

**ANÁLISE DOS IMPACTOS NA QUALIDADE DE SOFTWARE EM  
INSTITUIÇÕES FINANCEIRAS SEGUNDO A NORMA ISO/IEC 9126  
NA ADOÇÃO DAS PRÁTICAS DE TESTES DO MODELO CMMI**

**HEITOR LUIZ MURAT DE MEIRELLES QUINTELLA, D.Sc.**

Professor da Universidade Federal Fluminense, Centro Tecnológico, Escola de Engenharia.  
Rua Passo da Pátria, 156, Sala 329  
São Domingos, Niterói/RJ  
24210-240 - Brasil  
Telefone: (21) 2239-0405 Ramal: 5451 Fax: (21) 2717-6390  
E-mail: [hquintel@unisys.com.br](mailto:hquintel@unisys.com.br)

**RONI QUEIROZ DIAS, M.Sc.**

Universidade Federal Fluminense  
Coordenador de Qualidade de Software – MATERA Systems  
Rua do Ouvidor, 121 – 22º Andar  
Centro/RJ  
20040-030 - Brasil  
Telefone: (21) 3512-1000 Fax: (21) 2252-5712  
E-mail: [rdias@usa.com](mailto:rdias@usa.com)

**MANOELLA RODRIGUES TEIXEIRA**

Mestranda Universidade Federal Fluminense  
Telefone: (21) 9632 7521  
E-mail: [manoteixeira@yahoo.com.br](mailto:manoteixeira@yahoo.com.br)

## **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 (Capability Maturity Model Integration) em instituições financeiras. Dentro deste contexto, esta pesquisa buscou identificar por meio de uma amostra de 40 produtos que foram testados por um fornecedor o nível de aderência a estas práticas. Após esta análise, verificou-se junto aos clientes por meio de questionários dos modelos de avaliação de qualidade de software ISO/IEC 9126 seus respectivos impactos. Os referenciais teóricos utilizados são: Testes de Software, ISO/IEC 9126, o CMMI (Capability Maturity Model Integration) do SEI (Software Engineering Institute), da Carnegie Mellon University.

Palavras-chave: Fatores Humanos e Tecnológicos, Qualidade, 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(Capability Maturity Model Integration) in banking organizations. Inside this context, such research has searched for indentifying through 40 products that were tested by a supplier the level of tack to such practices. After this analise, it was noticed together with the client using questionnaires of the avaliation 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 (Capability Maturity Model Integration) of SEI (Software Engineering Institute), of the Carnegie Mellon University.

Key-words: Human and Technological Factors, Quality Management, Software Testing, CMMI, ISO/IEC 9126

## 1 INTRODUÇÃO

Na sociedade moderna, os softwares passaram a ter um papel vital na condução e execução dos negócios nas organizações, sendo em alguns casos parte intrínseca dos seus negócios. Segundo Rooijmans (1996) a quantidade de software incorporado em produtos de consumo duplica a cada 18 meses. Com a crescente demanda por novas tecnologias pelas empresas e pela sociedade em geral, esta importância tende a se multiplicar nos próximos anos. Diante deste cenário, a confiabilidade dos softwares passa a ser um fator crucial para os fabricantes junto aos seus clientes, que buscam fornecedores capazes de produzir softwares de alta qualidade em prazos cada vez menores.

Atualmente, existe uma crescente preocupação das empresas de software com a satisfação de seus clientes, com a produtividade de suas equipes e com os custos de seus projetos. Segundo Martin&McClure (1984) 90% dos softwares liberados possuem erros críticos. De acordo com o relatório “The Economic Impact of Inadequate Infrastructure for Software Testing” (NIST, 2002) estima-se que os custos com defeitos em softwares custem às empresas americanas um valor próximo de 1% do PIB dos Estados Unidos.

O objetivo do processo de teste é encontrar erros na implementação de um software e validar os requisitos definidos pelos usuários. Vale reafirmar que o processo de teste tem um papel chave na qualidade do produto de software. Porém, em muitos casos, o desenvolvimento e execução dos testes são feitos sem métodos estruturados. Além disso, frequentemente os testes são selecionados de forma aleatória e os casos de teste são desenvolvidos de uma forma não estruturada e não sistemática.

Diante do cenário apresentado, foi realizada uma pesquisa, que teve como objetivo avaliar se a adoção das práticas de testes estabelecidas pelo modelo CMMI (Capability Maturity Model Integration) no desenvolvimento de produtos de software impacta na qualidade sob a ótica dos clientes.

## 1.1. FORMULAÇÃO DA SITUAÇÃO PROBLEMA E QUESTÃO DA PESQUISA

O desenvolvimento de software com qualidade, dentro de prazos, custos e que satisfaça os usuários exigem dos fabricantes a melhoria dos processos da engenharia de software. Segundo dados de pesquisa realizada pelo Departamento de Comércio Americano mais de 60% das organizações reportaram grandes erros na utilização de softwares e 80% reportaram pequenos erros. Aliado a este patamar de qualidade, cabe ressaltar que apenas 1,3% dos projetos de softwares entregues avaliam a satisfação do cliente (ISBSG, 2003).

No Brasil, segundo dados do Ministério da Ciência e Tecnologia menos de 30% das empresas de desenvolvedoras de software do país possuíam processos estruturados de planejamento e documentação de testes. Estes números indicam um baixo grau de maturidade na área de testes conforme mostrado na tabela 1 a seguir.

MÉTODOS UTILIZADOS PARA DETECÇÃO DE DEFEITOS NO BRASIL		
Método	Nº de Empresas	%
Testes Funcionais	263	61,7%
Testes de Campo	255	59,9%
Testes de Aceitação	205	48,1%
Testes do Sistema Integrado	199	46,7%
Validação	192	45,1%
Testes de Integração	190	44,6%
Avaliação do Produto	179	42,0%
Avaliação da Usabilidade	149	35,0%
Testes de Unidade	130	30,5%
<b>Documentação Formal de Testes</b>	<b>115</b>	<b>27,0%</b>
<b>Planejamento Formal dos Testes</b>	<b>107</b>	<b>25,1%</b>
<b>Testes Estruturais</b>	<b>103</b>	<b>24,2%</b>
Inspeções Formais	87	20,4%
Revisões Estruturadas	66	15,5%
Outros	5	1,2%
Não adota métodos para detecção	31	7,3%
<b>Base</b>	<b>426</b>	

Tabela 1 - Métodos utilizados para detecção de defeitos no Brasil.

Fonte: Ministério da Ciência e Tecnologia (2001)

Os processos de testes de software descritos pelo CMMI se refletem atualmente no comportamento das empresas na busca em implantar ou mesmo melhorar o processo de teste utilizado. Ainda que as técnicas de teste de software mais utilizadas foram criadas por volta dos anos 80, as empresas têm uma grande dificuldade com a atividade de teste.

Em função desse cenário, temos o seguinte questionamento:

**Quais os impactos na qualidade dos produtos de software sob a ótica do cliente na adoção dos processos de testes baseados no modelo CMMI?**

## 1.2. HIPÓTESES E QUESTÕES DA PESQUISA

**Hipótese I** - A adoção dos processos de testes baseados no modelo CMMI (Capability Maturity Model Integration) aplicados no desenvolvimento de produtos de software impacta na qualidade sob a ótica do cliente.

Questões-chave: ISO/IEC 9126

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

Questões-chave: CMMI

1. Os produtos foram selecionados para verificação?
2. O ambiente de verificação foi devidamente estabelecido?
3. Os procedimentos e critérios de verificação foram estabelecidos?
3. A revisão pelos participantes foi devidamente preparada?
4. A revisão foi devidamente conduzida?
5. Os dados de revisão dos participantes foram analisados?
6. A verificação foi realizada?
7. Os resultados das verificações foram analisados e ações de correção foram identificadas?

8. Os produtos foram selecionados para validação?
9. O ambiente de validação foi devidamente estabelecido?
10. Os procedimentos e critérios de validação foram estabelecidos?
11. A validação foi realizada?
12. Os resultados das validações foram analisados e ações de correção foram identificadas?

## 2 REFERENCIAL TEÓRICO

### 2.1 TESTES DE SOFTWARE

Existem diversas definições para teste de software:

- processo de executar um programa ou sistema com a intenção de encontrar erros (Myers, 1979).
- qualquer atividade que a partir da avaliação de um atributo ou capacidade de um programa ou sistema seja possível determinar se ele alcança os resultados desejados (Hetzel, 1988).

O processo de teste envolve basicamente:

- Entradas: envolvendo a especificação de requisitos e do projeto do projeto;
- Processo: utilização dos planos de procedimentos de testes, bem como ferramentas para auxílio na depuração e no teste propriamente dito e;
- Saídas: avaliação dos resultados obtidos na fase anterior.



Figura 1 – Atividades do processo de teste.

Fonte: Inthurn (2001)

#### 2.1.1 OBJETIVOS DO TESTE

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 com isso 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 existem 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, sendo que quanto antes for descoberta a falha, menos custoso será 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 descobrir os erros 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.

### 2.1.2 TIPOS DE TESTE

Segundo (Hetzel,1988), (Myers,1979), (Beizer,1995), os tipos de testes podem ser definidos como:

- Testes Caixa Preta (Black Box): visam verificar a funcionalidade e a aderência aos requisitos, em uma visão externa ou do usuário, sem conhecimento do código e da lógica interna do componente testado.
- Testes Caixa Branca (White Box): visam avaliar as cláusulas de código, a lógica interna do componente codificado, as configurações e outros elementos técnicos.
- Testes Unitários: estágio mais baixo da escala de testes e são aplicados nos menores componentes de código criados.
- Testes de Integração: são executados em combinação de componentes para verificar se eles funcionam corretamente juntos.
- Testes de Regressão: visam 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.
- Testes Funcionais: grupos de testes que validam se o que foi especificado foi implementado.
- Teste de Qualidade de Código: 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: verificam a capacidade de retornar a uma versão anterior do software.
- Testes de Interoperabilidade: avaliam as condições de integração com outros softwares e /ou ambientes.
- Testes de Sobrevivência: avaliam a capacidade do software em continuar operando mesmo quando algum elemento (software ou hardware) fica inoperante ou para de funcionar.
- Testes Estéticos: avaliam toda a documentação do projeto, tais como modelos, requisitos, etc.



- Teste Embutido: avalia a capacidade de integração entre o hardware e o software.
- Teste de Conferência de Arquivos: verificam alterações nos arquivos usados.
- Testes Alfa: são executados quando o desenvolvimento está próximo a sua conclusão.
- Testes Beta: são executados quando o desenvolvimento e testes estão praticamente concluídos.
- Teste de Verificação de Sites Web: verificam problemas que possam haver no site como links inválidos, arquivos órfãos, ligações entre páginas (Molinari, 2003).

## 2.2 CMMI

O propósito do CMMI é fornecer orientação para melhorar os processos de uma organização e sua habilidade em gerenciar o desenvolvimento, aquisição e manutenção de produtos ou serviços.

O Modelo de Maturidade da Capabilidade de Software fornece às organizações de software um guia de como obter controle de seus processos para desenvolver e manter software e como evoluir em direção a uma cultura de engenharia de software e excelência de gestão. O CMMI foi projetado para guiar as organizações de software no processo de seleção das estratégias de melhoria, determinando a maturidade atual do processo e identificando as poucas questões mais críticas para a qualidade de software e melhoria do processo. Focando em um conjunto limitado de atividades e trabalhando agressivamente para concluí-las com êxito, a organização pode melhorar o processo de software na organização toda para possibilitar ganhos contínuos e duradouros na capabilidade do processo de software.

### 2.2.1. COMPONENTES E ESTRUTURA DO MODELO

O modelo CMMI é desenhado para descrever níveis discretos de melhoria de processos. Na representação em estágios, os níveis de maturidade estabelecem uma ordem recomendada para melhoria dos processos, organizando as áreas de processos. Estas, por sua vez, contém metas gerais e específicas, bem como práticas também genéricas e específicas. A figura a seguir mostra a estrutura e os componentes do modelo CMMI:

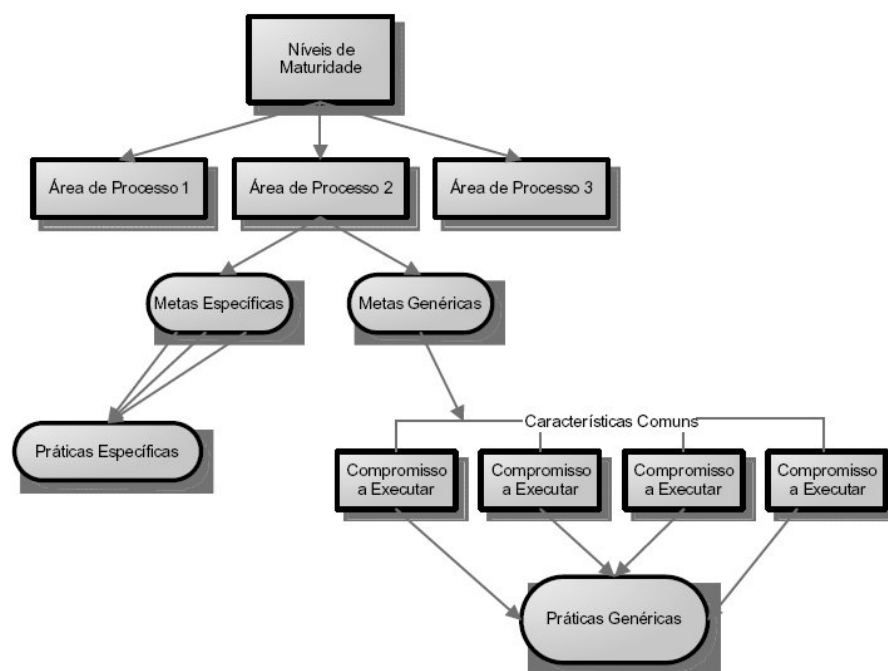


Figura 2 - Componentes do Modelo CMMI

Fonte: SEI (2001a)

### 2.2.2 NÍVEIS DE MATURIDADE

O CMMI fornece uma estrutura para organizar essas etapas evolutivas, em cinco níveis de maturidade, que coloca fundamentos sucessivos para a contínua melhoria do processo. Esses cinco níveis de maturidade definem uma escala ordinal para medir a maturidade de melhoria de processos de uma organização e para avaliar a sua capacidade do processo diversos.

Os níveis de maturidade são designados por números, de 1 a 5:

1. Inicial;
2. Gerenciado;
3. Definido;
4. Quantitativamente Gerenciado;
5. Em Otimização.

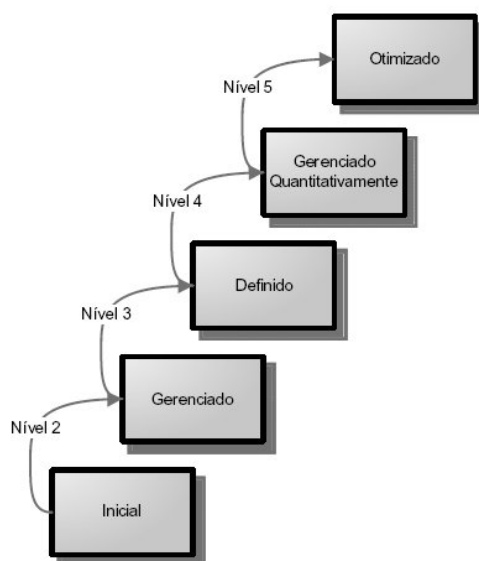


Figura 3 - Níveis de Maturidade na representação em estágios

Fonte: adaptado de SEI (1997)

### 2.2.3 ÁREAS DE PROCESSO RELACIONADAS A TESTES

No modelo CMMI, existem duas áreas direcionadas para as atividades de testes: Verificação e Validação.

#### 2.2.3.1 VERIFICAÇÃO

As metas, práticas da área de processo de Verificação do modelo podem ser identificadas conforme o quadro 1 a seguir:

Metas Específicas / Genéricas	Práticas Específicas / Genéricas
SG 1 Preparar para a Verificação	SP 1.1 Selecionar os produtos para verificação
	SP 1.2 Estabelecer o ambiente de verificação
	SP 1.3 Estabelecer os procedimentos e critérios de verificação
SG 2 Revisão pelos participantes	SP 2.1 Preparar revisão pelos participantes
	SP 2.2 Conduzir a revisão pelos participantes
	SP 2.3 Analisar os dados da revisão pelos participantes
SG 3 Verificar os produtos selecionados	SP 3.1 Realizar a verificação
	SP 3.2 Analisar os resultados da verificação e identificar as ações de correção

Quadro 1 – Características da Área de Processo Verificação

Fonte: Elaboração própria

### 2.2.3.2 VALIDAÇÃO

As metas, práticas da área de processo de Validação do modelo podem ser identificadas conforme o quadro 2 a seguir:

Metas Específicas / Genéricas	Práticas Específicas / Genéricas
SG 1 Preparar para a validação	SP 1.1 Selecionar os produtos para validação
	SP 1.2 Estabelecer o ambiente de validação
	SP 1.3 Estabelecer os procedimentos e critérios de validação
SG 2 Validar Produtos ou Componentes do Produto	SP 2.1 Realizar a validação
	SP 2.2 Analisar os resultados da validação

Quadro 2 – Características da Área de Processo Validação

Fonte: Elaboração própria

### 2.3 ISO/IEC 9126-1

A série de normas ISO/ IEC 9126 (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 pode 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 sub-características. As sub-características podem ser medidas por meio de métricas externas e internas.

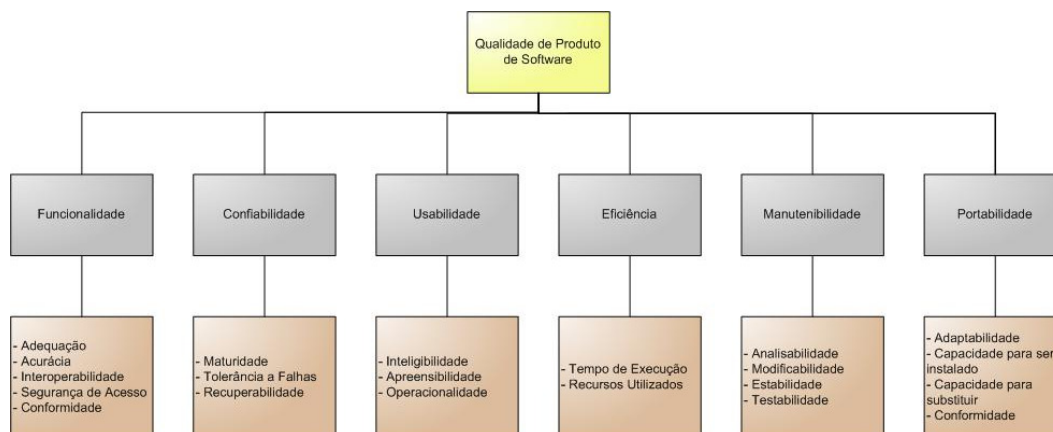


Figura 4 – Modelo de qualidade interna e externa segundo a ISO/IEC 9126- 1

Fonte: ISO (2001)

Definição 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.

Sub-características: 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.

Sub-características: 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.

Sub-características: Inteligibilidade, Apreensibilidade e Operacionalidade.

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

Sub-características: 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.

Sub-características: Analisabilidade, Modificabilidade, Estabilidade e Testabilidade.

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

Sub-características: Adaptabilidade, Capacidade para ser instalado, Capacidade para Substituir e Conformidade de Portabilidade.

### 3 METODOLOGIA

#### 3.1 MÉTODO DE ABORDAGEM

Para realização dessa pesquisa, foi utilizado o método hipotético dedutivo de Popper, que prega que o conhecimento científico evolui a partir de hipóteses.

A partir desta premissa, Popper constrói sua visão do método científico, o racionalismo crítico, e também do conhecimento em geral. Ambos evoluem por meio do que ele denomina de conjecturas e refutações. Isto significa que a busca do conhecimento se inicia com a formulação de hipóteses, que por sua vez procuram resolver problemas, e continua com tentativas de refutação dessas hipóteses, através de testes que envolvem observações ou experimentos.

#### 3.2. POPULAÇÃO / UNIVERSO E AMOSTRA

Adotou-se como universo da presente pesquisa o formado pela empresa Tools Software e seus clientes que são compostos por empresas do setor mercado financeiro. As instituições clientes avaliadas foram :GE Capital, IBI Bank, Banco Cruzeiro do Sul e Banco Máxima

Foram escolhidos 40 produtos da Tools Software disponibilizados para os quatro clientes selecionados.

A expectativa era relacionar um número maior de projetos, mais era necessário que os produtos e serviços possuísem o mesmo conjunto de características indicadas a seguir:

#### Produtos

- Arquitetura Cliente-Servidor
- Linguagem de Programação
- Tempo de Vida do Produto no Mercado
- N° de Funcionalidades

### 3.3. COLETA DE DADOS

O processo de coleta de dados no campo foi feito em duas etapas, conforme descrito:

A primeira etapa consistiu em pesquisa via e-mail com 20 analistas e/ou responsáveis pelo processo de testes, com o objetivo de identificar o nível de aderência às práticas de testes do modelo CMMI em cada produto e serviço de testes. Nesta etapa foi utilizado o questionário CMMI.

Foram definidos dois grupos de acordo com o seguinte critério: O produto pertencerá ao maior nível cujo score médio, obtido pelo CMMI, for maior ou igual a 3.

Desta forma, foi observado que:

- 12 produtos atingiram score médio  $< 3$ ;
- 28 produtos atingiram score médio  $\geq 3$

Foram selecionados, para fins de análise estatística, dois grupos:

- Grupo 1: 12 produtos (Não aderente às práticas do CMMI)
- Grupo 2: 28 produtos (Aderente às práticas do CMMI)

A segunda etapa consistiu em pesquisa via e-mail com os 40 usuários e/ou equipes de atendimento dos produtos de software visando avaliar as percepções que os usuários têm da qualidade dos produtos e serviços de testes de software prestado. Nesta etapa foi utilizado o questionário ISO/IEC 9126.

### 3.4. LIMITAÇÕES DO MÉTODO

Os dados obtidos através dos e-mail, não garantem que possam refletir a realidade, devido à existência de respostas distorcidas, causadas pelo grau de motivação do entrevistado, a falta de conhecimento sobre o assunto pesquisado, assim como, a inadequação do questionário (excessivo número de perguntas, escala utilizada e tempo, entre outros).

Há também a possibilidade dos respondentes aumentarem propositalmente os escores, de forma a não transmitir uma avaliação ruim de si próprios ou de suas empresas, mesmo com a indicação de que a pesquisa não tem por objetivo selecionar o entrevistado ou a empresa. Da mesma forma, o entrevistador exerce influência sobre as respostas dos entrevistados, assim como a apresentação e explicação dos itens, que podem ter influenciado no comportamento dos respondentes.

## 4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

### 4.1. MÉTODO ESTATÍSTICO

A análise estatística foi realizada pelo teste t de Student (Jerrold, 1999) para amostras independentes, para verificar se existe diferença significativa nos escores médios das características/aspectos da ISO/IEC 9126 nas questões individuais entre dois grupos estabelecidos pelo questionário do CMMI.

O critério de determinação de significância adotado foi o nível de 5%, ou seja, quando o p valor do teste estatístico for menor ou igual a 0,05, então existe significância estatística.

### 4.2 – ANÁLISE DOS RESULTADOS

#### 4.2.1 Análise 1 – Escores das Características e Sub-características dos Grupos

A tabela 2 fornece a média, erro padrão (EP), mínimo, máximo dos escores das características e sub-características da norma ISO/IEC 9126 segundo os grupos. O valor da estatística t e o



respectivo nível de significância (p valor) foram calculados para as vinte e duas sub-características.

CARACTERÍSTICA	QUESTÕES	GRUPO	n	MÉDIA	E.P.	MÍNIMO	MÁXIMO	t	p valor
FUNCIONALIDADE	Q01	1	12	2,67	0,49	2	3	2,495	<b>0,0171</b>
	Q01	2	28	3,21	0,69	2	4		
	Q02	1	12	2,67	0,49	2	3	2,477	<b>0,0178</b>
	Q02	2	28	3,07	0,47	2	4		
	Q03	1	12	2,83	0,39	2	3	2,209	<b>0,0332</b>
Q03	2	28	3,29	0,66	2	4			
Q04	1	12	2,33	0,49	2	3	2,164	<b>0,0368</b>	
Q04	2	28	3,04	1,07	2	5			
Q05	1	12	2,67	0,49	2	3	2,336	<b>0,0248</b>	
Q05	2	28	3,11	0,57	2	5			
CONFIABILIDADE	Q06	1	12	2,50	0,52	2	3	3,032	<b>0,0044</b>
	Q06	2	28	3,04	0,51	2	4		
	Q07	1	12	2,58	0,51	2	3	2,763	<b>0,0088</b>
	Q07	2	28	3,29	0,81	2	4		
Q08	1	12	2,83	0,58	2	4	2,209	<b>0,0332</b>	
Q08	2	28	3,29	0,60	2	5			
USABILIDADE	Q09	1	12	3,08	0,29	3	4	2,126	<b>0,0401</b>
	Q09	2	28	3,68	0,94	2	5		
	Q10	1	12	3,08	0,29	3	4	2,114	<b>0,0411</b>
	Q10	2	28	3,61	0,83	2	5		
	Q11	1	12	3,00	0,43	2	4	2,135	<b>0,0392</b>
Q11	2	28	3,57	0,88	2	5			
EFICIÊNCIA	Q12	1	12	2,75	0,45	2	3	2,304	<b>0,0268</b>
	Q12	2	28	3,21	0,63	2	4		
	Q13	1	12	2,83	0,39	2	3	2,209	<b>0,0332</b>
	Q13	2	28	3,29	0,66	2	4		
MANUTENIBILIDADE	Q14	1	12	2,92	0,51	2	4	2,131	<b>0,0396</b>
	Q14	2	28	3,50	0,88	2	5		
	Q15	1	12	2,83	0,58	2	4	2,147	<b>0,0382</b>
	Q15	2	28	3,43	0,88	2	5		
	Q16	1	12	2,58	0,51	2	3	2,807	<b>0,0079</b>
	Q16	2	28	2,96	0,33	2	3		
	Q17	1	12	2,58	0,51	2	3	2,879	<b>0,0065</b>
Q17	2	28	3,04	0,43	2	4			
PORTABILIDADE	Q18	1	12	3,17	0,39	3	4	2,078	<b>0,0445</b>
	Q18	2	28	3,61	0,69	3	5		
	Q19	1	12	2,92	0,29	2	3	2,154	<b>0,0376</b>
	Q19	2	28	3,36	0,68	2	5		
	Q20	1	12	3,17	0,72	2	4	2,078	<b>0,0445</b>
Q20	2	28	3,61	0,57	3	5			
Q21	1	12	3,25	0,45	3	4	2,069	<b>0,0454</b>	
Q21	2	28	3,82	0,90	3	5			

Tabela 2 - Análise das características e sub-característica segundo norma ISO/IEC 9126

#### 4.2.2 Análise 2 - Sub-características ISO/IEC 9126 mais impactadas

A tabela 3 fornece os escores médios das sub-características ordenados pelo valor t.

IND.	CARACTERÍSTICA	SUB-CARACTERÍSTICA	QUESTÕES	t	p valor
1º	CONFIABILIDADE	Maturidade	Q06	3,032	0,004
2º	MANUTENIBILIDADE	Testabilidade	Q17	2,879	0,007
3º	MANUTENIBILIDADE	Estabilidade	Q16	2,807	0,008
4º	CONFIABILIDADE	Tolerância à falhas	Q07	2,763	0,009
5º	FUNCIÓNALIDADE	Adequação	Q01	2,495	0,017
6º	FUNCIÓNALIDADE	Acurácia	Q02	2,477	0,018
7º	FUNCIÓNALIDADE	Segurança de acesso	Q05	2,336	0,025
8º	EFICIÊNCIA	Tempo de Execução	Q12	2,304	0,027
9º	FUNCIÓNALIDADE	Interoperabilidade	Q03	2,209	0,033
10º	CONFIABILIDADE	Recuperabilidade	Q08	2,209	0,033
11º	EFICIÊNCIA	Recursos Utilizados	Q13	2,209	0,033
12º	FUNCIÓNALIDADE	Conformidade	Q04	2,164	0,037
13º	PORTABILIDADE	Facilidade para Instalação	Q19	2,154	0,038
14º	MANUTENIBILIDADE	Modificabilidade	Q15	2,147	0,038
15º	USABILIDADE	Operacionalidade	Q11	2,135	0,039
16º	MANUTENIBILIDADE	Analisabilidade	Q14	2,131	0,040
17º	USABILIDADE	Inteligibilidade	Q09	2,126	0,040
18º	USABILIDADE	Apreensibilidade	Q10	2,114	0,041
19º	PORTABILIDADE	Adaptabilidade	Q18	2,078	0,045
20º	PORTABILIDADE	Facilidade para Substituição	Q20	2,078	0,045
21º	PORTABILIDADE	Conformidade com Portabilidade	Q21	2,069	0,045

Tabela 3 - Escores médios das sub-características ordenados pelo valor t – ISO/IEC 9126.

## 5 CONCLUSÕES E RECOMENDAÇÕES

Com base nas premissas listadas anteriormente e nos resultados da pesquisa executada, pôde-se estabelecer as seguintes análises quanto ao problema da pesquisa, hipóteses utilizadas e conclusões:

### 5.1. SOLUÇÃO DO PROBLEMA

O problema formulado nesta pesquisa foi: **Quais os impactos na qualidade dos produtos de software sob a ótica do cliente na adoção dos processos de testes baseados no modelo CMMI?**

Após análise estatística dos resultados obtidos em cada questão dos questionários aplicados, 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 sub-características observou-se que os escores médios das sub-características “Maturidade”, “Testabilidade” e “Estabilidade” são os de maior diferença entre os grupos avaliados. Observa-se também que as características de “Portabilidade” apresentam as menores diferenças entre os grupos avaliados.

De forma analítica, a resposta ao problema apresentado neste estudo poder ser formulada da seguinte forma: Os impactos na qualidade dos produtos de software sob a ótica do cliente na adoção dos processos de testes baseados no modelo CMMI são: Adequação, Acurácia, Interoperabilidade, Conformidade, Segurança de acesso, Maturidade, Tolerância à falhas, Recuperabilidade, Inteligibilidade, Apreensibilidade, Operacionalidade, Tempo de Execução, Recursos Utilizados, Analisabilidade, Modificabilidade, Estabilidade, Testabilidade, Adaptabilidade, Facilidade para Instalação, Facilidade para Substituição e Conformidade com Portabilidade.

## 5.2. VERIFICAÇÃO DAS HIPÓTESES

A metodologia aplicada nesta pesquisa tem como base o teste de falseabilidade das hipóteses levantadas através do método da hipótese nula, ou seja, pela aplicação de um teste estatístico adequado à natureza das variáveis e da amostra analisada de forma a verificar o grau de significância dos resultados obtidos.

Cada característica de qualidade de software do modelo ISO/IEC 9126 foi avaliada em relação a metodologia utilizada.

Com os resultados coletados no campo, tabulados e testados estatisticamente, a hipótese foi analisada com a possibilidade de ser validada - total ou parcialmente - ou refutada. Além disso, foram respondidas as questões-chave associadas. A seguir, estão relacionadas cada hipótese, sua análise com validação total, parcial ou refutação e a resposta para cada questão-chave.

### 5.2.1. Hipótese I

A adoção dos processos de testes baseados no modelo CMMI (Capability Maturity Model Integration) aplicados no desenvolvimento de produtos de software impacta na qualidade sob a ótica do cliente.

Resposta à Hipótese I: **A hipótese foi corroborada**, logo, conclui-se que a adoção dos processos de testes baseados no modelo CMMI (Capability Maturity Model Integration) aplicados no desenvolvimento de produtos de software impacta na qualidade sob a ótica do cliente.

Questões Chave – Questionário ISO/IEC 9126

#### **Funcionalidade**

1. O conjunto de funções satisfaz as necessidades explícitas e implícitas para a finalidade a que se destina o produto ?

**Resposta:** Analisando os escores médios das questões referentes a característica de qualidade **Funcionalidade** observou-se, que os escores médios das questões 1, 2 , 3 , 4 e 5 do grupo 2 foi significativamente maior que o grupo 1. Isto evidencia que, os usuários do grupo do grupo 2 percebem maior presença de características de qualidade no produtos avaliados, conforme mostrado na Tabela 3.

#### **Confiabilidade**

2. O desempenho se mantém ao longo do tempo e em condições estabelecidas? Ou seja, é tolerante a falhas?

**Resposta:** Analisando os escores médios das questões referentes a característica de qualidade **Confiabilidade** observou-se, que os escores médios das questões 6, 7 , e 8 do grupo 2 foi significativamente maior que o grupo 1. Isto evidencia que, os usuários do grupo 2 percebem maior presença de características de qualidade no produtos avaliados, conforme mostrado na Tabela 3.

#### **Usabilidade**

3. O software é fácil de usar?

**Resposta:** Analisando os escores médios das questões referente a característica de qualidade **Usabilidade** observou-se, que os escores médios das questões 9, 10 , e 11 do grupo 2 foi significativamente maior que o grupo 1. Isto evidencia que, os usuários do grupo do grupo 2 percebem maior presença de características de qualidade no produtos avaliados, conforme mostrado na Tabela 3.

### **Eficiência**

4. Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido pelo produto?

**Resposta:** Analisando os escores médios das questões referente a característica de qualidade **Eficiência** observou-se, que os escores médios das questões 12 e 13 do grupo 2 foi significativamente maior que o grupo 1. Isto evidencia que, os usuários do grupo do grupo 2 percebem maior presença de características de qualidade no produtos avaliados, conforme mostrado na Tabela 3.

### **Manutenibilidade**

5. É fácil corrigir, atualizar e alterar o software?

**Resposta:** Analisando os escores médios das questões referente a característica de qualidade **Manutenibilidade** observou-se, que os escores médios das questões 14, 15, 16 e 17 do grupo 2 foi significativamente maior que o grupo 1. Isto evidencia que, os usuários do grupo do grupo 2 percebem maior presença de características de qualidade no produtos avaliados, conforme mostrado na Tabela 3.

### **Portabilidade**

6. É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação?

**Resposta:** Analisando os escores médios das questões referente a característica de qualidade **Portabilidade** observou-se, que os escores médios das questões 18, 19, e 21 do grupo 2 foi significativamente maior que o grupo 1. Isto evidencia que, os usuários do grupo do grupo 2 percebem maior presença de características de qualidade no produtos avaliados, conforme mostrado na Tabela 3.

### 5.3. CONCLUSÕES

Após a verificação da hipótese e resposta às questões-chave, pôde ser feita uma análise dos resultados com relação à contextualização do problema e, a partir desta análise, fazer inferências sobre os impactos na qualidade dos produtos sob a ótica dos clientes com a adoção dos processos de testes baseados no modelo CMMI. Pode-se, a partir dos resultados encontrados, concluir que:

- A Confiabilidade dos produtos perante o cliente aumenta significativamente com a adoção dos processos de testes. Isto permite que empresas que investirem na adoção destes processos possam se beneficiar frente a seus concorrentes, por se tratar de um aspecto fundamental na seleção de fornecedores.
- A Manutenibilidade é uma característica altamente impactada com a adoção dos processos de testes. Levando em conta que estudos apontam o custo de manutenção como sendo responsável por 40% dos custos totais, pode-se afirmar que a aplicabilidade do modelo permite a redução de custos de desenvolvimento e manutenção de produtos.
- A adoção de um processo de testes estruturado permite melhor desenvolvimento das sub-características de Funcionalidade, por exigir durante sua execução dos testes que se tenha maior entendimento dos requisitos do produto frente ao cliente.
- Os processos de testes ainda oferecem poucos benefícios nas áreas de Usabilidade e Portabilidade. Cabe ressaltar que estas características estão em uma posição inferior nas prioridades do cliente em geral.
- A implantação de processos de melhoria de qualidade é mais que uma tendência nas empresas de software é um requisito primordial para enfrentar o mercado competitivo e atender às exigências de seus clientes.
- A metodologia do teste de software se reflete atualmente no comportamento das empresas na busca de implantar, ou mesmo melhorar o processo de teste utilizado.

- Torna-se necessário uma política de qualidade em todos os processos no desenvolvimento do software, aliada a uma equipe de testes organizada e treinada, de modo a auxiliar e minimizar os erros que possam passar despercebidos pelas equipes de desenvolvimento.

#### 5.4. SUGESTÕES PARA ESTUDOS FUTUROS

O tema desta pesquisa não se esgota neste trabalho, o estudo dos impactos da adoção dos processos de testes do modelo CMMI no desenvolvimento de produtos possui vários outros aspectos que são passíveis de uma investigação mais aprofundada. A seguir, são feitas algumas sugestões de futuros estudos, que podem complementar e aprofundar o trabalho aqui apresentado:

- Pesquisas que identifiquem qual a relação custo/benefício da adoção dos processos de testes de baseados no modelo CMMI.
- Estudos que avaliem o nível de aderência do modelo CMMI a outros modelos de maturidade de testes como o TPI, TMM, TOM e TAP.
- Pesquisas que busquem aprimorar a escolha nas estratégias de testes utilizadas pelos modelos de maturidade, objetivando a otimização do processo dos testes.
- Estudos que avaliem o nível de importância de cada característica de qualidade do modelo ISO/IEC 9126 junto aos clientes.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

DEUTSCH, M. Verification and Validation. In: PRESSMAN, Roger. Engenharia de Software. São Paulo: Makron Books, 1995. p. 780-781

GARTNER GROUP, Testing client/server applications: challenges, strategies and solutions for success. Stamford: 22 Nov. 1996. (Strategic Analysis Report).

HETZEL, Bill. The Complete Guide to Software Testing - Second Edition, 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.

\_\_\_\_\_. Characterizing the Software Process, IEEE Software, VoL.5, No.2, March, 1988, pp73-79.

HUMPHREY, W.S.; KITSON, D.H.; and GALE, J. A Comparison of U.S. and Japanese Software Process Maturity, Proceedings of the 13 th International Conference on Software Engineering. Austin, TX, 13-17 May 1991, pp.38-49.

IEEE, The Institute of Electrical and Electronics Engineers. IEEE Std 829: Standard for Software Test Documentation. New York: IEEE Computer Society, September, 1998.

INTHURN, Cândida. Qualidade e Teste de Software, Visual Books, 2001.

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

ISO, International Standart Organization; ISO/IEC 9126 – Software Engineering - Product Quality, 2001.

JERROLD H. Zar, Bioestatistical Analisis. Prentice-Hall, Inc, 4th ed., New Jersey, USA, 1999.

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

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

MARTIN J. & MCCLURE C., Software Maintenance: The Problem and Its Solutions' Prentice-Hall, 1984

MCT, Ministério da Ciência e Tecnologia. Programa Nacional de Qualidade de Software, Brasília, 2001.

MOLINARI, Leonardo. Testes de Software: Produzindo Sistemas Melhores e Mais Confiáveis, São Paulo: Érica, 2003.

MYERS, Glen. The Art of Software Testing. New York: Wiley, 1979.

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

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

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/cmmi/general/general.html>. Acesso em 10 agosto 2004.

ROOIJMANS, A. AND GENUCHTEN, M. Software Quality in Consumer Electronic Products, IEEE Software, January 1996, 55-64.

WATSON, R. T.; PITT, L. F.; CUNNINGHAM, C. J.; NEL, D; "User satisfaction and service quality of the IS department: closing the gaps". Journal of Information Technology. 1993