacidade para ser instalado, capacidade para substituir e conformidade de portabilidade.

A ISO/IEC: 9126 / 1991 (NBR: 13.596), pode ser definida como “um conjunto de atributos que têm impacto na capacidade do *software* de manter o seu nível de desempenho dentro de condições estabelecidas por um dado período de tempo [...]”.

Vale enfatizar que essa norma não apresenta as medidas para as características da qualidade. Ao invés disto, propõe que cada empresa desenvolva as suas próprias medidas.

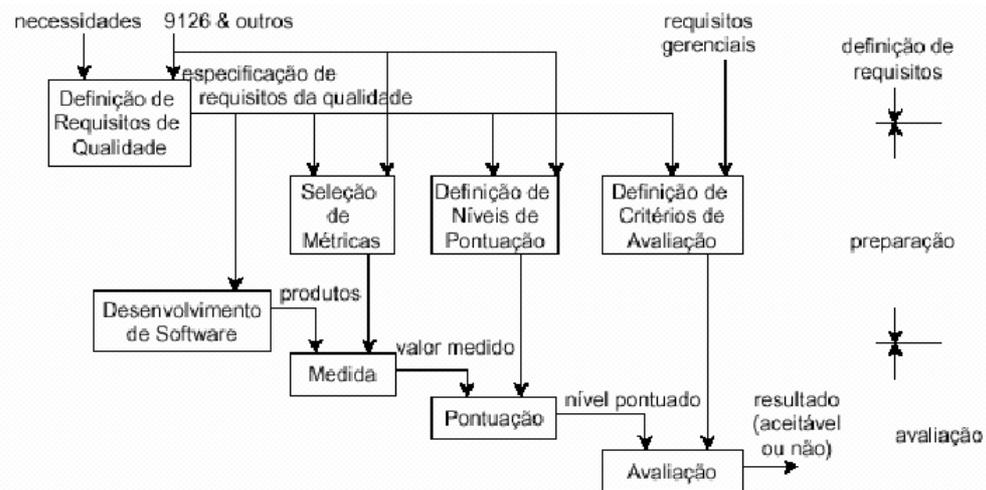


Figura 1 - Definição de requisitos de qualidade

Fonte: Disponível em: <<http://www.garcia.pro.br/06-02>> Acesso:20/03/06

## 2.2.2 - Do Capability Maturity Model (CMM) ao Capability Maturity Model Integration (SW-CMMI)

### 2.2.2.1 - O CMM

Em 1987, o SEI lançou uma breve descrição da estrutura de maturidade de processos e um questionário de maturidade (HUMPHREY, 1987). Com o questionário de maturidade, o SEI pretendia fornecer uma ferramenta simples para identificar áreas onde um processo de *software* da organização sendo avaliado precisava melhorar suas práticas. Depois de quatro anos de experiência com a estrutura de maturidade de processo e com a versão preliminar do questionário de maturidade, o SEI evoluiu a estrutura de maturidade de processo para o chamado *Capability Maturity Model* – CMM (PAULK *et al.*, 1991). Desde então, várias versões foram desenvolvidas e muitas

mudanças foram implementadas de acordo estudos e análises nas organizações aderentes ao modelo.

De acordo com ORLANDI (2000, p. 5): “o CMM representa uma proposição de recomendações para empresas cujo negócio seja a produção de *software*, que pretendam incrementar a capacidade (ou qualidade) do processo de desenvolvimento de sistemas”.

Em sua dissertação, Orlandi utilizou o modelo CMM, como trajetória para nortear as ações de implementação de um modelo de qualidade de *software* em organizações brasileiras. Assim, o CMM, segundo ORLANDI (2000, p.5):

Classifica as organizações produtoras de *software* em cinco grupos de maturidade e identifica, para grupo, as áreas-chave do processo desenvolvimento de sistemas que devem ser observadas na busca de melhoria de capacidade produtiva. Cada área deve possuir objetivos que precisam ser alcançados para que a organização seja considerada como tendo atingido determinado nível.

O Projeto CMMI surgiu para resolver os problemas de adequação das organizações as várias versões do CMM. Segundo NOGUEIRA (2006, p.32), ao modelo original, especificamente destinado à manutenção de *software*, denominado *Software Capability Maturity Model (SW-CMMM)*, sucederam-se outros quatro modelos que o complementavam:

- *Capability Maturity Model for Software (SA-CMM)*, orientado para as atividades associadas à aquisição de *software*, fornecendo um referencial para avaliar a maturidade de uma organização na seleção, aquisição e instalação de um produto de *software* desenvolvido por terceiros;
- *Systems Engineering Capability Model (SE CMM)*, destinado a determinar o nível de maturidade de uma organização em seu contexto de Engenharia de Sistemas, concebidos como algo mais amplo do que o *software* propriamente dito, englobando o *software*, o *hardware* e quaisquer outros elementos que dele venham a fazer parte.
- *The Integrated Product Development Capability Maturity Model (IPD-CMM)* mais abrangente que o modelo original, uma vez que incorpora outros processos relacionados ao ciclo de vida do *software*, tais como suporte, processos de fabricação etc.

- *People – Capability Maturity Model (P-CMM)*, cujo objetivo é determinar o nível de maturidade de uma organização em seus processos relacionados à gestão de Recursos Humanos envolvidos nas atividades de *software* englobando os processos de recrutamento e seleção de desenvolvedores, treinamento, remuneração etc.

Prendia-se a combinação desses modelos em uma única estrutura de melhoria para viabilizar seu uso de forma menos custosa para as organizações. Esses três modelos foram selecionados como fontes por causa de sua ampla difusão na indústria de *software* e nas comunidades da engenharia de sistemas e por causa de suas diferentes abordagens para melhorar os processos em uma organização (CHRISISSIS *et al.*, 2003).

Vale ressaltar que esse conjunto de modelos foi desenvolvido por uma mesma organização, mas apresentava diversas inconsistências entre seus componentes. Nesse sentido, sua utilização e disseminação revelaram pontos nos quais correções se faziam necessárias. Além disso, ficou evidente a necessidade de compatibilização do modelo com outros modelos e metodologias, pois muitos de seus usuários adotavam concomitantemente os padrões ISO 9001:2000 e ISO/IEC 15004 (ambos publicados depois da versão original do SW-CMM).

O *Capability Maturity Model Integration (CMMI)* foi desenvolvido pelo SEI (*Software Engineering Institute*), centro de pesquisa e desenvolvimento criado em 1984 pelo Departamento de Defesa dos Estados Unidos cuja missão é aprimorar a prática de engenharia de *software*. Localiza-se na *Carnegie Mellon University*, na Pennsylvania, EUA.

O SEI surgiu a partir de uma solicitação do governo americano de um método para avaliar a capacitação dos seus fornecedores de *software*. Nesse sentido, calcado na aplicação dos conceitos de gerência de processos do *Total Quality Management* para o desenvolvimento de *software*, desenvolveu-se um modelo estruturado em estágios de maturidade que estabelecia uma base sólida de engenharia e de gerência de projeto, visando o controle quantitativo do processo de *software*. O objetivo deste trabalho era ajudar as organizações a melhorar os seus processos de software.

A substituição do modelo CMM pelo modelo CMMI deu-se a partir de 2005, quando somente organizações cujo processo de avaliação se encontrava em curso foram ainda avaliadas segundo o modelo antigo (NOGUEIRA, 2006)<sup>1</sup>.

O modelo CMMI também foi pensado para suportar futuras integrações com outras disciplinas. Ele já foi desenvolvido para ser consistente e compatível com a ISO/IEC15504, comumente chamada de SPICE – *Software Process Improvement and Capability Determination*.

No âmbito da variedade de modelos, o CMMI passou a incorporar o conceito de disciplinas que remetem, de certa forma, aos antigos modelos independentes do CMM. São elas (CMU/SEI, 2002):

- *System Engineering–SE* (Engenharia de Sistemas).
- *Software Engineering* (Engenharia de *Software*).
- *Integrated Product and Process Development – IPPD* (Desenvolvimento Integrado de Produtos e Processos).
- *Supplier Sourcing–SS* (Avaliação e Análise de Fornecedor ou Subcontratado).

Todas essas disciplinas foram criadas para se complementarem coerente e consistentemente. A partir de sua integração, o CMMI é representado em cinco modelos distintos: o CMMI-SE; o CMMI-SW; o CMMI-SE/SW; o CMMI-SE/SW/IPPD e o CMMI-SE/SW/IPPD/SS. A seleção do modelo a ser adotado se dá de acordo com as características e necessidades da organização.

#### 2.2.2.2 - Estrutura do Modelo CMMI

O CMMI segue a mesma linha dos modelos iniciais propostos pelo SEI, sendo organizado em níveis de maturidade com componentes específicas para cada estágio. Níveis de maturidade consistem em um conjunto predefinido de áreas de processo (PA)

---

<sup>1</sup> Com o lançamento do CMMI, o SEI propunha dar suporte ao CMM até dezembro de 2003. Até esta data, as avaliações seriam ainda feitas. Após esta data, o CMM deveria substituir o CMMI, com treinamentos e avaliações oficiais do SEI apenas para este modelo. No entanto, muitas empresas não conseguiram cumprir este prazo e, algumas sequer realizaram a migração para o CMMI.

e são medidos pela satisfação das metas específicas e genéricas que se aplicam a cada conjunto de áreas de processo. As metas específicas aplicam-se a uma área de processo e descrevem o que deve ser implementado para satisfazer a área de processo. As metas genéricas são denominadas assim, porque as mesmas metas aparecem em múltiplas áreas de processo. A satisfação de uma meta genérica para uma área de processo significa maior controle no planejamento e implantação dos processos associados a esta área e, assim, indica se estes processos serão eficazes, passíveis de repetição e duradouros. As metas são usadas nas avaliações para determinar se uma área de processo é satisfeita.

Cada área é detalhada em práticas genéricas e específicas, que são os quesitos a serem cumpridos na implantação do modelo. As práticas genéricas fornecem institucionalização para assegurar que os processos associados com a área de processo sejam eficazes, repetíveis e duradouros. As específicas descrevem as atividades esperadas para resultar no estabelecimento das metas específicas de uma área de processo. Uma prática específica é uma atividade considerada importante no estabelecimento da meta específica associada. As práticas são categorizadas pelas metas e características comuns.

Antes de começar a usar o modelo CMMI, a organização deve mapear seus processos de acordo com as áreas de processo CMMI. Este mapeamento permite controlar o processo de melhoria na organização, ajudando a determinar o nível da organização em conformidade com o modelo em uso. Não é esperado que toda área de processo CMMI seja mapeada uma a uma para os processos da organização.

O *Framework* CMMI pode gerar diferentes modelos baseados nas necessidades da organização:

- CMMI-SW: Modelo que contém a disciplina de Engenharia de *Software*;
- CMMI-SE: Modelo que contém a disciplina de Engenharia de Sistemas;
- CMMI-SE/SW: Modelo que integra as disciplinas Engenharia de Sistema e Engenharia de *Software*;
- CMMI-SE/SW/IPPD: Modelo que integra Engenharia de Sistema, Engenharia de *Software* e Desenvolvimento de Produto e Processo Integrado (DPPI);

- CMMI-SE/SW/IPP/SS: Modelo que integra Engenharia de Sistema, Engenharia de *Software*, Desenvolvimento de Produto e Processo Integrado (DPPI) e Desenvolvimento com Subcontratação (*Supplier Sourcing*).

Para cada modelo do CMMI existem duas representações, por estágio e contínua. Essas representações descrevem como a organização pode seguir seu plano de melhorias de processos orientado aos modelos do CMMI. Na primeira, a empresa segue níveis de maturidade, na segunda ela possui nível de capacidade em áreas de processos específicas. A figura 2 ilustra a representação pro estágio.

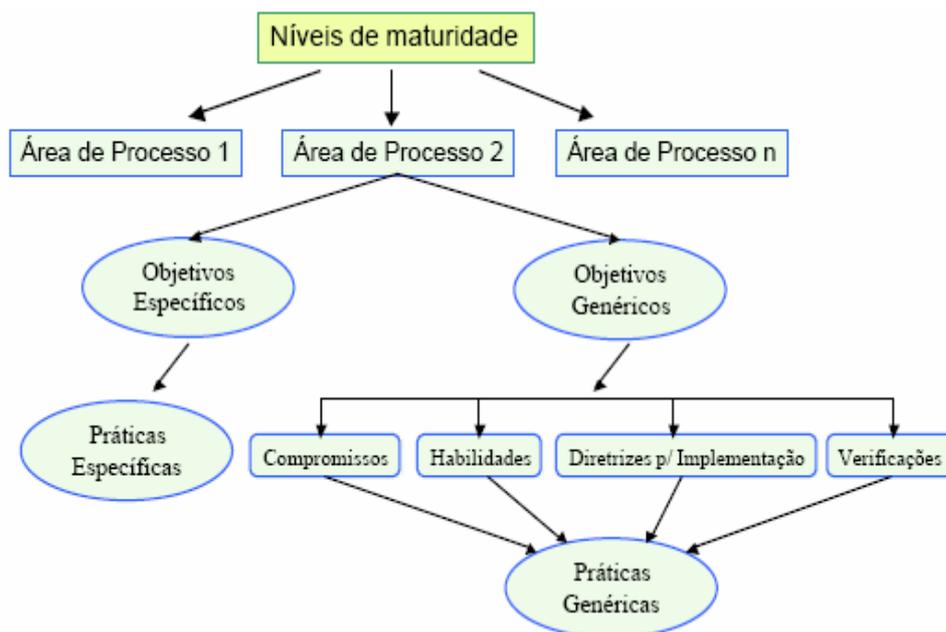


Figura 2 - Estrutura do Modelo CMMI por Estágio  
Fonte: CHRISSIS *et al.*, 2003.

Os graus de maturidade representam um caminho para o processo de melhoria indicando quais áreas de processo devem ser implantadas para se alcançar cada nível e ilustrando a evolução da melhoria para toda a organização. Assim, é possível prever o desempenho da organização dentro de uma dada disciplina ou conjunto de disciplinas. Cada nível estabelece uma parte importante do processo da organização. Nos modelos CMMI com representação em estágio, existem cinco níveis de maturidade designados pelos números de 1 a 5, (Fig. 3).

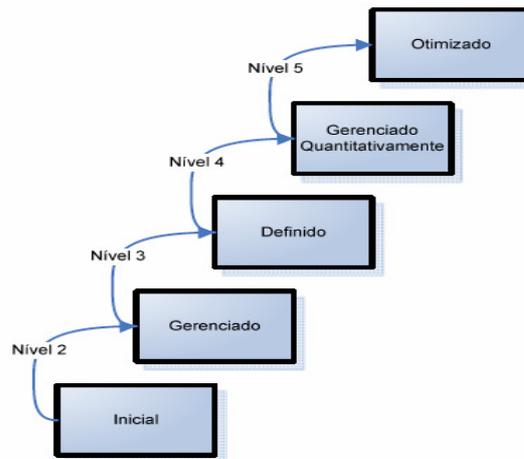


Figura 3 - Níveis de Maturidade do Modelo CMMI  
 Fonte: CHRISSIS *et al.*, 2003

#### Nível 1 – Inicial

No nível Inicial, a organização, de um modo geral, não fornece um ambiente estável para desenvolvimento e manutenção de *software*. Quando uma organização não dispõe de práticas de gestão bem estabelecidas, os benefícios das boas práticas de desenvolvimento de *software* são minados por um planejamento ineficiente e por um contexto no qual os compromissos são sempre reativos. Entende-se por compromissos reativos aqueles que são firmados como reação a algum acontecimento não previsto no planejamento.

Em meio a uma crise, os projetos tipicamente abandonam os procedimentos que foram planejados e partem para a codificação e testes. O sucesso depende inteiramente de se terem um gerente excepcional e uma equipe de *software* madura e eficiente. Ocasionalmente, gerentes de *software* eficientes, que dispõem de poder, podem resistir às pressões de recorrer a atalhos no processo de *software*, contudo, quando deixam o projeto, a segurança acaba. Isto implica afirmar que mesmo um forte processo de desenvolvimento não pode superar a instabilidade criada pela ausência de práticas sólidas de gestão.

A capacidade de processo de *software* em organizações de nível 1 é imprevisível, porque esse processo é constantemente modificado à medida que o trabalho progride. Cronogramas, orçamentos, funcionalidades e qualidade do produto são geralmente imprevisíveis. O desempenho depende da capacidade dos indivíduos e

varia com as suas habilidades, conhecimentos e motivações inatas. Raros são os processos de *software* estáveis em evidência e o desempenho só pode ser previsto através de habilidades individuais e não por meio da capacidade da organização.

#### Nível 2 – Gerenciado

No nível gerenciado, as políticas de gestão de projeto de *software* e os procedimentos para implementá-las são estáveis. As organizações, neste nível, estabeleceram todas as metas específicas e genéricas das áreas de processo do nível 2. O planejamento e a gestão de novos projetos baseiam-se na experiência adquirida em projetos similares. Um dos objetivos alcançados no nível 2 é a institucionalização dos processos para os projetos de *software*, possibilitando que as organizações repitam as práticas bem sucedidas desenvolvidas em projetos anteriores, apesar dos processos específicos implementados pelos projetos serem diferentes. Um processo efetivo pode ser caracterizado como hábil, documentado, robusto, treinado, medido e com capacidade de poder ser melhorado.

Os projetos nas organizações de nível 2 possuem controles básicos de gestão de *software*. Os compromissos realistas do projeto são baseados em resultados observados em projetos anteriores e nos requisitos do projeto atual. Os gerentes de *software* do projeto acompanham custos, cronogramas e funcionalidades do *software*, os problemas com compromissos são identificados quando surgem. Os requisitos e os produtos de trabalho de *software* desenvolvidos para satisfazê-los são armazenados de forma criteriosa e a integridade dos mesmos é controlada. Os padrões do projeto de *software* são definidos e a organização garante que eles sejam seguidos fielmente. Existindo subcontratados, o projeto de *software* trabalha em conjunto com os mesmos, visando fortalecer a relação cliente-fornecedor.

A capacidade de processo de *software* das organizações de nível 2 pode ser resumida como sendo disciplinada, uma vez que o planejamento e o acompanhamento de projeto de *software* são estáveis e os sucessos mais recentes podem ser repetidos. Os processos estão sob um controle efetivo de um sistema de gestão de projeto, seguindo planos realistas baseados no desempenho de projetos anteriores.

A Tabela 1 Mostra as áreas de processos de nível 2 e a finalidade de cada uma delas. Na seqüência, a Tabela 2 exhibe as metas específicas (SG) e genéricas (GG) de cada área de processo e com uma breve descrição.

Tabela 1. Áreas de Processos do Nível 2

| Áreas de Processos                                 | Finalidade   |
|--|--|
| Gerência de Requisitos (REQM)                      | Gerenciar os requisitos dos produtos e componentes de projeto identificando inconsistência entre os requisitos do projeto, o plano de projeto e os produtos trabalhos. |
| Planejamento de Projeto (PP)                       | Estabelecer e manter o plano que define as atividades do projeto.  |
| Controle e Monitoramento de Projeto (PMC)          | Prover o entendimento do progresso do projeto tomando medidas corretivas quando o desenvolvimento do projeto desvia-se significativamente do plano de projeto.         |
| Gerência de Acordos com Fornecedores (SAM)         | Definir um processo formal de aquisição de produtos junto aos fornecedores.  |
| Medição e Análise (MA)                             | Desenvolver a capacidade de medição fornecendo informações que servirão de suporte a gerência.   |
| Garantia da Qualidade do Processo e Produto (PPQA) | Fornecer uma gestão com visibilidade apropriada sobre os processos utilizados e produtos desenvolvidos pelo projeto de software.                                       |
| Gerência de Configuração (CM)                      | Estabelecer e manter a integridade dos produtos do projeto ao longo de todo o ciclo de vida.   |

Fonte: Próprio autor

Tabela 2. Metas para PAs do Nível 2

| PA   | Metas   | Descrição   |
|------|---|---|
| REQM | SG - Gerenciar Requisitos                     | Os requisitos são gerenciados e as inconsistências entre os planos de projeto e os produtos de trabalho são identificadas.  |
|      | GG - Institucionalizar um Processo Gerenciado | Um processo gerenciado é um processo executado, que é planejado e executado com concordância com a política da empresa. Esta meta está presente em todas PAs do nível 2.                              |
|      | GG - Institucionalizar um Processo Definido   | Um processo definido é um processo gerenciado que é suportado pela organização com um conjunto padrão de processos definidos, acordados e alinhados. Esta meta está presente em todas PAs do nível 2. |
| PP   | SG - Estabelecer Estimativas                  | Estimativas de planejamento de projeto são estabelecidas e mantidas.  |
|      | SG - Desenvolver Plano de Projeto             | Os compromissos do projeto são estabelecidos e mantidos.  |
| PMC  | SG - Monitorar o Projeto e o Plano            | Monitora o progresso e os resultados do projeto de encontro ao que está definido no plano.  |

|      |  |   |
|------|--|---|
| SAM  | SG- Gerenciar Ações Corretivas                               | Gerencia as ações corretivas tomadas quando o projeto desvia-se do planejado.   |
|      | SG - Estabelecer Acordos com Fornecedores                    | Acordos com fornecedores são estabelecidos e mantidos.  |
|      | SG - Satisfazer Acordos com Fornecedores                     | Acordos feitos com fornecedores são satisfeitos tanto por eles quanto pelo projeto.   |
| MA   | SG - Alinhar Medidas e Análises as Atividades                | As medidas e análise são alinhadas a necessidades de informação e aos objetivos.  |
|      | SG - Fornecer Resultados de Medidas                          | Fornecer resultados de medidas realizadas endereçando as informações pretendidas<br>Verificar aderência aos processos e avaliar o desempenho dos trabalhos quanto à aplicação desses. |
| PPQA | SG - Avaliação Objetiva dos Processos e Produtos de Trabalho |   |
| CM   | SG- Visão introspectiva dos Processos                        | Atividades não seguidas são comunicadas e resoluções definidas.   |
|      | SG - Estabelecer <i>Baselines</i>                            | Estabelecer <i>baselines</i> para o desenvolvimento de produtos.  |
|      | SG- Registro de Mudanças de Controle                         | Mudanças nos produtos trabalhados são registradas e controladas.  |

Fonte: Próprio autor

### Nível 3 – Definido

No nível Definido, o processo padrão de desenvolvimento e manutenção de *software* global da organização é documentado, incluindo o desenvolvimento de *software* e a gestão de processos, estando estes últimos integrados em um todo coerente. As organizações, neste nível, estabeleceram todas as metas específicas e genéricas das áreas de processo listadas pelo nível 2 e nível 3. Os processos, no nível 3, são utilizados para apoiar os gerentes de *software* e o pessoal técnico, contribuindo para que possam atuar de forma mais efetiva. Ao padronizar seus processos de *software*, a organização exercita práticas de engenharia de *software* efetivas. Existe uma equipe responsável pelas atividades de processo de *software* da organização, por exemplo, a equipe de processo de engenharia de *software*. Um programa de treinamento é implementado em toda a organização para garantir que os gerentes e

funcionários tenham os conhecimentos e as habilidades necessárias ao cumprimento de suas funções.

Os projetos adaptam o processo de *software* padrão da organização para desenvolver seus próprios processos de *software* definidos, que consideram as características únicas de cada projeto. Um processo bem definido caracteriza-se como possuidor de critérios de disponibilidade, entradas, padrões e procedimentos para realização do trabalho, mecanismos de verificação (revisões por pares), saídas e critérios de finalização. Como o processo de *software* é bem definido, a gerência tem uma boa percepção do progresso técnico em todos os projetos.

A capacidade de processo de *software* das organizações de nível 3 pode ser resumida como padronizada e consistente, porque tanto as atividades gestão como a engenharia de *software* são estáveis e repetíveis. Nas linhas de produtos estabelecidas, a qualidade de *software* é acompanhada, além do custo, cronograma e funcionalidades estarem sob controle. Esta capacidade de projeto baseia-se em um entendimento global da organização com relação às atividades, regras e responsabilidades decorrentes do processo de *software* definido.

A Tabela 1.3 mostra as áreas de processos de nível 3 e a finalidade de cada uma delas. Em seguida, a Tabela 4 exhibe as metas específicas (SG) e genéricas (GG) de cada área de processo.

Tabela 3. Áreas de Processos do Nível 3

|   |   |
|---|---|
| Áreas de Processos                            | Finalidade  |
| Desenvolvimento de Requisitos (RD)            | Produzir e analisar os requisitos dos clientes, produtos e componentes dos produtos.  |
| Solução Técnica (TS)                          | Projetar, desenvolver e implementar soluções para os requisitos dentro do ciclo de vida do projeto.   |
| Integração de Produto (PI)                    | Desenvolver o produto a partir dos seus componentes, assegurar que a integração funciona como planejado e entregar o produto.               |
| Verificação (VER)                             | Garantir que o produto está de acordo com a especificação de requisitos.  |
| Validação (VAL)                               | Demonstrar que o produto ou seus componentes funcionam como esperado no ambiente pretendido.  |
| Foco do Processo na Organização (OPF)         | Planejar e implementar melhorias de processos na organização baseados nas suas fragilidades e forças.                                       |
| Definição do Processo Organizacional (OPD)    | Estabelecer e manter um conjunto de processos usável por toda organização.  |
| Treinamento da Organização (OT)               | Desenvolver nos colaboradores da empresa habilidades e conhecimentos que possibilitem realizar seus trabalhos de forma efetiva e eficiente. |
| Gerência de Projeto Integrada (IPM)           | Estabelecer e gerenciar o envolvimento dos relevantes stakeholders com o projeto de acordo com os processos da organização.                 |
| Gerência de Risco (RSKM)                      | Identificar potenciais problemas antes de ocorrerem definindo estratégias para contorná-los ou minimizar seus impactos.                     |
| Análise e Resolução de Decisões (DAR)         | Analisar possíveis decisões usando um processo formal de avaliação que identifica critérios pré-estabelecidos.                              |
| Ambiente Organizacional para Integração (OEI) | Prover infraestrutura para integração de produtos e processo de desenvolvimento (IPPD) bem como gerência de pessoas para integração.        |

Fonte: Próprio autor

Tabela 4. Metas para PA'S de Nível 3

| PA | Metas                                       | Descrição   |
|----|---|---|
| RD | SG - Desenvolver Requisitos do Cliente      | Necessidades, restrições e expectativas dos <i>stakeholders</i> são coletadas e traduzidas como requisitos do cliente.  |
|    | SG - Desenvolver Requisitos do Produto      | Os requisitos do cliente são refinados e transformados nos requisitos do produto.   |
|    | SG - Análise e Validação dos Requisitos     | Os requisitos são analisados e validados e uma definição dos requisitos de funcionalidade é Um processo definido é um processo gerenciado que é suportado pela organização com um conjunto padrão de processos definidos, acordados e alinhados. Esta meta está presente em todas PAs do nível 3. |
|    | GG - Institucionalizar um Processo Definido |   |

|     |   |   |
|-----|---|---|
| PA  | Metas   | Descrição   |
|     | GS - Selecionar os Componentes do Produto                                 | Produtos ou componentes do produto são selecionados como uma solução alternativa.   |
| TS  | GS - Desenvolver o Projeto  | Desenvolver o projeto do produto.   |
|     | GS - Implementar o Projeto do Produto                                     | Desenvolver o produto a partir do seu projeto e das documentações de suporte.   |
|     | SG - Preparar para Integração do Produto                                  | A preparação para a integração do produto é conduzida.  |
| PI  | SG- Garantir a Compatibilidade das Interfaces                             | Verificar se as interfaces internas e externas dos componentes do produto são compatíveis.  |
|     | SG- Montar os Componentes do Produto e Entregar o Produto                 | Verificar se os componentes do produto podem ser integrados, montar o produto, verificar, validar e entregar o produto.                       |
|     | SG - Preparar para Verificação  | A preparação para verificação é conduzida.  |
| VER | SG - Realizar Revisão por Pares   | Revisão por pares é realizada em alguns produtos de trabalho selecionados.  |
|     | SG - Verificar os Produtos de Trabalho Selecionados                       | Os produtos de trabalhos selecionados para verificação são comparados com a especificação de requisitos.                                      |
|     | SG - Preparar para Validação  | Preparação para validação é conduzida.  |
| VAL | SG - Validar Produto ou Componentes de Produto                            | Os produtos e componentes de produtos são validados para garantir que serão apropriados para o ambiente de operação pretendido.               |
|     | SG - Determinar oportunidades de Melhoria de Processos                    | Fragilidades, forças e oportunidades de melhoria de processos para a organização são identificados periodicamente bem como suas necessidades. |
| OPE | SG - Planejamento e Implementação das Atividades de Melhoria de Processos | Melhorias são planejadas e implementadas, os recursos são identificados e as experiências relacionadas são incorporadas aos processos.        |
| OPD | SG - Estabelecer os Recursos para os Processos da Organização             | Um conjunto de recursos necessários para a melhoria de processos é estabelecido e mantido.  |
|     | SG - Estabelecer Treinamentos das Capacidades da Organização              | Treinamento das capacidades de gerência e técnica de toda a organização é estabelecido e mantido.   |
| OT  | SG - Prover Treinamento Necessário  | Treinamento individual de funcionários é realizado para esse desenvolver suas tarefas mais eficientemente.                                    |

|      | Metas  | Descrição   |
|------|--|---|
| PA   | SG - Uso de Processos Definidos de Projeto                           | O projeto é conduzido usando processos definidos que foram traçados pela organização a partir de processos padrões. |
| IPM  | SG - Coordenação e Colaboração com os <i>Stakeholders</i> Relevantes | Coordenação e Colaboração com os <i>stakeholders</i> relevantes para o projeto são conduzidas.                      |
|      | SG - Preparação para Gerência de Risco                               | A preparação para gerência de risco é conduzida.  |
| RSKM | SG - Identificação e Análise dos Riscos                              | Riscos são identificados e analisados para determinar sua relativa importância.                                     |
|      | SG - Mitigar Riscos  | Os riscos são controlados e atenuados para diminuir os seus impactos.   |
| DAR  | SG - Avaliar Alternativas  | Decisões são tomadas baseadas na avaliação de alternativas usando critérios pré-estabelecidos.                      |
|      | SG - Prover Infraestrutura de IPPD                                   | Prover uma infraestrutura que maximize a produtividade das pessoas e favoreça as necessidades de integração.        |
| OEI  | SG - Gerenciar Pessoas para Integração                               | Pessoas são orientadas para prover um ambiente de colaboração para IPPD.  |

Fonte: Próprio autor

#### Nível 4 - Quantitativamente Gerenciado

No nível Quantitativamente Gerenciado, a organização estabelece metas quantitativas de qualidade para os produtos e processos de software. As organizações neste nível estabeleceram todas as metas específicas das áreas de processo listadas pelo nível 2, nível 3 e nível 4 e as metas genéricas listadas pelo nível 2 e nível 3. A produtividade e a qualidade são medidas nas atividades importantes de processo de *software* em todos os projetos, como parte de um programa organizacional de medições. Um banco de dados de processo de *software*, que abrange a organização toda, é utilizado para coletar e analisar os dados disponíveis dos processos de *software* definidos dos projetos. No nível 4, os processos de *software* são instrumentalizados com medições consistentes e bem definidas. Essas medições estabelecem as bases para avaliar os processos e os produtos de *software* do projeto.

Os projetos possuem controle sobre seus produtos e processos, reduzindo a variação no desempenho desses últimos, para atingir limites quantitativos aceitáveis.

Os riscos decorrentes da introdução de um novo domínio de aplicação são conhecidos e cuidadosamente gerenciados.

A capacidade de processo de *software* das organizações de nível 4 pode ser resumida como sendo previsível, porque o processo é medido e opera dentro de limites mensuráveis. A capacidade de processo desse nível permite que a organização preveja as tendências na qualidade dos produtos e dos processos dentro das fronteiras quantitativas desses limites. Quando esses limites são excedidos, são tomadas providências para corrigir a situação.

A Tabela 5 mostra as áreas de processos de nível 4 e a finalidade de cada uma delas. Na seqüência, a Tabela 6 exibe as metas específicas (SG) e genéricas(GG) de cada área de processo e com uma breve descrição.

Tabela 5. Áreas de Processos do Nível 4

| Áreas de Processos                           | Finalidade   |
|--|--|
| Desempenho dos Processos da Organização(OPP) | Estabelecer e manter o entendimento quantitativo do desempenho dos processos padrões da organização dando suporte a qualidade e ao controle da efetividade dos seus processos. |
| Gerência Quantitativa de Projeto(PQM)        | Estabelecer a gerência quantitativa de projeto para melhorar a qualidade e desempenho.   |

Fonte: Próprio autor.

Tabela 6. Metas das PAs do Nível 4

| PA  | Metas   | Descrição   |
|-----|---|---|
| OPP | SG - Estabelecer Desempenho de <i>Baselines</i> e Modelos | <i>Baselines</i> e modelos que caracterizam o desempenho esperado dos processos padrões da organização são estabelecidos e mantidos.  |
|     | GG - Institucionalizar um Processo Definido               | Um processo definido é um processo gerenciado que é suportado pela organização com um conjunto padrão de processos definidos, acordados e alinhados. Esta meta está presente em todas PAs do nível 4. |
| PQM | SG - Gerência Quantificável de Projeto                    | O projeto é quantitativamente gerenciado usando a qualidade e o desempenho de processos.  |
|     | SG– Estatística de Desempenho de Subprocessos             | O desempenho de subprocessos de processos definidos está estatisticamente gerenciado.   |

Fonte: Próprio autor.

### Nível 5 – Otimizado

No nível Otimizado, toda a organização está voltada para a melhoria contínua de processo. As organizações, neste nível, estabeleceram todas as metas específicas das áreas de processo listadas pelo nível 2, nível 3, nível 4 e nível 5 e as metas genéricas listadas pelo nível 2 e nível 3. A organização tem meios de identificar as oportunidades de melhoria e fortalecer o processo de maneira pró-ativa, com o objetivo de prevenir a ocorrência de falhas. Os dados sobre a efetividade de processo de software são utilizados para realizar análises de custo benefício das novas tecnologias e das mudanças propostas no processo de software da organização. As inovações decorrentes das melhores práticas de engenharia de software são identificadas e transferidas para a organização toda.

As equipes de projeto de software das organizações de nível 5 analisam as falhas para determinar suas causas. Os processos de software são revistos para prevenir tipos de falhas já conhecidas que normalmente se repetem e as lições aprendidas são disseminadas para outros projetos.

A capacidade de processo de software das organizações de nível 5 pode ser resumida como sendo de melhoria contínua porque estas estão continuamente se esforçando para melhorar a abrangência de sua capacidade de processo, melhorando dessa forma o desempenho dos processos de seus projetos. As melhorias ocorrem através de avanços incrementais nos processos já existentes e através de inovações que utilizam novos métodos e tecnologias.

A Tabela 1.7 mostra as áreas de processos desse nível e a finalidade de cada uma delas. Na seqüência, a Tabela 1.8 exhibe as metas específicas (SG) e genéricas (GG) de cada área de processo e com uma breve descrição.

Tabela 7. Áreas de Processos do Nível 5

| Áreas de Processos                             | Finalidade  |
|--|---|
| Desenvolvimento e Inovação Organizacional(OID) | Inovar e desenvolver melhorias de forma incremental, de acordo com os objetivos da organização, que permitam medir o desempenho dos seus processos e tecnologias. |
| Análise e Resolução de Causa(CAR)              | Identificar as causas de defeitos e problemas e tomar decisões para prevenir que esses não aconteçam no futuro.   |

Fonte: Próprio autor.

Tabela 8. Metas das PAs de Nível 5

| PA  | Metas                                       | Descrição   |
|-----|---|---|
|     | SG - Selecionar Melhorias                   | Selecionar melhorias de processos e tecnologias que contribuirão com a qualidade e o desempenho dos processos.  |
| OID | SG - Desenvolver Melhorias                  | Medidas de melhorias de processos e tecnologias da organização são contínuas e sistematicamente tomadas.  |
|     | GG - Institucionalizar um Processo Definido | Um processo definido é um processo gerenciado que é suportado pela organização com um conjunto padrão de processos definidos, acordados e alinhados. Esta meta está presente em todas PAs do nível 5. |
|     | SG - Determinar as Causas dos Defeitos      | As principais causas dos defeitos e problemas são sistematicamente determinadas.  |
| CAR | SG - Prevenir as Causas dos Defeitos        | Tomar decisões para prevenir que os defeitos e problemas já ocorridos voltem a acontecer.   |

Fonte: Próprio autor.

Desde 1991, outros modelos de maturidade surgiram, TickIT [TickIT], BOOTSTRAP (KUJAVA, 1993,1994), ISO12207, ISO15204, dentre outros, todos abordando várias disciplinas, principalmente as de engenharia de sistemas, engenharia de *software*, aquisição de *software*, gerenciamento e desenvolvimento de processos integrados.

Embora estes modelos se mostrassem úteis a muitas organizações, o uso de múltiplos modelos mostrou-se inadequado. As diferenças entre as disciplinas específicas desses modelos, suas arquiteturas, conteúdo e abordagem limitaram a habilidade destas organizações de focar com sucesso seus processos de melhorias. A aplicação de vários modelos não-integrados em uma organização pode onerá-la em termos de treinamento, das avaliações e das atividades da melhoria.

#### Processo de Software Brasileiro (MPS.BR):

De acordo com Nogueira (2006), a indústria de desenvolvimento de *software* no Brasil, ao longo da última década, vem demonstrando uma evolução no que tange à questão da qualidade de *software*. Entretanto, o modelo que apresentava maior aceitação é o NBR ISO: 9.001 que, como já foi discutido, não foi desenvolvido

especificamente para o contexto de *software*, o que resulta em problemas para sua interpretação e adaptação a esse segmento. A utilização dos demais modelos apresenta uma série de barreiras no Brasil.

A primeira barreira é o custo, pois o CMMI, por seu custo elevado, fica, quase sempre, restrito às grandes empresas do mercado, em sua maioria filiais de corporações multinacionais (WEBER *et al.*, 2005).

Para minorar a situação, a Sociedade para a Promoção da Excelência do Software Brasileiro (SOFTEX) apoia o desenvolvimento do MPS.BR (Melhoria de Processo do Software Brasileiro). SOFTEX, 2005.

De acordo com Weber *et al.* (*apud* NOGUEIRA, 2006, p.38):

Do ponto de vista conceitual, quatro elementos merecem destaque. Em primeiro lugar, foi concebido de modo a prover um modelo de referência evolutivo, nos mesmos moldes dos modelos CMMI e ISO/IEC 15504, possibilitando que as organizações que o adotem possam desenvolver um plano de desenvolvimento de longo prazo. Apresenta um escalonamento de níveis de maturidade com uma maior granularidade nos níveis iniciais que aquela utilizada nos modelos citados, permitindo uma estratégia de implantação e evolução mais adequada às condições do mercado brasileiro. Em terceiro lugar, é plenamente aderente e compatível com todos os demais modelos citados [...] Finalmente, a operacionalização de sua institucionalização se traduz em um Modelo de Negócio que incorpora tanto os princípios utilizados pelo Software *Engineering Institute* – SEI para a avaliação de organizações conforme o modelo CMMI, quanto aqueles recomendados pela *International Organization for Standardization* – ISO para as avaliações de conformidade de terceira parte (independência entre agente avaliador e agente implementador e reavaliação periódica). Além disso, o Modelo de Negócio contempla a possibilidade de estratégias de implantação específicas para empresas específicas ou para grupos de empresas cooperadas- estratégia esta que facilita sobremaneira sua utilização por empresas de pequeno porte por dividir os custos e dispor de linhas de financiamento

Segundo Nogueira (2006), a estrutura do projeto é formada por 3 componentes: o Modelo de Referência, o Modelo de Avaliação e o Modelo de Negócio. Cada um deles descrito por documentos adquire *software* e guias específicos. Soma-se a estes um Guia de Aquisição destinado às organizações relacionadas a essa atividade (SOFTEX, 2005).

O Modelo de Referência (MR-MPS) define uma estrutura que combina os conceitos de maturidade de uma organização, definidos pelo CMMI, com o de capacidade de processo estabelecido pela ISO/IEC 15504. No MR-MPS, níveis de

maturidade “são uma combinação entre processos e capacidade de processos” (SOFTEX. 2005).

No MR-MPS, a definição dos processos reproduz a forma utilizada pela norma ISO/IEC 12207, onde a descrição dos processos inclui a declaração de seu propósito e dos resultados esperados a partir da efetiva implementação do processo em questão. No modelo, o termo propósito descreve o principal objetivo a ser atingido durante a execução do processo. Além disso, o modelo incorpora informações adicionais acerca dos processos que são, segundo a Softex (2005 p. 27): “referências que podem ajudar na definição do processo pela organização. Em geral, elas citam o processo ou o Subprocessos da NBR ISO/IEC 12207 que estão correlacionados ao processo do MR-MPS, os quais possuem uma descrição das atividades, tarefas ou melhores práticas para implementar o processo”.

A capacidade do processo, por outro lado, é um conjunto de atributos de processo descrito em termos de resultados os quais proporcionam o atendimento dos atributos de processo. A capacidade estabelece o grau de refinamento e institucionalização com que o processo é executado na organização. O modelo estabelece ainda que “a capacidade do processo possui cinco atributos de processos (AP) [...] Cada AP está detalhada em termos dos resultados para alcance completo do atributo” (SOFTEX, 2005). O Quadro a seguir apresenta a relação dos atributos de processos com suas respectivas descrições.

| Atributo | Característica                                       | Descrição   |
|----------|--|---|
| AP 1.1.  | O processo é executado.                              | O processo atinge seu propósito.  |
| AP 2.1.  | O processo é gerenciado                              | O atributo é uma medida da extensão na qual a execução do processo é gerenciada.  |
| AP 2.2.  | Os produtos de trabalho do processo são gerenciados. | O atributo é uma medida da extensão na qual os produtos do trabalho produzidos pelo processo são gerenciados apropriadamente. |
| AP 3.1.  | O processo é definido.                               | O atributo é uma medida da extensão na qual um processo-padrão é mantido para apoiar a implementação do processo definido.    |
| AP       | O processo está                                      | O atributo é uma medida da extensão na qual o processo-   |

|      |               |  |
|------|---------------|--|
| 3.2. | implementado. | padrão é efetivamente implementado como um processo definido para atingir seus resultados. |
|------|---------------|--|

Quadro 1 MR-MPS: atributos de processo  
Fonte: NOGUEIRA (2006 p.40).

De acordo com Nogueira (2006), a partir de seu arcabouço evolutivo, com base nos pressupostos acima, são estabelecidos níveis de maturidade que indicam estágios de evolução dos processos de uma dada organização e permitem prever seu desempenho em uma ou mais disciplinas.

Vale ressaltar que a sistemática de classificação em níveis de maturidade assemelha-se àquela estabelecida no modelo CMMI, sendo que a diferença está no escalonamento dos níveis. Assim:

Em uma correlação entre estes modelos tendo como ponto de referência as “Áreas de Processo (AP)” do CMMI *vis-à-vis* os “Processos” do MR-MPS, pode-se afirmar que o equivalente ao nível 2 do CMMI se apresenta de certo modo desdobrado em 2 níveis de maturidade no modelo MR-MPS: o mesmo se dá com o nível 3 do CMMI, que é decomposto em 3 níveis distintos (NOGUEIRA, 2006, p. 41).

Dessa forma, o modelo MR-MPS propõe 7 níveis de maturidade, Parcialmente Gerenciado (G), Gerenciado (F), Parcialmente Definido (E), Largamente Definido (D), Definido (C), Gerenciado Quantitativamente (B) e Em Otimização (A). A situação definida na Representação por Estágios do CMMI como Nível 1 e na Representação Contínua ou no modelo ISO/IEC 15504, como nível 0, denominados por letras de A a G, sendo o nível A o mais elevado de todos.

| Nível                          | Processo   | Capacidade                               |
|--------------------------------|--|--|
| (em otimização)                | Inovação e implantação na organização.<br>Análise e Resolução de Causas.                 | AP1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2.  |
| (gerenciado quantitativamente) | Desempenho do processo organizacional.<br>Gerência Quantitativa do Projeto               | AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2. |
| (Definido)                     | Análise de Decisão e Resolução<br>Gerência de Riscos                                     | AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2. |
| (Largamente definido)          | Desenvolvimento de requisitos<br>Solução técnica<br>Integração do produto<br>Verificação | AP 1.1, AP 2.1 e AP 2.1.                 |

|                           | Validação  |  |
|---------------------------|--|--|
| (Parcialmente definido)   | Treinamento<br>Avaliação e melhoria do processo organizacional<br>Definição do processo organizacional<br>Adaptação do processo para gerência do projeto | AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2. |
| (Gerenciado)              | Medição<br>Gerência de configuração<br>Aquisição<br>Garantia de qualidade  | AP 1.1, AP 2.1 e AP 2.2.                 |
| (parcialmente gerenciado) | Gerência de Requisitos<br>Gerência de Projeto  | AP 1.1 e AP 2.1.                         |

Quadro 2 MR-MPS: níveis de maturidade  
Fonte: NOGUEIRA, (2006, p.41).

Ainda Nogueira (2006) observa que, para cada um dos níveis de maturidade, são estabelecidos perfis de processos que devem ser contemplados pela organização com a respectiva capacidade de processo, definida em termos de atributo do processo (AP). A classificação de uma organização em um determinado nível de maturidade “decorre do atendimento de processos correspondentes a esse nível de maturidade” (SOFTEX, 2005). Em seu caráter evolutivo, o modelo é cumulativo, ou seja, a fim de atender aos requisitos de um dado nível, a organização deve, também, atender às exigências de todos os níveis predecessores para os processos relacionados.

Cabe enfatizar o esforço brasileiro na elaboração do modelo MR-MPS que, dada a sua concepção, pode ser implantado de forma menos impactante do que os demais modelos, além de permitir uma visibilidade dos resultados em prazos mais curtos. O modelo MR-MPS representa uma opção mais adequada às necessidades de micros e pequenas empresas, parcela significativa do mercado brasileiro de desenvolvimento de *software*. Acrescente-se a isso o fato de ser possível, a partir de um único modelo, buscar não apenas evidenciar externamente o nível de maturidade da organização, mas também dispor de um *framework* para a melhoria dos processos da organização.

De acordo com BARRETO (2006), o MPS.BR define um processo específico para a Garantia da Qualidade do Processo e do Produto, além dos processos Verificação e Validação que, também, se preocupam com a qualidade do produto de *software*.

Nesse sentido, a autora comenta que, apesar de a primeira avaliação MPS.BR ter sido realizada em setembro de 2005, até janeiro de 2006, cinco empresas já haviam sido avaliadas com sucesso pelo MPS.BR. Dentre essas, uma foi avaliada no nível G, em Pernambuco, três no nível F, sendo uma no estado do Rio de Janeiro e duas no estado de São Paulo e uma no nível E, também, no estado do Rio de Janeiro (MPS.BR, 2005). Nesse contexto, Barreto (2006) enfatiza que, além dessas avaliações já concluídas, outras estão em andamento ou serão iniciadas em breve.

Dessa forma, pode-se afirmar que existem várias abordagens que visam a melhorar a qualidade do processo de *software* que, além de proporcionar melhor qualidade aos produtos desenvolvidos, podem trazer outros benefícios às organizações.

BARRETO (2006) enfatiza que algumas empresas, visualizando esses benefícios, têm investido na melhoria da qualidade de seus processos, mas ainda há muito a ser feito para que se possa falar de uma indústria de *software* de alta qualidade no Brasil.

Em dezembro de 2006, foi apresentado o Guia de Implementação do MPS.BR - Melhoria de Processo do *Software* Brasileiro – Parte 4: Nível D (Versão 1.0), que contém orientações para a implementação do Nível D do Modelo de Referência MR-MPS (ANEXO).

### 2.3 - TESTES DE VERIFICAÇÃO E VALIDAÇÃO DE SOFTWARE:

Verificação é o ato de revisar, testar, verificar e documentar itens, processos, serviços ou documentos que estão ou não conforme os requisitos especificados. Validação é a avaliação do software no final do processo de desenvolvimento para assegurar a conformidade com os requisitos de usuário. Validação é, portanto, o fim da verificação (ESA, 1991, pp. 2, 21).

De acordo com GRANDCHAMP (2002), a verificação é essencial para assegurar a qualidade do produto; pode ser cara e consumir grande parte do tempo do projeto.

Por isso, essa atividade deve aparecer no Plano de Gerenciamento do Projeto de *Software*. As atividades de verificação incluem:

- Revisões técnica, ensaios e inspeções de *software*;
- Verificação de requisitos determinados pelos usuários;
- Verificação do projeto de componentes determinados pelos requisitos de *software*;
- Verificação de algoritmos;
- Testes de integração;
- Testes de sistema;
- Testes de aceitação.

Os processos de teste são utilizados para exercitar ou avaliar um sistema ou componente, de forma manual ou automática para confirmar se os requisitos especificados foram satisfeitos ou identificar diferenças entre os resultados esperados e atuais.

GRANDCHAMP (2002) considera que a quantidade de tempo gasto com os testes de *software* é freqüentemente subestimada. Estas atividades devem ser cuidadosamente especificadas para que possam ser adequadamente orçadas. Os gastos aumentam com o número de erros presentes antes do início da fase de testes. Métodos mais baratos de remoção de erros, como inspeção e ensaio devem sempre ser experimentadas antes do início dos testes.

O autor ressalta que os testes de *software* incluem as seguintes atividades:

- Planejamento da abordagem e alocação de recursos;
- Detalhamento da abordagem através de vários tipos de testes;
- Definição dos dados de entrada, predição dos resultados e condições de execução em cada especificação de teste;
- Início da seqüência de testes pela equipe de testes amparada por um procedimento;
- Registro das execuções em um relatório de testes.
- Os testes devem ser produzidos nos mesmos padrões do *software* que será entregue.

Como se pode notar, verificação de *software* não é simples, uma vez que várias avaliações devem ser realizadas ao longo de um projeto e cada avaliação requer planejamento, controle e uso de técnicas de verificação adequadas. O planejamento da verificação pode ser iniciado no planejamento do projeto e refinado ao longo do projeto, devendo estar integrado ao plano do projeto.

O teste *de software* (TS) é uma das atividades de verificação e validação com o objetivo de “avaliar uma característica ou recurso de um programa ou sistema” (HETZEL, 1987, p.6), sendo entendida como elemento crítico para a GQS (ROCHA *et al.*, 2001).

Para a execução da tarefa de TS, a literatura aponta diversas técnicas e metodologias, cuja importância reside no fato de que, com elas, far-se-ão o planejamento e desenho dos casos de teste.

De acordo com MYERS (1979), faz-se essencial um bom planejamento para que se obtenham bons casos de teste; do contrário, corre-se o risco de obter um conjunto de casos que não encontrará erros (o verdadeiro objetivo da atividade), ou um número excessivo de casos, o que resultaria em elevação dos custos do processo de desenvolvimento e em uma relação custo-benefício não compensadora. Nesse sentido, recomenda-se que as técnicas sejam combinadas no planejamento de um teste rigoroso de um programa, pois as mesmas não são excludentes, mas sim, complementares, ao mesmo tempo em que possuem diferentes forças e fraquezas.

### **2.3.1 - Objetivos dos Testes:**

O objetivo da atividade de teste pode ser entendido da seguinte forma:

- No início de cada fase verificar se esta etapa do projeto reflete exatamente os requisitos e definições da fase imediatamente anterior, para garantir que o produto encomendado e o gerado pela atividade de desenvolvimento do *software* será o mesmo por meio dos diferentes níveis de refinamento do projeto;

- Verificar se não existe erros de lógica no projeto e código, no fluxo de dados, no entendimento de requisitos, de codificação, tipográficos ou de interface em todas as fases do projeto;
- Identificar e interferir na presença do erro, iniciando-se a depuração, pois, quanto antes for descoberta a falha, menos custos serão dispendidos para adequá-la;
- Ter em mente que, uma vez que “errar é humano” e atividade de desenvolvimento de *software* é um exercício complexo, os erros existem e devem ser descobertos, portanto o sucesso em um teste consiste em descobri-los e corrigi-los.

Os testes têm como objetivo a detecção de defeitos e existem diversos meios de tornar mais eficientes e efetivos os esforços relacionados aos testes. O teste de software é um elemento crítico da garantia de qualidade de *software* e representa a revisão final da especificação, projeto e geração de código.

### **2.3.2 - Níveis de Teste:**

Divide-se basicamente em três níveis (ou fases), de acordo com as etapas do processo de desenvolvimento nas quais ela está sendo aplicada (ROCHA e cols.,2001). De acordo com HERBERT (1999) encontram-se comumente:

- teste de unidade: teste isolado das menores unidades (módulos) que constituem o programa, na busca de identificar erros de lógica e comparar a função de um módulo com alguma definição de funcionalidade ou interface do mesmo (ROCHA e cols.,2001; PETERS e cols.,2001);
- teste de integração: identifica erros de programação relacionados à interface ou suposições errôneas sobre os módulos conforme eles se vão integrando e formando módulos maiores (ROCHA e cols.,2001).
- teste de sistema: teste do sistema como um todo, comparando o mesmo com seus objetivos originais (MYERS, 1979). Os erros revelados por este tipo de

teste são os que não podem ser detectados no teste individual de módulos ou integração, como erros de funcionais e características de desempenho que não estejam de acordo com a especificação (PETERS *et al.*, 2001; ROCHA *et al.*, 2001).

### 2.3.3 - Tipos de Testes

#### 2.3.3.1 - Teste Estático Versus Teste Dinâmico:

A literatura apresenta uma classificação das formas de teste de acordo com a forma de execução do mesmo. Nesse sentido, para PRESSMAN (2000) e ROCHA *et al.*(2001) o TS é uma atividade essencialmente dinâmica e ambas as abordagens seriam complementares. HETZEL (1987) e PETERS (2001), por sua vez, entendem que a revisão de *software* é uma atividade estática de TS, e não um complemento da mesma.

O teste estático, também, conhecido como teste não-computacional e revisão de *software*. Esse tipo de teste não envolve a execução propriamente dita do programa. As técnicas de teste estático se baseiam na leitura ou na inspeção visual de programas por um grupo de pessoas com o objetivo de encontrar erros. O custo de correção de erros encontrados por este método parece ser mais baixo, porque a origem precisa do erro é localizada, enquanto que os testes dinâmicos acusam apenas sintomas dos erros. Entretanto, estes métodos não são eficazes na detecção de erros de modelagem ou análise de requisitos. Como exemplos, citam-se as inspeções de código, o percorrimto e as avaliações por pares ou *peer-reviews* (MYERS, 1979).

O teste dinâmico tem por objetivo encontrar erros no *software* por meio de execução do mesmo. Testes estáticos e dinâmicos são complementares por

encontrarem tipos diferentes de erros, e a detecção dos mesmos vai sofrer se um ou outro não estiver presente (MYERS, 1979; ROCHA *et al.* 2001).

Os testes estáticos e dinâmicos são complementares, por encontrarem tipos diferentes de erros, e a detecção dos mesmos vai sofrer se um ou outro não estiver presente (MYERS, 1979; ROCHA *et al.*, 2001).

### 2.3.3.2 - Técnicas de testes dinâmicos

#### 2.3.3.2.1 - *Teste estrutural ou teste de caixa-branca*

Esta estratégia de TS permite examinar a estrutura interna de um programa e desenhar casos de teste a partir do exame dessa estrutura. PRESSMAN (1995, p. 794) argumenta que “erros lógicos e pressuposições incorretas são inversamente proporcionais à probabilidade de que um caminho de programa seja executado”, assim como erros de sintaxe que não são detectados até que os testes se iniciem. No entanto, uma das principais falhas dessa estratégia é que ela não garante que o programa esteja de acordo com a especificação e não revelaria problemas funcionais, erros de saída de dados ocasionados por certos valores de entrada (MYERS, 1979). Pelo uso dos métodos desta estratégia, pode-se desenhar casos de teste segundo os critérios utilizados em cada técnica, ou seja:

- abrangência de instrução: objetiva a execução de cada instrução do código pelo menos uma vez (PETERS e cols., 2001). Existem técnicas e métricas que podem auxiliar na visualização do fluxo de controle do programa e o número de caminhos independentes a serem percorridos durante o teste, como o grafo de fluxo (PRESSMAN, 1995) que descreve o fluxo de controle lógico do programa na forma de grafos onde cada círculo representa uma ou mais instruções de código fonte sem ramificação.

- abrangência de ramificação: põe à prova todas condições lógicas contidas em um programa, forçando todos os valores de saída possíveis (PETERS e cols., 2001).
- abrangência de ramificação/condição (PETERS e col.,2001), que consiste em testar todas as combinações possíveis de condições em decisões.
- teste de fluxo de dados: este método de teste define os caminhos a serem percorridos em um programa a partir das localizações das definições e usos de variáveis no mesmo. Tendo identificado as definições e usos das variáveis, são estabelecidas *associações definição-uso* (ou associação DU), e a estratégia de teste de fluxo de dados exige que cada associação DU seja coberta pelo menos uma vez (PRESSMAN, 1995).

#### 2.3.3.2.2 - Teste funcional ou teste de caixa-preta (*black box*)

O testador se preocupa em encontrar circunstâncias nas quais o programa não se comporta de acordo com as especificações funcionais, considerando as especificações funcionais como a origem mais imediata de casos de teste, os dados de teste são derivados somente das mesmas, não considerando a estrutura interna do programa. O teste de caixa preta tende a ser aplicado durante as últimas etapas da atividade de teste (PRESSMAN, 1995). As principais estratégias deste tipo de teste são:

- Particionamento de equivalência: esta técnica permite derivar casos de teste a partir da divisão do domínio de entrada de um programa em classes de dados equivalentes (PRESSMAN, 1995). Os casos de teste devem ser derivados a partir das classes de equivalência identificadas. (MYERS, 1979);
- Análise de valores limites: considerando que “*um número maior de erros tende a ocorrer nas fronteiras do domínio de entrada*” (PRESSMAN, 1995, p. 819), esta técnica foi desenvolvida para possibilitar casos de teste que ponham à prova os valores limites de entrada. Esta é uma técnica de teste que complementa o particionamento de equivalência, pois ao invés de

selecionar um elemento qualquer de uma classe de equivalência, os casos de testes derivados trabalham nas extremidades da classe (MYERS, 1979).

- Grafo de causa-efeito: esta técnica é um método sistemático de gerar casos de teste que representem combinações de condições de entrada e suas ações correspondentes a partir da tradução de uma especificação em uma rede de lógica booleana (MYERS, 1979; PRESSMAN, 1995).

### 2.3.3.3 - Testes Unitários:

Estágio mais baixo da escala de testes e são aplicados nos menores componentes de código criados.

- Testes de Integração: executados em combinação de componentes para verificar se eles funcionam corretamente juntos. Identificação de erros de programação relacionados à interface ou suposições errôneas sobre os módulos, conforme eles vão sendo integrados e formando módulos maiores (ROCHA *et al.*, 2001; MYERS, 1979). Para realizar testes de integração, existem duas abordagens, cada uma com suas vantagens e desvantagens, para guiar o sentido de integração dos módulos, conforme eles vão sendo testados. São elas as abordagens *top-down* e *bottom-up* (MYERS, 1979).

- Testes de Regressão

Visa garantir que o software permaneça intacto depois de novos testes serem realizados.

- Testes de Carga

Visam avaliar a resposta de um *software* sob uma pesada carga de dados, repetição de certas ações de entrada de dados, entrada de valores numéricos grandes, consultas complexas a base de dados, grande quantidades de usuários simultâneos para verificar o nível de escalabilidade.

- Testes back-to-back

O mesmo teste executado em versões diferentes do *software* e os resultados são comparados.

- Testes de Configuração

Verificam se o software está apto a funcionar em diferentes versões ou configurações de ambientes (*hardware e software*).

- Testes de Usabilidade

Verificam o nível de facilidade de uso do software pelos usuários.

- Testes de Instalação

Verificam o processo de instalação parcial, total ou atualização do *software*.

- Testes de Segurança

Validam a capacidade de proteção do software contra acessos interno ou externo não autorizados.

- Testes de Recuperação

Validam a capacidade e qualidade da recuperação do software após falhas de hardware ou problemas externos.

- Testes de Compatibilidade

Validam a capacidade do software de executar em um particular ambiente de *hardware/software/sistema* operacional ou rede.

- Testes de Desempenho/Performance

Visam garantir que o sistema atende os níveis de desempenho e tempo de resposta acordados com usuários e definidos nos requisitos.

- Teste de Qualidade de Código

Trata-se de grupos de testes com o intuito de verificar o código fonte dos programas em consonância com padrões, melhores práticas, instruções não executadas e outros.

- Testes de Alterações

Visam rastrear alterações de programas durante o processo de teste.

- Testes de Recuperações de Versões

Os testadores verificam a capacidade de retornar a uma versão anterior do *software*.

- Testes de Interoperabilidade

Os testadores avaliam as condições de integração com outros *softwares* e/ou ambientes.

- Testes de Sobrevivência

Os testadores avaliam a capacidade do *software* em continuar operando mesmo quando algum elemento (*software* ou *hardware*) fica inoperante ou pára de funcionar.

- Testes Estéticos

Os testadores avaliam toda a documentação do projeto, tais como modelos, requisitos etc.

- Teste Embutido

Os testadores avaliam a capacidade de integração entre o *hardware* e o *software*.

- Teste de Conferência de Arquivos

Os testadores verificam alterações nos arquivos usados.

- Testes Alfa

Executados quando o desenvolvimento está próximo a ser concluído.

- Testes Beta

Executados, quando o desenvolvimento e testes estão praticamente concluídos.

- Teste de Verificação de Sites Web

Os testadores verificam problemas no *site* como *links* inválidos, arquivos órfãos, ligações entre páginas (MOLINARI, 2003).

## CAPÍTULO III

### 3. FUNDAMENTOS DA PESQUISA

Com referência nas normas internacionais da área de qualidade de *software*, neste capítulo, será visto um método para avaliar qualidade de produto de *software* pronto, disponível para utilização, com o foco de visão do usuário, tendo referência no Processo de Avaliação estabelecido pela Norma NBR ISO/IEC 9126 e utilizando o Método de Avaliação de Qualidade de Produtos de *Software* - MEDE-PROS®.

Compreende, ainda, a definição do instrumento de coleta de dados, a escolha do local da pesquisa e da empresa a ser utilizada como estudo de caso. Contém também o método de análise e os problemas encontrados na coleta de dados.

#### 3.1 - MEDE-PROS®

Tem a Missão de apoiar os desenvolvedores e outros interessados na melhoria da qualidade de *software*, por meio da geração, absorção e disponibilização de tecnologia. Não se configura como uma ferramenta de domínio público. Ele está registrado junto à Fundação Biblioteca Nacional (MEDE-PROS®, 1997) e pertence ao Centro de Pesquisas Renato Archer (CenPRA, MINISTÉRIO DA CIÊNCIA E TECNOLOGIA) instituição associada ao Ministério da Ciência e Tecnologia, que tem a finalidade de desenvolver e implementar pesquisas científicas e tecnológicas no setor de informática, sendo que o domínio e a disseminação do conhecimento tecnológico é o seu foco de atuação, para isso credencia laboratórios para execução de avaliações de produtos de *software* utilizando: INSOFT; UNISINOS; ITS; UEL; USP - São Carlos.

Nem todas as informações sobre o MEDE-PROS® podem ser apresentadas devido ao Termo de Sigilo. Apesar dessa restrição, a qualidade deste trabalho não foi

prejudicada, pois o insumo principal, de ambos, é a suportar a crescente demanda por Avaliação da Qualidade de Produtos de Software.

O CenPRA fornece aos avaliadores instrumentos para avaliação de software, por meio de métodos que definem as características e atributos de modelo de qualidade, tendo como destaque, o Método de Avaliação da Qualidade de Produto de Software (MEDE-PROS)

### 3.1.1 - Base Teórica do MEDE-PROS®

O MEDE-PROS® foi desenvolvido com o objetivo de fornecer subsídios para a melhoria da qualidade de produtos de *software*, segundo a visão do usuário final, verificando está atendendo os padrões das Normas Internacionais de Qualidade:

- NBR 13596 (ISO 9126) *software*;
- NBR ISO/IEC 12119 (descrição do produto);
- ISO 9241 (*interfaces*);
- ANSI IEEE 1063 (documentação);
- NBR ISO/IEC 12119 (Embalagem).

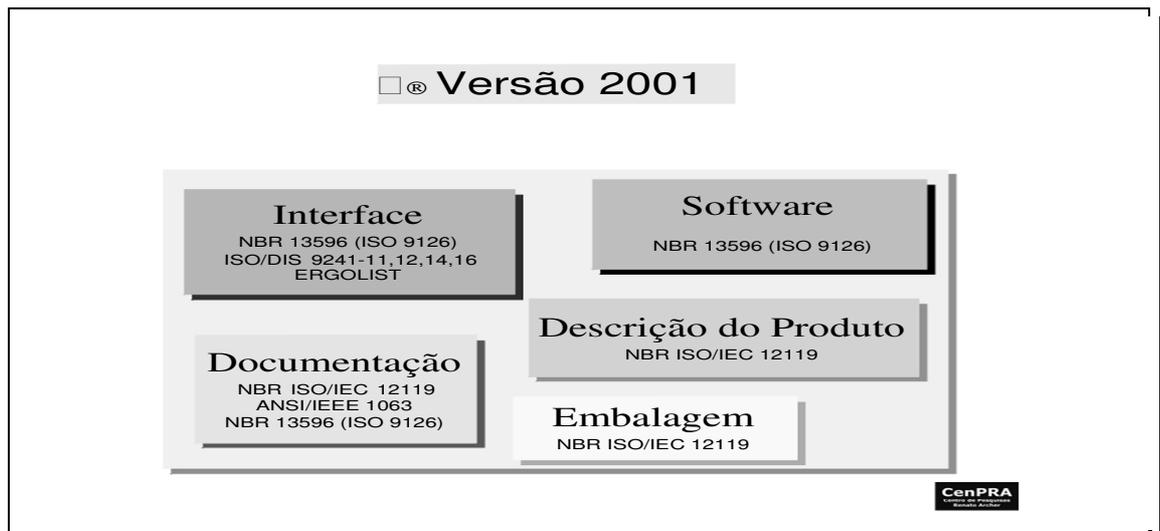


Figura 4 - Mede Pros  
Fonte: Cenpra / MCT

O resultado sobre a qualidade do produto são Relatórios de Avaliação dos aspectos positivos e dos aspectos a serem revistos de acordo com a visão do usuário.

De acordo com TSUKUMO (1995), o estabelecimento de Normas para a Avaliação da Qualidade de Produto de *Software* possibilita a adoção de uma base conceitual comum, capaz de permitir a aceitação universal de avaliações e certificações de produtos.

Nessa linha, a ISO/IEC: 9126 (NBR 13596, 1996) – “Tecnologia de Informação – Avaliação de produto de *software* – Características de qualidade e diretrizes para o seu uso” – define as características de qualidade de *software* que devem estar presentes em todos os produtos de *software*: Funcionalidade, Confiabilidade, Eficiência, Usabilidade, Manutenibilidade e Portabilidade, servindo como referência na avaliação de produtos de *software*, pois cobre os aspectos mais importantes.

As seis características de qualidade de *software*, segundo a Norma ISO/IEC 9126, são apresentadas a seguir. Uma preocupação da Norma foi definir características com um mínimo de sobreposição de conceitos entre elas e por isso são desdobradas em subcaracterísticas, como mostra a Tabela 8, a seguir.

Os atributos, mencionados na definição de ‘funcionalidade’, caracterizam ‘o quê’ o *software* faz para satisfazer as necessidades implícitas e explícitas do usuário. Os atributos mencionados nas demais características de qualidade caracterizam principalmente ‘quando’ e ‘como’ o *software* procura satisfazê-las.

Tabela 8 – Características e subcaracterísticas de qualidade segundo a norma 9126

| Características | Subcaracterísticas  | Descrições  |
|-----------------|---------------------|---|
| Funcionalidade  | Adequação           | Presença de um conjunto de funções e sua apropriação para as tarefas especificadas. |
|                 | Acurácia            | Geração de resultados ou efeitos corretos.  |
|                 | Interoperabilidade  | Capacidade de interagir com outros sistemas especificados.                          |
|                 | Conformidade        | Estar de acordo com normas, convenções e regulamentações.                           |
|                 | Segurança de acesso | Capacidade de evitar acesso não autorizado a programas e dados.                     |

|                  |                                     |  |
|------------------|-------------------------------------|--|
| Confiabilidade   | Maturidade                          | Freqüência de falhas por defeitos.   |
|                  | Tolerância a falhas                 | Capacidade de manter o nível de desempenho em caso de falha no <i>software</i> ou de violação nas interfaces.            |
|                  | Recuperabilidade                    | Capacidade de restabelecer seu desempenho e restaurar dados após falha.  |
| Usabilidade      | Inteligibilidade                    | Atributos do <i>software</i> que evidenciam o esforço do usuário para reconhecer o conceito lógico e sua aplicabilidade. |
| Usabilidade      | Apreensibilidade                    | Atributos do <i>software</i> que evidenciam o esforço do usuário para aprender sua aplicação.                            |
|                  | Operacionalidade                    | Atributos do <i>software</i> que evidenciam o esforço do usuário para sua operação e controle da sua operação.           |
| Eficiência       | Comportamento em relação ao tempo   | Tempo de resposta, de processamento e velocidade na execução de funções.   |
|                  | Comportamento em relação a recursos | Quantidade de recursos utilizados e duração de seu uso.  |
| Manutenibilidade | Analisabilidade                     | Esforço necessário para diagnosticar deficiência e causas de falhas.   |
|                  | Modificabilidade                    | Esforço necessário para realizar modificações e remoção de defeitos.   |
|                  | Estabilidade                        | Ausência de riscos de efeitos inesperados ocasionados por modificações.  |
|                  | Testabilidade                       | Facilidade de ser testado.   |
| Portabilidade    | Adaptabilidade                      | Capacidade de ser adaptado a ambientes diferentes.   |
|                  | Capacidade para ser instalado       | Esforço necessário para a instalação.  |
|                  | Conformidade                        | Consonância com padrões ou convenções de portabilidade.  |
|                  | Capacidade para Substituir          | Capacidade e esforço necessário para substituir outro <i>software</i> .  |

Fonte: NOGUEIRA, (2006, p.41).

A ISO/IEC 9126 propicia um entendimento dos conceitos que definem as diversas características e subcaracterísticas da qualidade de produto de *software*. Na prática, entretanto, ainda não facilita o suficiente a definição dos requisitos de qualidade a partir dela (ROCHA, 2001), sendo esse um dos motivos para a criação e utilização do MEDE-PROS®.

ISO/IEC: 12119 (NBR ISO/IEC 12119, 1998) – “Tecnologia de Informação – Pacotes de *Software* – Teste e requisitos de qualidade”.

Nesse sentido, da mesma forma que a Norma ISO/IEC 9126 foi traduzida para a Norma NBR 13596, a Norma ISO/IEC 12119 também o foi, recebendo a denominação NBR ISO/IEC 12119.

A NBR ISO/IEC: 12119 estabelece requisitos de qualidade para um tipo particular de produtos de *software* denominado “pacotes de *software*”, fornecendo instruções de como testá-los em relação aos requisitos definidos. Como exemplos de pacotes de *software* citam-se processadores de texto, planilhas eletrônicas, bancos de dados, *software* gráficos, programas para funções técnicas ou científicas e programas utilitários.

De acordo com essa Norma, o pacote de *software* a ser testado ou a ter sua qualidade avaliada deve dispor dos seguintes elementos: Descrição do Produto, Documentação do Usuário e Programas e Dados.

- Descrição do Produto: um documento impresso (*folder*) ou disponível nas páginas da internet, que estabelece as propriedades do produto de *software* a ser adquirido por um potencial comprador, antes da aquisição, com o propósito de orientar na avaliação da adequação do produto. Caso o produto de *software* não disponha da descrição, considera-se uma não-conformidade para o processo de avaliação da qualidade do produto. A Norma considera para a avaliação dos requisitos de qualidade da Descrição do Produto:
  - Requisitos gerais sobre o conteúdo – a Descrição do Produto deve existir ser inteligível completa e apresentar boa organização e apresentação, devendo incluir os seguintes aspectos:

- Identificações e indicações – a descrição do produto deve ser identificada podendo ser, por exemplo, Descrição Funcional ou Informação de Produto; a descrição do produto deve conter a identificação: do produto devendo ter no mínimo o nome do produto e uma versão ou data; do fornecedor, contendo no mínimo o nome e endereço de, no mínimo, um fornecedor do produto; das tarefas que podem ser executadas, utilizando-se o produto; dos requisitos de *hardware* e *software* para colocar o produto em uso; das interfaces com outros produtos, caso o produto faça interface com algum; dos componentes físicos do produto a ser fornecidos, em particular todos os documentos impressos que acompanham o produto e todos os meios de armazenamento de dados; da possibilidade ou não da instalação do produto ser conduzida pelo usuário; do fornecimento ou não de suporte para a operação do produto; do fornecimento ou não de manutenção no produto e em caso do fornecimento, o que está incluído.
- Declarações sobre funcionalidade – a descrição do produto deve fornecer uma visão geral das funções disponíveis para o usuário do produto, os dados necessários e as facilidades oferecidas; no caso do produto ser limitado por valores limite específicos, estes devem ser fornecidos na descrição do produto; informações para evitar o acesso não-autorizado a programas e dados devem estar na descrição do produto, caso o produto possua esse recurso.
- Declarações sobre confiabilidade – a descrição do produto deve incluir informações sobre procedimentos para preservação de dados, por exemplo, que é possível efetuar *backup* através de funções do sistema operacional, que a entrada de dados é verificada, que o produto dispõe de função para recuperação de erro, dentre outros.
- Declarações sobre usabilidade – a descrição do produto deve especificar o tipo de interface com o usuário (linha de comando, *menu* ou janelas), o conhecimento necessário para a utilização do produto, a possibilidade do usuário adaptar o produto a suas necessidades e o que é necessário,

proteção contra infrações a direitos autorais, quando esta dificultar a usabilidade do produto.

- Declarações sobre eficiência – a descrição do produto pode especificar o comportamento do produto em relação ao tempo.
- Declarações sobre manutenibilidade – a descrição do produto pode conter declarações sobre manutenibilidade.
- Declarações sobre portabilidade - a descrição do produto pode conter declarações sobre portabilidade.
- Documentação do Usuário: Conjunto completo de documentos disponível em forma impressa ou não, fornecido ao usuário para a utilização do produto de *software*.

Para a avaliação dos requisitos de qualidade da Documentação do Usuário, são considerados pela Norma os seguintes aspectos:

- Completitude – a documentação do usuário deve conter as informações necessárias para o uso do produto. Todas as funções estabelecidas na descrição do produto e todas as funções do programa que os usuários tenham acesso devem ser completamente descritas na documentação do usuário. Todas as outras informações incluídas na descrição do produto devem estar na documentação do usuário também.
- Correção – todas as informações na documentação do usuário devem estar corretas, sem ambigüidades e erros.
- Consistência – a documentação do usuário não deve apresentar contradições internas entre suas partes e com a descrição do produto.
- Inteligibilidade – a documentação do usuário deve ser inteligível pela classe de usuários que, normalmente, executa a tarefa a ser atendida pelo produto, utilizando, por exemplo, uma seleção apropriada de termos, exibições gráficas, explicações detalhadas e citando fontes úteis de informação.
- Visão Geral – a documentação do usuário deve apresentar boa apresentação e organização, de tal modo que quaisquer relacionamentos sejam facilmente identificados, por exemplo, por índices analítico e remissivo.

- Programas e Dados: Conjunto completo de programas de computador e dados, fornecidos para a utilização do produto de *software*. Esse conjunto é parte integrante do produto de *software*.

Para a avaliação dos requisitos de qualidade para Programas e Dados, a Norma considera os seguintes aspectos:

- Funcionalidade – são avaliados o processo de instalação do produto de *software*; verificada a existência e execução de todas as funções descritas na documentação do usuário e as declarações contidas na descrição do produto, em particular, programas e dados que devem estar de acordo com todos os requisitos definidos em qualquer documento de requisitos citados na descrição do produto.
- Confiabilidade – é avaliada a capacidade do produto de *software* em manter a integridade dos programas e dados, permitindo ao usuário controlar o sistema compreendendo hardware e *software*.
- Usabilidade – é avaliado se perguntas, mensagens e resultados dos programas são inteligíveis; se as mensagens de erro fornecem informações detalhadas explicando a sua causa ou forma de correção; se é possível ao usuário, quando estiver trabalhando com os programas, descobrir qual função está sendo executada; se os programas fornecem ao usuário informações claramente visíveis e fáceis de serem lidas, guiando o usuário por informações codificadas e agrupadas adequadamente; se, quando necessário, os programas alertam o usuário; se as mensagens dos programas são projetadas de forma que o usuário as possa diferenciar facilmente pelo tipo, por exemplo: confirmação, solicitações, advertências ou mensagens de erro.
- Eficiência – é avaliado se o produto está em conformidade com as declarações de eficiência citadas em sua descrição de produto.
- Manutenibilidade – é avaliado se o produto está em conformidade com as declarações de manutenibilidade citadas em sua descrição de produto.
- Portabilidade – é avaliado se o produto está em conformidade com as declarações de portabilidade citadas em sua descrição de produto.

A série de normas ISO/IEC: 14598 é um conjunto de normas que define o processo de avaliação de produtos de *software*. (WEBER, 2001).

- ISO/IEC: 14598-1 – Visão Geral – apresenta toda a estrutura de funcionamento da série de normas para avaliação da qualidade dos produtos de *software*, além de definir os termos técnicos utilizados nesse modelo. Fornece, também, os conceitos e o funcionamento do processo de avaliação da qualidade de qualquer tipo de *software*, para utilização por desenvolvedores (incluindo gerentes, analistas de requisitos, projetistas de *software*, implementadoras e equipe de garantia da qualidade), por adquirentes e por avaliadores de *software* independentes. De maneira geral, pode ser utilizada por pessoas envolvidas no desenvolvimento, padronização e uso de tecnologia de avaliação.
- ISO/IEC: 14598-2 – Planejamento e Gestão – apresenta requisitos, recomendações e orientações para uma função de suporte ao processo de avaliação dos produtos de *software*. O suporte está relacionado ao planejamento e gerenciamento de um processo de avaliação de *software* e a tecnologia necessária, incluindo: desenvolvimento, aquisição, padronização, controle, transferência e realimentação do uso de tecnologia de avaliação no âmbito da organização.
- ISO/IEC: 14598-3 – Processo para Desenvolvedores – destina-se ao uso, durante o processo de desenvolvimento e manutenção de *software*, enfocando a seleção e registro de indicadores que possam ser medidos e avaliados a partir dos produtos intermediários, obtidos nas fases de desenvolvimento de sistemas, com o objetivo de prever a qualidade do produto final a ser desenvolvido, de modo a orientar a tomada de decisões técnicas e gerenciais ao longo do processo de desenvolvimento.
- ISO/IEC: 14598-4 – Processo para Adquirentes – estabelece um processo sistemático para avaliação de produtos de *software* tipo pacote, produtos de *software* sob encomenda, ou ainda modificações em produtos já existentes. O objetivo da avaliação pode ser a comparação entre diversas alternativas de produtos existentes no mercado, ou a tentativa de garantir que um produto

desenvolvido ou modificado sob encomenda atenda aos requisitos inicialmente especificados. A norma utiliza o processo de avaliação definido genericamente na ISO/IEC 14598-1.

- ISO/IEC: 14598-5 – Processo para Avaliadores – fornece orientações para a implementação prática da avaliação de produto de *software*, quando diversas partes necessitam entender, aceitar e confiar em resultados da avaliação. O processo descrito define as atividades necessárias para analisar os requisitos da avaliação de modo a especificar, projetar e executar as atividades da avaliação e para se obter a conclusão sobre a avaliação de qualquer tipo de produto de *software*.
- ISO/IEC: 14598-6 – Documentação de Módulos de Avaliação – define a estrutura e o conteúdo da documentação a ser usada na descrição dos módulos de avaliação. Explica como desenvolver módulos de avaliação e como validá-lo. Um Módulo de Avaliação é um conjunto de instruções e dados usados para a avaliação. Especifica os métodos de avaliação aplicáveis para avaliar as características de qualidade; define também os procedimentos elementares de avaliação e o formato do relatório de apresentação dos resultados das medições resultantes das aplicações dos métodos e das técnicas. O uso de módulos de avaliação produzidos e validados, conforme a norma deve garantir que as avaliações de produtos de *software* possam ser:
  - Repetíveis – um mesmo produto, avaliado mais de uma vez, com a mesma especificação de avaliação e o mesmo avaliador deve apresentar resultados com um mínimo de variação.
  - Reprodutíveis – um mesmo produto, avaliado mais de uma vez, com a mesma especificação de avaliação e diferentes avaliadores deve apresentar resultados com um mínimo de variação.
  - Imparciais – a avaliação deve ser livre de tendências.
  - Objetivas ou empíricas – a avaliação deve ser baseada em fatos reais.

### 3.1.2 - Estrutura do MEDE-PROS®

O MEDE-PROS ® é formado por três componentes:

- Lista de Verificação
- Manual do Avaliador
- Modelo de Relatório de Avaliação

A Lista de Verificação constitui ferramenta de avaliação, utilizada pelos avaliadores durante o processo de avaliação da qualidade de produtos de *software*. Foi elaborada, tomando-se como base as Normas de qualidade já apresentadas, além das (ISO/DIS 9241, 1996), (ISO/DIS 9241-10, 1996), (ISO/DIS 9241-11, 1997), (ISO/DIS 9241-12, 1997), (ISO/DIS 9241-14, 1997), (ISO/DIS 9241-16, 1997), (ANSI/IEEE 1063, 1987) e do método (ERGOLIST, 1997).

A Lista de Verificação é composta por um conjunto de atributos e esses, por um conjunto de questões.

Em um processo de avaliação podem ser avaliados: Embalagem, Descrição do Produto, Documentação do Usuário, Interface e *Software*. Entretanto, nem sempre os produtos submetidos a um processo de avaliação apresentam todos esses elementos. Portanto, a Embalagem é um meio físico que acondiciona a mídia e documentos impressos.

Para a Descrição do Produto, pode-se utilizar a definição presente na norma NBR ISO/IEC 12119, já descrita no item anterior. A Documentação do Usuário é o conjunto completo de documentos, disponível ao usuário na forma impressa ou não, fornecida para auxiliar na utilização dos produtos de *software*.

A Interface permite que as informações sejam transferidas entre o usuário e os componentes de *hardware* ou *software* de um sistema computacional. O *Software* são as instruções (programas de computador) que, quando executadas pelo usuário, produzem a função e o desempenho esperados.

A Lista de Verificação é organizada pelos seguintes elementos do produto de *software*: Embalagem, Descrição do Produto, Documentação do Usuário, Interface e

*Software*. Para cada um desses elementos, são considerados os requisitos contidos nas respectivas normas de qualidade associadas.

A aplicação desse método permite a realização de uma avaliação completa ou parcial. Em uma avaliação completa, são avaliados os seguintes elementos de um produto de *software*: Embalagem, Documento de Descrição do Produto, Documentação do Usuário, Interface e *Software*. Como o método está estruturado por esses elementos, é possível excluir do processo de avaliação qualquer um deles se o solicitante da avaliação assim desejar, realizando-se uma avaliação parcial.

Ainda, a condição de método de avaliação genérico, faz com que ele trate as características de qualidade de forma geral, ficando, muitas vezes, a cargo da experiência do avaliador a realização de alguns testes mais específicos, principalmente aqueles relacionados com as funcionalidades do sistema, o que torna a avaliação bastante subjetiva no contexto de requisitos funcionais do *software*.

Na presente Dissertação, como a avaliação do produto de *software* baseia-se na comparação do produto contra alguns requisitos, ou ainda contra necessidades explícitas e implícitas dos usuários, o trabalho realizado durante a elaboração do MEDE-PROS® foi o de transformar os requisitos de qualidade presentes nas normas de qualidade em atributos e esses em um conjunto de questões, de tal forma que, avaliando o atributo através do conjunto de questões associadas fosse possível julgar o atendimento ou não do requisito.

Para a elaboração do questionário da pesquisa foram utilizadas as características e subcaracterísticas da norma ISO/IEC: 9126, atual padrão mundial para a Qualidade de produtos de *software*, apresentadas no quadro 8.

O modelo de qualidade é utilizado para identificar os atributos dos requisitos de avaliação. As subcaracterísticas de funcionalidades auxiliam na construção dos itens.

Para estabelecer os requisitos de usabilidade, para os componentes interface e documentação do usuário, foi utilizada como base a lista de itens de verificação já construídos para o método de avaliação de produto do MEDEPROS®, desenvolvido com base no modelo de qualidade da NBR: 9126.

Entretanto, apesar da norma definir cada característica e Subcaracterística de qualidade, ela não define como medi-las. Há a necessidade, então, de um trabalho de

definição das métricas a serem aplicadas em cada uma das subcaracterísticas de qualidade. Como, por exemplo: Característica de qualidade de funcionalidade:

- Subcaracterística Adequação:- quantidade de funções atendidas, que poderão ser subdivididas em desejáveis e obrigatórias.
- Subcaracterística Segurança de Acesso:- definição de notas a serem atribuídas de acordo com o atendimento de segurança de acesso a dados e funções; somente a funções; somente a dados; ou não possui segurança de acesso.

A definição da forma de medição das características e subcaracterísticas de qualidade também podem ser efetuadas de acordo com o que se quer obter do produto a ser adquirido, ou seja, pode-se definir a medição somente para aquelas características que são determinantes para o produto de software desejado. Aliado a esta medição, a separação das características de qualidade permite a atribuição de pesos de forma a reforçar a importância desta ou daquela característica.

Neste estudo, devem ser verificadas as seguintes questões-chave em relação ao método de avaliação de produto MEDE-PROS®, desenvolvido com base no modelo de qualidade da ISO/IEC: 9126.

- Em relação à Funcionalidade: O conjunto de funções satisfaz as necessidades explícitas e implícitas para a finalidade a que se destina o produto?
- Em relação à Confiabilidade: O desempenho se mantém ao longo do tempo e em condições estabelecidas, ou seja, não é tolerante a falhas?
- Em relação à Usabilidade: O *software* é fácil de usar?
- Em relação à Eficiência: Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido pelo produto?
- Em relação à Manutenibilidade: É fácil corrigir, atualizar e alterar o *software*?
- Em relação à Portabilidade: É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?

Existe, também, a necessidade de se verificarem algumas questões-chave em relação ao CMMI:

- Os produtos foram selecionados para verificação?

- O ambiente de verificação foi devidamente estabelecido?
- Os procedimentos e critérios de verificação foram estabelecidos?
- A revisão pelos participantes foi devidamente preparada?
- A revisão foi devidamente conduzida?
- Os dados de revisão dos participantes foram analisados?
- A verificação foi realizada?
- Os resultados das verificações foram analisados e ações de correção foram identificadas?
- Os produtos foram selecionados para validação?
- O ambiente de validação foi devidamente estabelecido?
- Os procedimentos e critérios de validação foram estabelecidos?
- A validação foi realizada?

### 3.2 - PERFIL DO AMBIENTE DA PESQUISA

O universo da pesquisa é formado pela empresa X, fundada em 1988, tornou-se empresa líder no fornecimento de soluções corporativas de Tecnologia da Informação. Auxilia seus clientes na melhoria dos seus processos de gestão empresarial e de gestão da informação, trazendo expertise em tecnologia, qualidade no gerenciamento de projetos de TI, sempre através da combinação do conhecimento técnico com o conhecimento do negócio dos clientes.

O portfólio de clientes consiste prioritariamente nas 200 maiores empresas do mercado brasileiro, com forte atuação nos setores de Finanças, Indústria & Serviços, Telecom, Energia e Governo.

A Empresa X está estruturada para prestar serviços em todo o território nacional, através de escritórios nas grandes cidades brasileiras. A empresa conta também com uma subsidiária em Frankfurt, Alemanha, visando atender à crescente demanda por serviços de tecnologia da informação no mercado europeu, proporcionando forte base

de conhecimento com relação às tendências tecnológicas e econômicas dos segmentos de mercado em que atua.

A Empresa X conta com um corpo de consultores altamente especializados nas mais diversas práticas tecnológicas e um corpo gerencial voltado para soluções segmento-específico, reunindo um elenco capaz de desenvolver soluções diferenciadas para as necessidades dos clientes.

Como empresa de serviços orientada por mercados, a Empresa X tem no relacionamento com seus clientes e na satisfação de seus colaboradores sua principal plataforma de alavancagem para sustentar o seu forte crescimento histórico - da ordem de 30% por ano. A empresa entende que a melhor maneira de aumentar a competitividade é oferecer serviços sob medida e específicos para os mercados em que atua.

Através de soluções personalizadas, a Empresa X atende de forma inovadora ao ciclo de vida tecnológico, baseado em atividades sinérgicas que garantem o atendimento com excelência operacional e flexibilidade, adotando a filosofia *Plan, Build e Run* para que todas as etapas relativas às complexas necessidades de tecnologia sejam atendidas, com alto padrão de qualidade, conhecimento e experiência, garantindo assim, o sucesso dos projetos de sistemas de informação:

*Plan* (planejamento)

Alinhamento das necessidades de tecnologia da informação com a estratégia de negócio, foco nos processos operacionais / gerenciais e de gestão de informação.

*Build* (implementação)

Implementação, desenvolvimento e/ou manutenção de sistemas que reflitam os processos operacionais definidos e melhor se adequem à gestão da informação corporativa.

*Run* (operação)

Suporte e manutenção a sistemas já em estágio de operação.

### **3.2.1 - População e Amostra**

A escolha da Empresa X pesquisada foi por conveniência, porque possui várias características relevantes para o estudo em questão. Entre elas pode-se destacar o ambiente de desenvolvimento de *software* com participantes do processo fisicamente distribuídos nos clientes e a utilização de tecnologia de ponta, tanto de *hardware* como *software*.

A seleção da Empresa X pesquisada ocorreu de acordo com a base instalada do produto de *software* Y, o Tempo de relacionamento, os serviços que possuem e os seguintes conjuntos de características:

- Arquitetura Cliente-Servidor
- Linguagem de Programação
- Tempo de Vida do Produto no Mercado (5 Anos)
- Nº de Funcionalidades

A aplicação do questionário nas empresas-cliente ocorreu através de entrevistas individuais com o gerente de informática; um integrante da equipe de desenvolvimento e um integrante da equipe de usuários do Software.

O questionário foi construído seguindo a escala de avaliação de LIKERT. Este tipo de construção de perguntas foi desenvolvido por Rensis Likert e publicado em artigo em 1932 (“*A Technique for the Measurement of Attitudes*”, *Archives of Psychology* 140: pp. 1-55).

Esta forma de construção de perguntas busca a avaliação do entrevistado em relação a uma determinada característica, objeto, opinião ou estímulo, utilizando a medida de pontos em uma escala. Esta forma de mensuração é conhecida também como itens de Likert, sendo a de 5 pontos a mais utilizada, exemplo: muito bom, bom, regular, ruim e muito ruim, por exemplo.

### 3.3 - ANÁLISE E DISCUSSÃO DOS RESULTADOS:

Este item visa avaliar a percepção de usuários quanto aos *softwares* produzidos e utilizados. O questionário foi enviado para 100 empresas-cliente, por email, de forma a avaliar diversos aspectos relativos à sua utilização.

O questionário foi dividido em duas partes; características funcionais e características gerais. Uma análise crítica dos dados coletados foi realizada a partir do uso da escala de Likert de 5 pontos para qualificar a apreciação dos produtos de *software*.

A primeira parte apresenta os resultados tabulados e a segunda, uma análise multivariada que busca identificar fatores latentes associados à avaliação dos informantes. Para realizar a segunda parte da análise, foi usada a análise de classes latentes, cuja metodologia se encontra no anexo 1. A análise de dados foi com base no retorno da resposta de 50 usuários e culminou em diversos quadros síntese, onde foram apresentadas as conclusões da análise crítica desenvolvida na forma de lições aprendidas a partir do caso apresentado.

### **3.3.1 - Análise Descritiva**

Os atributos funcionais foram avaliados segundo as seguintes características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Os resultados da apuração são apresentados em tabelas e gráficos a seguir.

#### **3.3.1.1 - Parte I – Atributos Funcionais:**

- Funcionalidade:

A funcionalidade foi avaliada segundo as seguintes questões:

- As funções solicitadas (ou necessárias) são realizadas de modo apropriado?

- O *software* realiza o que foi proposto de forma correta?
- O *software* interage facilmente com os sistemas especificados?
- O *software* está de acordo com as normas, convenções ou regulamentações previstas em leis?
- O *software* evita acessos não autorizados, acidentais ou deliberados a programas e dados?

Tabela 9: As funções solicitadas (ou necessárias) são realizadas de modo apropriado?

| Resposta         | N° de informantes |     | Acumulado % |
|------------------|-------------------|-----|-------------|
| 2 Raramente      | 5                 | 10  | 10          |
| 3 Às vezes       | 15                | 30  | 40          |
| 4 Frequentemente | 30                | 60  | 100         |
| Total            | 50                | 100 |             |

Fonte: Pesquisa ISO9126.

Nesse caso, a maioria dos respondentes (60%) considera que as funções solicitadas são realizadas de modo apropriado.

Tabela 10: O software realiza o que foi proposto de forma correta?

| Resposta         | N° de informantes |     | Acumulado % |
|------------------|-------------------|-----|-------------|
| 2 Raramente      | 5                 | 10  | 10          |
| 3 Às vezes       | 5                 | 10  | 20          |
| 4 Frequentemente | 40                | 80  | 100         |
| Total            | 50                | 100 |             |

Fonte: Pesquisa ISO9126.

80% dos entrevistados consideram que o *software* realiza o que foi proposto de forma correta; 10% entendem que às vezes, apenas e os restantes 10% que o faz raramente.

Tabela 11: O *software* interage facilmente com os sistemas especificados?

| Resposta         | N° de informantes |     | Acumulado % |
|------------------|-------------------|-----|-------------|
| 2 Raramente      | 5                 | 10  | 10          |
| 3 Às vezes       | 15                | 30  | 40          |
| 4 Frequentemente | 30                | 60  | 100         |
| Total            | 50                | 100 |             |

Fonte: Pesquisa ISO 9126.

60% dos entrevistados responderam que o *software* interage facilmente com os sistemas especificados; 30% responderam que o *software* às vezes interage facilmente e 10% responderam que o *software* raramente interage facilmente com os sistemas especificados.

Tabela 12: O software está de acordo com as normas, convenções ou regulamentações previstas em leis?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 4 Frequentemente | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

Todos os entrevistados atestam que o *software* está de acordo com as normas, convenções ou regulamentações previstas em leis.

Tabela 13: O software evita acessos não autorizados, acidentais ou deliberados a programas e dados?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 4 Frequentemente | 50                | 100 | 100         |

Fonte: Pesquisa ISO9126.

Todos (100%) dos respondentes garantem que o software evita acessos não-autorizados, acidentais ou deliberados a programas e dados.

Tabela 14: Indicadores estatísticos

| Estatísticas de posição  | Mediana        | Moda           | Percentil      |                |
|--|----------------|----------------|----------------|----------------|
|  |                |                | 25             | 75             |
| As funções solicitadas (ou necessárias) são realizadas de modo apropriado? | Frequentemente | Frequentemente | Às vezes       | Frequentemente |
| O <i>software</i> realiza o que foi proposto de forma correta?             | Frequentemente | Frequentemente | Frequentemente | Frequentemente |
| O <i>software</i> interage facilmente com os sistemas especificados?       | Frequentemente | Frequentemente | Às vezes       | Frequentemente |

|  |                    |                    |                |                |
|--|--------------------|--------------------|----------------|----------------|
| O <i>software</i> está de acordo com as normas, convenções ou regulamentações previstas em leis? | Freqüentement<br>e | Freqüenteme<br>nte | Freqüentemente | Freqüentemente |
| O <i>software</i> evita acessos não autorizados, acidentais ou deliberados a programas e dados?  | Freqüentement<br>e | Freqüenteme<br>nte | Freqüentemente | Freqüentemente |

Fonte: Pesquisa ISO9126.

O exposto permite concluir que, quanto à funcionalidade, a avaliação dos informantes revelou na maior parte das vezes os programas correspondem às expectativas “freqüentemente”. No caso de adequação a normas e leis e ao impedimento de acessos não autorizados houve unanimidade nas respostas. As opções Nunca e Sempre não foram assinaladas por nenhum informante.

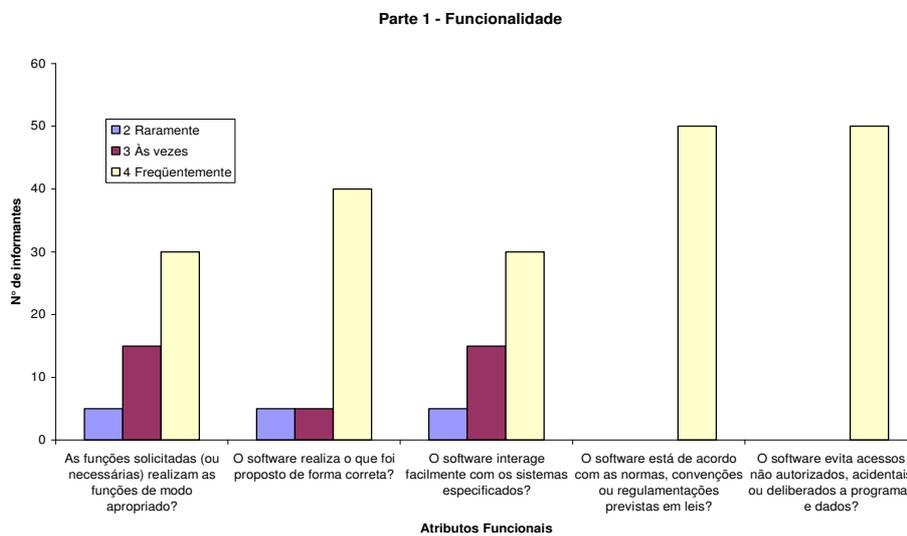


Gráfico 1

Fonte: Pesquisa ISO 9126.

- **Confiabilidade:**

A confiabilidade foi avaliada segundo as seguintes questões:

- Com que freqüência o *software* apresenta falhas?
- Como o *software* reage à ocorrência de falhas?

- O *software* é capaz de restabelecer a operação em caso de falha?

Tabela 15: Com que frequência o *software* apresenta falhas?

| Resposta | N° de informantes | %   | Acumulado % |
|----------|-------------------|-----|-------------|
| 1 Nunca  | 50                | 100 | 100         |

Fonte: Pesquisa ISO9126.

Todos os respondentes afirmam que o *software* nunca apresenta falhas.

Tabela 16: Como o *software* reage à ocorrência de falhas?

| Resposta | N° de informantes | %   | Acumulado % |
|----------|-------------------|-----|-------------|
| 1 Nunca  | 50                | 100 | 100         |

Fonte: Pesquisa ISO9126.

De acordo com os respondentes, o *software* nunca apresentou falhas.

Tabela 17: O *software* é capaz de restabelecer a operação em caso de falha?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 2 Raramente      | 9                 | 18  | 18          |
| 3 Às vezes       | 31                | 62  | 80          |
| 4 Frequentemente | 10                | 20  | 100         |
| Total            | 50                | 100 |             |

Fonte: Pesquisa ISO9126.

62% dos respondentes afirmaram que, às vezes, o *software* é capaz de restabelecer a operação em caso de falha; 20%, frequentemente, o *software* restabelece a operação em caso de falha e 18% que, raramente, o *software* restabelece a operação em caso de falha.

Tabela 18: Indicadores estatísticos

| Estatísticas de posição  | Mediana  | Moda     | Percentil |          |
|--|----------|----------|-----------|----------|
|  |          |          | 25        | 75       |
| Com que frequência o <i>software</i> apresenta falhas?                 | Nunca    | Nunca    | Nunca     | Nunca    |
| Como o <i>software</i> reage à ocorrência de falhas?                   | Nunca    | Nunca    | Nunca     | Nunca    |
| O <i>software</i> é capaz de restabelecer a operação em caso de falha? | Às vezes | Às vezes | Às vezes  | Às vezes |

Fonte: Pesquisa ISO9126.

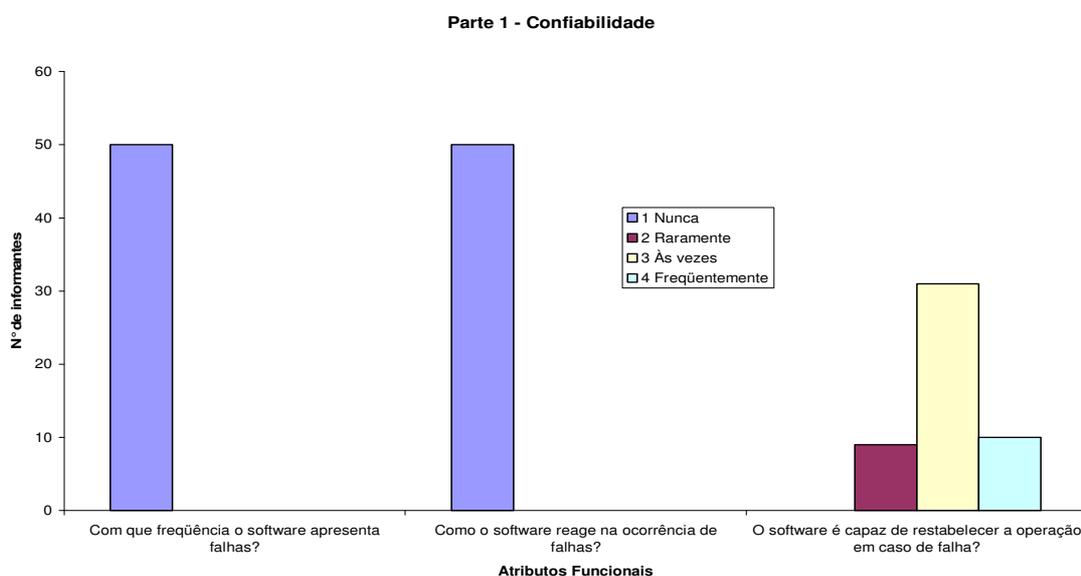


Gráfico 2

Fonte: Pesquisa ISO9126.

Na questão da confiabilidade, mais uma vez, os informantes foram unânimes em duas questões. Nos casos de ocorrência de falhas e reação a elas a opção “nunca” foi assinalada por todos.

Quanto à capacidade de restabelecer as operações após falhas, a maioria dos informantes declarou que, “às vezes”, os programas apresentam essa capacidade.

- Usabilidade:

A usabilidade foi avaliada segundo as seguintes questões:

- É fácil entender o conceito e a aplicação?
- É fácil aprender a usar o *software*?
- É fácil operar e controlar o *software*?

Tabela 19: É fácil entender o conceito e a aplicação?

| Resposta    | Nº de informantes | %   | Acumulado % |
|-------------|-------------------|-----|-------------|
| 2 Raramente | 5                 | 10  | 10          |
| 3 Às vezes  | 45                | 90  | 100         |
| Total       | 50                | 100 |             |

Fonte: Pesquisa ISO9126.

90% dos respondentes afirmaram que, às vezes, é fácil entender o conceito e a aplicação; 10%, ao contrário, disseram que raramente é fácil entender o conceito e a aplicação.

Tabela 20: É fácil aprender a usar o *software*?

| Resposta    | N° de informantes | %   | Acumulado % |
|-------------|-------------------|-----|-------------|
| 2 Raramente | 5                 | 10  | 10          |
| 3 Às vezes  | 45                | 90  | 100         |
| Total       | 50                | 100 |             |

Fonte: Pesquisa ISO9126.

90% dos respondentes consideraram que, às vezes, é fácil aprender a usar o *software*. Os restantes 10% disseram que, raramente, é fácil aprender a usar o *software*.

Tabela 21: É fácil operar e controlar o *software*?

| Resposta    | N° de informantes | %   | Acumulado % |
|-------------|-------------------|-----|-------------|
| 2 Raramente | 10                | 20  | 20          |
| 3 Às vezes  | 40                | 80  | 100         |
| Total       | 50                | 100 |             |

Fonte: Pesquisa ISO9126.

80% dos informantes consideraram fácil operar e controlar o *software*. Os restantes 20% disseram que, raramente, é fácil operar e controlar o *software*.

Tabela 22: Indicadores estatísticos

| Estatísticas de posição                           | Mediana  | Moda     | Percentil |          |
|---|----------|----------|-----------|----------|
|   |          |          | 25        | 75       |
| É fácil entender o conceito e a aplicação ?       | Às vezes | Às vezes | Às vezes  | Às vezes |
| É fácil aprender a usar o <i>software</i> ?       | Às vezes | Às vezes | Às vezes  | Às vezes |
| É fácil de operar e controlar o <i>software</i> ? | Às vezes | Às vezes | Às vezes  | Às vezes |

Fonte: Pesquisa ISO9126.

A usabilidade é considerada, “às vezes”, fácil. Para as três perguntas os informantes declararam, em sua maioria, que, “às vezes”, é fácil entender o conceito da aplicação, aprender a usar e operar e controlar o *software*.

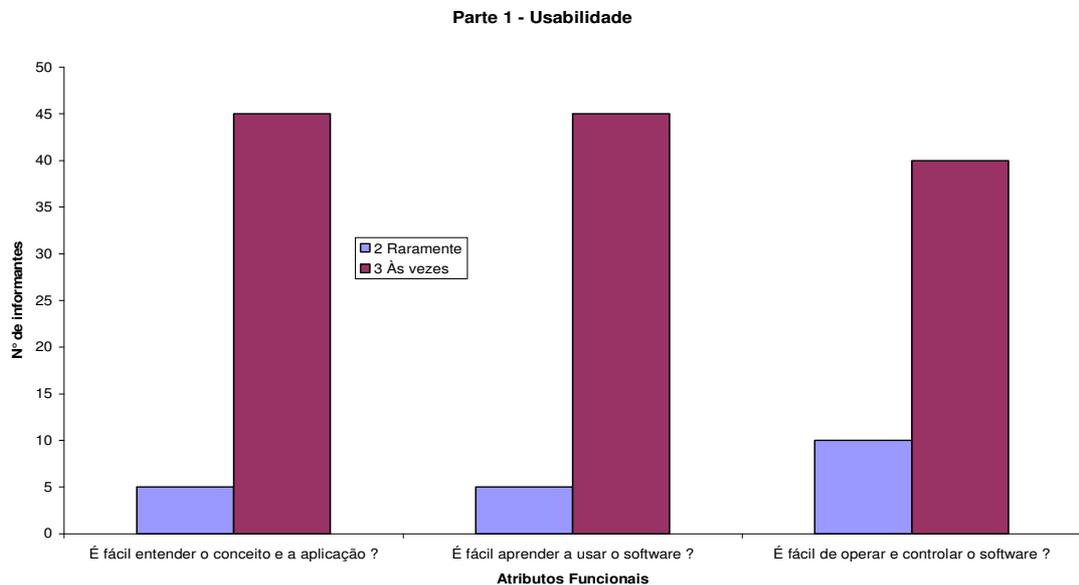


Gráfico 3

Fonte: Pesquisa ISO9126.

- **Eficiência:**

A eficiência foi avaliada segundo as seguintes questões:

- Os tempos utilizados são compatíveis com o nível de desempenho requerido pelo produto?
- Os recursos utilizados são compatíveis com o nível de desempenho requerido pelo produto?

As duas perguntas foram respondidas unanimemente pelos informantes, com a opção “às vezes”.

Tabela 23: Os tempos utilizados são compatíveis com o nível de desempenho requerido pelo produto?

| Resposta   | N° de informantes | %   | Acumulado % |
|------------|-------------------|-----|-------------|
| 3 Às vezes | 50                | 100 | 100         |

Fonte: Pesquisa ISO9126.

100% dos informantes consideraram que os tempos utilizados são, às vezes, compatíveis com o nível de desempenho requerido pelo produto.

Tabela 24: Os recursos utilizados são compatíveis com o nível de desempenho requerido pelo produto?

| Resposta   | N° de informantes | %   | Acumulado % |
|------------|-------------------|-----|-------------|
| 3 Às vezes | 50                | 100 | 100         |

Fonte: Pesquisa ISO9126.

100% dos informantes afirmaram que, às vezes, os recursos utilizados são compatíveis com o nível de desempenho requerido pelo produto.

Tabela 25: Indicadores estatísticos

| Estatísticas de posição  | Mediana  | Moda     | Percentil |          |
|--|----------|----------|-----------|----------|
|  |          |          | 25        | 75       |
| Os tempos utilizados são compatíveis com o nível de desempenho requerido pelo produto?   | Às vezes | Às vezes | Às vezes  | Às vezes |
| Os recursos utilizados são compatíveis com o nível de desempenho requerido pelo produto? | Às vezes | Às vezes | Às vezes  | Às vezes |

Fonte: Pesquisa ISO9126.

75% dos informantes consideraram que os tempos e os recursos utilizados são compatíveis, às vezes, com o nível de desempenho requerido pelo produto.

- **Manutenibilidade:**

A manutenibilidade foi avaliada segundo as seguintes questões:

- É fácil encontrar uma falha quando a mesma ocorre? É fácil determinar uma alteração?
- É fácil modificar, adaptar e remover defeitos? É fácil executar uma alteração?
- Existem grandes riscos de ocorrência de *bugs* quando se faz uma alteração?
- É fácil testar quando se fazem alterações?

Quanto à manutenção *do software* os informantes declararam “às vezes” ser fácil a identificação de falhas (80%), “raramente” ser fácil remover o defeito (80%) e “nunca” existirem riscos de *bugs*, quando é feita uma alteração (90%). Todos informaram que “raramente” é fácil testar alterações.

Tabela 26: É fácil encontrar uma falha quando a mesma ocorre? É fácil determinar uma alteração?

| <b>Resposta</b> | <b>N° de informantes</b> | <b>%</b>   | <b>Acumulado %</b> |
|-----------------|--------------------------|------------|--------------------|
| 2 Raramente     | 10                       | 20         | 20                 |
| 3 Às vezes      | 40                       | 80         | 100                |
| <b>Total</b>    | <b>50</b>                | <b>100</b> |                    |

Fonte: Pesquisa ISO9126.

80% dos informantes afirmaram que, às vezes, é fácil encontrar uma falha quando a mesma ocorre; 20% disseram que, raramente, é fácil encontrar uma falha quando a mesma ocorre.

Tabela 27: É fácil modificar, adaptar e remover defeitos? É fácil executar uma alteração?

| <b>Resposta</b> | <b>N° de informantes</b> | <b>%</b>   | <b>Acumulado %</b> |
|-----------------|--------------------------|------------|--------------------|
| 2 Raramente     | 40                       | 80         | 80                 |
| 3 Às vezes      | 10                       | 20         | 100                |
| <b>Total</b>    | <b>50</b>                | <b>100</b> |                    |

Fonte: Pesquisa ISO 9126.

80% dos informantes disseram que, raramente, é fácil modificar, adaptar e remover defeitos e executar uma alteração; 20% afirmaram que, às vezes, é fácil modificar, adaptar e remover defeitos e executar uma alteração.

Tabela 28: Existem grandes riscos de ocorrência de bugs, quando se faz uma alteração?

| <b>Resposta</b> | <b>N° de informantes</b> | <b>%</b>   | <b>Acumulado %</b> |
|-----------------|--------------------------|------------|--------------------|
| 1 Nunca         | 45                       | 90         | 90                 |
| 2 Raramente     | 5                        | 10         | 100                |
| <b>Total</b>    | <b>50</b>                | <b>100</b> |                    |

Fonte: Pesquisa ISO 9126.

90% dos informantes consideraram que nunca existem grandes riscos de ocorrência de *bugs*, quando se faz uma alteração; 10% raramente existem grandes riscos de ocorrência de *bugs*, quando se faz uma alteração.

Tabela 29: É fácil testar quando se fazem alterações?

| <b>Resposta</b> | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|-----------------|--------------------------|----------|--------------------|
| 2 Raramente     | 50                       | 100      | 100                |

Fonte: Pesquisa ISO 9126.

Todos os informantes (100%) consideraram que, raramente, é fácil testar quando se fazem alterações.

Tabela 30: Indicadores estatísticos

| Estatísticas de posição  | Mediana   | Moda      | Percentil |           |
|--|-----------|-----------|-----------|-----------|
|  |           |           | 25        | 75        |
| É fácil encontrar uma falha quando a mesma ocorre? É fácil determinar uma alteração? | Às vezes  | Às vezes  | Às vezes  | Às vezes  |
| É fácil modificar, adaptar e remover defeitos? É fácil executar uma alteração?       | Raramente | Raramente | Raramente | Raramente |
| Existem grandes riscos de ocorrência de bugs quando se faz uma alteração?            | Nunca     | Nunca     | Nunca     | Nunca     |
| É fácil testar quando se fazem alterações?   | Raramente | Raramente | Raramente | Raramente |

Fonte: Pesquisa ISO 9126.

100% dos informantes entenderam que, às vezes, é fácil encontrar uma falha quando a mesma ocorre e determinar uma alteração; 25% que, raramente, é fácil; 100% entenderam que nunca é fácil e 100%, também afirmaram que, raramente é fácil testar quando se fazem alterações.

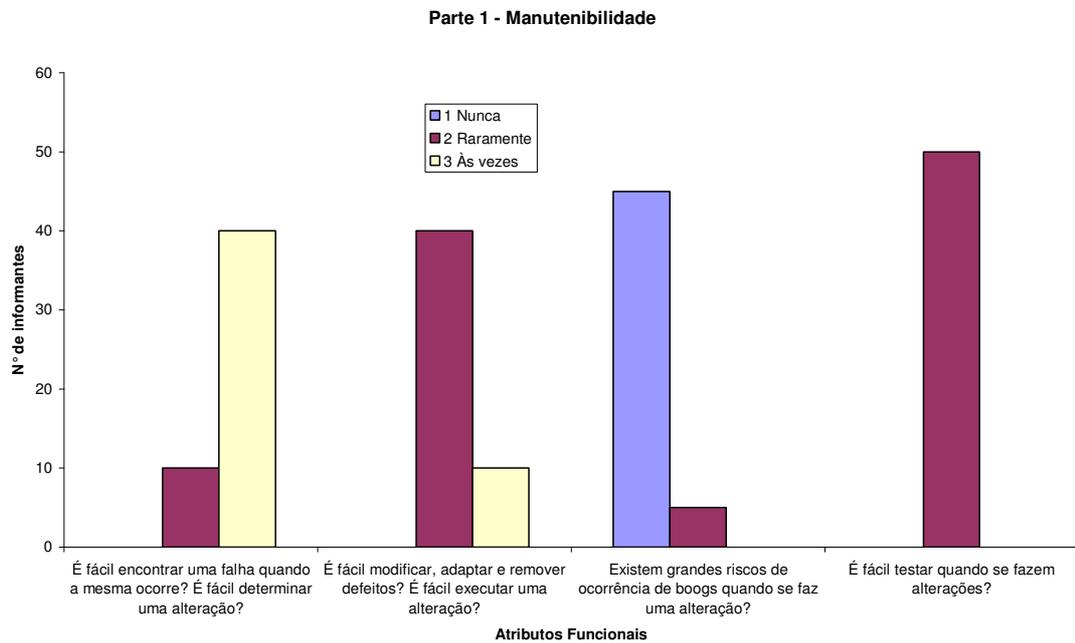


Gráfico 4

Fonte: Pesquisa ISO9126.

- Portabilidade:

A portabilidade foi avaliada segundo as seguintes questões:

- É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no *software* considerado?
- É fácil instalar em outros ambientes?
- É fácil usá-lo para substituir outro *software*?
- Está de acordo com padrões ou convenções de portabilidade?

Todos os informantes foram unânimes quanto a essas questões. É freqüentemente” fácil para o *software* se adaptar a outros ambientes, “às vezes” instalar em outros ambientes, “às vezes”, usá-lo como substituto de outro programa, e, “às vezes”, encontra-se nos padrões de portabilidade.

Tabela 31 - É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no *software* considerado?

| Resposta         | Nº de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 4 Frequentemente | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes consideraram que é fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no *software* considerado?

Tabela 32: É fácil instalar em outros ambientes?

| <b>Resposta</b> | N° de informantes | %   | Acumulado % |
|-----------------|-------------------|-----|-------------|
| 3 Às vezes      | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes afirmaram que é fácil instalar em outros ambientes.

Tabela 33: É fácil usá-lo para substituir outro software?

| <b>Resposta</b> | N° de informantes | %   | Acumulado % |
|-----------------|-------------------|-----|-------------|
| 3 Às vezes      | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes afirmaram que é fácil usá-lo para substituir outro *software*.

Tabela 34: Está de acordo com padrões ou convenções de portabilidade?

| <b>Resposta</b> | N° de informantes | %   | Acumulado % |
|-----------------|-------------------|-----|-------------|
| 3 Às vezes      | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes declararam-se de acordo com padrões ou convenções de portabilidade.

Tabela 35: Indicadores estatísticos

| <b>Estatísticas de posição</b>  | <b>Mediana</b> | <b>Moda</b>    | <b>Percentil</b> |                |
|---|----------------|----------------|------------------|----------------|
|   |                |                | 25               | 75             |
| É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no <i>software</i> considerado? | Freqüentemente | Freqüentemente | Freqüentemente   | Freqüentemente |
| É fácil instalar em outros ambientes?   | Às vezes       | Às vezes       | Às vezes         | Às vezes       |
| É fácil usá-lo para substituir outro <i>software</i> ?  | Às vezes       | Às vezes       | Às vezes         | Às vezes       |
| Está de acordo com padrões ou   | Às vezes       | Às vezes       | Às vezes         | Às vezes       |

convenções de portabilidade?

Fonte: Pesquisa ISO 9126.

### 3.3.1.2 - Parte 2 – Questões Gerais

Ao todo 22 perguntas foram feitas para caracterizar o processo de teste de software, segundo a mesma lógica dos atributos funcionais e a escala de Likert. Destas 10, foram respondidas de forma unânime, com a escolha de uma opção.

A tabulação e gráficos são apresentados a seguir.

Tabela 36: Existem equipamentos (*hardware* e *software*) avançados para o suporte ao processo de teste?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 4 Frequentemente | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes consideraram que existem equipamentos (*hardware* e *software*) avançados para o suporte ao processo de teste.

Tabela 37: A empresa possui uma área dedicada à execução e suporte aos processos de testes?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 4 Frequentemente | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes afirmaram que a empresa possui uma área dedicada à execução e suporte aos processos de testes.

Tabela 38 Há empregados organizados na execução das atividades de testes?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 4 Frequentemente | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos empregados organizados na execução das atividades de testes.

Tabela 39 Há elementos materiais relacionados com o serviço (planos, procedimentos e relatórios) visualmente atraentes?

| Resposta   | N° de informantes | %   | Acumulado % |
|------------|-------------------|-----|-------------|
| 3 Às vezes | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes responderam que há elementos materiais relacionados com o serviço (planos, procedimentos e relatórios) visualmente atraentes.

Tabela 40 Os projetos de testes agendados são executados na data acordada?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 2 Raramente      | 5                 | 10  | 10          |
| 3 Às vezes       | 25                | 50  | 60          |
| 4 Freqüentemente | 20                | 40  | 100         |
| Total            | 50                | 100 |             |

Fonte: Pesquisa ISO 9126.

50% dos informantes disseram que, às vezes, os projetos de testes agendados são executados na data acordada; 40% afirmaram que, freqüentemente, isso acontece e 10% declararam que, raramente, ocorre.

Tabela 41: Na identificação de erros, os analistas de testes demonstram interesse na resolução dos mesmos?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 2 Raramente      | 5                 | 10  | 10          |
| 3 Às vezes       | 15                | 30  | 40          |
| 4 Freqüentemente | 30                | 60  | 100         |
| Total            | 50                | 100 |             |

Fonte: Pesquisa ISO 9126.

60% dos respondentes acreditam que, freqüentemente, na identificação de erros, os analistas de testes demonstram interesse na resolução dos mesmos. No entanto, 30% demonstram interesse às vezes e 10% raramente o demonstram.

Tabela 42: Os testes são realizados corretamente na primeira vez?

| Resposta    | N° de informantes | %   | Acumulado % |
|-------------|-------------------|-----|-------------|
| 2 Raramente | 5                 | 10  | 10          |
| 3 Às vezes  | 45                | 90  | 100         |
| Total       | 50                | 100 |             |

Fonte: Pesquisa ISO 9126.

Tabela 43: O projeto de testes é concluído segundo o tempo prometido?

| Resposta    | N° de informantes | %   | Acumulado % |
|-------------|-------------------|-----|-------------|
| 2 Raramente | 10                | 20  | 20          |
| 3 Às vezes  | 40                | 80  | 100         |
| Total       | 50                | 100 |             |

Fonte: Pesquisa ISO 9126.

80% dos informantes afirmaram que o projeto de testes é concluído segundo o tempo prometido; 20% que raramente o projeto de testes é concluído segundo o tempo prometido.

Tabela 44: A equipe de testes mantém um histórico de trabalhos sem erros?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 4 Frequentemente | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes declararam que a equipe de testes mantém um histórico de trabalhos sem erros.

Tabela 45: A equipe de testes comunica aos clientes quando irá concluir os testes programados?

| Resposta         | N° de informantes | %   | Acumulado % |
|------------------|-------------------|-----|-------------|
| 3 Às vezes       | 5                 | 10  | 10          |
| 4 Frequentemente | 45                | 90  | 100         |
| Total            | 50                | 100 |             |

Fonte: Pesquisa ISO 9126.

90% dos informantes declararam que, frequentemente, a equipe de testes comunica aos clientes quando concluirá os testes; 10% que, raramente, a equipe faz tal comunicado.

Tabela 46: Os analistas de testes prestam um serviço rápido a seus clientes?

| Resposta    | N° de informantes | %   | Acumulado % |
|-------------|-------------------|-----|-------------|
| 2 Raramente | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos informantes responderam que, raramente, os analistas de testes prestam um serviço rápido a seus clientes.

Tabela 47: Os analistas de testes estão sempre dispostos a ajudar os clientes?

| Resposta   | N° de informantes | %   | Acumulado % |
|------------|-------------------|-----|-------------|
| 3 Às vezes | 50                | 100 | 100         |

Fonte: Pesquisa ISO 9126.

100% dos respondentes informaram que às vezes os analistas de testes estão dispostos a ajudar os clientes.

Tabela 48: Há analistas de testes sempre disponíveis para ajudar os clientes?

| <b>Resposta</b> | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|-----------------|--------------------------|----------|--------------------|
| 1 Nunca         | 50                       | 100      | 100                |

Fonte: Pesquisa ISO 9126.

100% dos informantes afirmaram que nunca há analistas disponíveis para ajudar os clientes.

Tabela 49: Há analistas de testes que transmitem, por meio de seu comportamento, confiança a seus clientes?

| <b>Resposta</b>  | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|------------------|--------------------------|----------|--------------------|
| 3 Às vezes       | 10                       | 20       | 20                 |
| 4 Frequentemente | 40                       | 80       | 100                |
| Total            | 50                       | 100      |                    |

Fonte: Pesquisa ISO 9126.

80% dos informantes consideraram que existem analistas de testes que transmitem, por meio de seu comportamento, confiança a seus clientes.

Tabela 50: Os analistas de testes fazem com que o cliente se sinta seguro com os processos de testes adotados pela empresa?

| <b>Resposta</b>  | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|------------------|--------------------------|----------|--------------------|
| 4 Frequentemente | 50                       | 100      | 100                |

Fonte: Pesquisa ISO 9126.

100% dos informantes garantiram que, frequentemente, os analistas de testes fazem com que o cliente se sinta seguro com os processos de testes adotados pela empresa.

Tabela 51: Há analistas de testes que se preocupam com a segurança do ambiente e dados utilizados para testes?

| <b>Resposta</b>  | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|------------------|--------------------------|----------|--------------------|
| 1 Nunca          | 3                        | 6        | 6                  |
| 2 Raramente      | 9                        | 18       | 24                 |
| 3 Às vezes       | 12                       | 24       | 48                 |
| 4 Freqüentemente | 26                       | 52       | 100                |
| Total            | 50                       | 100      |                    |

Fonte: Pesquisa ISO 9126.

52% dos analistas de testes se preocupam freqüentemente com a segurança do ambiente e dados utilizados para testes; 24% se preocupam às vezes; 18% se preocupam raramente e 6% nunca o fazem.

Tabela 52: Há analistas de testes com conhecimentos suficientes para responder às perguntas dos clientes?

| <b>Resposta</b>  | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|------------------|--------------------------|----------|--------------------|
| 1 Nunca          | 13                       | 26       | 26                 |
| 2 Raramente      | 15                       | 30       | 56                 |
| 3 Às vezes       | 16                       | 32       | 88                 |
| 4 Freqüentemente | 6                        | 12       | 100                |
| Total            | 50                       | 100      |                    |

Fonte: Pesquisa ISO 9126.

32% dos analistas de testes às vezes possuem conhecimentos suficientes para responder às perguntas dos clientes; 26% nunca o possuem; 30% raramente podem responder às perguntas dos clientes e, apenas 12% podem, freqüentemente, responder aos clientes.

Tabela 53: Há analistas de testes com bom relacionamento com as outras áreas do projeto e da organização?

| <b>Resposta</b>  | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|------------------|--------------------------|----------|--------------------|
| 1 Nunca          | 7                        | 14       | 14                 |
| 2 Raramente      | 7                        | 14       | 28                 |
| 3 Às vezes       | 7                        | 14       | 42                 |
| 4 Freqüentemente | 29                       | 58       | 100                |
| Total            | 50                       | 100      |                    |

Fonte: Pesquisa ISO 9126.

58% dos analistas de testes freqüentemente possuem bom relacionamento com as outras áreas do projeto e da organização; 14% raramente se relacionam com as

outras áreas do projeto e da organização; outros 14% nunca se relacionam e, finalmente, os 14% restantes, às vezes o fazem.

Tabela 54: Há horários de execução dos processos de testes convenientes para todos os clientes?

| <b>Resposta</b> | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|-----------------|--------------------------|----------|--------------------|
| 0               | 50                       | 100      | 100                |

Fonte: Pesquisa ISO 9126.

Nenhum dos informantes respondeu a esta questão.

Tabela 55: A maioria dos participantes do grupo de testes participa das atividades de teste em tempo integral?

| <b>Resposta</b>  | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|------------------|--------------------------|----------|--------------------|
| 1 Nunca          | 1                        | 2        | 2                  |
| 2 Raramente      | 9                        | 18       | 20                 |
| 3 Às vezes       | 34                       | 68       | 88                 |
| 4 Freqüentemente | 6                        | 12       | 100                |
| Total            | 50                       | 100      |                    |

Fonte: Pesquisa ISO 9126.

68% às vezes participam das atividades de teste em tempo integral; 18% raramente o faz; 12% freqüentemente participam e 2% nunca participam dessas atividades em tempo integral.

Tabela 56: Os participantes do grupo de testes preocupam-se com os interesses de seus clientes?

| <b>Resposta</b>  | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|------------------|--------------------------|----------|--------------------|
| 1 Nunca          | 7                        | 14       | 14                 |
| 2 Raramente      | 15                       | 30       | 44                 |
| 3 Às vezes       | 21                       | 42       | 86                 |
| 4 Freqüentemente | 7                        | 14       | 100                |
| Total            | 50                       | 100      |                    |

Fonte: Pesquisa ISO9126.

42% dos participantes do grupo de testes preocupam-se às vezes com os interesses de seus clientes; 30% raramente o fazem; 14 % nunca se preocupam e os outros 14% freqüentemente o fazem.

Tabela 57: Há empregados que compreendem as necessidades específicas de seus clientes?

| <b>Resposta</b>  | <b>N° de informantes</b> | <b>%</b> | <b>Acumulado %</b> |
|------------------|--------------------------|----------|--------------------|
| 2 Raramente      | 16                       | 32       | 32                 |
| 3 Às vezes       | 17                       | 34       | 66                 |
| 4 Freqüentemente | 17                       | 34       | 100                |
| Total            | 50                       | 100      |                    |

Fonte: Pesquisa ISO 9126.

34% dos empregados compreendem freqüentemente as necessidades específicas de seus clientes; outros 34% as compreendem às vezes; 32% raramente o fazem.

Tabela 58: Indicadores estatísticos

| <b>Estatísticas de posição</b>  | <b>Mediana</b> | <b>Moda</b>    | <b>Percentil</b> |                |
|---|----------------|----------------|------------------|----------------|
|   |                |                | 25               | 75             |
| Existem equipamentos (hardware e software) avançados para o suporte ao processo de teste?                     | Freqüentemente | Freqüentemente | Freqüentemente   | Freqüentemente |
| A empresa possui uma área dedicada à execução e suporte aos processos de testes?                              | Freqüentemente | Freqüentemente | Freqüentemente   | Freqüentemente |
| Há empregados organizados na execução das atividades de testes?   | Freqüentemente | Freqüentemente | Freqüentemente   | Freqüentemente |
| Há elementos materiais relacionados com o serviço (planos, procedimentos e relatórios) visualmente atraentes? | Às vezes       | Às vezes       | Às vezes         | Às vezes       |
| Os projetos de testes agendados são executados na data acordada?  | Às vezes       | Às vezes       | Às vezes         | Freqüentemente |
| Na identificação de erros, os analistas de testes demonstram interesse na resolução dos mesmos?               | Freqüentemente | Freqüentemente | Às vezes         | Freqüentemente |

|  |                |                |                    |                |
|--|----------------|----------------|--------------------|----------------|
| Os testes são realizados corretamente na primeira vez?   | Às vezes       | Às vezes       | Às vezes           | Às vezes       |
| O projeto de testes é concluído segundo o tempo prometido?   | Às vezes       | Às vezes       | Às vezes           | Às vezes       |
| A equipe de testes mantém um histórico de trabalhos sem erros?   | Freqüentemente | Freqüentemente | Freqüentemente     | Freqüentemente |
| A equipe de testes comunica aos clientes quando irá concluir os testes programados?                          | Freqüentemente | Freqüentemente | Freqüentemente     | Freqüentemente |
| Os analistas de testes prestam um serviço rápido a seus clientes?  | Raramente      | Raramente      | Raramente          | Raramente      |
| Os analistas de testes estão sempre dispostos a ajudar os clientes?  | Às vezes       | Às vezes       | Às vezes           | Às vezes       |
| Há analistas de testes que transmitem, por meio de seu comportamento, confiança a seus clientes?             | Freqüentemente | Freqüentemente | Freqüentemente     | Freqüentemente |
| Os analistas de testes fazem que o cliente se sinta seguro com os processos de testes adotados pela empresa? | Freqüentemente | Freqüentemente | Freqüentemente     | Freqüentemente |
| Há analistas de testes que se preocupam com a segurança do ambiente e dados utilizados para testes?          | Freqüentemente | Freqüentemente | Raramente/Às vezes | Freqüentemente |
| Há analistas de testes com conhecimentos suficientes para responder às perguntas dos clientes?               | Raramente      | Às vezes       | Nunca              | Às vezes       |

|   | Freqüenteme<br>nte | Freqüentement<br>e          | Raramente | Freqüenteme<br>nte |
|---|--------------------|-----------------------------|-----------|--------------------|
| Há analistas de testes com bom relacionamento entre as outras áreas do projeto e da organização.?   |                    |                             |           |                    |
| Há horários de execução dos processos de testes convenientes para todos os clientes?                | 0                  | 0                           | 0         | 0                  |
| A maioria dos participantes do grupo de testes participa das atividades de teste em tempo integral? | Às vezes           | Às vezes                    | Às vezes  | Às vezes           |
| Os participantes do grupo de testes preocupam-se com os interesses de seus clientes?                | Às vezes           | Às vezes                    | Raramente | Às vezes           |
| Há empregados que compreendem as necessidades específicas de seus clientes?                         | Às vezes           | Às vezes/Freqüent<br>emente | Raramente | Freqüenteme<br>nte |

Fonte: Pesquisa ISO 9126.

Os projetos de testes agendados são executados na data acordada?

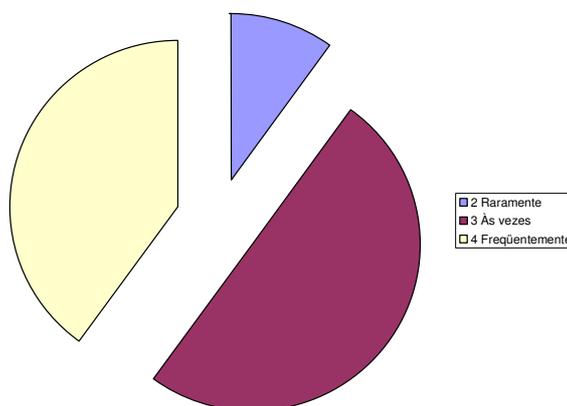


Gráfico 5

Fonte: Pesquisa ISO 9126.

Na identificação de erros, os analistas de testes demonstram interesse na resolução dos mesmos?

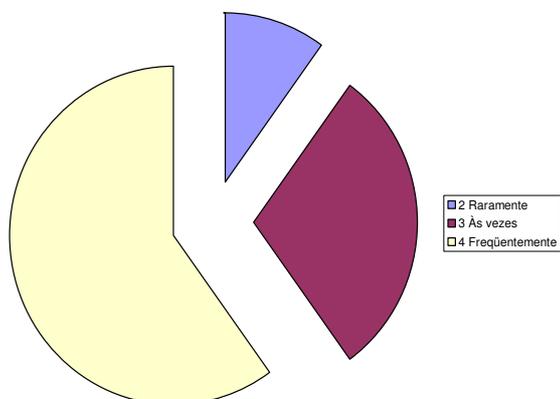


Gráfico 6  
Fonte: Pesquisa ISO 9126.

Os testes são realizados corretamente na primeira vez?

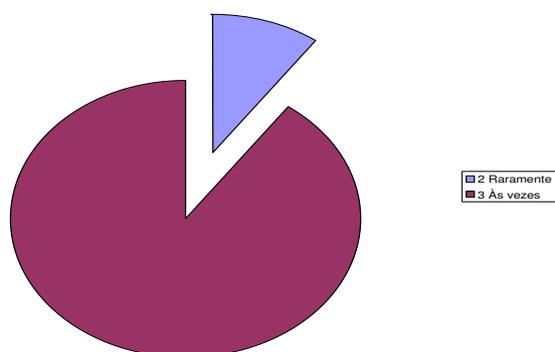


Gráfico 7  
Fonte: Pesquisa ISO 9126.

O projeto de testes é concluído segundo o tempo prometido?

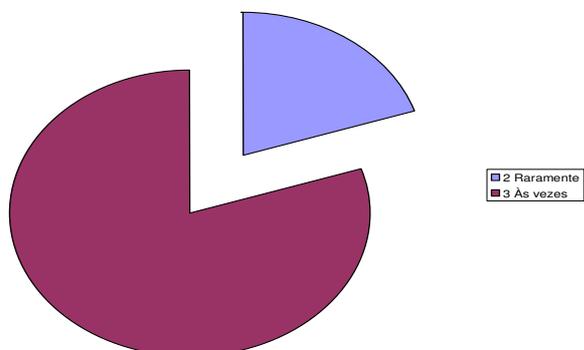


Gráfico 8  
Fonte: Pesquisa ISO 9126

A equipe de testes comunica aos clientes quando irá concluir os testes programados?

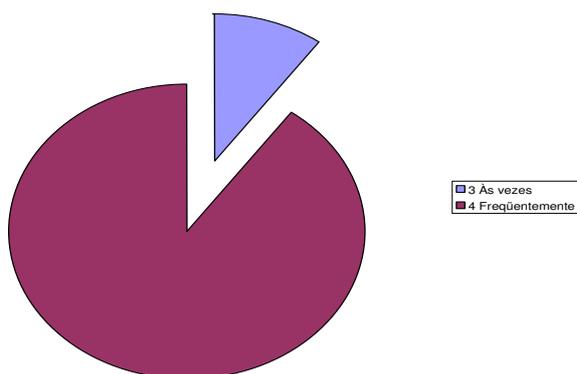


Gráfico 9  
Fonte: Pesquisa ISO 9126.

Há analistas de testes que transmitem, por meio de seu comportamento, confiança a seus clientes?

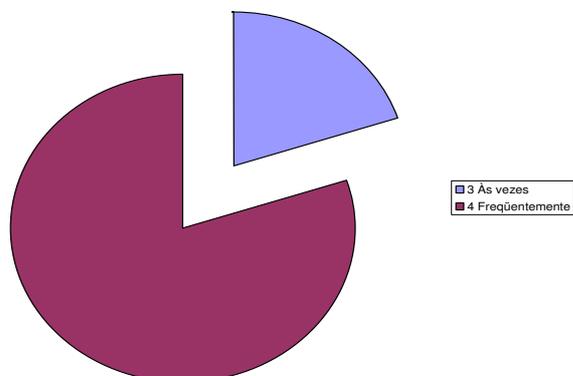


Gráfico 10  
Fonte: Pesquisa ISO 9126.

Há analistas de testes que se preocupam com a segurança do ambiente e dados utilizados para testes?

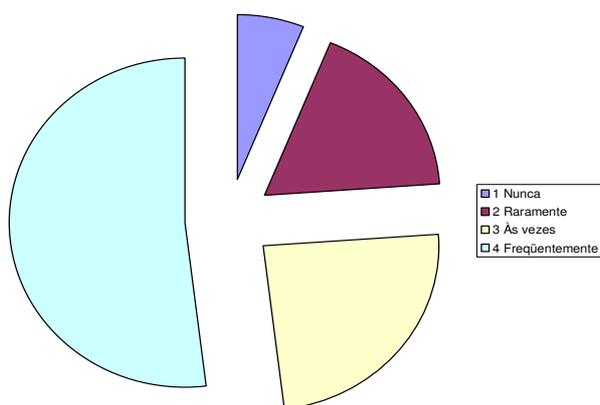


Gráfico 11  
Fonte: Pesquisa ISO9126.

Há analistas de testes com conhecimentos suficientes para responder às perguntas dos clientes?

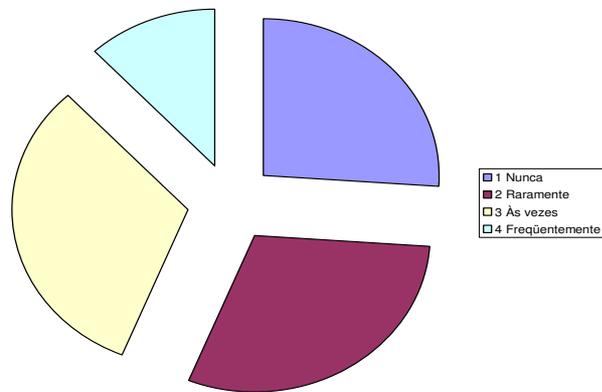


Gráfico 12  
Fonte: Pesquisa ISO9126.

Há analistas de testes com bom relacionamento entre as outras áreas do projeto e da organização?

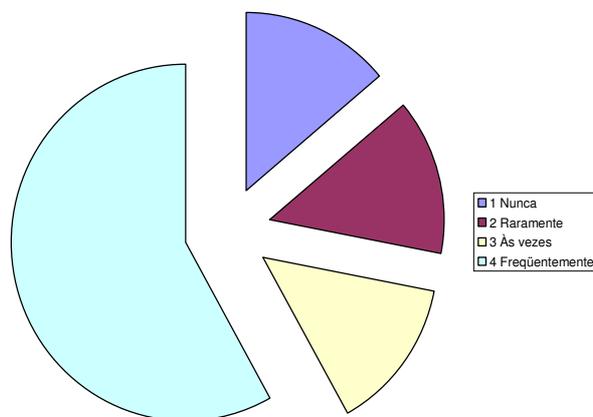


Gráfico 13  
Fonte: Pesquisa ISO9126.

A maioria dos participantes do grupo de testes participa das atividades de teste em tempo integral?

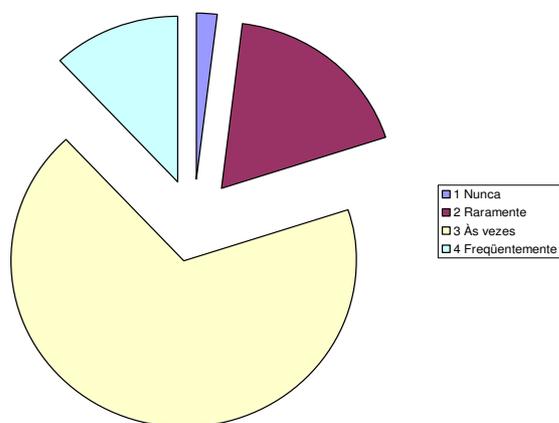


Gráfico 14  
Fonte: Pesquisa ISO9126

Os participantes do grupo de testes preocupam-se com os interesses de seus clientes?

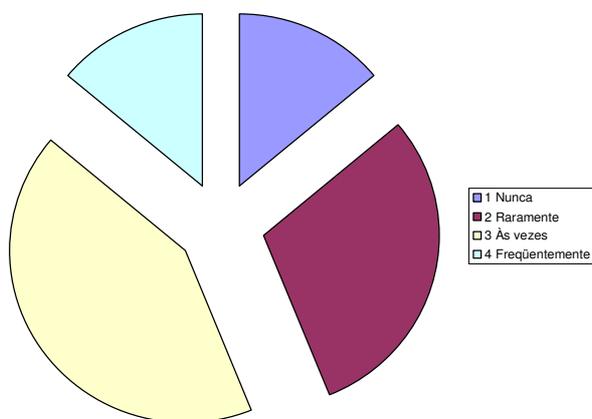


Gráfico 15  
Fonte: Pesquisa ISO 9126.

Há empregados que compreendem as necessidades específicas de seus clientes?

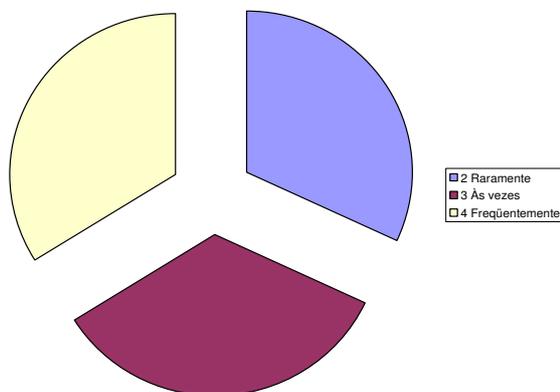


Gráfico 16  
Fonte: Pesquisa ISO9126.

Após análise dos resultados obtidos em cada questão, concluiu-se que todas as características de qualidade segundo a norma ISO/IEC 9126 são impactadas positivamente sob a perspectiva do usuário com a adoção dos processos de testes baseados no modelo CMMI.

Analisando as subcaracterísticas observou-se que os escores são impactadas positivamente sob a perspectiva do usuário.

### 3.3.1.3 - Parte 3 – Análise de Classes Latentes

A análise de classes latentes é uma técnica multivariada que abrange análise de cluster, análise fatorial e modelos de regressão.

Nesse estudo, o objetivo é avaliar a opinião de gerentes em relação a atributos funcionais no desenvolvimento e uso de *softwares*. Essa avaliação é feita segundo uma gama de características. Essas características, por sua vez, estão relacionadas a dimensões não-mensuráveis diretamente ou observáveis, sendo denominadas de variáveis latentes. Observam-se essas diversas “manifestações” por meio de quesitos,

ligados à funcionalidade, confiabilidade, usabilidade etc., denominadas de variáveis observáveis. A análise de classes latentes será utilizada, nesse caso, para explicar a associação existente em uma tabela de contingência multidimensional através dos níveis (classes) das variáveis latentes, tal qual uma análise fatorial com variáveis categóricas. Conseqüentemente, os níveis das variáveis latentes poderão ser interpretados. Além disso, a técnica permite a obtenção da probabilidade de cada informante ser classificado em cada um dos níveis das variáveis latentes, possibilitando a formação de grupos, tal qual uma análise de cluster.

A análise de classes latentes será realizada apenas para os atributos funcionais. A partir dos resultados é feita uma análise em conjunto com os dados de atributos gerais. Para a análise foram apenas consideradas as questões em que não houve unanimidade. A unanimidade impede que haja variabilidade a ser analisada.

#### Modelo de Classes Latentes

A partir dos resultados apresentados no anexo 1 foi escolhido o modelo de fator latente com 3 classes. Esse modelo mostrou-se adequado do ponto de vista de ajuste, e a partir dele:

- Interpreta-se cada classe do fator, identificando as características destas classes; e
- Faz-se uma classificação dos informantes, alocando-os na classe em que for maior a sua probabilidade de pertença.

A tabela a seguir apresenta uma análise das classes do fator encontrado.

Tabela 59: Probabilidades de cada classe e probabilidades condicionais

| Item | Pergunta  | Opções    | Fator latente                                  |  |  |
|------|---|-----------|--|--|--|
|      |   |           | Classe 1                                       | Classe 2                                       | Classe 3                                       |
|      |   |           | $P(Q_i = \text{opção}/\text{Fator Latente}=1)$ | $P(Q_i = \text{opção}/\text{Fator Latente}=2)$ | $P(Q_3 = \text{opção}/\text{Fator Latente}=3)$ |
|      | Probabilidades por classe                           |           | 0.6  | 0.2  | 0.2  |
|      | Probabilidades condicionais                         |           |  |  |  |
| 1    | 1. As funções solicitadas (ou necessárias) realizam | Raramente | 0  | 0  | 0.5  |
| 1    |   | Às vezes  | 0  | 1  | 0.5  |

|    |  |                |        |   |     |
|----|--|----------------|--------|---|-----|
| 1  | as funções de modo apropriado?   | Freqüentemente | 1      | 0 | 0   |
| 2  | 2. O software realiza o que foi proposto de forma correta?                               | Raramente      | 0      | 0 | 0.5 |
| 2  |  | Às vezes       | 0      | 0 | 0.5 |
| 2  |  | Freqüentemente | 1      | 1 | 0   |
| 3  | 3. O software interage facilmente com os sistemas especificados?                         | Raramente      | 0      | 0 | 0.5 |
| 3  |  | Às vezes       | 0      | 1 | 0.5 |
| 3  |  | Freqüentemente | 1      | 0 | 0   |
| 8  | 8. O software é capaz de restabelecer a operação em caso de falha?                       | Raramente      | 0      | 0 | 0.9 |
| 8  |  | Às vezes       | 0.6667 | 1 | 0.1 |
| 8  |  | Freqüentemente | 0.3333 | 0 | 0   |
| 9  | 9. É fácil entender o conceito e a aplicação?  | Raramente      | 0      | 0 | 0.5 |
| 9  |  | Às vezes       | 1      | 1 | 0.5 |
| 10 | 10. É fácil aprender a usar o software?  | Raramente      | 0      | 0 | 0.5 |
| 10 |  | Às vezes       | 1      | 1 | 0.5 |
| 11 | 11. É fácil de operar e controlar o software?  | Raramente      | 0      | 0 | 1   |
| 11 |  | Às vezes       | 1      | 1 | 0   |
| 14 | 14. É fácil encontrar uma falha quando a mesma ocorre? É fácil determinar uma alteração? | Raramente      | 0      | 0 | 1   |
| 14 |  | Às vezes       | 1      | 1 | 0   |
| 15 | 15. É fácil modificar, adaptar e remover defeitos? É fácil executar uma alteração?       | Raramente      | 0.6667 | 1 | 1   |
| 15 |  | Às vezes       | 0.3333 | 0 | 0   |
| 16 | 16. Existem grandes riscos de ocorrência de boogs quando se faz uma alteração?           | Nunca          | 0.8333 | 1 | 1   |
| 16 |  | Raramente      | 0.1667 | 0 | 0   |

Fonte: Elaboração própria.

A partir da tabela acima foram identificadas às características de cada classe do fator latente em linhas gerais:

- Classe 1: avaliação positiva de funcionalidade, neutra (às vezes) de confiabilidade e usabilidade e negativa (menos que as demais classes) para manutenibilidade.
- Classe 2: avaliação neutra de funcionalidade, confiabilidade e usabilidade e negativa (intermediária) para manutenibilidade.
- Classe 3: avaliação negativa de funcionalidade, neutra (às vezes) de confiabilidade e usabilidade e negativa (mais que as demais classes) para manutenibilidade.

• **Tabulação dos Atributos Gerais pelas Classes Latentes dos Informantes**

A seguir são tabulados os dados de atributos gerais segundo classes do fator latente (apenas os atributos gerais em que as respostas não foram unânimes).

Tabela 60: Classes do fator latente (Atributos Gerais)

| Resposta         | Classe do fator latente |               |               | Total          |
|------------------|-------------------------|---------------|---------------|----------------|
|                  | 1                       | 2             | 3             |                |
| 2 Raramente      | 0                       | 0             | 5             | 5              |
|                  | 0.00%                   | 0.00%         | 100.00%       | 100.00%        |
| 3 Às vezes       | 10                      | 10            | 5             | 25             |
|                  | 40.00%                  | 40.00%        | 20.00%        | 100.00%        |
| 4 Frequentemente | 20                      | 0             | 0             | 20             |
|                  | 100.00%                 | 0.00%         | 0.00%         | 100.00%        |
| <b>Total</b>     | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO 9126.

Tabela 53: Na identificação de erros, os analistas de testes demonstram interesse na resolução dos mesmos?

Tabela 61: Classes do fator latente (ERROS)

| Resposta         | Classe do fator latente |           |           | Total     |
|------------------|-------------------------|-----------|-----------|-----------|
|                  | 1                       | 2         | 3         |           |
| 2 Raramente      | 0                       | 0         | 5         | 5         |
|                  | 0.00%                   | 0.00%     | 100.00%   | 100.00%   |
| 3 Às vezes       | 0                       | 10        | 5         | 15        |
|                  | 0.00%                   | 66.70%    | 33.30%    | 100.00%   |
| 4 Frequentemente | 30                      | 0         | 0         | 30        |
|                  | 100.00%                 | 0.00%     | 0.00%     | 100.00%   |
| <b>Total</b>     | <b>30</b>               | <b>10</b> | <b>10</b> | <b>50</b> |

|  |               |               |               |                |
|--|---------------|---------------|---------------|----------------|
|  | <b>60.00%</b> | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |
|--|---------------|---------------|---------------|----------------|

Fonte: Elaboração própria. Pesquisa ISO 9126.

Tabela 62: Os testes são realizados corretamente na primeira vez?

| Resposta     | Classe do fator latente |               |               | Total          |
|--------------|-------------------------|---------------|---------------|----------------|
|              | 1                       | 2             | 3             |                |
| 2 Raramente  | 0                       | 0             | 5             | 5              |
|              | 0.00%                   | 0.00%         | 100.00%       | 100.00%        |
| 3 Às vezes   | 30                      | 10            | 5             | 45             |
|              | 66.70%                  | 22.20%        | 11.10%        | 100.00%        |
| <b>Total</b> | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|              | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO 9126.

Tabela 63: O projeto de testes é concluído segundo o tempo prometido?

| Resposta     | Classe do fator latente |               |               | Total          |
|--------------|-------------------------|---------------|---------------|----------------|
|              | 1                       | 2             | 3             |                |
| 2 Raramente  | 0                       | 0             | 10            | 10             |
|              | 0.00%                   | 0.00%         | 100.00%       | 100.00%        |
| 3 Às vezes   | 30                      | 10            | 0             | 40             |
|              | 75.00%                  | 25.00%        | 0.00%         | 100.00%        |
| <b>Total</b> | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|              | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO 9126.

Tabela 64: A equipe de testes comunica aos clientes quando irá concluir os testes programados?

| Resposta         | Classe do fator latente |               |               | Total          |
|------------------|-------------------------|---------------|---------------|----------------|
|                  | 1                       | 2             | 3             |                |
| 3 Às vezes       | 0                       | 0             | 5             | 5              |
|                  | 0.00%                   | 0.00%         | 100.00%       | 100.00%        |
| 4 Frequentemente | 30                      | 10            | 5             | 45             |
|                  | 66.70%                  | 22.20%        | 11.10%        | 100.00%        |
| <b>Total</b>     | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO 9126.

Quanto à realização de testes, tabelas 52 a 56, os informantes da classe 1 do fator latente avaliam de forma mais positiva a execução dos testes. Nas duas outras classes a avaliação é menos favorável, sendo negativa para os informantes classificados na classe 3.

Tabela 65: Há analistas de testes que transmitem, por meio de seu comportamento, confiança a seus clientes?

| Resposta         | Classe do fator latente |               |               | Total          |
|------------------|-------------------------|---------------|---------------|----------------|
|                  | 1                       | 2             | 3             |                |
| 3 Às vezes       | 0                       | 0             | 10            | 10             |
|                  | 0.00%                   | 0.00%         | 100.00%       | 100.00%        |
| 4 Freqüentemente | 30                      | 10            | 0             | 40             |
|                  | 75.00%                  | 25.00%        | 0.00%         | 100.00%        |
| <b>Total</b>     | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO 9126.

O comportamento dos analistas também é avaliado positivamente entre os informantes da classe 1, em relação aos demais informantes.

Tabela 66: Há analistas de testes que se preocupam com a segurança do ambiente e dados utilizados para testes?

| Resposta         | Classe do fator latente |               |               | Total          |
|------------------|-------------------------|---------------|---------------|----------------|
|                  | 1                       | 2             | 3             |                |
| 1 Nunca          | 3                       | 0             | 0             | 3              |
|                  | 100.00%                 | 0.00%         | 0.00%         | 100.00%        |
| 2 Raramente      | 8                       | 1             | 0             | 9              |
|                  | 88.90%                  | 11.10%        | 0.00%         | 100.00%        |
| 3 Às vezes       | 9                       | 2             | 1             | 12             |
|                  | 75.00%                  | 16.70%        | 8.30%         | 100.00%        |
| 4 Freqüentemente | 10                      | 7             | 9             | 26             |
|                  | 38.50%                  | 26.90%        | 34.60%        | 100.00%        |
| <b>Total</b>     | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO 9126.

No quesito segurança do ambiente de dados, a classe 2 e 3 de informantes revelaram-se mais satisfeita com a atuação dos analistas que a classe 1. Quanto ao conhecimento dos analistas, na hora de responder aos usuários, todas as classes fizeram avaliações mais negativas que positivas.

Tabela 67: Há analistas de testes com conhecimentos suficientes para responder às perguntas dos clientes?

| Resposta         | Classe do fator latente |               |               | Total          |
|------------------|-------------------------|---------------|---------------|----------------|
|                  | 1                       | 2             | 3             |                |
| 1 Nunca          | 8                       | 3             | 2             | 13             |
|                  | 61.50%                  | 23.10%        | 15.40%        | 100.00%        |
| 2 Raramente      | 10                      | 2             | 3             | 15             |
|                  | 66.70%                  | 13.30%        | 20.00%        | 100.00%        |
| 3 Às vezes       | 9                       | 3             | 4             | 16             |
|                  | 56.30%                  | 18.80%        | 25.00%        | 100.00%        |
| 4 Frequentemente | 3                       | 2             | 1             | 6              |
|                  | 50.00%                  | 33.30%        | 16.70%        | 100.00%        |
| <b>Total</b>     | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO9126.

O relacionamento com outras atividades foi avaliado positivamente independentemente da classe do informante.

Tabela 68: Há analistas de testes com bom relacionamento entre as outras áreas do projeto e da organização?

| Resposta         | Classe do fator latente |               |               | Total          |
|------------------|-------------------------|---------------|---------------|----------------|
|                  | 1                       | 2             | 3             |                |
| 1 Nunca          | 5                       | 1             | 1             | 7              |
|                  | 71.40%                  | 14.30%        | 14.30%        | 100.00%        |
| 2 Raramente      | 5                       | 1             | 1             | 7              |
|                  | 71.40%                  | 14.30%        | 14.30%        | 100.00%        |
| 3 Às vezes       | 5                       | 1             | 1             | 7              |
|                  | 71.40%                  | 14.30%        | 14.30%        | 100.00%        |
| 4 Frequentemente | 15                      | 7             | 7             | 29             |
|                  | 51.70%                  | 24.10%        | 24.10%        | 100.00%        |
| <b>Total</b>     | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO9126.

Tabela 69: A maioria dos participantes do grupo de testes participa das atividades de teste em tempo integral?

| Resposta    | Classe do fator latente |       |         | Total   |
|-------------|-------------------------|-------|---------|---------|
|             | 1                       | 2     | 3       |         |
| 1 Nunca     | 0                       | 0     | 1       | 1       |
|             | 0.00%                   | 0.00% | 100.00% | 100.00% |
| 2 Raramente | 0                       | 0     | 9       | 9       |
|             | 0.00%                   | 0.00% | 100.00% | 100.00% |

|                  |               |               |               |                |
|------------------|---------------|---------------|---------------|----------------|
| 3 Às vezes       | 25            | 9             | 0             | 34             |
|                  | 73.50%        | 26.50%        | 0.00%         | 100.00%        |
| 4 Freqüentemente | 5             | 1             | 0             | 6              |
|                  | 83.30%        | 16.70%        | 0.00%         | 100.00%        |
| <b>Total</b>     | <b>30</b>     | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b> | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO 9126.

Tabela 70: Os participantes do grupo de testes preocupam-se com os interesses de seus clientes?

| Resposta         | Classe do fator latente |               |               | Total          |
|------------------|-------------------------|---------------|---------------|----------------|
|                  | 1                       | 2             | 3             |                |
| 1 Nunca          | 3                       | 2             | 2             | 7              |
|                  | 42.90%                  | 28.60%        | 28.60%        | 100.00%        |
| 2 Raramente      | 9                       | 4             | 2             | 15             |
|                  | 60.00%                  | 26.70%        | 13.30%        | 100.00%        |
| 3 Às vezes       | 14                      | 3             | 4             | 21             |
|                  | 66.70%                  | 14.30%        | 19.00%        | 100.00%        |
| 4 Freqüentemente | 4                       | 1             | 2             | 7              |
|                  | 57.10%                  | 14.30%        | 28.60%        | 100.00%        |
| <b>Total</b>     | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO9126.

Tabela 71: Há empregados que compreendem as necessidades específicas de seus clientes?

| Resposta         | Classe do fator latente |               |               | Total          |
|------------------|-------------------------|---------------|---------------|----------------|
|                  | 1                       | 2             | 3             |                |
| 2 Raramente      | 10                      | 3             | 3             | 16             |
|                  | 62.50%                  | 18.80%        | 18.80%        | 100.00%        |
| 3 Às vezes       | 10                      | 3             | 4             | 17             |
|                  | 58.80%                  | 17.60%        | 23.50%        | 100.00%        |
| 4 Freqüentemente | 10                      | 4             | 3             | 17             |
|                  | 58.80%                  | 23.50%        | 17.60%        | 100.00%        |
| <b>Total</b>     | <b>30</b>               | <b>10</b>     | <b>10</b>     | <b>50</b>      |
|                  | <b>60.00%</b>           | <b>20.00%</b> | <b>20.00%</b> | <b>100.00%</b> |

Fonte: Elaboração própria. Pesquisa ISO9126.

A relação com clientes é avaliada na maioria dos casos como algo que ocorre “às vezes”, não havendo diferenças significativas entre as classes latentes.

## CAPÍTULO IV

### 4. CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES:

A linha de pesquisa em qualidade de *software* ainda engatinha no Brasil e no mundo, embora esteja claro para os profissionais do ramo que propostas de ações e relatos de experiências sobre este tema interessam fortemente aos centros de pesquisa e os meios acadêmicos. Há o consenso de que o caminho para a excelência do *software* brasileiro passa pela qualidade. É por isso que avaliar os indicadores de qualidade e produtividade bem como difundir as novas tecnologias é um dos pilares para a consolidação da qualidade do *software* no Brasil.

Neste sentido, o objetivo geral deste estudo, ou seja, a avaliação da percepção dos impactos na qualidade dos produtos de *software* com a adoção das práticas baseadas nas áreas de processo de Verificação e Validação do modelo CMMI foi medir, de forma qualitativa a qualidade do produto de *software* como definido na ISO: 9.126. Assim, a importância de algumas características varia de acordo com o tipo de *software*, no entanto, todos eles devem atender aos requisitos do grupo de manutenção. Sendo assim, o controle sistemático sobre as modificações realizadas será fundamental para o sucesso da construção e posterior manutenção de um *software*.

Modelos de gestão de processo propõem atividades de garantia da qualidade para a obtenção de melhores produtos nas organizações de desenvolvimento de *software*.

Embora a atual norma ISO 9126 enumere as características e subcaracterísticas de um *software*, ela ainda não define como dar uma nota a um *software* em cada um destes itens. Se não houver familiarização com o processo de avaliação de *software*, pode-se ter dificuldades em tentar utilizar a norma. Para avaliar um *software* segundo esta norma, deve-se tentar atribuir valores (como se fossem notas ou conceitos) a cada uma das subcaracterísticas.

Algumas características podem ser realmente medidas, como o tempo de execução de um programa, número de linhas de código, número de erros encontrados em uma sessão de teste ou o tempo médio entre falhas. Nestes casos, é possível utilizar uma técnica, uma ferramenta ou um *software* para realizar medições. Em outros casos, a característica é tão subjetiva que não existe nenhuma forma óbvia de medi-la.

Nada ajuda mais a avaliar características de um produto do que um avaliador experiente, que já realizou esta tarefa diversas vezes e em diversas empresas diferentes. Afinal, medir é comparar com padrões e um avaliador experiente terá maior sensibilidade do que um profissional que acaba de ler uma norma pela primeira vez.

A atividade de teste, por si só, não é capaz de tornar o produto livre defeitos. Contudo, quando formalmente feita pode reduzir o número de não conformidades e ainda fornecer dados importantes para uma estimativa da confiabilidade do produto. Sua implementação exige, além da capacitação técnica da equipe de desenvolvimento para torná-la capaz de gerar casos de testes consistentes e de alta probabilidade de evidenciar erros, uma estrutura organizacional que permita incorporar a atividade de testes às atividades do ciclo de desenvolvimento e a participação de pessoas isentas na execução dos testes.

A avaliação da percepção dos impactos na qualidade dos produtos realizada neste estudo aponta que os usuários se preocupam com os procedimentos de testes e de segurança, na busca pelo “zero erro”, embora resultados mais flexíveis tenham sido detectados.

Ao mesmo tempo, a metodologia estudada mostra a necessidade de padrões mínimos de qualidade, sustentando seu resultado pelo uso de procedimentos e medidas bem definidos, garantindo equidade no tratamento dos produtos e fornecendo informações objetivas.

Espera-se que essa metodologia aplicada nas avaliações de produtos para o confirme que é capaz de mostrar a percepção dos impactos na qualidade dos produtos de *software* em relação à especificação dos requisitos.

Finalmente, acredita-se que a metodologia desenvolvida na adaptação para a avaliação dos impactos na qualidade dos produtos de *software* nas indústrias, segundo a norma ISO/IEC: 9.126, com a adoção das práticas de testes do modelo CMMI permite

sugerir ou não a adoção desse procedimento em outras empresas ou novas formas de gestão.

## REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 13596** – tecnologia de informação: avaliação de produto de *software* - características de qualidade e diretrizes para o seu uso. Rio de Janeiro, 1996.

BAHIA, A. S. **Avaliação da complexidade de *software* científico**. Tese. COPPE/UFRJ, Rio de Janeiro, 1992.

BARRETO, A. S. **Apoio à verificação de *software* em ambientes de desenvolvimento de *software* orientados à organização**. Dissertação. COPPE/UFRJ. Rio de Janeiro, 2006.

BEIZER, B. **Software testing techniques**. 2. ed. Scottsdale: Coriolis Group, 1990.

BELCHIOR, A.D. **Qualidade de *software* financeiro**. Tese. COPPE/UFRJ, Rio de Janeiro, 1992.

BOEHM, B.; BROWN, J.R.; KASPAR, H.; LIPOW, M.; McLEOD, G.; MERRIT, M., **Characteristics of software quality**. North Holland, 1978.

BLASCHECK, J.R. **Planejamento estratégico de sistemas de informação**. Tese. COPPE/UFRJ, Rio de Janeiro, 1995.

BRASIL. Ministério da Ciência e Tecnologia. Certificação ISO dos Sistemas da Qualidade. In: \_\_\_\_\_. **Qualidade e produtividade no setor de *software***. Disponível em: <<http://www.mct.gov.br/Temas/info/Dsi/qualidad/certiso.htm>>. Acesso em: 25 out. 2006.

BRASIL. Ministério da Ciência e Tecnologia. Indicadores e metas da qualidade e produtividade em software: valores observados para 1995/1999 e metas estabelecidas para 2001. In: \_\_\_\_\_. Programa brasileiro da qualidade e produtividade em *software*. **PBQP *software***: melhoria da qualidade e produtividade nos serviços. Disponível em: <<http://www.mct.gov.br/Temas/info/Dsi/PBQP/Indic.htm>>. Acesso em: 20/10/06.

BRYMAN, A. **Research methods and organization studies**. London: Unwin Hyman, 1989.

CAMPOS, F. **Qualidade de aplicações hipermídia**. Dissertação COPPE/UFRJ. Rio de Janeiro, 1994.

CERVO, Amado; BERVIAN, Pedro A. **Metodologia científica**. São Paulo Prentice Hall, 2002.

CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. **CMMI**: guidelines for process integration and product improvement. Addison-Wesley, 2003.

COELHO, M. A. **Mps Br**: melhoria do processo de software brasileiro. Disponível em:< [www.abramti.org.br](http://www.abramti.org.br)>. Acesso: 01 jun.2008.

DEMO, Pedro. **Introdução à metodologia da ciência**. São Paulo: Atlas, 1987.

DIAS, R. Q. **Adoção das práticas de testes do modelo CMMI na melhoria da qualidade dos produtos e serviços de testes de software em instituições financeiras**. Dissertação. Universidade Federal Fluminense. Rio de Janeiro, 2005.

D'OLIVEIRA, F.M., **Planejamento de testes na homologação de software**: uma abordagem orientada ao cliente, Dissertação. Universidade Católica de Brasília, Brasília, 2003.

DROMEY, R.G. Cornering the chimera, *IEEE Software*. V.13, n.: 33-43, 1996.

FIORINI, S.T.; STAA, A.V. e BAPTISTA, R.M. Engenharia de software com CMM. Rio de Janeiro: Brasport,1998.

FUGETTA, Affonso. Software Processo: a roadmap. In The future of software engineering 2000. Proceeding of the 20<sup>nd</sup>. **International Conference on Software engineering**, p. 27-34. The association for computing machinery: Anthony Fiukelstein, 2000.

GALIN, D., AVRAHAMI, M. Do SQA programs work: CMM Works: a meta analysis. In: **IEEE International Conference on Software: Science, Technology & Engineering (SwSTE'05)**: 95-100, 2005.

GARCIA, L.F. Qualidade e auditoria de *software*. Disponível em: <<http://www.garcia.pro.br/06-02>> Acesso em :20/03/2006.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 1999

\_\_\_\_\_. **Métodos e técnicas de pesquisa social**. São Paulo: Atlas, 1999.

GOLDENBERG, Mirian. **A arte de pesquisar**. Rio de Janeiro: Record, 2000.

GOODE, W. J.; HATT, P. K. **Métodos em pesquisa social**. São Paulo: Editora Nacional, 1979, pp. 237-268.

HERBERT, Juliana Silva. **Teste cooperativo de software**. Doutorado em Ciência da Computação. Porto Alegre: Instituto de Informática. Universidade Federal do Rio Grande do Sul, 1999.

HETZEL, B. **The complete guide to software testing**. 2 ed. John Wiley & Sons, 1988.

HUMPHREY, W.S. **Characterizing the software process**: a maturity framework, Software Engineering Institute, CMU/SEI-87-TR-11, ADA182895, June 1987.

IEEE, The Institute of electrical and electronics engineers. IEEE STD 610.12-1990: **Standard Glossary of Software Engineering Terminology**. New York: IEEE Computer Society, 1990.

IEEE, The institute of electrical and electronics engineers. IEEE Std 829: Standard for **Software Test Documentation**. New York: IEEE Computer Society, September, 1998.

INTHURN, C. **Qualidade e teste de software**. São Paulo: Visual Books, 2001.

ISBSG. **International Software Benchmarking Standards Group**: The Benchmarking, Release 8, 2003.

International Standart Organization. **ISO/IEC 9126**: software engineering: Product Quality, 2001.

JUNG, H., KIM, S., CHUNG, C. **Measuring software product quality**: a survey of ISO/IEC 9126, IEEE Software, v. 21, n. 5: 88-92, 2004.

KITCHENHAM, B.A., PFLEEGER, S.L. principles of survey research. Part 6: data analysis. ACM SIGSoft Software engineering. IEEE **Transactions on software engineering**. V. 28, n.8, p. 721-734

KUJAVA, P. Bootstrap: Europe's assessment method. **IEEE Software**, vol 10, N. 3, 1993.

\_\_\_\_\_. **Software process assessment and improvement**: The Bootstrap Approach. Blackwell, 1994.

McCALL, J.L. *et al.* **Factors in software quality**. National Technical Information Service, 1977.

MADU, C.N., KUEI, C.H., JACOB, R.A. **An empirical assessment of the influence of quality dimensions on organizational performance**. *International journal of production research*, v. 34, n. 7: 1943-62, 1996.

MOLINARI, L. **Testes de software**: produzindo sistemas melhores e mais confiáveis. São Paulo: Érica, 2003.

MYERS, G. **The art of software testing**. New York: Wiley, 1979.

NIST, Instituto Nacional de Padrões e Tecnologia. **The Economic Impact of Inadequate Infrastructure for Software Testing**, 2002.

NOGUEIRA, M.O. **Qualidade no setor de software brasileiro**: uma avaliação das práticas das organizações. Tese. Universidade Federal do Rio de Janeiro. UFRJ, 2006.

ORLANDI, T.R.C. **Processo de implantação de gestão de qualidade de software em empresas nacionais**: estudo de caso do Tribunal de Contas da União. Dissertação

(Mestrado em Informática). Universidade Católica de Brasília - UCB. Brasília, dezembro de 2000.

OSTERWEIL, L. Software process are software too. In **Proceedings of the International Conference on Software Engineering**, 9: 2-13. Monterey, USA, 1987.

PAULK, M.C.; CURTIS, B.; CHRISISSIS, M.B. **Capability Maturity Model for Software**. Software Engineering Institute, CMU/SEI-91-TR-24, ADA240603, 1991.

PAULK, M. C.; Curtis, B.; Chrissis, M.B.; Weber, C.V. **Key practices of the capability maturity model**, Version 1.1. Pittsburgh, Software Engineering Institute, 1993.

PETERS, J.F., PEDRYCZ, W. **Engenharia de software: teoria e prática**. Rio de Janeiro: Campus, 2001.

PFLEEGER, S.L **Engenharia de software: teoria e prática**, 2 ed. São Paulo: Prentice Hall, 2004.

PRESSMAN, R. **Engenharia de software**. São Paulo: Makron Books, 1995.

ROCHA, A.R.C. **Um Modelo para avaliação da qualidade de especificações**. Tese. PUC/RJ, Rio de Janeiro, Brasil, 1983.

ROCHA, A.R.C. **Análise e projeto estruturado de sistemas**. São Paulo: Campus, 1987.

ROCHA, A.R.C., PALERMO, S. *Software quality assurance in HEP*, North Holland, *Computer physics communications* 57, 1989.

ROCHA, A.R.C., MALDONADO, J. C., WEBER, K.C. *et al.* **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001.

SEI. **Capability Maturity Model Integration for Systems Engineering. Software Engineering, and Integrated Product and Process Development (CMMI-SE/SW/IPPD)**, version 1.1. Software Engineering Institute, Carnegie Mellon, 2001a.

\_\_\_\_\_. **CMMI General Information**. Software Engineering Institute, Carnegie Mellon, 2004. Disponível em <<http://www.sei.cmu.edu/cmml/general/general.html>>. Acesso em 10/04/2006.

SILVA FILHO, R.C. **Uma abordagem para avaliação de propostas de melhoria em processo de software**. Dissertação de mestrado. COPPE/UFRJ. Rio de Janeiro, 2006.

SILVA, Edna Lúcia; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. Florianópolis: Universidade federal de Santa Catarina, 2001. [www.projetos.inf.ufsc.br/arquivos/Metodologia% 20da% 20 Pesquisa %20 a% 20edicao.pdf](http://www.projetos.inf.ufsc.br/arquivos/Metodologia%20da%20Pesquisa%20a%20edicao.pdf). Acesso em 05 de maio 2008. Às 17h 23min.

SOMMERVILLE, I. **Software Process Models**: ACM Computing Survey. V. 28, n.1, p. 160-180. The Association for Computing Machinery (ACM), New York, 1996.

STRONG, D.M., LEE, Y.W., WANG, R.Y. Data quality in context. **Communications of the ACM**, v. 40, n. 9, p. 103-110, 1997.

VERGARA, Sylvia Constant. **Projetos e relatórios de pesquisa em administração**. 3. ed. São Paulo: Atlas, 2000.

YIN, R. K. **Case study research**: design and methods. Thousand Oaks: SAGE Publications, 1994.

## **ANEXOS**

## ANEXO A Análise de Classes Latentes

A análise de classes latentes é uma técnica multivariada que abrange análise de cluster, análise fatorial e modelos de regressão.

Nesse estudo, o objetivo é avaliar a opinião de gerentes em relação a atributos funcionais no desenvolvimento e uso de softwares. Essa avaliação é feita segundo uma gama de características. Essas características por sua vez estão relacionadas a dimensões que não são mensuráveis diretamente ou observáveis, sendo denominadas de variáveis latentes. Observam-se estas diversas “manifestações” por meio de quesitos, ligados a funcionalidade, confiabilidade, usabilidade, etc., que são denominadas de variáveis observáveis. A análise de classes latentes será utilizada, nesse caso, para explicar a associação existente numa tabela de contingência multidimensional através dos níveis (classes) das variáveis latentes, tal qual uma análise fatorial com variáveis categóricas. Conseqüentemente, os níveis das variáveis latentes poderão ser interpretados. Além disso, a técnica permite a obtenção da probabilidade de cada informante ser classificado em cada um dos níveis das variáveis latentes, possibilitando a formação de grupos, tal qual uma análise de cluster.

### Metodologia

#### Estimação

A suposição básica do modelo de variáveis latentes é de que em cada classe latente as variáveis observáveis sejam independentes (localmente independentes).

Consideremos um modelo com quatro variáveis observáveis A, B, C e D e 1 variável latente X:

- A – variável observável com I níveis –  $i=1, \dots, I$ ;
- B - variável observável com J níveis –  $j=1, \dots, J$ ;
- C – variável observável com K níveis –  $k = 1, \dots, K$ ;
- D – variável observável com L níveis –  $l=1, \dots, L$ ;
- X – variável latente com T níveis –  $t =1, \dots, T$ .

Dessa forma, a probabilidade de observar um município com o perfil  $A=i, B=j, C=k, D=l$  pertencente à classe  $t$  da variável latente X é dada por:

(1)

Conseqüentemente, a probabilidade de observar um município com o perfil  $A=i$ ,  $B=j$ ,  $C=k$  é dada por:

(2)

A probabilidade de observar  $n_{ijkl}$  municípios com perfil  $\{i, j, k, l\}$  é dada por:

(3)

A função de verossimilhança  $L$  de contagens para todos os perfis de respostas possíveis é dada por:

(4)

e o correspondente log da verossimilhança:

(5)

Os estimadores de  $\theta$  podem ser obtidos maximizando-se a expressão (5). Entretanto, como apenas as contagens são observadas e as classes latentes não ( $X=t$ ), a maximização de (5) torna-se complexa.

Uma possível alternativa seria estimar a informação faltante (classe latente) e, então, maximizar o log da verossimilhança com dados completados. O algoritmo EM - Expectation-Maximization (Dempster, Laird & Rubin, 1977) consiste na repetição sucessiva dos seguintes passos:

- Passo E - calcula a esperança do log da verossimilhança, fornecendo as freqüências esperadas para a estrutura com dados completos;
  - Passo M - maximiza a verossimilhança completa, obtendo as estimativas dos parâmetros do modelo;
- até que ocorra a convergência das estimativas.

Medidas de Qualidade do Ajuste

Para se avaliar o grau de adequabilidade do modelo são utilizadas as seguintes medidas:

- i. LL – Log-Likelihood, é uma medida da variabilidade total dos dados observados.
- ii. L2 – Avalia o quanto as freqüências de uma cela estimadas através do modelo ( $\hat{f}_{ijkl}$ , onde  $f_{ijkl}$ ) encontram-se próximas das freqüências observadas  $f_{ijkl}$ :

Quando  $L2=0$ , tem-se o ajuste perfeito e  $L2=0$ .

iii. BICLL – Avalia o grau de ajuste através do L2, ajustado pelo número de parâmetros estimados no modelo:

$$\text{BICLL} = L2 + \ln(N)M$$

onde N é o total de observações e M corresponde à contagem de T-1 probabilidades das classes latentes distintas. Quanto menor o valor de BIC, melhor o modelo.

Usualmente a estatística L2 tem distribuição de Qui-quadrado com graus de liberdade igual ao número de celas de toda a tabela de contingência menos o número total de parâmetros estimados menos 1.

No presente estudo o tamanho da amostra é muito pequeno em relação ao número de possíveis combinações de resposta. Este tipo de problema implica em que a distribuição de L2 não seja Qui-quadrado. Para avaliar a qualidade do ajuste utilizam-se as outras estatísticas e por meio de Bootstrap verifica-se se o modelo encontrado corresponde a um modelo adequado.

Antes de fazer a análise pelo método de bootstrap escolheu-se um modelo a partir das demais estatísticas. A tabela a seguir apresenta os dados.

Tabela: Estatísticas de ajuste de modelos

| Modelo        | L2       | BIC      | % de variância explicada |   |
|---------------|----------|----------|--------------------------|---|
| Nulo 1 classe |          | 417.8835 | -                        | - |
| Lca2          | 292.7012 | -19869.9 | 29.96                    |   |
| Lca3          | 108.311  | -19995.6 | 74.08                    |   |
| Lca4          | 95.1403  | -19950.1 | 77.23                    |   |

**Fonte:** Elaboração própria.

A partir destas tentativas optou-se pelo modelo com um fator latente de três classes, pois é o que possui o menor BIC.

#### Bootstrap

A metodologia de bootstrap desenvolvida inicialmente por Efrom (1973) consiste da simulação de dados a partir de resultados da amostra com o objetivo de avaliar se uma determinada estatística não pode ser considerada discrepante em relação aos dados observados.

Neste estudo o objetivo é avaliar se o modelo escolhido corresponde a um modelo adequado para os dados. Para fazer esta avaliação foram realizadas 100 amostras de tamanho 50 com reposição da matriz de respostas. Para cada uma das 100 amostras foram ajustados modelos tais quais o modelo escolhido e verificado se a estatística L2 do modelo “real” não se diferenciava das demais de forma significativa (estava entre os percentis 5% e 95%). Se a estatística de ajuste do modelo escolhido estiver neste intervalo adota-se o modelo para análise, caso contrário tenta-se um novo modelo. Os resultados são apresentados abaixo.

Tabela: Estatísticas de bootstrap

| Estatística | “Real”  | Percentil 5% | Percentil 95% |
|-------------|---------|--------------|---------------|
| L2          | 108.311 | 98.0329      | 253.6849      |
| BIC         | 280.44  | 270.1619     | 425.8139      |

Fonte: Elaboração própria.

A partir destes resultados optou-se pelo modelo de fator latente com 3 classes.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)