

UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE ENGENHARIA MECÂNICA E DE PRODUÇÃO
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**ANÁLISE DE MODELOS DE MELHORIA NO
DESENVOLVIMENTO DE *SOFTWARE***

Santa Bárbara d'Oeste
2000

UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE ENGENHARIA MECÂNICA E DE PRODUÇÃO
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**ANÁLISE DE MODELOS DE MELHORIA NO
DESENVOLVIMENTO DE *SOFTWARE***

Guilherme Augusto Spiegel Gualazzi

Orientador: Prof. Dr. Néocles Alves Pereira

Santa Bárbara d'Oeste
2000

UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE ENGENHARIA MECÂNICA E DE PRODUÇÃO
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**ANÁLISE DE MODELOS DE MELHORIA NO
DESENVOLVIMENTO DE *SOFTWARE***

Guilherme Augusto Spiegel Gualazzi

Orientador: Prof. Dr. Néocles Alves Pereira

Dissertação apresentada ao Programa de Pós Graduação em Engenharia de Produção, da Faculdade de Engenharia Mecânica e de Produção, da Universidade Metodista de Piracicaba - UNIMEP, como requisito para obtenção ao Título de Mestre em Engenharia de Produção.

Santa Bárbara d'Oeste
2000

Gualazzi, Guilherme Augusto Spiegel

G899a Análise de modelos de melhoria no desenvolvimento de
software./ Guilherme Augusto Spiegel Gualazzi. - Santa
Bárbara D'Oeste, SP:[s.n.], 1999.

Orientador : Néocles Alves Pereira.

Dissertação (Mestrado) – Universidade Metodista de
Piracicaba, Faculdade de Engenharia Mecânica e de Produção, Programa

ANÁLISE DE MODELOS DE MELHORIA NO DESENVOLVIMENTO DE *SOFTWARE*

Guilherme Augusto Spiegel Gualazzi

Dissertação de Mestrado defendida e aprovada, em 25 de setembro
de 2000, pela Banca Examinadora constituída pelos Professores:

Prof. Dr. Néocles Alves Pereira
UNIMEP

Prof. Dr. Paulo Augusto Cauchick Miguel
UNIMEP

Prof. Dr. Fernando Celso de Campos
Centro Universitário de Araraquara

Dedico este trabalho àquele
que me incentivou, meu eterno professor,
amigo e pai, Ilacyr.

Pai, subi mais um degrau...
com você.

AGRADECIMENTOS

Agradeço aos meus pais e às minhas irmãs, e à família de minha esposa, pelo apoio, incentivo e paciência. Sem vocês, todo meu esforço teria sido em vão.

Agradeço minha irmã, Priscila, que com sua experiência ajudou-me na revisão deste trabalho, no momento em que eu mais precisava.

Agradeço à minha amiga, parceira, namorada, esposa e cúmplice, Érica, pela paciência, apoio e amor, e por se doar a este projeto de vida, tanto quanto eu.

Agradeço a todos os meus amigos que, de alguma forma, me deram força para que eu continuasse sempre em frente. Peço que me desculpem pela longa ausência junto a vocês.

Agradeço a meu orientador, Prof. Dr. Néocles, que em momentos de dificuldade ensinou-me a seguir em frente, dando um passo de cada vez, sem nunca parar, por menor que fosse o passo.

“O objetivo primário de realizarmos medições no tocante ao desenvolvimento de *software* é obter níveis cada vez maiores de qualidade, considerando o projeto, o processo e o produto, visando à satisfação plena dos clientes ou usuários, a um custo economicamente compatível”.

FERNANDES, 1995.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS.....	viii
LISTA DE FIGURAS.....	xi
LISTA DE TABELAS.....	xii
GLOSSÁRIO.....	xiii
RESUMO.....	xvii
ABSTRACT.....	xviii
1 INTRODUÇÃO.....	1
1.1 A Proposta do MCT.....	2
1.2 Objetivos.....	6
1.3 Estrutura do Trabalho.....	6
2 QUALIDADE E <i>SOFTWARE</i>	7
2.1 Ferramentas Gerenciais e Operacionais da Qualidade.....	8
2.1.1 TQM - Gerenciamento da Qualidade Total.....	8
2.1.2 QFD - Desdobramento da Função Qualidade.....	9
2.1.3 Qualidade em Serviços.....	10
2.1.4 As Ferramentas da Qualidade.....	11
2.2 Sistemas da Qualidade Baseados nas Normas ISO.....	11
2.2.1 ISO 9000-3 (1993) – Diretrizes para a aplicação da ISO 9001 (1994).....	13
2.2.2 ISO 12207 - Processos do Ciclo de Vida do <i>Software</i>	14
2.3 Qualidade de <i>Software</i>	17
2.3.1 Engenharia, Qualidade de Processo e Produto de <i>Software</i>	17

2.4 Considerações Finais.....	21
3 MODELOS DE MATURIDADE DA CAPACIDADE DO PROCESSO DE <i>SOFTWARE</i>	22
3.1 CMM - O Modelo de Maturidade da Capacidade.....	23
3.1.1 Nível 1 - O Nível Inicial.....	25
3.1.2 Nível 2 - O Nível Disciplinado.....	25
3.1.3 Nível 3 - O Nível Definido.....	26
3.1.4 Nível 4 - O Nível Gerenciado.....	27
3.1.5 Nível 5 - O Nível Otimizado.....	28
3.2 Entendendo os Níveis Gerenciado e Otimizado.....	28
3.3 Visibilidade no Processo de <i>Software</i>	30
3.4 Definição Operacional do CMM.....	32
3.4.1 Áreas-chave de Processo.....	33
3.4.2 Características Comuns e Práticas-base.....	39
3.5 Usando o CMM.....	41
3.6 Outros Modelos de Definição, Avaliação e Melhoria de Processos de <i>Software</i>	42
3.6.1 O Modelo Trillium.....	43
3.6.2 ISO 15504 (SPICE).....	48
3.7 Análise das Normas e Modelos Apresentados.....	51
3.8 Considerações Finais.....	53
4 ESTUDO DE CASOS.....	54

4.1 Modelos de SPI no Departamento de Defesa Americano e em Organizações Comerciais (MCGIBBON <i>et al.</i> , 1999).....	54
4.1.1 Benefícios Financeiros da Melhoria de Processo de <i>Software</i> (SPI)...	55
4.1.2 Os Benefícios Secundários do SPI.....	57
4.2 Uma Pesquisa Sistemática da Melhoria de Processo, seus Benefícios e Fatores (GOLDENSON <i>et al.</i> , 1995).....	58
4.2.1 Sobre as Avaliações de Implementações do CMM.....	59
4.2.2 Benefícios.....	60
4.2.3 Sobre os Fatores de Sucesso de Implementação do CMM.....	62
4.3 A Experiência da Raytheon Electronic Systems na Melhoria do Processo de <i>Software</i> (HALEY <i>et al.</i> , 1995).....	63
4.3.1 Custo da Qualidade.....	64
4.3.2 Produtividade de <i>Software</i>	66
4.3.3 Índice de Desempenho de Custo.....	68
4.3.4 Qualidade Global do Produto.....	68
4.3.5 Pessoal.....	70
4.3.6 Um Sumário das Atividades.....	70
4.4 O Uso do Modelo CMM e ISO 12207 na Celepar (MACHADO <i>et al.</i> , 1997).....	72
4.4.1 O Contexto da Celepar.....	73
4.4.2 Análise da MDS frente ao Modelo CMM.....	74
4.4.3 Proposta de Evolução da MDS.....	75
4.4.4 A Conclusão da Celepar.....	76
4.5 Uma Experiência com a Norma ISO 15504 (SALVIANO <i>et al.</i> , 1999)....	77

4.5.1 Projeto de Melhoria dos Processos de <i>Software</i> da Empresa.....	77
4.5.2 Execução da Avaliação dos Processos e Planejamento da Melhoria..	78
4.6 Considerações Finais.....	80
5 ANÁLISE DOS CASOS ESTUDADOS.....	81
5.1 Comparação entre os Casos Estudados.....	81
5.2 Considerações a respeito do uso de modelos de SPI.....	89
5.3 Considerações Finais.....	95
6 CONCLUSÃO.....	96
REFERÊNCIAS BIBLIOGRÁFICAS.....	100
BIBLIOGRAFIA CONSULTADA.....	105
ANEXOS.....	108
ANEXO 1 - Indicadores da Qualidade e Produtividade em <i>Software</i>	108
ANEXO 2 - Comparativo das Normas e Modelos Apresentados.....	114

LISTA DE ABREVIATURAS E SIGLAS

ASQ	- <i>American Society for Quality</i> - Sociedade Americana para a Qualidade
CAC	- <i>Cost At Completion</i> - Custo Atual na Conclusão.
CASE	- <i>Computer Aided Software Engineering</i> - Engenharia de Software Ajudada pelo Computador. Ferramenta utilizada na documentação de projetos de <i>software</i> .
CELEPAR	- Companhia de Informática do Paraná - órgão subsidiado pelo estado do Paraná.
CMM	- <i>Capability Maturity Model</i> - Modelo de maturidade da capacidade. Modelo desenvolvido pelo CMU/SEI, destinado à garantir a qualidade do processo de <i>software</i> .
CMU	- <i>Carnegie Mellon University</i>
CTI	- Centro Tecnológico para Informática - órgão pertencente ao Ministério da Ciência e Tecnologia.
DACS	- <i>DoD Data & Analysis Center for Software</i> - Centro de Dados & Análise para <i>Software</i> do Departamento de Defesa dos EUA.
DSI	- <i>Delivered Source Instructions</i> - Linhas de Código-Fonte Distribuídas.
FP	- <i>Function Points</i> - Pontos por função
IDD	- <i>Interface Design Document</i> - Documento do Projeto de Interface

ISO	- International Organization for Standardization - Organização Internacional de Normalização.
KDSI	- Milhares de Linhas de Código-fonte Distribuídas.
KPA	- Key Process Area - Área-chave de processo.
LOC	- <i>Lines Of Code</i> - Linhas de código-fonte.
MDS	- Modelo de Desenvolvimento de Serviços, da Celepar.
PSP	- <i>Personal Software Process</i> - Processo de <i>Software</i> Pessoal.
QFD	- <i>Quality Function Deployment</i> - Desdobramento da Função Qualidade. Método que ajuda a identificar exigências do usuário e incorporá-las ao projeto.
RAPID	- <i>Raytheon Advanced Prototyping, Integration, and Demonstration</i> - Prototipação, Integração e Demonstração Avançada da Raytheon.
RES	- <i>Raytheon Electronic Systems</i> .
RPAC	- Roteiro de Acompanhamento de Projetos.
RPRE	- Roteiro de Projeto Preliminar.
RREV	- Roteiro de Revisão.
SCE	- <i>Software Capability Evaluation</i> - Avaliação da Capacidade de <i>Software</i> .
SDD	- <i>Software Design Document</i> - Documento do Projeto de <i>Software</i>
SEI	- <i>Software Engineering Institute</i> - Instituto de Engenharia de <i>Software</i> , da <i>Carnegie Mellon University</i> .

- SEL - *Software Engineering Laboratory* - Laboratório de Engenharia de *Software* da RES.
- SEPG - *Software Engineering Process Group* - Grupo de Processo de Engenharia de *Software*, criado no Nível 3, do CMM.
- SPA - *Software Process Assessment* - Avaliação do Processo de *Software*.
- SPI - *Software Process Improvement* - Melhoria do Processo de *Software*.
- SPICE - *Software Process Improvement Capability dEtermination* - Determinação da Capacidade de Melhoria do Processo de *Software*. Futura norma ISO 15504.
- SPIN - *Software Process Improvement Network* - Grupo de Melhoria do Processo de *Software*. Grupo formado por estudiosos e usuários de métodos, ferramentas e modelos voltados para a melhoria de processos de desenvolvimento de *softwares*.
- STR - *Software Trouble Reports* - Relatório de Problemas de *Software*.
- TQM - *Total Quality Management* - Gerenciamento da Qualidade Total. É uma ferramenta gerencial que possui várias idéias embutidas em seus conceitos: "melhoria contínua", "zero defeitos", "fazer certo pela primeira vez", "os empregados próximos da situação sabem mais como melhorá-la".

LISTA DE FIGURAS

2.1 Visão Geral dos Processos - ISO 12207.....	15
2.2 Métricas Externas - Funcionalidade e Conveniência.....	20
2.3 Métricas Externas - Confiabilidade e Maturidade.....	20
2.4 Métricas Internas – Rastreabilidade.....	20
3.1 Os Cinco Níveis de Maturidade de Processo de <i>Software</i>	24
3.2 - O Diagrama de Trilogia de Juran.....	28
3.3 - Visibilidade dos processos a cada nível de maturidade.....	31
3.4 - A Estrutura do CMM.....	34
3.5 - As KPA's através dos Níveis de Maturidade.....	35
3.6 - Um Exemplo de Prática-Base.....	40
3.7 - A Arquitetura do Modelo Trillium.....	45
3.8 - Avaliação do Processo de <i>Software</i>	49
4.1 - Retrabalho e Custos de Desenvolvimento em função do tamanho do programa.....	56
4.2 - Duração da Programação em função do tamanho do programa.....	57
4.3 - Custo da Qualidade Versus Tempo.....	64
4.4 - Aumento da Produtividade de <i>Software</i> Versus Tempo.....	67
4.5 - Alcançando a Previsibilidade de Projeto.....	68
4.6 - Densidade de Defeito Versus Tempo.....	69

LISTA DE TABELAS

1.1 Gestão de Qualidade.....	3
1.2 Certificação do Sistema da Qualidade.....	4
1.3 Procedimento Para a Qualidade em Software.....	5
2.1 Normas ISO relacionadas à qualidade de <i>software</i>	12
2.2 Processos do Ciclo de Vida do <i>Software</i>	16
2.3 Características da Qualidade de <i>Software</i>	19
3.1 Características Comuns e Práticas-base.....	39
3.2 Áreas de Capacidade Através dos Níveis do Trillium.....	45
3.3 <i>Roadmaps</i> através das áreas de capacidade.....	47
3.4 - Descrição de Categorias de Processo.....	50
3.5 - Descrição dos níveis de capacitação.....	51
4.1 Comparando todas as métricas da melhoria de processo.....	55
4.2 Processos selecionados da ISO 15504.....	79
5.1 Aspectos de Melhoria Versus Estudo de Casos.....	84
5.2 Relação entre Aspectos de Melhoria Apontados no Estudo de Casos..	90
5.3 Outras Relações entre os Aspectos de Melhoria.....	93

GLOSSÁRIO

Alinhamento Organizacional - os indivíduos da organização compartilham de uma mesma visão, cultura e o entendimento dos objetivos de negócio, para que possam executar suas funções de modo mais efetivo.

Benchmarking – “... é a arte de descobrir como e por que algumas empresas podem desempenhar muito mais tarefas do que outras. Podem-se comparar dez diferenças em termos de qualidade, velocidade e desempenho em custos de uma empresa média *versus* outra de classe mundial” (KOTLER, 1998).

Bootstrap - é uma metodologia genérica para estimar a variabilidade em estatística

Ciclo de vida do *software* - obsolescência programada.

Cleanroom - processo de desenvolvimento onde se enfatiza a construção correta do programa. Prioridade 1: prevenir falhas em vez de removê-las. Prioridade 2: fornecer uma certificação estatística da qualidade do *software*.

Código-fonte - programa de computador; conjunto de instruções codificadas através de linguagem de programação.

Conformidade do produto - aspecto de melhoria que observa se um produto de *software* encontra-se em conformidade com o projeto e com as necessidades do usuário.

Cumprimento de Prazos e Custos - aspecto de melhoria que observa a capacidade de um processo em executar projetos dentro dos limites quantitativos "tempo" e "custo".

Densidade de defeito - aspecto de melhoria referente à quantidade de defeitos/milhares de linhas de código-fonte.

Desenvolvedor - profissional relacionado ao desenvolvimento do produto de *software*.

Engenharia de *software* - é um conjunto de disciplinas técnicas e gerenciais que sistematizam o desenvolvimento, a operação, a manutenção e o descarte de produtos de *software*, possuindo qualidade predeterminada, dentro de prazos e custos estimados, com progresso controlado, e operando satisfatória e economicamente em ambientes reais.

Feedback - resposta.

Lead Assessor - esta certificação qualifica um auditor a atuar na avaliação de empresas segundo as normas ISO 9000.

Marketing – “A venda focaliza-se nas necessidades do vendedor; *marketing* nas necessidades do comprador. A venda está preocupada com a necessidade do vendedor transformar seu produto em dinheiro; *marketing* com a idéia de satisfazer às necessidades do consumidor por meio do produto e de um conjunto de valores associados com a criação, entrega e, finalmente, seu consumo” (KOTLER, 1998).

Maturidade - aumento da capacidade do processo de *software*.

Modelo de maturidade da capacidade - procedimentos gerenciais que visam a melhoria e, conseqüentemente, o aumento da capacidade do processo de *software*.

Modelo de SPI - termo que generaliza modelos de melhoria do processo de *software*.

Organização de *software* - empresa que tem como principal atividade o desenvolvimento de *softwares*.

Pacote de *software* - *software* de prateleira, vendido como um produto embalado.

Processo de *software* - toda e qualquer atividade que esteja relacionada ao *software*, desde o primeiro contato com o usuário até a implantação, treinamento e manutenção do *software*.

Produtividade – "... é a relação entre saídas e entradas, onde as entradas são o trabalho, material, capital e utilidades" (JURAN & GRAYNA, 1991).

Produto de *software* - terminologia que define o *software* como produto final de um processo de produção.

Qualidade de *software* - termo que generaliza todo e qualquer esforço para a melhoria do processo e produto de *software*.

Qualidade do processo de *software* - termo que engloba todas as características de qualidade pertinentes a atividades e ao processo de desenvolvimento de *software*.

Qualidade do produto de *software* - termo que engloba todas as características de qualidade de um produto de *software*.

Repetível - é a capacidade, no caso deste trabalho, de repetir experiências de sucesso ocorridas em projetos de *software* anteriores.

Retrabalho - é a repetição de um trabalho em decorrência do defeito de um produto ou falha no processo; aspecto negativo para a melhoria do processo de *software*.

Reuso - técnica utilizada para otimização das atividades do processo de *software*. Exemplo: reutilizar códigos-fonte ou procedimentos de sucesso de outros projetos.

Riscos do SPI - aspecto de melhoria que classifica o grau de risco que um determinado aspecto pode representar à melhoria do processo. Divide-se em: alto, médio e baixo

Software embutido - *software* desenvolvido com o propósito de definir as funções de equipamentos de uso específico, como *paggers*, celulares e outros equipamentos de telecomunicações.

Trillium - Modelo de melhoria do processo de *software* desenvolvido pela Bell Canada, empresa canadense de telecomunicações.

Turnover - rotatividade de mão-de-obra.

GUALAZZI, Guilherme A. S. ***Análise de Modelos de Melhoria no Desenvolvimento de Software***. Santa Bárbara d'Oeste: UNIMEP, 2000. 100p. Dissertação (Mestrado) – Faculdade de Engenharia Mecânica e de Produção, Universidade Metodista de Piracicaba, 2000

ANÁLISE DE MODELOS DE MELHORIA NO DESENVOLVIMENTO DE SOFTWARE

RESUMO:

Este trabalho tem por objetivo a análise de modelos gerenciais de desenvolvimento de *software*, com vistas a permitir que organizações interessadas possam utilizá-los em suas atividades, conseqüentemente, o trabalho organiza um texto preliminar que introduz o tema de forma acessível a estudantes de graduação em engenharia de produção, em ciência da computação e outros cursos similares. São apresentados cinco casos de utilização de modelos de melhoria de desenvolvimento de *software*, relatados através de pesquisas realizadas por três instituições norte-americanas, e duas organizações brasileiras, onde serão analisados diversos aspectos de melhoria referentes a essas utilizações. Os resultados dos casos apresentados indicam que esses modelos contribuem positivamente para a melhoria de processos de desenvolvimento de *software*, principalmente os casos ocorridos nos EUA, demonstrando grandes chances de êxito em empresas nacionais, principalmente, com relação a um dos modelos considerados, o qual é denominado CMM (*Capability Maturity Model*).

PALAVRAS-CHAVE:

Qualidade, *software*, modelo de maturidade, desenvolvimento de *software*.

GUALAZZI, Guilherme A. S. *Análise de Modelos de Melhoria no Desenvolvimento de Software*. Santa Bárbara d'Oeste: UNIMEP, 2000. 100p. Dissertação (Mestrado) – Faculdade de Engenharia Mecânica e de Produção, Universidade Metodista de Piracicaba, 2000

ANALYSIS OF IMPROVEMENT MODELS IN SOFTWARE DEVELOPMENT

ABSTRACT:

This work aims an analysis of managerial models of software development, in order to enable that interested organizations can use them in their activities, consequently, organize a preliminary text that introduce the theme in a accessible way to graduation students in production engineering, in computation science and other similar courses. Five cases of use of improvement models of software development are presented, reported by research accomplished by three North American institutions, and two Brazilian organizations. Several concerning improvement aspects of this cases are analyzed. The results of the presented cases indicate that the models contribute positively to the improvement of software development processes, mainly in the US cases, demonstrating great chances of success in Brazilian companies, mainly those cases related to the CMM (Capability Maturity Model).

KEYWORDS:

Quality, software, maturity model, software development.

1 INTRODUÇÃO

As empresas de qualquer natureza jurídica, quanto aos seus resultados, podem ser classificadas em geradoras de produtos ou prestadoras de serviços. Entretanto, independente dessa tipologia, todas elas se assemelham quanto às suas entradas, necessárias para poderem desenvolver suas atividades específicas. As entradas são caracterizadas por energia, dinheiro, insumos e “informações”, desde dados até conhecimentos, essenciais aos processos de tomada de decisões e como suporte ao processo administrativo, que inclui o planejamento, o controle, a previsão e a coordenação. Nesse sentido, as informações devem ser confiáveis, precisas, discretas e completas, caso contrário comprometem os resultados ou as implicações das decisões que ali ocorrem.

A celeridade dos processos de inovação tecnológica, a conectividade mundial criada pela Internet e o processo *negocial* cada vez mais automático e rápido para *software*, combinado com o elevado conteúdo tecnológico do produto, definem o desafio que enfrenta o *software* brasileiro, cujo problema, centra-se na indução e na manutenção de uma base tecnológica e de negócios de classe mundial, sem o que o perfil da indústria brasileira será sempre muito baixo.

Do ponto de vista tecnológico, ainda não se atingiu a massa crítica que garantiria a visibilidade mundial da qualidade do *software* brasileiro, e isso parece acontecer por uma série de razões. Uma delas é que a quantidade de novas empresas geradas a partir de centros de ensino superior, de ciência e

tecnologia é ainda muito pequeno, e o número de alianças entre empresas e esses centros é irrisório. Mas é o conhecimento do domínio de aplicação, muito mais do que a capacitação em engenharia de *software*, que parece ser o principal vetor de desenvolvimento de uma indústria de *software* de sucesso.

As empresas do setor de *software* parecem esquecer-se de que, para se alcançar a qualidade, é preciso pessoal cultural e tecnicamente qualificado, e parecem não entender que essa qualificação é investimento e não despesa. Autores como BIO (1985) e FELICIANO NETO (1996) já identificavam em suas obras, deficiências relativas a abordagens pouco criteriosas por parte dos desenvolvedores em relação aos usuários.

A qualidade do produto de *software* é a consequência de seu processo. Hoje, existem ferramentas que auxiliam o processo a analisar a melhor forma de estruturar uma empresa para desenvolver seus produtos com boa qualidade. Essas ferramentas são *Modelos de Maturidade da Capacidade de Software*, que atuam como guias para a melhoria. São critérios para medir os níveis de maturidade e o grau de qualidade do processo de desenvolvimento de *software* na empresa.

1.1 A Pesquisa do MCT

Para contextualizar o presente trabalho, foram utilizadas três pesquisas, relativas aos anos de 1995, 1997 e 1999, do Ministério da Ciência e Tecnologia (1995, 1997, 2000), denominadas "Qualidade no Setor de *Software* Brasileiro", contando com a participação de 445 empresas em 1995, 589 em 1997 e, 446 em 1999, onde a informática é fator chave. A síntese desta

pesquisa pode ser observada no Anexo 1. Seguem abaixo os principais aspectos:

A. Quanto à Gestão da Qualidade

Esta seção apresenta uma síntese dos principais esforços na gestão da qualidade em software, quanto à estratégias, planos, uso de recursos e controles.

A Tabela 1.1 apresenta uma síntese dos principais procedimentos para a Gestão da Qualidade, disponibilizada em ordem decrescente, ou seja, a partir dos procedimentos mais utilizados aos menos utilizados.

TABELA 1.1 GESTÃO DA QUALIDADE

FONTE: Ministério da Ciência e Tecnologia (1995, 1997, 2000), adaptado pelo autor

	1995	1997	1999
Inclusão sistemática de metas ou diretrizes para a qualidade nos planos	38,9%	40,4%	42,6%
Coleta sistemática de indicadores da qualidade de produtos e serviços	25,1%	29,5%	-
Elaboração sistemática de planos estratégicos ou planos de metas	21,7%	27%	31,2%
Implantação de programa da qualidade total	11%	18%	26,5%
Apropriação sistemática de custos da qualidade	4%	6%	13,3%

A Tabela 1.1 apresenta apenas percentuais de utilização sistemática dos recursos acima citados. Assim, o percentual restante destes recursos ficam distribuídos entre utilização eventual, em estudo ou não utilização.

Os recursos apontados na Tabela 1.1, são considerados importantes na gestão da qualidade de software. Com exceção de 'Indicadores da Qualidade', que não são apontados na pesquisa 1999, os demais recursos apresentam progressão em sua utilização. Recursos que, mesmo

apresentando-se com baixa utilização, vêm progredindo são 'Programa da Qualidade Total' e 'Custos da Qualidade'.

Das empresas envolvidas com o setor de informática, 8, 45 e 74 empresas, respectivamente, relacionadas às pesquisas de 1995, 1997 e 1999, obtiveram certificação do sistema da qualidade, distribuídas conforme demonstrado pela Tabela 1.2.

TABELA 1.2 CERTIFICAÇÃO DO SISTEMA DA QUALIDADE

FONTE: Ministério da Ciência e Tecnologia (1995, 1997, 2000), adaptado pelo autor

	1995	1997	1999
Empresas certificadas	8	45	74
Certificação ISO 9001	8	35	63
Certificação ISO 9002	1	11	16
<i>Software</i> explicitado no certificado	1	15	39

B. Quanto aos Procedimentos para Qualidade em *Software*

Nesta seção é feito um confronto entre os resultados de 1995, 1997 e 1999, referentes aos procedimentos para qualidade em *software* mais utilizados.

A Tabela 1.3, exibida a seguir, mostra uma síntese das evoluções e involuções de alguns dos principais procedimentos para a qualidade de *software*, nos anos de 1995, 1997 e 1999. Estes procedimentos dizem respeito a métodos de engenharia de *software* utilizados na prevenção e detecção de defeitos, e outras práticas de engenharia de *software*.

A Tabela 1.3 encontra-se classificada em ordem decrescente, ou seja, partindo dos procedimentos mais utilizados aos menos utilizados, onde observam-se baixos índices, nas três pesquisas, para procedimentos como 'Inspeção formal', 'Estimação de confiabilidade', 'JAD', 'Medições da qualidade'

e 'QFD'. O procedimento 'Análise de requisitos' apresentou uma queda significativa na pesquisa de 1999.

TABELA 1.3 PROCEDIMENTOS PARA A QUALIDADE EM SOFTWARE

FONTE: Ministério da Ciência e Tecnologia (1995, 1997, 2000), adaptado pelo autor

Procedimento	1995	1997	1999
Testes de sistema	62,2%	66,6%	46,7%
Testes de campo	58,2%	60,6%	59,9%
Controles de versão	53,7%	55,5%	71,8%
Testes funcionais	48,8%	55,9%	61,7%
Testes de aceitação	47,6%	47,2%	48,1%
Análise de requisitos	47,4%	35,5%	14,3%
Prototipação	46,5%	44,0%	43,9%
Programação orientada a objeto	43,4%	36,7%	43,7%
Reuso de código	37,3%	18,7%	24,4%
Inspeção formal	10,1%	17,0%	14,8%
Estimação de confiabilidade	10,1%	5,4%	6,3%
JAD - <i>Joint Application Design</i>	9,9%	7,8%	8,5%
Gerência de projetos	-	39,9%	42,3%
Medições da qualidade	-	8,1%	12,2%
QFD - <i>Quality Function Deployment</i>	-	1,9%	1,9%

'Gerência de projetos', mesmo não tendo sido abordado na pesquisa de 1995, apresentou um índice razoável de utilização nas pesquisas de 1997 e 1999, tendo ocorrido o mesmo para os demais procedimentos.

D. Quanto ao Atendimento a Clientes

Houve uma redução de 6,8% para 4,8% e, depois, para 4,7%, respectivamente, entre as pesquisas 1995, 1997 e 1999, quanto à empresas que não utilizam estruturas de atendimento e resolução de reclamações.

Quanto ao uso sistemático de dados de pesquisa ou reclamações na revisão de projetos ou na especificação de novos produtos ou serviços, as pesquisas de 1995, 1997 e 1999 apresentaram, respectivamente, os percentuais de 41%, 44% e 44,3%.

1.2. Objetivos

Considerando o exposto, é preciso que as empresas utilizem procedimentos gerenciais voltados à qualidade de *software*.

Assim, o objetivo do trabalho é analisar um método gerencial de desenvolvimento de *software* com vistas a permitir que organizações interessadas possam utilizá-lo em suas atividades.

Como consequência deste objetivo, o trabalho organiza um texto que introduz o tema de forma acessível a estudantes de graduação em engenharia de produção, em ciência da computação e outros cursos similares.

1.3 Estrutura do Trabalho

O Capítulo 2 apresenta conceitos e ferramentas de qualidade, e normas internacionais capazes de auxiliar na qualidade do desenvolvimento de *software*. Já, o Capítulo 3, descreve ferramentas mais específicas para a qualidade de *software*, denominadas modelos de maturidade da capacidade, como possíveis alternativas para a qualidade de *software*. A ênfase, nesse capítulo, é dada ao modelo CMM (*Capability Maturity Model*).

O Capítulo 4 relata experiências na implementação de modelos de melhoria da qualidade de *software* e resultados. Em seguida, no Capítulo 5, são confrontados os principais aspectos de melhoria apontados em cada caso, descritos no Capítulo 4, e analisadas as relações entre estes aspectos.

Finalmente, o Capítulo 6 encerra este trabalho concluindo todo o exposto e apresentando sugestões para estudo futuro.

2 QUALIDADE E SOFTWARE

Numa época em que a qualidade é exigida e observada em todo e qualquer produto e serviço, a produção de *softwares*, ainda que um pouco tarde e em escala ainda reduzida, não poderia ignorar essa tendência. Conforme observado no Capítulo 1, são baixos os índices de atividades que garantam a qualidade dos processos e produtos de *software*.

A qualidade está sempre associada aos conceitos de completeza e correção. A qualidade dos resultados dos sistemas de informação pressupõe a qualidade dos ambientes computacional e organizacional. Enquanto, nos processos industriais, cada agente tem suas operações e técnicas bem delineadas e definidas, o desenvolvimento de *software* depende de "artesãos intelectuais". Na linha de produção industrial, os processos já não dependem mais tanto de quem o faz, mas de como são feitos. Seguindo esse modelo, os objetivos dos programas de qualidade voltados ao desenvolvimento de *softwares* têm se fixado na diminuição da dependência da qualidade em relação ao trabalho desses "artesãos", e na determinação de pontos de controle para avaliar a qualidade do processo durante e depois da sua execução. Neste sentido, este capítulo faz uma revisão de procedimentos de qualidade, tais como TQM, normas, medição da qualidade e ferramentas mais operacionais.

2.1 Ferramentas Gerenciais e Operacionais da Qualidade

A Qualidade só pode ser conhecida quando avaliada. Para isso, é preciso basear-se em dados extraídos de processos e produtos que, quando cruzados entre si, resultam em valiosas informações capazes de auxiliar em tal avaliação. Mas, para que se obtenha bons resultados são necessários procedimentos que os garantam. Nesta seção, serão detalhadas algumas das principais ferramentas gerenciais e operacionais, hoje utilizadas na melhoria contínua de processos, algumas das quais, aplicam-se à avaliação da qualidade de *software*.

2.1.1 TQM - Gerenciamento da Qualidade Total

TQM é um amplo conjunto de procedimentos que aumenta as vantagens competitivas de uma empresa, possibilitando uma melhoria constante de seus produtos e serviços, resultando em clientes mais fiéis que voltam para adquirir mais bens e serviços, definem CORTADA & QUINTELLA (1994). Várias idéias encontram-se embutidas nos conceitos de TQM: "melhoria contínua", "zero defeitos", "fazer certo pela primeira vez", "os empregados próximos da situação sabem mais como melhorá-la".

Ainda, segundo CORTADA & QUINTELLA (1994), existem alguns pontos importantes, em TQM, para se trabalhar na aplicação do zelo pelos clientes e funcionários, tais como o *feedback* da satisfação e de sugestões de clientes e funcionários, compensação, recompensas e avaliações de desempenho, *benchmarking* e educação para vitalidade tecnológica e qualidade. Enfim, TQM preconiza a existência de um plano diretor abrangente para melhorar constantemente a qualidade numa organização, com ênfase na importância da abrangência da abordagem da qualidade a todos os cargos e níveis de uma organização.

O TQM, em empresas de desenvolvimento de *software*, possui o mesmo significado que nas empresas de manufatura e de serviços. PESSÔA (1995) afirma que ao gerenciar "... seus processos, a empresa de *software* necessita se mover na direção da prevenção de defeitos, em oposição à antiga postura de detecção de defeitos, que via a qualidade como função da inspeção final (testes) dos programas".

2.1.2 QFD - Desdobramento da Função Qualidade

Considerando que o investimento na prevenção é melhor do que a despesa com a correção, é fundamental que se procure incorporar e garantir a qualidade de um produto desde a sua concepção e seu projeto. Assim, o QFD, conforme AKAO (1990), é o desdobramento, passo a passo, das funções ou operações que compõem a qualidade do produto. É uma maneira de converter os "requisitos do consumidor" em "características de qualidade do produto" (ou parâmetros do projeto), "características de qualidade do processo" e em "métodos de controle do processo e da qualidade".

O QFD ajuda a identificar exigências do usuário que não foram tratadas pelo desenvolvedor de *software*. Além de destacar tais omissões, o QFD documenta também exigências que são avaliadas pelo usuário e recebem pouca atenção de tais características. Trabalhos direcionados para a aplicação do QFD no desenvolvimento de *software* são descritos por BARNETT & RAJA (1995), ELBOUSHI & SHERIF (1997), FUNG, LAO & IP (1999) e, KALARGEROS & GAO (1998).

2.1.3 Qualidade em Serviços

As companhias que fornecem bons serviços são capazes de prestar serviços efetivamente porque aplicam princípios básicos da qualidade em serviços. Esses princípios, descritos por DENTON (1990), é que são o cerne dos serviços com qualidade. São eles:

- 1) Visão gerencial;
- 2) Desenvolver um nicho estratégico;
- 3) A alta administração deve demonstrar apoio;
- 4) Entender o seu negócio;
- 5) Aplicar os fundamentos operacionais;
- 6) Entender, respeitar e monitorar o cliente;
- 7) Usar tecnologia apropriada;
- 8) A necessidade de inovar;
- 9) Contratar as pessoas certas;
- 10) Fornecer treinamento com base no perfil;
- 11) Definir padrões, medir desempenho e agir;
- 12) Estabelecer incentivos.

KAPLAN (1996) defende que a forma de manter estes princípios está relacionada ao uso do TQM. Quanto ao foco no cliente, o QFD demonstra ser uma excelente ferramenta de manutenção.

Sendo a produção de *software* uma prestação de serviço, é importante observar a manutenção dos princípios acima citados no decorrer do processo de desenvolvimento do *software*.

2.1.4 As Ferramentas da Qualidade

Afim de garantir a qualidade do desenvolvimento do produto é imprescindível a utilização de metodologias e técnicas que permitam uma melhor organização de idéias e fatos, dando maior objetividade ao processo de obtenção de dados e à análise que se fizer necessária.

Assim, BRASSARD (1985) e OLIVEIRA (1996) defendem que é fundamental que se tenha o domínio sobre a aplicação de ferramentas como Fluxograma, Diagrama de Causa-Efeito, Folha de Verificação, Diagrama de Pareto, Histograma, Diagrama de Dispersão, Capabilidade do Processo, Gráfico de Controle (ou Carta de Controle), assim como outros gráficos encontrados facilmente em planilhas eletrônicas.

Não existe a ferramenta totalmente capaz de solucionar todos os problemas. Caberá aos profissionais a arte de combiná-las, criando novas abordagens e possibilidades. A qualidade depende muito mais de pessoas comprometidas com o desenvolvimento de todas as suas potencialidades, do que de um conjunto de técnicas.

2.2 Sistemas da Qualidade Baseados nas Normas ISO

De acordo com MIGUEL (2000), a série ISO 9000 é composta por normas que trazem em sua essência a gestão da qualidade em projetos/desenvolvimento, produção, instalação, assistência técnica, inspeção e ensaios, exigindo diretrizes sólidas e sistêmicas. As normas têm caráter genérico podendo ser introduzidas em diversos ramos de atividades.

FERNANDES (1995) descreve que a série ISO 9000 fornece orientações sobre como estruturar o Sistema da Qualidade da empresa, de forma que a mesma possa evidenciar a qualidade de seu processo de produção de bens e serviços, demonstrando assim que é capaz de suprir as necessidades dos clientes conforme requisitos previamente estabelecidos.

A Tabela 2.1 fornece uma visão geral das principais normas envolvidas com produtos e processos de desenvolvimento de software.

TABELA 2.1 - NORMAS ISO RELACIONADAS À QUALIDADE DE SOFTWARE.

FONTE: Qualidade de *Software*, 1998.

Norma	Descrição
ISO 9126-1 (1997) ISO 9126-2 (1997) ISO 9126-3 (1996)	Características da qualidade de produtos de <i>software</i> .
ISO 14598 (1996)	Guias para avaliação de produtos de <i>software</i> , baseados na utilização prática da norma ISO 9126.
ISO 12119 (1994)	Características de qualidade de pacotes de <i>software</i> .
ISO 12207-1 (1994)	Processo do ciclo de vida do <i>software</i> . Norma para a qualidade do processo de desenvolvimento de <i>software</i> .
NBR ISO 9001 (1994)	Sistemas de qualidade - modelo para garantia de qualidade em projeto, desenvolvimento, instalação e assistência técnica.
NBR ISO 9000-3 (1993)	Gestão e garantia de Qualidade. Aplicação da norma ISO 9001 ao processo de desenvolvimento de <i>software</i> .
SPICE (1998)	Futura norma ISO 15504 para a avaliação de processo de desenvolvimento de <i>software</i> .

A seguir, serão descritas as normas ISO 9000-3 (1993) e ISO 12207 (1994), relacionadas ao processo de desenvolvimento de *software*, foco deste trabalho. A descrição dessas duas normas tem o objetivo de auxiliar no entendimento de uma análise comparativa, assim como, sua relação com modelos de melhoria de produção de *software*, descritos no capítulo seguinte.

2.2.1 ISO 9000-3 (1993) – Diretrizes para a aplicação da ISO 9001 (1994)

Em junho de 1993 foi criada a norma ISO 9000-3 (1993) com diretrizes para aplicação da ISO 9001 (1994) ao desenvolvimento, fornecimento e manutenção de *software*. Esta norma se espelha nos itens da ISO 9001 (1994) fazendo a necessária adaptação. Para cada item da ISO 9001 existe um correspondente na ISO 9000-3 que o detalha e o adequa ao *software*.

Destina-se a fornecer orientação quando um contrato entre duas partes exigir a demonstração da capacidade de um fornecedor em desenvolver, fornecer e manter produtos de *software*.

Suas diretrizes destinam-se a descrever os controles e métodos sugeridos para a produção de *software* que atendam aos requisitos do comprador, evitando-se não conformidades em todos os estágios, desde o desenvolvimento até a manutenção.

As diretrizes desta norma (ISO 9000-3, 1993) são aplicáveis em situações contratuais para produtos de *software*, quando:

- a) O contrato exigir, especificamente, esforço de projeto, e os requisitos do produto forem indicados principalmente em termos de desempenho, ou precisarem ser estabelecidos;
- b) A confiança puder ser obtida através da demonstração adequada da capacidade de desenvolvimento, fornecimento e manutenção de um determinado fornecedor.

TSUKUMO *et al.* (1996) descreve que esta norma é dividida em três partes principais:

- *Estrutura*: descreve aspectos organizacionais relacionados ao sistema de qualidade, relatando responsabilidades e ações relativas à qualidade que devem ser tomadas por fornecedor e comprador.

- *Atividades do ciclo de vida*: descreve as atividades de desenvolvimento de *software*, definindo que o desenvolvimento de *software* deve ser feito conforme cada modelo de ciclo de vida e que as atividades relacionadas à qualidade devem ser planejadas e implementadas de acordo com a natureza deste modelo.
- *Atividades de suporte*: descreve as atividades que apoiam as atividades do ciclo de vida de desenvolvimento.

TSUKUMO *et al.* (1996) afirma que mesmo definindo e padronizando este conjunto de atividades, a melhoria de processo de *software* não é enfocada de maneira direta pela série ISO 9000. Esta norma apenas recomenda que ações corretivas e preventivas sejam tomadas. Já em outros modelos, como o CMM e o *Trillium*, por exemplo, voltados para a melhoria de processo de *software* esta questão é explicitamente enfocada.

2.2.2 ISO 12207 - Processos do Ciclo de Vida do *Software*

Esta norma estabelece os processos, atividades e tarefas a serem aplicados durante a aquisição, fornecimento, desenvolvimento, operação e manutenção de *software*. Conforme descrito por TSUKUMO *et al.* (1996), a Norma apresenta uma definição abrangente e orienta sua adaptação aos projetos de *software* implementados numa organização. Sua estrutura foi concebida de maneira que fosse flexível, modular e pudesse ser adaptável à necessidade do produto a ser gerado. Para isto, ela baseia-se em dois princípios básicos: modularidade e responsabilidade. Entende-se modularidade como uma característica existente na definição dos processos, que possuem o mínimo de acoplamento e o máximo de coesão. Define-se responsabilidade pela designação de uma parte envolvida para executar um processo, facilitando a aplicação da norma em um projeto, onde várias pessoas podem estar

legalmente envolvidas.

MACHADO *et al.* (1997) descreve que a estrutura da norma é composta por um conjunto de processos, atividades e tarefas que pode ser adaptado de acordo com os projetos de *software*. A norma não especifica como executar estes processos, deixando esta tarefa para a organização. Uma das formas para isso é através da adoção de modelos de maturidade, descritos no capítulo seguinte.

Esta Norma define dezessete processos do ciclo de vida de *software* e os organiza em três classes: processos primários, processos de apoio e processos organizacionais. Cada classe contém os processos definidos e os possíveis usuários, conforme ilustra a Figura 2.1.

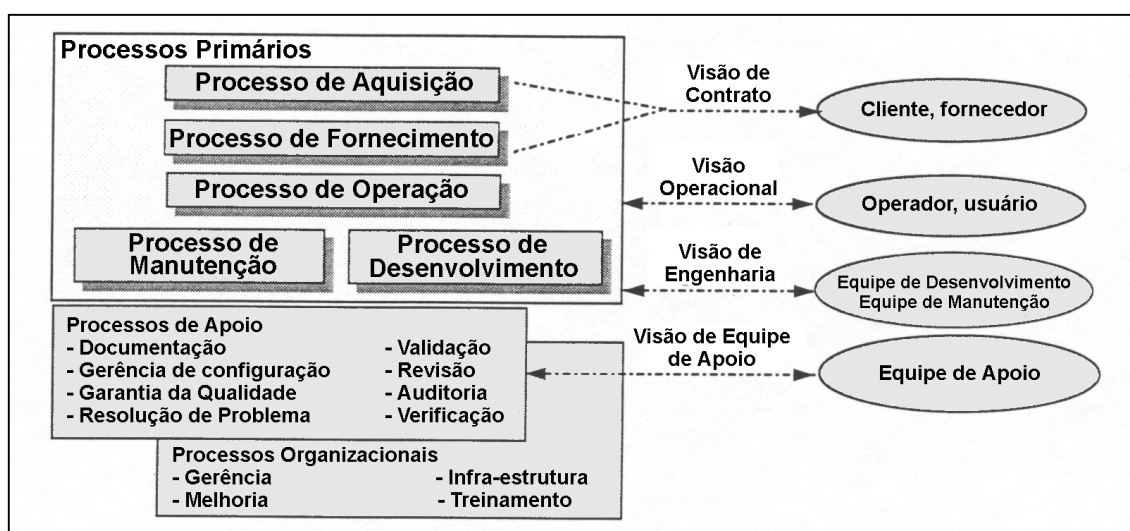


FIGURA 2.1 - VISÃO GERAL DOS PROCESSOS - ISO 12207.

FONTE: TSUKUMO, A.N. *et al.* (1996).

A Tabela 2.2 descreve as atividades e tarefas que podem ser executadas durante o ciclo de vida do *software* em processos fundamentais, processos de apoio e processos organizacionais, ilustrados pela Figura 2.1. De acordo com TSUKUMO *et al.* (1996), a importância desta norma está em estabelecer uma estrutura de classificação de processos, normalizando a terminologia.

TABELA 2.2 PROCESSOS DO CICLO DE VIDA DO SOFTWAREFONTE: GRAHL, E. A. *et al.* (1997, p. 34), adaptado pelo autor.

Classes	Processos do Ciclo de Vida do Software
Processos Primários: são 5 processos que atendem as partes fundamentais (pessoa ou organização) durante o ciclo de vida do <i>software</i> .	Processo de Aquisição: define as atividades de uma organização que adquira um sistema, produto de <i>software</i> ou serviço de <i>software</i> .
	Processo de Fornecimento: define as atividades de uma organização que forneça o sistema, produto de <i>software</i> ou serviço de <i>software</i> ao adquirente.
	Processo de Desenvolvimento: define as atividades da organização que define e desenvolve o produto de <i>software</i> .
	Processo de Operação: define as atividades do operador, provendo serviço de operação de um sistema computacional no seu ambiente de funcionamento.
	Processo de Manutenção: define as atividades do mantenedor, provendo os serviços de manutenção do produto de <i>software</i> , isto é, gerenciando as modificações havidas para mantê-lo atualizado e em perfeita operação, até sua descontinuação.
Processos de Apoio: auxiliam um outro processo como parte integrante, com um propósito distinto.	Processo de Documentação: define as atividades para registro da informação produzida por um processo de ciclo de vida.
	Processo de Gerência de Configuração: define as atividades de gerência de configuração.
	Processo de Garantia da Qualidade: define as ações para garantir objetivamente que os produtos e processos de <i>software</i> estejam em conformidade com seus requisitos especificados e aderem aos planos estabelecidos.
	Processo de Verificação: define as atividades para verificação dos produtos de <i>software</i> , em profundidade variável, dependendo do projeto de <i>software</i> .
	Processo de Validação: define as atividades para validação dos produtos de <i>software</i> do projeto de <i>software</i> .
	Processo de Revisão Conjunta: define as atividades para avaliação da situação e dos produtos de uma atividade. Pode ser empregado por qualquer uma das duas partes, onde uma delas (parte revisora) revisa a outra parte (parte revisada) em um fórum conjunto.
	Processo de Auditoria: define as atividades para determinar a conformidade com requisitos, planos e contrato. Pode ser empregado por qualquer das duas partes, onde uma (parte auditora) audita os produtos de <i>software</i> ou atividades da outra (parte auditada).
	Processo de Resolução de Problema: define um processo para análise e remoção dos problemas (incluindo não-conformidades), independente da sua natureza ou origem, identificados durante o desenvolvimento, a operação, a manutenção ou outros processos.
Processos Organizacionais: são empregados para estabelecer e implementar uma estrutura e melhorar continuamente esta estrutura e os processos.	Processo de Gerência: define as atividades básicas da gerência, incluindo gerência de projeto, durante um processo de ciclo de vida.
	Processo de Infra-estrutura: define as atividades básicas para o estabelecimento da estrutura de apoio de um processo de ciclo de vida.
	Processo de Melhoria: define as atividades básica que a organização executa para estabelecer, medir, controlar e melhorar seu processo de ciclo de vida.
	Processo de Treinamento: define as atividades para prover pessoal adequadamente treinado.

2.3 Qualidade de *Software*

A demanda da vida moderna fez com que usuários de *software* tornassem-se mais exigentes, obrigando o uso de ferramentas, por parte dos desenvolvedores, que assegurassem a qualidade dos produtos e processos de *software*.

Apesar de alguns especialistas em qualidade afirmarem o contrário, é possível medir o produto e o processo de *software*, embora alguns dados de grande relevância sejam subjetivos. GIL (1995) aponta Indicadores de Qualidade (IQ) em informática capazes de medir esforços de qualidade, quantificar custos e projeções da qualidade, com relativa precisão.

2.3.1 Engenharia, Qualidade de Processo e Produto de *Software*

A Engenharia de *Software* surgiu com o objetivo de melhorar a qualidade do produto, propor modelos, métodos e técnicas para aplicação nas diversas fases de desenvolvimento do *software*. A avaliação da qualidade de *software*, nas duas visões (processo e produto), se insere nesse esforço.

Avaliar e julgar processo e produto são abordagens necessárias e complementares. TSUKUMO *et al.* (1997) define que "a visão de processos de *software* propicia uma estrutura para a harmonização das várias disciplinas da Engenharia de *Software*, englobando não apenas as atividades de desenvolvimento mas todas as atividades necessárias para a sua produção, incluindo avaliação".

A qualidade de *software* é determinada através da qualidade dos processos utilizados para o seu desenvolvimento. Assim, esta melhoria

contribui com a elaboração de modelos de definição, avaliação e melhoria de processos de *software*.

De acordo com TSUKUMO *et al.* (1997), as características de um *software*, são definidas pelos requisitos do usuário, a fim de permitir o exame de seu atendimento. Por outro lado, AZUMA (1996) define que a qualidade de um produto é avaliada baseada numa lista de características pré-definidas, conforme mostrado na Tabela 2.3. Essa lista fornece a base para especificar exigências de qualidade e avaliar a qualidade de um produto. Assim, avaliar a qualidade de um *software* é verificar o quanto os requisitos são atendidos.

Um processo de avaliação para o exame sistemático da qualidade do produto de *software* pode ser encontrado na Série ISO 14598 (1996), que utiliza como referência, em uma de suas partes, a Norma ISO 9126-1 (1997).

Segundo AZUMA (1996), uma característica de qualidade é uma visão específica de qualidade e, um atributo pode ser considerado como uma propriedade mensurável abstrata ou física de uma entidade. Medição é um processo para atribuir um valor a um atributo de certa entidade; medida é um valor atribuído a um resultado de medição e; métrica é uma escala, associada a regras e métodos, a ser aplicada para um processo de medição.

TABELA 2.3 - CARACTERÍSTICAS DA QUALIDADE DE SOFTWARE

FONTE: TSUKUMO, A. *et al.* (1997, p. 186), adaptado pelo autor.

Características de Qualidade	Subcaracterísticas
Funcionalidade - evidencia que o conjunto de funções atendem às necessidades explícitas e implícitas para a finalidade a que se destina o produto	Adequação - presença de conjunto de funções e sua apropriação para as tarefas
	Acurácia - geração de resultados ou efeitos corretos
	"Interoperabilidade" - capacidade de interagir com outros sistemas
	Conformidade - estar de acordo com normas, convenções, regulamentações
	Segurança de Acesso - capacidade de evitar acesso não autorizado a programas e dados
Confiabilidade - evidencia que o desempenho se mantém ao longo do tempo e em condições estabelecidas	Maturidade - frequência de falhas
	Tolerância a falhas - manter o nível de desempenho em caso de falha
	"Recuperabilidade" - capacidade de restabelecer e restaurar dados após falha
"Usabilidade" - evidencia a facilidade para a utilização do <i>software</i>	"Inteligibilidade" - facilidade de entendimento dos conceitos utilizados
	"Apreensibilidade" - facilidade de aprendizado
	Operacionalidade - facilidade de operar e controlar a operação
Eficiência - evidencia que os recursos e os tempos envolvidos são compatíveis com o nível de desempenho requerido para o produto	Comportamento em relação ao tempo - tempo de resposta, de processamento
	Comportamento em relação a recursos - quantidade de recursos utilizados
"Manutenibilidade" - evidencia que há facilidade para correções, atualizações e alterações	"Analisabilidade" - facilidade de diagnosticar deficiências e causas de falhas
	"Modificabilidade" - facilidade de modificação e remoção de defeitos
	Estabilidade - ausência de riscos de efeitos inesperados
	"Testabilidade" - facilidade de ser testado
Portabilidade - evidencia que é possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação	Adaptabilidade - capacidade de ser adaptado a ambientes diferentes
	Capacidade para ser instalado - facilidade de instalação
	Conformidade - acordo com padrões ou convenções de portabilidade
	Capacidade para substituir - substituir outro <i>software</i>

Um valor obtido por medição, tal como as escalas LOC (*Lines Of Code* - Linhas de Código) ou FP (*Function Points* - Pontos de Função), é uma medida do tamanho do programa. A regra e método para contar o LOC do programa é uma métrica.

Existem três tipos de métricas descritas por AZUMA (1996):

1) *Métricas externas*: são aquelas usadas para medir características externas.

Alguns exemplos são mostrados nas Figuras 2.2 e 2.3.

$$\text{Proporção de Mudança da Especificação Funcional} = \frac{\text{O Número de Funções Alteradas}}{\text{O Número de Funções Implementadas}}$$

$$\text{Proporção de Pedido de Mudança} = \frac{\text{O Número de Pedidos de Usuários por Mudança}}{\text{Escala de Produto (LOC ou O Número de Funções)}}$$

FIGURA 2.2 - MÉTRICAS EXTERNAS - FUNCIONALIDADE E CONVENIÊNCIA.

FONTE: AZUMA, M. (1996, p. 150).

$$\text{Tempo Significativo para Falha} = \frac{\text{Tempo de Operação Total}}{\text{O Número de Falhas Observadas}}$$

$$\text{Densidade de Erro do Produto} = \frac{\text{O Número de Erros no Produto}}{\text{Volume do Produto}}$$

FIGURA 2.3 - MÉTRICAS EXTERNAS - CONFIABILIDADE E MATURIDADE.

FONTE: AZUMA, M. (1996, p. 150).

2) *Métricas internas*: são aquelas usadas para medir características internas.

Alguns exemplos são mostrados na Figura 2.4.

$$\text{Proporção de Especificações de Função Rastreável} = \frac{\text{O Número de Funções em Especificação até a Fase Atual}}{\text{O Número de Funções em Especificação até a Fase Anterior}}$$

$$\text{Proporção de Item Verificável} = \frac{\text{O Número de Itens Verificáveis}}{\text{O Número de Itens Verificados}}$$

FIGURA 2.4 - MÉTRICAS INTERNAS - RASTREABILIDADE.

FONTE: AZUMA, M. (1996, p. 151).

3) *Métricas básicas*: em muitos casos, quando uma medida é usada para avaliação ou seleção, ela deve ser normalizada. Como uma característica de qualidade é propriedade abstrata, muitas medidas são indiretas (envolve a medição de um ou mais atributos), como uma proporção, que é derivada de medidas de atributos. Existem, ainda, algumas medidas que são frequentemente usadas para compor várias medidas indiretas. Essa classe de medidas é chamada de medidas básicas, e as métricas correspondentes, de métricas básicas. LOC é um exemplo de métricas básicas.

As visões de processo e produto são necessárias e complementares, conclui TSUKUMO *et al.* (1997), pois se o processo dá uma expectativa de geração de produtos melhores, não se tem a garantia da qualidade do produto porque sempre há fatores imponderáveis e imprevisíveis que escapam ao controle do processo de produção e que podem afetar o processo final. Mais ainda, sendo o desenvolvimento de *software* concentrado em atividades de projeto, está mais sujeito a erros e fatores imponderáveis.

Conclui, ainda, que as duas visões objetivam garantir a qualidade do *software* e ambas interferem no processo de desenvolvimento, realimentando-o com os resultados obtidos.

2.4 Considerações Finais

Considerando-se que "produto de *software*" é resultado de uma produção intelectual, que merece e deve ser controlada, mensurada e quantificada, abordou-se neste capítulo, ferramentas gerenciais e operacionais que apresentam relação direta com o processo de produção de *software*.

Algumas destas ferramentas descritas, tradicionalmente, são utilizadas em outros processos de produção, como o TQM, o QFD e as sete ferramentas da qualidade. Outros controles, também utilizados tradicionalmente em outros processos de produção, também mereceram destaque. Exemplo desta situação diz respeito a controle dos custos da qualidade que, segundo JURAN & GRAYNA (1991), são utilizados para evidenciar a importância interna da qualidade e vendê-la aos gerentes e diretores da companhia, os quais têm constante preocupação com a administração de recursos financeiros. O uso desse controle de custos da qualidade poderá ser observado no próximo capítulo em conjunto com um dos modelos de melhoria descritos, o CMM.

Conforme descrito anteriormente, as normas ISO 9000-3 (1993) e ISO 12207-1 (1994) são descritas neste capítulo por relacionarem-se com modelos de melhoria do processo de produção de *software*, descritos no capítulo seguinte.

Também, a qualidade em serviços, processos e produtos, foram abordados com a finalidade de facilitar a compreensão e interpretação das documentações e resultados de implementações destes mesmos modelos de melhoria. Tal abordagem justifica-se pelo fato de a produção de *software* ser uma prestação de serviço, na qual se deve observar a manutenção dos princípios da qualidade em serviços no decorrer do processo de desenvolvimento do *software*.

3 MODELOS DE MATURIDADE DA CAPACIDADE DO PROCESSO DE *SOFTWARE*

Um modelo de maturidade classifica o sistema da qualidade de uma equipe ou empresa produtora de *software*, com base na capacidade que ela possui em seu processo de desenvolvimento (MCT, 1995). Modelos de maturidade podem abranger o desenvolvimento de *software* e o pessoal envolvido, a aquisição de recursos, engenharia de sistemas, desenvolvimento de produtos integrados, enfim, o processo de *software*.

Os modelos mais importantes de que se tem conhecimento são o CMM (*Capability Maturity Model*), a metodologia *Bootstrap*, a engenharia de *software Cleanroom*, o PSP (*Personal Software Process*), o Trillium e o SPICE (*Software Process Improvement Capability Determination*). No entanto, serão abordados neste capítulo, os modelos CMM, por se tratar do foco de estudo deste trabalho, Trillium, por se tratar de um modelo com finalidade específica, e SPICE, por se tratar de uma generalização do modelo CMM.

Este capítulo apresenta, inicialmente, o modelo desenvolvido pelo Instituto de Engenharia de *Software*, da *Carnegie Mellon University* - o CMM. Este modelo é tido como foco de estudo deste trabalho: (1) por ser o primeiro sistema desenvolvido, (2) para servir de referência para uma análise comparativa entre modelos derivados, apresentados em seguida e, (3) para servir de referência para os estudos de casos.

Daqui em diante, será utilizado o termo "modelo de SPI" (*Software Process Improvement*) - termo que generaliza modelos de maturidade da capacidade, ou melhoria, do processo de *software*.

3.1 CMM - O Modelo de Maturidade da Capacidade

Nesta seção será descrito o Modelo de Maturidade da Capacidade, que em inglês é denominado *Capability Maturity Model* (CMM), como será referido daqui em diante. O CMM é tido como modelo referencial para outros modelos de maturidade, estes, adaptados a outros fins específicos, como organizações, processos ou produtos. Este modelo é tomado, neste trabalho, como objeto principal de estudo.

Em novembro de 1986, o *Software Engineering Institute* (SEI), da *Carnegie Mellon University*, abrevia-se CMU/SEI (1997), começou a desenvolver uma estrutura de maturidade de qualidade que ajudaria as organizações a melhorarem seus processos de *software*. Este esforço foi iniciado em resposta a um pedido para prover o governo federal dos Estados Unidos com um método para avaliar a capacidade de suas contratadas para o desenvolvimento de *softwares*.

O CMU/SEI (1997) conclui que o CMM é "uma aplicação dos conceitos de gerenciamento de processo do TQM para *software*. O TQM pode ser definido como uma aplicação de métodos quantitativos e recursos humanos para melhorar os materiais e serviços fornecidos como *inputs* à uma organização e todos os processos dentro da organização. A meta do TQM é satisfazer as necessidades do cliente, agora e no futuro" (ver item 2.1.1).

Ainda, segundo o CMU/SEI (1997), quando se realiza a melhoria do processo de *software*, esse esforço passa a permear todo o contexto da organização, alinhando-se às necessidades de negócio e com qualquer esforço

do TQM já existente. Este alinhamento terá um efeito crítico sobre o sucesso do esforço de melhoria do processo de *software*.

Conceitualmente, o CMM é um modelo destinado a orientar a construção de uma série de ferramentas úteis na melhoria do processo de *software*, inclusive de um questionário de maturidade. O CMM proporciona uma estrutura para organizar passos evolutivos em cinco níveis sucessivos de maturidade. Estes cinco níveis definem uma escala ordinal para medir e avaliar o grau de maturidade atingido pelo processo de *software* de uma organização. Os níveis também ajudam-na a priorizar seus esforços de melhoria contínua.

Cada nível inclui um conjunto de metas de processo que, quando satisfeitas, resultam em um aumento da capacidade do processo de *software* da organização (*maturidade*). Organizando o CMM nos cinco níveis mostrados na Figura 3.1, priorizam-se ações de melhoria para aumentar a maturidade do processo de *software*, na medida em que maior grau de maturidade é obtido pela organização a cada novo passo.

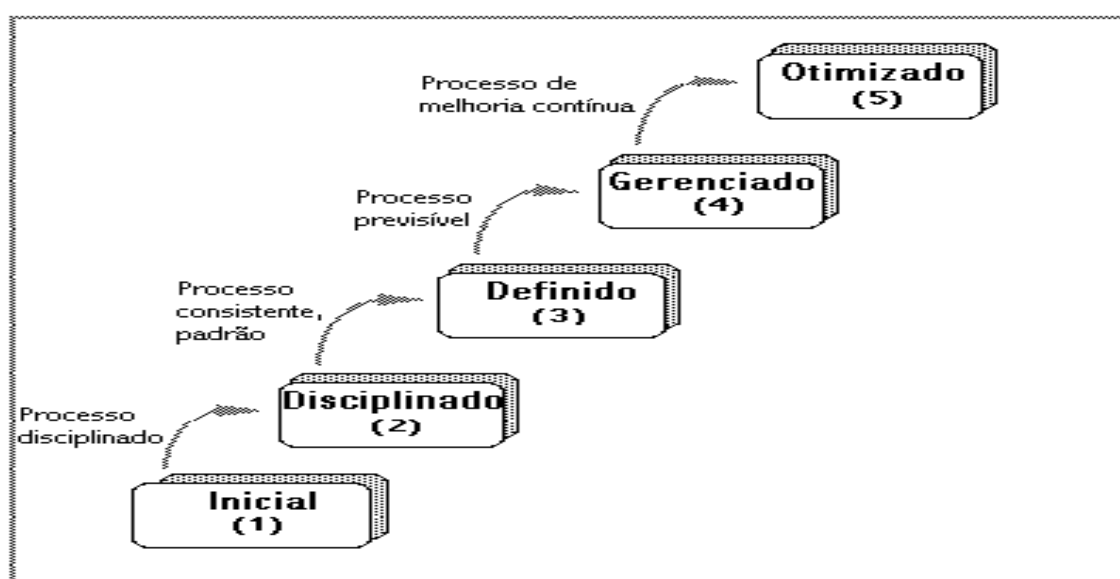


FIGURA 3.1 - OS CINCO NÍVEIS DE MATURIDADE DE PROCESSO DE SOFTWARE.
FONTE: CARNEGIE MELLON UNIVERSITY/Software Engineering Institute (1997, p. 16).

Os níveis de maturidade de 2 a 5 referem-se às atividades executadas pela organização para estabelecer ou melhorar o processo de *software*. A caracterização comportamental de Nível 1 é incluída para estabelecer uma base de referência para a avaliação das melhorias de processo obtidas em cada novo nível.

3.1.1 Nível 1 - O Nível *Inicial*

Neste nível, a organização não proporciona ambiente estável para desenvolvimento e manutenção de *software*. Durante as crises, procedimentos planejados são abandonados. Os projetos limitam-se a codificar e testar o produto. O sucesso depende de gerentes experientes e de uma equipe de *software* amadurecida.

Prazo, custo, funcionalidade e qualidade de produto são geralmente impossíveis de prever. O desempenho depende da capacidade dos indivíduos e varia com suas habilidades inatas, conhecimento e motivações.

As organizações, neste nível, são freqüentemente caracterizadas como tendo processos parcialmente organizados ou até mesmo caóticos. Elas desenvolvem seus produtos de trabalho nem sempre dentro do orçamento e cronograma fixados. O sucesso no Nível 1 depende da competência das pessoas da organização.

3.1.2 Nível 2 - O Nível *Disciplinado*

Neste nível, são estabelecidas as políticas para administrar e implementar os projetos. O gerenciamento de novos projetos são baseados em

experiências obtidas do *feedback* de projetos similares. O objetivo, no Nível 2, é institucionalizar o gerenciamento de projetos de *software*, permitindo que a organização repita práticas de sucesso desenvolvidas em projetos anteriores.

O Nível 2 provê a estrutura para o Nível 3 porque enfoca a gerência, que age para melhorar seus processos antes de tentar resolver assuntos técnicos e organizacionais, típicos do Nível 3.

Resumindo, este nível utiliza as melhores práticas existentes.

3.1.3 Nível 3 - O Nível *Definido*

No Nível Definido, o processo de desenvolvimento e manutenção de *software* pela organização é padronizado e documentado. Um programa de treinamento para toda a empresa é implementado a fim de assegurar que os funcionários e gerentes tenham os conhecimentos e as habilidades exigidas para cumprir seus respectivos papéis. O padrão de procedimentos da organização, até então em uso, é adaptado para responder às novas exigências específicas de cada novo projeto.

Um processo de *software* definido contém um coerente, integrado e bem definido conjunto de processos de engenharia e gerenciamento de *software*. A capacidade de processo do Nível 3 pode ser assumida como padrão e consistente, quando as atividades de gerenciamento e engenharia de *software* são estáveis e podem ser repetidas.

As práticas de sucesso do Nível 2 são, no Nível 3, documentadas e padronizadas, e deverão compor o processo de *software* padrão, ou "definido", da organização. A integração, neste caso, significa que as saídas de uma

tarefas fluem suavemente para as entradas da próxima. Quando há incompatibilidade entre tarefas, elas são identificadas e tratadas na fase de planejamento, ao invés da fase de execução do processo.

Resumindo, este nível documenta, padroniza e integra o processo de *software*.

3.1.4 Nível 4 - O Nível Gerenciado

No Nível Gerenciado, a organização estabelece metas de qualidade quantitativas para produtos e processos. Produtividade e qualidade mensuráveis são medidas importantes como parte de um programa de avaliação organizacional. Processos de desenvolvimento de *software* são instrumentalizados com medidas bem definidas e consistentes, que estabelecem a estrutura quantitativa para avaliá-los. Neste nível, a organização alcança o controle sobre seus produtos e processos, estreitando as variações de seus desempenhos para dentro de limites aceitáveis.

A capacidade de processo de *software* pode ser resumida como previsível porque o processo é medido e opera dentro de limites quantitativos.

Cabe citar aqui, que o uso das sete ferramentas da qualidade podem e devem ser utilizadas. Um exemplo desse uso pode ser observado no capítulo seguinte, no relato dos resultados da implementação do modelo CMM. Outros modelos também possuem atividades de medição de indicadores e, por sua vez, cabe-lhes o uso destas ferramentas.

Resumindo, este nível realiza a mensuração qualitativa e quantitativa do processo de *software*.

3.1.5 Nível 5 - O Nível *Otimizado*

Neste nível, são identificadas fraquezas com o propósito de prevenir a ocorrência de defeitos. Nele, são identificadas inovações que exploram a melhor prática de engenharia de *software* e transferidas ao longo da organização.

Resumindo, este nível visa a melhoria contínua do processo de *software*.

3.2 Entendendo os Níveis Gerenciado e Otimizado

Muitas características de Níveis 4 e 5 estão baseadas nos conceitos de controle estatístico de processo (CMU/SEI, 1997), como ilustra a Figura 3.2. O Diagrama de Trilogia de Juran (JURAN & GRZYNA, 1991) ilustra os objetivos primários da gerência de processo.

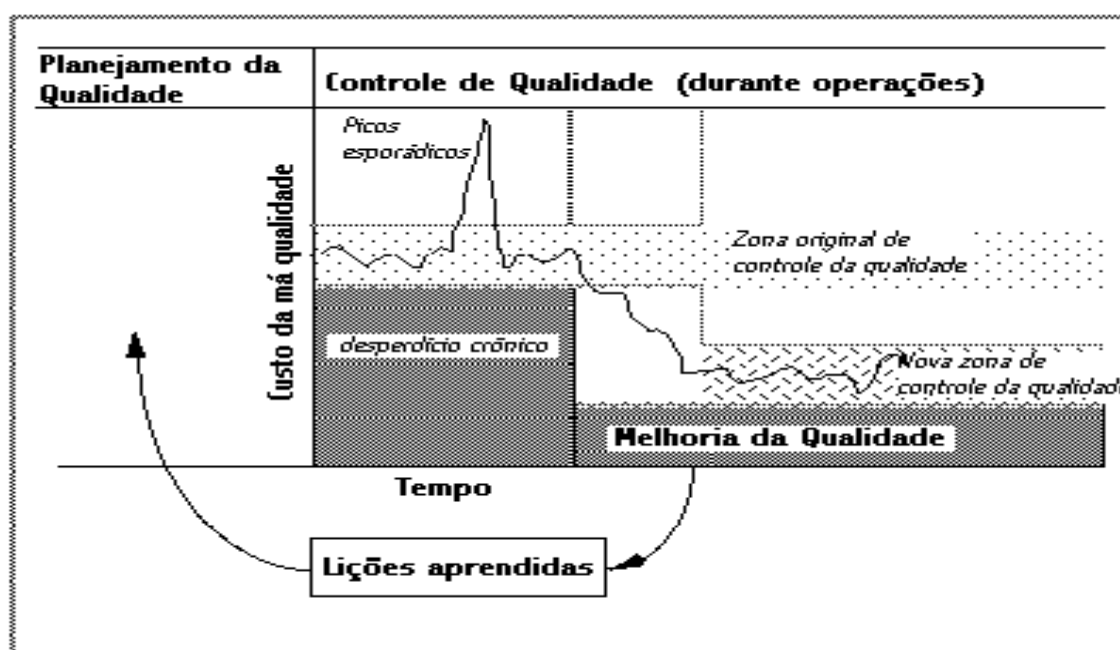


FIGURA 3.2 - O DIAGRAMA DE TRILOGIA DE JURAN.

FONTE: JURAN, J. M. & GRZYNA, F. M. (1991, p. 28), adaptado pelo autor.

Como em outros modelos de qualidade, o propósito do planejamento de qualidade de *software* é prover os desenvolvedores com os meios adequados para a obtenção de produtos que satisfaçam as necessidades do cliente. Nas condições usuais, imperantes nas organizações que não dispõem de tais modelos, as forças operacionais produzem o produto, mas algum tipo de retrabalho pode ser exigido devido a deficiências de planejamento e execução. O “controle de qualidade” é mais do tipo corretivo ou é realizado para prevenir coisas piores. Picos esporádicos de custo da má qualidade no processo, como mostrado na Figura 3.2, representam o início de atividades conflituosas. No controle corretivo, o desperdício crônico é que provê as informações para melhoria da qualidade.

Uma das responsabilidades da gerência, caracterizadas por JURAN & GRAYNA (1991), é o controle do processo, focado no Nível 4. O processo é gerenciado de forma a operar estavelmente dentro de uma zona de controle de qualidade. Há, inevitavelmente, um pouco de desperdício crônico e podem haver picos de má qualidade nos resultados medidos, que precisam ser controlados, mas o sistema total é geralmente estável, porque o propósito de controlar as causas – e não os efeitos, entra em jogo. Devido ao processo ser estável e medido, quando alguma circunstância excepcional acontece, a causa da variação pode ser rapidamente identificada e tratada.

Outra responsabilidade da gerência é a melhoria contínua do processo, enfocada no Nível 5. Uma nova orientação é estabelecida para reduzir o desperdício crônico. As lições aprendidas a cada melhoria são aplicadas no planejamento de processos futuros. Há desperdício crônico, mas

simplesmente devido a variações casuais, ainda não previstos. Mas o desperdício é sempre inaceitável. Então, esforços organizados para removê-los resultam em melhoria do processo, atacando "causas comuns" de ineficiência, e conseguindo impedir que o desperdício ocorra.

É previsto que organizações que alcançam os níveis mais altos do CMM tenham um processo que é capaz de produzir *softwares* extremamente seguros dentro de custos e prazos previsíveis. Como cresce o entendimento dos níveis mais altos de maturidade, as áreas-chave de processo, descritas adiante, serão refinadas e as outras podem ser somadas ao modelo.

O CMM, segundo o CMU/SEI (1997), é derivado de idéias aplicáveis aos processos de fabricação repetitiva. Mas processos de *software* não são dominados por assuntos de replicação como um processo industrial.

3.3 Visibilidade no Processo de *Software*

Os engenheiros de *software* podem revelar a capacidade de discernimento do estado e do desempenho de um projeto na medida em que dispõem de informações sobre esses fatores, o que ocorre facilmente em pequenos projetos. Porém, em grandes projetos, esse discernimento passa a depender de sua experiência pessoal e de informações, porquanto, sem elas, eles têm sua percepção reduzida e, por isso, passam a investir em revisões periódicas para obter as informações sobre o progresso que precisam monitorar. A Figura 3.3 ilustra o nível de visibilidade no estado e desempenho do projeto dispostos à administração a cada nível de maturidade do processo. Cada nível de maturidade incrementalmente provê melhor visibilidade.

No Nível 1, o processo de *software* é uma "caixa preta" e a visibilidade fica limitada. Após a etapa de atividades ser pobremente definida, os gerentes têm um tempo extremamente curto para estabelecer o estado do progresso do projeto e das atividades. Exigências e alguns resultados de produto fluem no processo de *software* de uma maneira descontrolada.

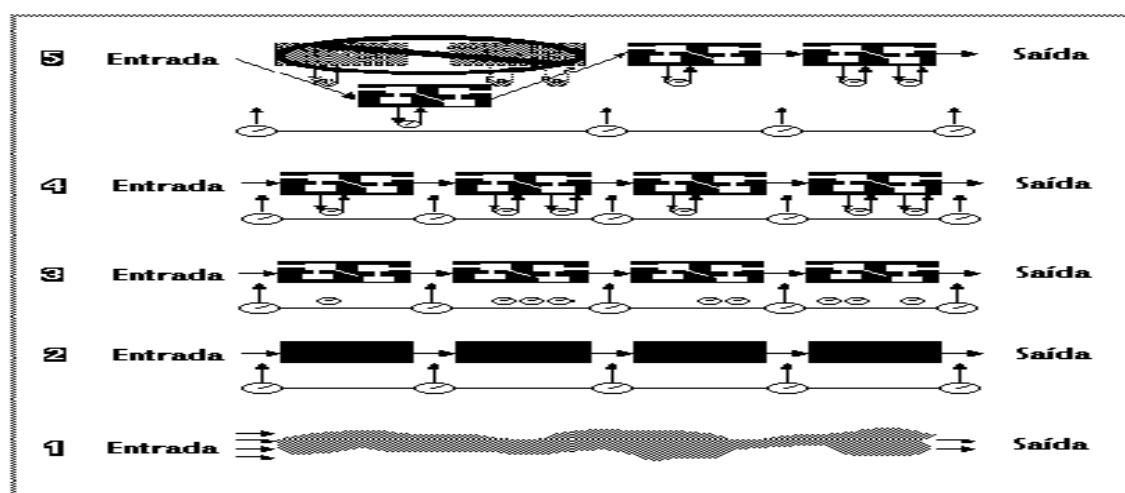


FIGURA 3.3 - VISIBILIDADE DOS PROCESSOS A CADA NÍVEL DE MATURIDADE
FONTE: CARNEGIE MELLON UNIVERSITY/*Software Engineering Institute* (1997, p. 23)

No Nível 2, as exigências do cliente são conhecidas, produtos de trabalho são controlados, e práticas básicas de gerência de projeto são estabelecidas. O processo de construção do *software* pode ser visto como uma sucessão de caixas pretas que permitem visibilidade de gerência apenas nos pontos de transição, como fluxos de atividades entre caixas. Embora a gerência possa não saber os detalhes do que está acontecendo dentro da caixa, os produtos, verificados nos postos de fiscalização entre caixas, informam o andamento do projeto.

No Nível 3, a estrutura interna das caixas (as tarefas do processo de desenvolvimento de *software*) é visível. A estrutura interna representa o modo como o processo de *software* padrão da organização foi aplicado a projetos

específicos. Gerentes e engenheiros entendem seus papéis e responsabilidades dentro desse processo e como suas atividades interagem ao nível apropriado de detalhe. A gerência prepara-se para riscos que podem surgir. Agentes externos ao projeto podem captar rápido e precisamente as mudanças na situação porque processos definidos oferecem boa visibilidade.

No Nível 4, são instrumentalizados e controlados quantitativamente os processos de *software*. Gerentes podem medir progressos e problemas. Eles dispõem de uma ampla base informativa para tomar decisões. Sua capacidade para prever resultados crescem continuamente e, concomitantemente, a variabilidade no processo cresce menos ou reduz.

No Nível 5, novos e melhorados modos de construção de *software* são continuamente experimentados, de uma maneira controlada, para melhorar a produtividade e a qualidade. A mudança disciplinada é um modo de vida, conforme atividades ineficientes ou propensas a defeitos são identificadas e substituídas ou revisadas. Os gerentes podem estimar e localizar quantitativamente o impacto e efetividade da mudança.

3.4 Definição Operacional do CMM

A operacionalização do CMM é projetada para apoiar diferentes formas de uso. Existem, pelo menos, quatro formas de uso suportados pelo CMM, segundo o CMU/SEI (1997):

- 1) Equipes de avaliação o usarão para identificar pontos fortes e fracos na organização.

- 2) Equipes de avaliação o usarão para identificar riscos, na seleção e monitoramento de contratos e contratados.
- 3) Os gerentes e pessoal técnico o usarão para identificar as atividades necessárias para planejar e implementar programas de melhoria do processo de *software* da organização.
- 4) Grupos de melhoria de processo o usarão como um guia para ajudá-los a definir e melhorar o processo de *software* em suas organizações.

Devido aos diversos usos do CMM, este deve ser exaustivamente decomposto e detalhado para evidenciar os processos e estruturas que caracterizam a maturidade e a capacidade de um processo de *software*.

3.4.1 Áreas-chave de Processo

Com exceção do Nível 1, como mostrado na Figura 3.4, cada nível de maturidade é decomposto em áreas-chave de processo ou KPA's - *Key Process Area*, como serão referidas daqui em diante, que indicam as atividades que a organização deve focar para atingir a melhoria do seu processo de *software*. KPA's, portanto, identificam os assuntos que devem ser tratados para alcançar seu nível de maturidade.

Cada KPA identifica um grupo de atividades relacionadas que, quando executadas coletivamente, realizam as metas consideradas importantes para aumentar a capacidade de processo. As KPA's são definidas para constituir um nível de maturidade específico, como mostrado na Figura 3.5. Os caminhos para alcançar as metas de uma KPA podem diferir, por

projeto, conforme seus domínios de aplicação ou ambientes. Não obstante, todas as metas devem ser alcançadas para que as KPA's sejam satisfeitas.

Quando as metas de uma KPA são realizadas continuamente através dos projetos, pode-se dizer que a organização tem sua capacidade de processo institucionalizada naquela área-chave.

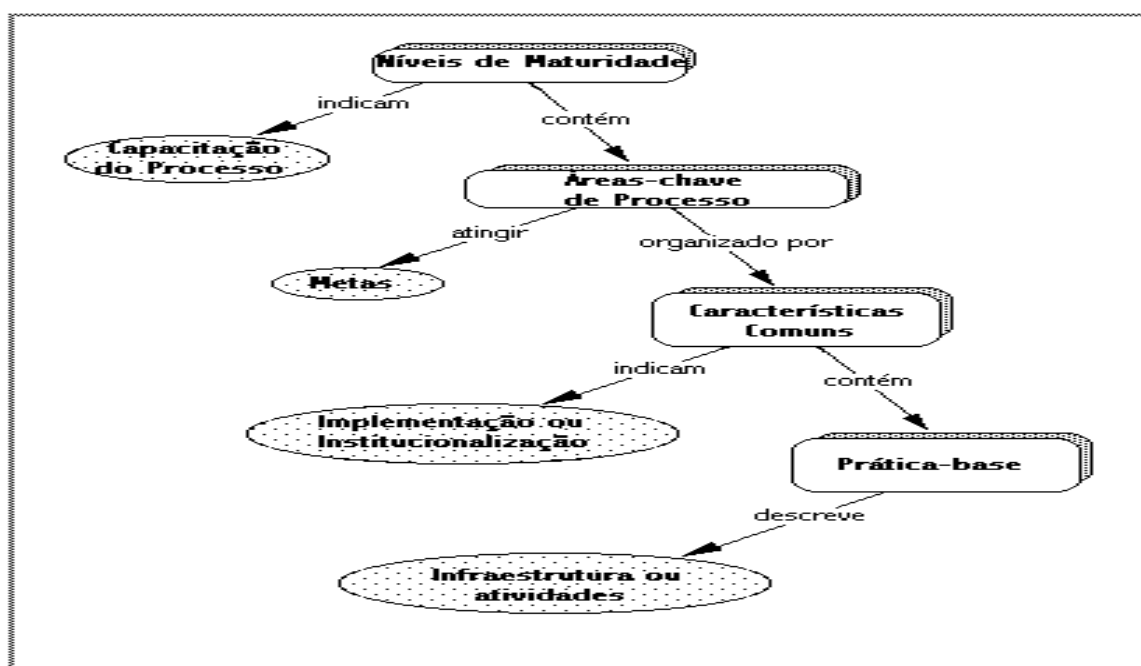


FIGURA 3.4 - A ESTRUTURA DO CMM

FONTE: The Capability Maturity Model for *Software* (1998)

O adjetivo "chave" implica que existem áreas de processo que não são chave para alcançar um nível de maturidade. O CMM não descreve todas as áreas de processo que estão envolvidos com desenvolvimento e manutenção de *software*. Ele apenas identifica e descreve como chave aquelas determinantes da capacidade de processo.

Cada KPA possui metas que, por sua vez, resumem as práticas-base que podem ser usadas para determinar se uma organização ou projeto implementaram a KPA efetivamente. As metas significam o ambiente, limites e intento de cada KPA.

As práticas específicas a serem executadas em cada KPA evoluirão conforme a organização alcance níveis mais altos de maturidade de processo. Por exemplo, muitos dos projetos que estimam capacidades descritas na KPA “Planejamento de Projeto de *Software*”, no Nível 2, têm que evoluir para manipular os dados de projeto adicionais disponíveis nos Níveis 3, 4, e 5. O “Gerenciamento de *Software* Integrado” no Nível 3 é a evolução do “Planejamento de Projeto de *Software*” e “Visão Geral e Acompanhamento do Projeto de *Software*” no Nível 2, conforme o projeto é gerenciado usando um processo de *software* definido.

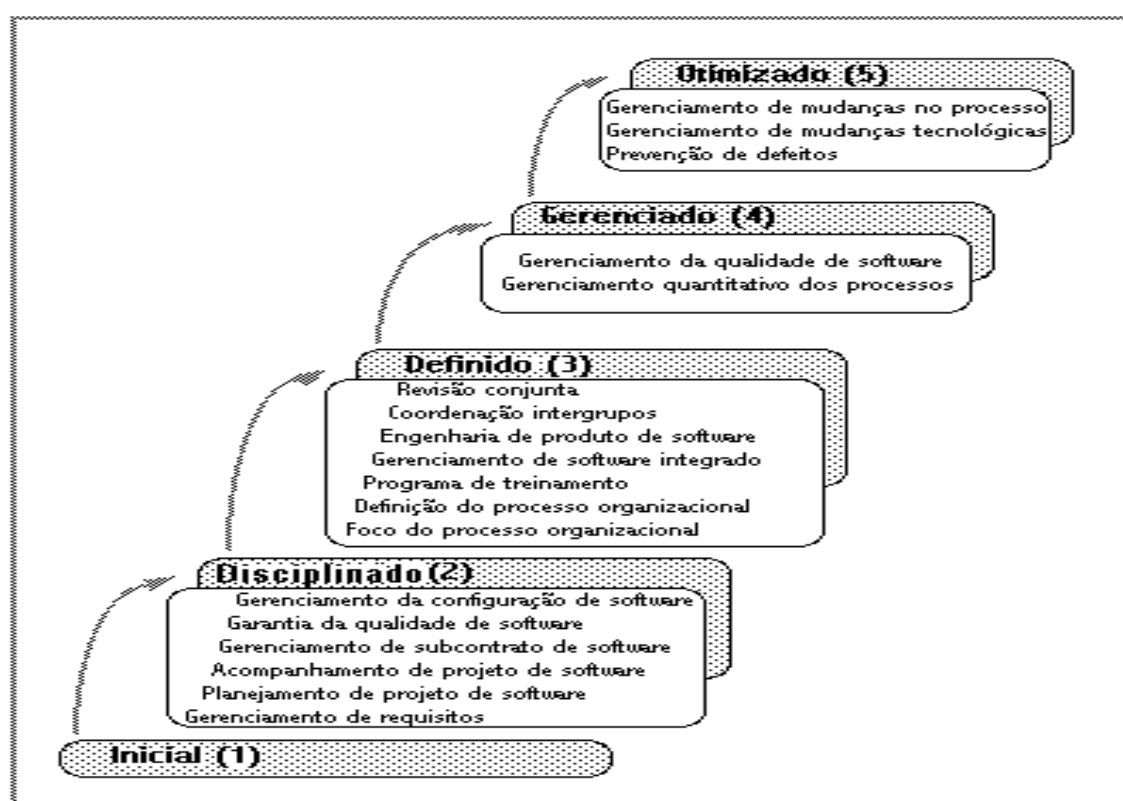


FIGURA 3.5 - AS KPA'S ATRAVÉS DOS NÍVEIS DE MATURIDADE

FONTE: The Capability Maturity Model for *Software* (1998)

As KPA's do nível 2 enfocam o estabelecimento de controles básicos de gerência de projeto. São seis: *Gerenciamento de Requisitos*, *Planejamento do Projeto de Software*, *Visão Geral e Acompanhamento do Projeto de*

Software, Gerenciamento de Subcontratados de Software, Garantia de Qualidade do Software, Gerenciamento de Configuração de Software.

Com relação às KPA's do Nível 2, CHRISTEL (1992) descreve que algumas metas do QFD assemelham-se às metas referentes a conclusão de requisitos, propostas pelo SEI. São elas:

- treinar a gerência.
- capturar a " voz de cliente ", isto é, capturar o que o cliente gosta em lugar do problema, permitindo planejar decisões a serem traçadas por trás da necessidade do cliente.
- desenvolvimento horizontal, isto é, "trabalho de grupo" em lugar de "auto-promoção".

O QFD fortalece o processo de desenvolvimento atual e alcança a produção desejada eficazmente. Fortalece o processo de desenvolvimento por:

- definir antes objetivos claros baseados em demandas de mercado e negócios;
- focar simultaneamente sobre tecnologias de produto e processo;
- priorizar os assuntos-chave para melhor alocação de recurso; e
- melhorar a comunicação e equipes de trabalho.

Alcança a produção desejada eficazmente por satisfazer as necessidades do cliente com o produto, e por fornecer um produto que tenha uma margem competitiva (CHRISTEL, 1992).

Vantagens de aplicar o QFD na fase de conclusão de requisitos, incluem:

- enfatizar projetos para qualidade através de necessidades do cliente;

- promover a criação de equipes;
- melhorar a comunicação de funções cruzadas;
- tratar antes itens de alta prioridade;
- preservar conhecimento nos documentos do QFD (promover o reuso);
- reduzir custos através de diminuição de problemas no início;
- reduzir o tempo de desenvolvimento do produto (em parte pela eliminação virtual das últimas mudanças de engenharia);
- melhorar a confiabilidade do projeto;
- aumentar a satisfação do cliente.

Desvantagens de aplicar o QFD na fase de conclusão de requisitos incluem o seguinte:

- Mais trabalho deve ser feito nos estágios de planejamento.
- É mais difícil mudar de direção depois que, uma vez começado, muito trabalho foi feito e todas as interrelações tiveram que ser revisitadas e revisadas para seguir numa determinada direção.
- Muitos passos no método esboçado para planejamento do QFD são aplicáveis ao desenvolvimento de um produto competitivo, ou seja, o uso de pesquisa de mercado e análise de produtos competidores.
- O método esboçado do QFD não indica o processo pelo qual as exigências do cliente e características de controle do produto, decompostos, são derivados. Outros métodos podem ser usados em conjunto com o QFD para descrever como a decomposição do problema e da arquitetura poderiam ocorrer durante a fase de conclusão de requerimentos.

- O método QFD não fornece condições de parar sobre a decomposição de exigências de clientes, isto é, o detalhamento ideal de exigências de clientes não é especificada.

CHRISTEL (1992) conclui que, apesar destas desvantagens, muitos dos aspectos do QFD deveriam ser incorporados à fase de conclusão de requisitos, proposta pelo SEI/CMU.

As KPA's do Nível 3 tratam de assuntos organizacionais e de projeto, conforme a organização estabelece uma infra-estrutura que institucionalize efetiva engenharia de *software* e gerenciamento de processo para todos os projetos. São sete: *Foco do Processo Organizacional, Definição do Processo Organizacional, Programa de Treinamento, Gerenciamento de Software Integrado, Engenharia do Produto de Software, Coordenação Intergrupos, Revisão Conjunta.*

No Nível 4, as KPA's enfocam o estabelecimento de uma compreensão quantitativa do processo e produtos do *software* em construção. As duas KPA's, neste nível, *Gerenciamento do Processo Quantitativo* e *Gerenciamento da Qualidade de Software*, são altamente interdependentes.

No Nível 5, as KPA's cobrem os assuntos que a organização e os projetos têm que encaminhar para implementar a melhoria, de forma contínua e mensurável. São três: *Prevenção de Defeito, Gerenciamento de Mudança Tecnológica, Gerenciamento de Mudanças no Processo.*

3.4.2 Características Comuns e Práticas-base

Por conveniência, as KPA's são organizadas segundo elementos comuns – os atributos que indicam se a implementação e institucionalização de cada uma delas são efetivas, repetíveis e duradouras. São cinco características comuns e cada uma possui suas práticas-base a serem realizadas, como mostra a Tabela 3.1 (Qualidade de *Software*, 1998).

TABELA 3.1 CARACTERÍSTICAS COMUNS E PRÁTICAS-BASE

FONTE: Qualidade de *Software* (1998).

Característica Comum	Descrição	Práticas-base relacionadas
Compromisso de realizar	Atitudes a serem tomadas pela organização para garantir que o processo se estabeleça e seja duradouro.	Estabelecimento de políticas e apadrinhamento de um gerente experiente.
Capacidade de realizar	Pré-requisitos que devem existir no projeto ou na organização para implementar o processo de forma competente.	Alocação de recursos, definição da estrutura organizacional e de treinamento.
Atividades realizadas	Papéis e os procedimentos necessários para implementar uma KPA.	Estabelecimento de planos e procedimentos, realização e supervisão do trabalho e ações corretivas.
Medições e análise	Necessidade de medir o processo e analisar as medições.	Realização de medições para determinar a efetividade das atividades realizadas.
Implementação com Verificação	Passos para assegurar que as ações sejam realizadas de acordo com o processo estabelecido.	Revisão, auditoria e garantia de qualidade.

As práticas-base relacionadas às características comuns das *Atividades Realizadas*, descrevem o que deve ser implementado para estabelecer uma capacidade de processo.

Cada KPA é descrita em termos de práticas-base que satisfazem suas metas. As práticas-base descrevem a infra-estrutura e atividades que contribuem para a implementação e institucionalização efetiva da KPA.

A descrição de cada prática-base, também chamadas de *top-levels*, é feita por uma única sentença, freqüentemente seguida por uma descrição mais detalhada que pode incluir exemplos e elaboração. As práticas-base declaram as políticas fundamentais, procedimentos e atividades para suas respectivas KPA's. A Figura 3.6 fornece um exemplo da estrutura que está por baixo de uma prática-base para a KPA "Planejamento do Projeto de *Software*".

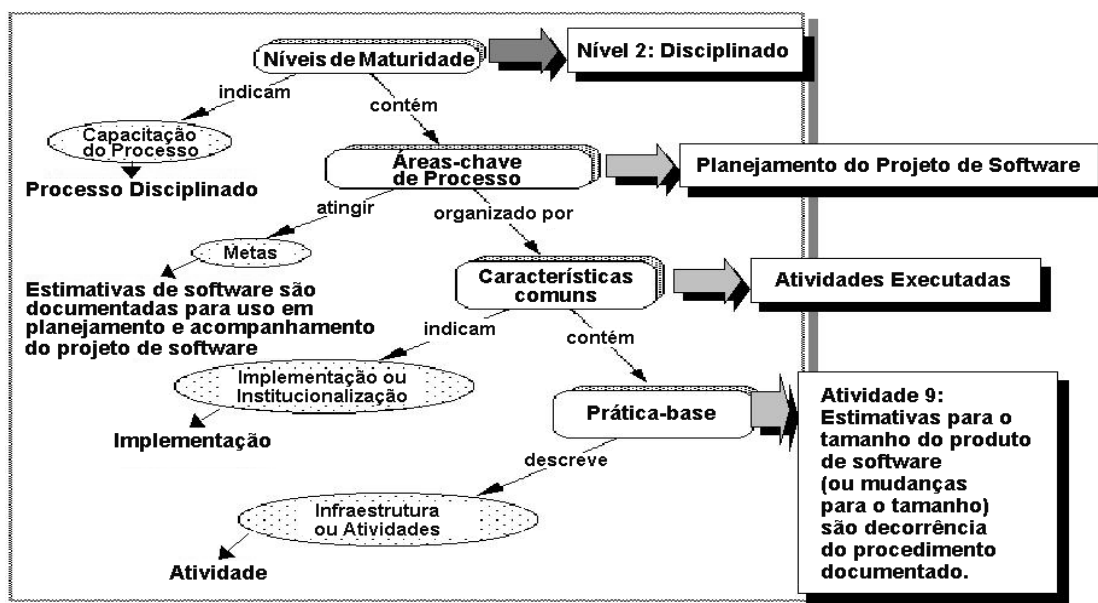


FIGURA 3.6 - UM EXEMPLO DE PRÁTICA-BASE

FONTE: The Capability Maturity Model for *Software* (1998), adaptado pelo autor.

Conforme ilustrado na Figura 3.6, a organização deve estabelecer procedimentos normatizados e documentados para assegurar previsibilidade com relação ao desenvolvimento e produto de *software*. Se estes procedimentos não são desenvolvidos a partir de um processo documentado, eles podem variar demasiadamente ou ao sabor de empirismos. A descrição

detalhada do que deve ser esperado em cada procedimento parte de históricos de tamanho de dados e de revisão das estimativas.

As práticas-base descrevem “o que” será feito, e não “como” será feito. Elas devem ser interpretadas no sentido de avaliar racionalmente se as metas da KPA são efetivamente, embora talvez diferentemente, alcançadas.

3.5 Usando o CMM

O CMM estabelece uma lista de critérios que descrevem as características de empresas de *software* maduras. Este critério pode ser usado para as empresas melhorarem seus processos de desenvolvimento e manutenção de *software* ou por organizações governamentais ou comerciais para avaliar os riscos de um projeto de *software* contratado com terceiros.

Esta seção enfoca dois métodos desenvolvidos pelo *Software Engineering Institute* (CMU/SEI, 1997) para avaliar a maturidade de uma organização na execução do processo de *software*, em termos de avaliação do processo de *software* e avaliação da capacidade de *software*, descritos a seguir:

- SPA (*Software Process Assessment*) – *avaliação do processo de software* – determina o estado atual do processo de desenvolvimento de *software* de uma organização, as prioridades dos assuntos relacionados que ela enfrenta e obtém o apoio organizacional para sua melhoria.
- O SCE (*Software Capability Evaluation*) – *avaliação da capacidade de software* – identifica e qualifica contratados para executar o trabalho de *software* ou, monitora o estado do processo de desenvolvimento de

software usado. Pode também ser usado internamente na preparação para uma avaliação externa.

Estes critérios não são, por si só, suficientes para o gerenciamento de uma avaliação. Qualquer um, desejando aplicar o CMM por estes métodos, deve obter informações adicionais sobre avaliação e treinamento de avaliação.

Resumidamente, os métodos de avaliação de processo de *software* e avaliação da capacidade de *software* (CMU/SEI, 1997):

- usam o questionário de maturidade como um trampolim para visitar o local,
- usam o CMM como um mapa que guia a investigação sobre o local,
- identificam pontos fortes e fracos do processo de *software*,
- derivam um perfil baseado em uma análise da satisfação das metas dentro da KPA, e
- apresentam os resultados dessa avaliação em termos de descobertas e de perfil da KPA.

3.6 Outros Modelos de Definição, Avaliação e Melhoria de Processos de *Software*

A seguir, serão apresentadas descrições sucintas de outros modelos de maturidade, que tratam da questão da qualidade do processo e do produto de *software*, a fim de oferecer referenciais para a melhor compreensão do modelo enfocado, o CMM.

Conforme descrito por TSUKUMO *et al.* (1996), pode-se observar que todos os modelos são fortemente influenciados por princípios de qualidade

similares e que os primeiros influenciaram os mais recentes, como, por exemplo, os modelos Trillium e SPICE, que foram baseados no modelo CMM.

3.6.1 O Modelo Trillium

O Modelo Trillium (Trillium, 1997) cobre todos os aspectos do ciclo de desenvolvimento de *software*, atividades de suporte e desenvolvimento de produto e sistema e um número significativo de atividades relacionadas a *marketing*.

Embora o Trillium tenha sido projetado para ser aplicado a sistemas de *software* embutidos como sistemas de telecomunicações, muito do modelo pode ser aplicado diretamente a desenvolvimento de *hardware*, bem como a outros segmentos da indústria de *software* - como Sistemas de Informação de Gerenciamento (MIS – *Management Information Systems*).

Mesmo o Modelo Trillium estando baseado no CMM, versão 1.1, em relação a este, possui algumas diferenças significativas, tais como (Trillium, 1997):

- modelo de arquitetura baseado em *roadmaps* (mapas de rota), em lugar de KPA's,
- perspectiva de produto, em lugar de *software*,
- capacidade mais abrangente de inclusão de assuntos, e
- enfoque de cliente, maturidade tecnológica e orientação de telecomunicações.

O Trillium possui a seguinte estrutura (Trillium, 1997):

A. A escala do Trillium

A escala do Trillium mede níveis de 1 a 5, que podem ser caracterizados do seguinte modo:

- 1) *Desestruturado (Risco Elevado)*: o processo de desenvolvimento é caótico. Projetos freqüentemente não incluem metas de qualidade ou prazos. O sucesso, enquanto possível, está baseado em indivíduos em lugar de infraestrutura organizacional.
- 2) *Projeto Orientado (Risco Médio)*: é alcançado o sucesso de projeto individual através de forte controle e planejamento de gerenciamento de projeto, com ênfase em gerenciamento de requisitos, técnicas de estimativa e gerenciamento de configuração.
- 3) *Definido e Processo Orientado (Risco Baixo)*: são definidos e utilizados processos à nível organizacional, embora a customização de projeto ainda seja permitida. Processos são controlados e melhorados. Exigências da ISO 9001, como treinar e examinar processo interno, estão incorporadas.
- 4) *Gerenciado e Integrado (Risco Muito Baixo)*: a instrumentação e análise de processo é usada como um mecanismo-chave para sua melhoria. Gerenciamento de mudança de processo e programas de prevenção de defeito são integrados aos processos.
- 5) *Totalmente Integrado (Risco Mínimo)*: metodologias formais são extensivamente usadas. Repositórios organizacionais para desenvolvimento histórico e processo são utilizados e efetivos.

B. Áreas de capacidade

O Modelo Trillium consiste em Áreas de Capacidade, *Roadmaps* (mapas de rota) e Práticas, conforme ilustrado na Figura 3.7.

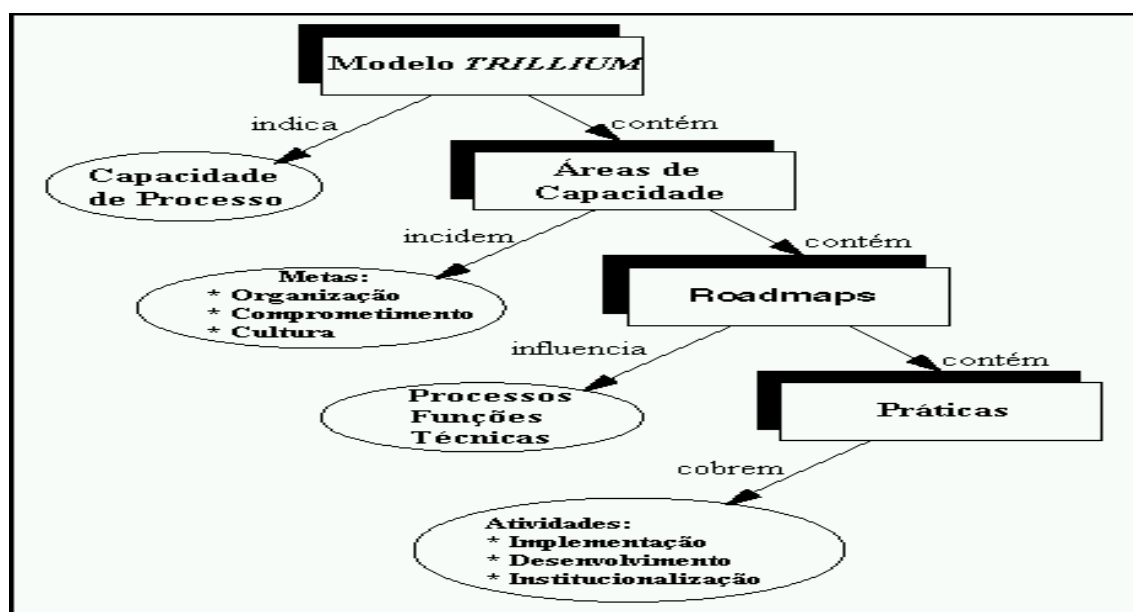


FIGURA 3.7 - A ARQUITETURA DO MODELO TRILLIUM

FONTE: Trillium Model Description (1997).

Cada Área de Capacidade contém práticas nos múltiplos níveis do Trillium. Por exemplo, o Gerenciamento mede os níveis 2 a 4 enquanto o Sistema de Qualidade mede os níveis 2 a 5. A medida de cada Área de Capacidade é mostrado na tabela seguinte.

TABELA 3.2 ÁREAS DE CAPACIDADE ATRAVÉS DOS NÍVEIS DO TRILLIUM

FONTE: Trillium Model Description (1997).

Área de Capacidade Trillium	Contém práticas no nível			
	2	3	4	5
Qualidade de Processo Organizacional - QPO	X	X	X	
Gerenciamento e Desenvolvimento de RH - RH	X	X	X	
Processo	X	X	X	X
Gerenciamento - GER	X	X	X	
Qualidade de <i>Software</i> - QS	X	X	X	X
Práticas de Desenvolvimento de Sistemas - PDS	X	X	X	X
Ambiente de Desenvolvimento - AD	X	X	X	X
Suporte a Clientes - SC	X	X	X	

C. Nível de capacidade e *roadmaps* (mapas de rota)

A organização deve satisfazer um mínimo de 90% dos critérios, em cada uma das 8 Áreas de Capacidade naquele nível. Níveis 3, 4 e 5 requerem a realização dos níveis mais baixos do Trillium, isto é, assim como no CMM, não podem ser saltados níveis.

Cada Área de Capacidade incorpora um ou mais *roadmaps*. Um *roadmap* é um conjunto de práticas relacionadas que enfocam uma área ou necessidade organizacional ou um elemento específico dentro do processo de desenvolvimento do produto. Cada *roadmap* representa uma capacidade significativa. A Tabela 3.3 lista os *roadmaps* contidos dentro de cada área de capacidade, como também a distribuição de práticas por nível de maturidade e área de capacidade.

Dentro de um determinado *roadmap*, o nível das práticas está baseado em seu respectivo grau de maturidade. Uma organização amadurece pelos níveis de *roadmap*.

Para efeito de comparação, vale lembrar que no CMM cada nível possui suas KPA's (Áreas-chave de Processo), que por sua vez, possuem suas características comuns e práticas-base. No Trillium, cada Área de Capacidade possui seus níveis, que por sua vez, possuem seus *roadmaps*.

TABELA 3.3 ROADMAPS ATRAVÉS DAS ÁREAS DE CAPACIDADE

FONTE: Trillium Model Description (1997).

Áreas de Capacidade do Trillium	Roadmaps (Mapas de Rota)	Número de práticas por nível				Total
		2	3	4	5	
Qualidade do Processo Organizacional	Gerenciamento da Qualidade Engenharia de Processo de Negócios	10	20	5	0	35
Gerenciamento e Desenvolvimento de Recursos Humanos	Gerenciamento e Desenvolvimento de Recursos Humanos	9	42	1	0	52
Processo	Definição de Processo Gerenciamento de Tecnologia Engenharia & Melhoria do Processo Medições	16	55	24	4	99
Gerenciamento	Gerenciamento de Projeto Gerenciamento de Sub-contratados Relação Cliente-Fornecedor Gerenciamento de Requerimentos Orçamento	74	29	4	0	107
Sistema de Qualidade	Sistema de Qualidade	14	15	2	2	33
Práticas de Desenvolvimento	Processo de Desenvolvimento Técnicas de Desenvolvimento Documentação Interna Verificação & Validação Gerenciamento de Configuração Reutilização Gerenciamento de Confiabilidade	41	49	15	5	110
Ambiente de Desenvolvimento	Ambiente de Desenvolvimento	4	6	1	1	12
Suporte a Cliente	Sistema de Resposta a Problemas Engenharia de Usabilidade Modelagem do Custo do Ciclo de Vida Documentação do Usuário Treinamento do Usuário	25	30	5	0	60
Total:		193	246	57	12	508

3.6.2 ISO 15504 (SPICE)

O SPICE (1998) - *Software Process Improvement and Capability Determination* (Melhoria do Processo de *Software* e Determinação da Capacidade) é uma norma que cobre todos os aspectos da Qualidade do Processo de *Software*.

Depois da criação do CMM, começaram a surgir outros modelos que compartilhavam as mesmas idéias, mas indo por caminhos diferentes. A ISO, então, criou um projeto determinando uma norma para avaliação de processos, voltado especificamente para a área de *software*, objetivando a padronização e harmonização de todos os modelos existentes. Este projeto – denominado SPICE (1998), idealiza reunir todo o conhecimento existente sobre avaliação de processos, com dois objetivos:

- Melhoria: avaliar os processos e identificar pontos de melhoria.
- Capacidade: avaliar graus de risco ao se fazer negócios com terceiros.

Essa avaliação de processo também serve para dar uma idéia do grau de maturidade em que se encontra a empresa contratada. Para isso, o SPICE criou um modelo que é uma generalização do modelo CMM.

Conforme a Figura 3.9, a determinação da capacidade do processo ocupa-se com analisar a capacidade proposta contra um perfil designado, para identificar os riscos envolvidos. A capacidade proposta pode estar baseada nos resultados das avaliações de processos relevantes, ou pode estar baseada em uma avaliação com a finalidade de estabelecer a capacidade proposta.

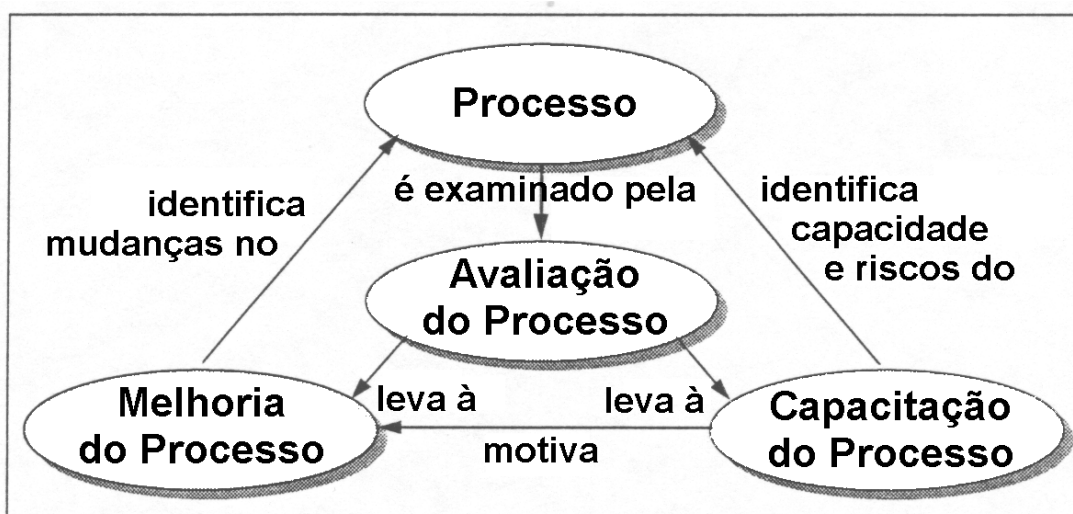


FIGURA 3.8 – AVALIAÇÃO DO PROCESSO DE SOFTWARE
 FONTE: SPICE (1998).

O SPICE foi projetado para satisfazer as necessidades de compradores, fornecedores e assessores, e suas exigências individuais dentro uma única fonte. Os benefícios advindos de seu uso incluem:

- Compradores: habilidade para determinar a capacidade atual e potencial dos processos de *software* de um fornecedor.
- Fornecedores: habilidade para determinar a capacidade atual e potencial de seus próprios processos de *software* e para definir áreas e prioridades para melhoria de processo de *software*.
- Assessores: estrutura que define todos os aspectos para gerenciar avaliações.

O padrão é projetado para prover resultados de avaliação que são repetíveis, objetivos, comparáveis dentro de contextos similares e capazes de serem usados para melhoria ou determinação de capacidade de processo. Pode ser usado por organizações envolvidas em planejamento, gerenciamento, monitoramento, controle e melhoria de aquisição, fornecimento, desenvolvimento, operação,

evolução e suporte de *software*.

Cada processo é descrito por práticas básicas que são as atividades essenciais do processo específico. Os processos são agrupados dentro de cinco categorias de processo conforme exibido na tabela a seguir.

TABELA 3.4 - DESCRIÇÃO DE CATEGORIAS DE PROCESSO

FONTE: SPICE (1998).

Categoria de Processo	Descrição: Processos que
Cliente-Fornecedor	Impactam diretamente sobre o cliente
Engenharia	Especificam, implementam, ou mantêm um sistema e um produto de <i>software</i>
Projeto	Determinam o projeto, coordenam e gerenciam os recursos
Suporte	Habilitam e suportam a execução de outros processos no projeto
Organização	Determinam os objetivos de negócio da organização, seus processos de desenvolvimento, produto e recursos que ajudarão a organização a atingí-los

O objetivo do SPICE é avaliar a capacidade da organização em cada processo e permitir sua melhoria. O modelo de referência inclui seis níveis de capacitação. Cada um dos processos mencionados acima deve ser classificado nos níveis descritos na Tabela 3.5.

TABELA 3.5 - DESCRIÇÃO DOS NÍVEIS DE CAPACITAÇÃOFONTE: Qualidade de *Software* (1998).

Nível	Descrição
0 Incompleto	Há falha em realizar o objetivo do processo.
1 Realizado	O objetivo do processo é atingido, embora não como planejado. As ações devem ser realizadas quando necessárias. Os produtos são referenciais do atendimento dos objetivos do processo.
2 Gerenciado	O processo produz os produtos com Qualidade aceitável e dentro do prazo. Isto é feito de forma planejada e controlada. Os produtos estão de acordo com padrões e requisitos.
3 Estabelecido	O processo é realizado e gerenciado de forma definida. Ações implementadas são versões adaptadas do processo padrão.
4 Previzível	O processo é realizado de forma consistente, objetiva e dentro dos limites de controle. Medidas da realização do processo são analisadas, levando a um entendimento quantitativo da capacitação do processo e da habilidade de predição.
5 Otimizado	O processo é otimizado para atender necessidades atuais e futuras do negócio, atinge os objetivos propostos e pode ser repetido. São estabelecidos objetivos quantitativos de eficácia e eficiência. A monitoração constante é conseguida através de <i>feedback</i> quantitativo e a melhoria é conseguida pela análise dos resultados. A otimização do processo envolve o uso piloto de idéias e tecnologias inovadoras, além da mudança de processos ineficientes.

3.7 Análise das Normas e Modelos Apresentados

A análise que se segue nesta seção foi adaptada do estudo realizado por TSUKUMO *et al.* (1996), inserindo-se aspectos do modelo Trillium, assim como outros aspectos observados nos demais modelos e normas. A síntese desta análise pode ser observada no Anexo 2.

O CMM, Trillium e SPICE destacam-se como modelos mais apropriados para organizações de grande porte, os demais podem ser aplicados em organizações de porte variado. Com exceção da ISO 9000-3, todos os demais modelos estabelecem explicitamente tipos de processos agrupados em níveis ou categorias.

Com exceção da ISO 12207-1, todos os demais modelos tratam da avaliação e melhoria do processo de *software*, direta ou indiretamente, diferindo apenas na abordagem. Especificamente, a ISO 9000-3 não mostra como promover a melhoria da qualidade, mas como certificar se o processo de uma organização possui um nível adequado do seu sistema da qualidade.

A ISO 9000-3, o CMM e o Trillium exigem que as organizações satisfaçam todos os aspectos definidos no modelo para comprovação da capacidade. No modelo SPICE a capacidade é definida para cada processo como uma porcentagem de adequação a cada um dos níveis. Conseqüentemente, a melhoria no CMM e no Trillium implica em atender todos os aspectos do nível superior, enquanto no SPICE são definidos quais processos devem ser melhorados e quanto, de acordo com os objetivos da organização.

Como aspecto positivo, a ISO 12207-1 traz como contribuição uma taxonomia útil para classificação de processos de qualquer organização. No entanto, não passa disso, apenas uma definição de uma taxonomia.

O aspecto positivo da ISO 9000-3 é o fato de ser uma norma bem difundida e por ter seu valor reconhecido entre as organizações. No entanto, não apoia diretamente a melhoria contínua de processos e pode induzir à colocação da certificação como objetivo principal da aplicação da Norma.

O CMM estabelece diretrizes para melhoria contínua dos processos de *software* de uma organização. Contudo, dá pouca consideração à diversidade das organizações e é de difícil aplicação em organizações pequenas.

O modelo Trillium traz como contribuição a expansão e flexibilização do modelo SEI/CMM e, além do CMM, incorpora critérios e normas internacionais. Entretanto, é de difícil aplicação em organizações pequenas, devido ao grande número de práticas a serem executadas. Tal inferência se impôs por três fatos:

- Ter sido desenvolvido pela *Bell Canada*, do setor de telecomunicações - uma empresa de grande porte que tomou-se como referencial.
- Impor e exigir grande volume de atividades e informações, nem sempre disponíveis e factíveis às pequenas empresas.
- Basear-se totalmente no SEI/CMM que, segundo estudo realizado pelo CTI (TSUKUMO *et al.*, 1996), é de difícil aplicação em pequenas organizações, característica também herdada pelo Trillium.

A contribuição do SPICE baseia-se na expansão e flexibilização de vários modelos anteriores como o CMM, Trillium e *Bootstrap*. Contudo, a amplitude e dimensão do próprio modelo e a grande quantidade de informação requer um esforço que dificulta a sua aplicação por organizações de pequeno porte. Segundo TSUKUMO *et al.* (1996), "Informações mais recentes sobre os resultados dos experimentos... indicam que uma nova estruturação (visando a sua transformação em Norma), tornará a aplicação mais flexível, facilitando o uso, mesmo em pequenas organizações".

3.8 Considerações Finais

Ao contrário do capítulo anterior, onde o objetivo era propiciar uma aproximação aos conceitos de qualidade de *software*, visando produto e

processo, este capítulo focou a qualidade do processo de *software*, destacando três modelos: o CMM, o Trillium e o SPICE. Isso não significa que a qualidade do produto de *software* não tenha tanta importância quanto o processo, ao contrário, deve-se considerar que a qualidade do produto é consequência de um processo com qualidade.

Os modelos estudados foram escolhidos com base nos seguintes critérios:

- 1) CMM - O primeiro modelo de maturidade criado, do qual todos os demais derivaram;
- 2) Trillium - Um modelo com finalidade específica de aplicação;
- 3) SPICE (ISO 15504) - Um modelo com reconhecimento internacional.

A análise comparativa, adaptada da obra de TSUKUMO *et al.* (1996), teve como objetivo propiciar o entendimento sobre os diversos aspectos positivos e negativos dos modelos e normas ligados à qualidade de *software*, assim como, a relação entre os mesmos.

4 ESTUDO DE CASOS

Neste capítulo são apresentadas cinco implementações de modelos de melhoria de desenvolvimento de *software*, ocorridas em diversas empresas, relatadas através de pesquisas realizadas pelas instituições norte-americanas CMU/SEI (*Carnegie Mellon University/Software Engineering Institute*) e DACS (*DoD Data & Analysis Center for Software*) e pelas organizações brasileiras CTI (Fundação Centro Tecnológico para Informática) e CELEPAR (Companhia de Informática do Paraná).

Será apresentado um resumo de cada caso, apenas com o objetivo de delineá-los para uma melhor compreensão da análise realizada no capítulo seguinte.

4.1 Modelos de SPI no Departamento de Defesa Americano e em Organizações Comerciais (MCGIBBON *et al.*, 1999)

Este caso, relatado pelo DACS (*DoD Data & Analysis Center for Software*), baseou-se nos resultados da implementação de modelos de SPI (*Software Process Improvement*). São apresentadas formas de se estimar os custos e benefícios da referida implementação, sob diversos aspectos no desenvolvimento de *software*, permitindo gerar a Tabela 4.1.

Como já foi colocado no Capítulo 3, SPI se refere a modelos, em geral, de melhoria de desenvolvimento de *software*. Os autores deste caso

utilizaram mais de uma alternativa, porém com destaque ao CMM, conforme apresentado na seção 2 do texto referência.

É importante registrar que o texto referência não apresenta a implementação em si dos modelos de SPI utilizados, fazendo apenas algumas menções a respeito ao longo do texto.

Cabe aqui esclarecer melhor a métrica Probabilidade de Alto Risco, apontada na tabela a seguir. Sem SPI, a probabilidade de alto risco concentra-se nas faixas Alto e Médio, tornando o custo elevado para essa métrica. Com SPI, a probabilidade de alto risco mantém-se baixa, mantendo o custo, para esta métrica, pouco mais de 90% menor do que uma organização sem SPI.

TABELA 4.1 COMPARANDO TODAS AS MÉTRICAS DA MELHORIA DE PROCESSO.
FONTE: MCGIBBON, T. *et al.* (1999, p. 26).

Métrica	Sem Melhoria	Com SPI	Melhoria
Benefícios Primários			
Custo total de desenvolvimento	\$ 2.886.543	\$ 780.174	\$ 2.106.370
Custo total de retrabalho	\$ 619.369	\$ 26.080	\$ 593.288
Duração média de cronograma	27 meses	17 meses	10 meses
Defeitos de versão divulgada	15% do total de defeitos	< 5% do total de defeitos	80%
Benefícios Secundários			
Vendas projetadas	\$ 10.000.000	\$ 10.500.000	\$ 500.000
Penalidades/Gratificação	- \$ 50.000	\$ 50.000	\$ 100.000
Custo anual de "Turnover"	\$ 615.000	\$ 102.500	\$ 512.500
Reincidência de negócios	\$ 1.000.000	\$ 5.000.000	\$ 4.000.000
Custo da melhoria		\$ 373.000	- \$ 373.000
Probabilidade de alto risco			
Alto	\$ 412.500	\$ 0	
Médio	\$ 1.678.125	\$ 0	
Baixo	\$ 0	\$ 175.000	

4.1.1 Benefícios Financeiros da Melhoria de Processo de *Software* (SPI)

O SPI, significativamente, reduz a quantia de tempo e esforço exigido para desenvolver *software*, o número de defeitos induzidos dentro de

um sistema, os custos e tempo para encontrar defeitos introduzidos, os custos de manutenção sobre produtos de *software* e, melhora a produtividade da equipe de desenvolvimento.

Análises adicionais têm sido executadas para observar o impacto sobre este modelo para vários tamanhos de programas computacionais, medidos através de número de linhas de código - LOC. Do trabalho desenvolvido, os autores puderam estimar os custos de retrabalho para diferentes modelos de processo, variando em função do tamanho do programa, conforme ilustrado pela Figura 4.1. As economias de custo são proporcionais ao tamanho do programa e são mostrados como uma porcentagem de métodos tradicionais na legenda do gráfico.

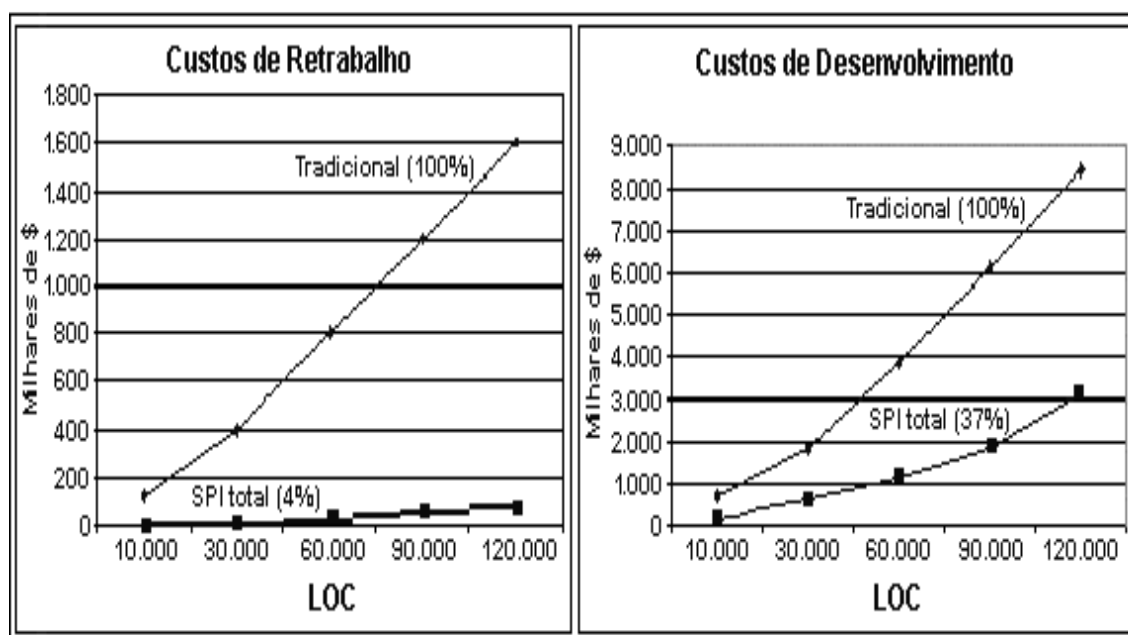


FIGURA 4.1 - CUSTOS DE RETRABALHO E DESENVOLVIMENTO EM FUNÇÃO DO TAMANHO DO PROGRAMA

FONTE: MCGIBBON *et al.* (1999, p. 28).

Similarmente, os autores comparam a variação do custo de desenvolvimento de projetos tradicionais com outros modelos de processo em

função do tamanho do programa. Foram descobertas economias de custo por serem proporcionais ao tamanho do programa.

A Figura 4.2 mostra o impacto sobre o prazo de vários modelos de processo. Novamente, a melhoria de prazos foi descoberta por ser proporcional ao tamanho do programa.

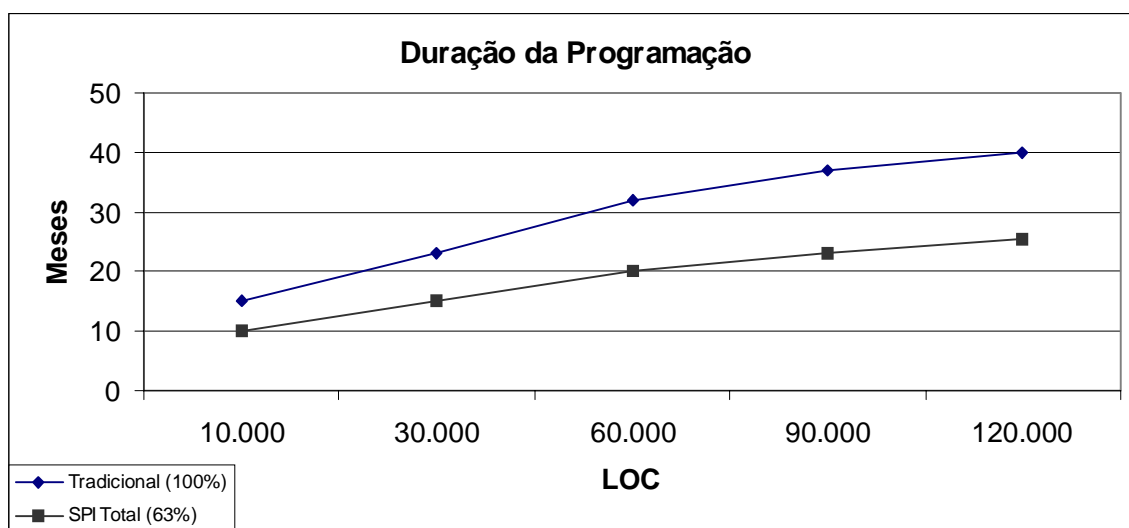


FIGURA 4.2 - DURAÇÃO DA PROGRAMAÇÃO EM FUNÇÃO DO TAMANHO DO PROGRAMA

FONTE: MCGIBBON *et al.* (1999, p. 29).

4.1.2 Os Benefícios Secundários do SPI

O SPI incide sobre as seguintes áreas:

- Vendas Projetadas com e sem Melhoria: Este fator é usado principalmente em organizações desenvolvendo produtos comerciais e para medir o aumento em vendas de produtos.
- Média do Histórico de Penalidades/Gratificações e Média de Gratificações Projetadas: Estas métricas são apenas executadas por organizações onde são recompensadas gratificações e prêmios de remuneração pelo alto

desempenho e entrega no prazo ou onde penalidades são incorridas para baixo desempenho.

- Custo Anual de "Turnover" com e sem Melhoria: Considerando que a melhoria de processo aumenta o moral do empregado, o *turnover* deve ser considerado como dramático. As economias de *turnover* menores poderiam, sozinhas, pagar as melhorias propostas.
- Custos de Desenvolvimento com e sem desenvolvedores-chave, Duração do Cronograma com e sem desenvolvedores-chave: Aumentados os custos de *turnover*, mais adiante está o impacto significativo da possibilidade de perder alguns dos desenvolvedores-chave de uma organização.
- Reincidência de Negócios com e sem Melhoria: Melhorar a satisfação de clientes deveria resultar em reincidência de negócios.
- Probabilidade de Alto Risco com e sem Melhoria: Estas duas métricas selecionadas são as mais interessantes e potencialmente mais poderosas, porque elas sumarizam e atribuem uma probabilidade para todos os fatores que necessitam ser considerados antes de selecionar um método sobre o outro.

4.2 Uma Pesquisa Sistemática da Melhoria de Processo, seus Benefícios e Fatores (GOLDENSON *et al.*, 1995)

Este estudo de caso baseou-se em resultados de pesquisa realizada pela *Carnegie Mellon University*, através de questionários aplicados no período de 1992 e 1993, nos Estados Unidos e Canadá. Estes questionários foram respondidos por 138 organizações sendo que, destas, 56 haviam sido

submetidas a avaliações de implementações do CMM. Os objetivos principais da pesquisa eram três: 1) descrever o que acontece aos esforços de melhoria de processo após as avaliações de implementações do CMM; 2) entender o quanto possível sobre porquê muitos esforços de melhoria eram melhores que outros; 3) aprender mais sobre a relação entre maturidade de processo e desempenho organizacional.

Segundo este estudo, organizações com poucos desenvolvedores de *software* parecem beneficiar-se da maturidade de processo mais alta, da mesma maneira que fazem organizações maiores.

4.2.1 Sobre as Avaliações de Implementações do CMM

Os respondentes da pesquisa vêem suas avaliações de processo de *software* como tendo sido altamente úteis para guiar seus esforços subseqüentes de melhoria de processo.

Os respondentes relatam que as avaliações fazem um bom trabalho identificando pontos fortes de suas organizações assim como suas fraquezas. Baseado em suas experiências de avaliações, uma grande maioria (mais de 80%) acredita que o CMM tenha fornecido um mapa que permitiu a identificação de quais melhorias de processo devem ser "agarradas" primeiro. Só 10% pensam agora que suas avaliações ou o CMM os fizeram negligenciar assuntos de melhoria de processo importantes.

Entretanto, havia dificuldades. Aproximadamente 1/4 dos respondentes relataram que as descobertas e recomendações elencadas por suas respectivas avaliações eram muito ambiciosas para serem alcançadas

num período de tempo razoável. Grande número dos respondentes disseram que eles precisavam de mais assistência e direção sobre *como* alcançar melhoria tangível nas áreas identificadas por suas avaliações. Sabendo que melhorar não é o bastante, eles precisavam de mais direção sobre como continuar a fazer com que as melhorias de fato acontecessem. Há evidência que, esses que informaram tais dificuldades, de fato fizeram menos progresso em seus esforços subsequentes de melhoria de processo.

Porém, ao todo, a maioria dos respondentes acredita que o dinheiro e o esforço dedicados às suas avaliações estavam bem gastos, e que as avaliações tiveram um impacto positivo e significativo em suas organizações.

4.2.2 Benefícios

Os respondentes da pesquisa relataram que uma quantia significativa de progresso aconteceu desde que suas avaliações foram conduzidas. A maioria dos respondentes (56%) relatou que suas organizações tiveram pelo menos um sucesso moderado tratando as descobertas e recomendações que foram elevadas por suas avaliações. 31% relataram sucesso significativo ou sucesso marcado ao longo de suas organizações. Somente 14% disseram que tiveram pequeno, se algum, sucesso apreciável.

A vasta maioria dos respondentes relatou ter seguido suas avaliações com planos de ação e equipes de ação de processo para levar adiante esses planos. Quase 3/4 disseram que suas organizações tinham implementado mudanças de processo em projetos de demonstração como resultado de suas avaliações.

Os respondentes relataram que o suporte para melhoria de *software* tem melhorado entre o gerenciamento, pessoal técnico e patrocinadores de avaliações de suas organizações, bem como aquelas que participaram diretamente nas avaliações.

A evidência da pesquisa sugere que uma boa porção de progresso tenha sido feito desde as avaliações. Existem muitas evidências pequenas, realmente, de que as avaliações tenham tido um impacto negativo sobre o progresso da melhoria de processo. Poucos dos respondentes (4%) disseram que suas avaliações tinham sido contra-produtivas. Contrário a algumas críticas, mais de 80% dos respondentes disseram que suas organizações de processo de *software* não tinham se tornado mais burocráticas e que a criatividade técnica não havia sido reprimida desde suas avaliações. De fato, nos setores comercial e governamental, havia evidência de que organizações mais maduras tinham menos papelada de requerimentos do que organizações menos maduras.

Ainda, detectaram mais que um pequeno desânimo sobre o ritmo da melhoria de processo. Aproximadamente 25% dos respondentes disseram que "nada tem mudado muito" desde a avaliação. Quase metade disse "ter estado muito desiludido pela falta de melhoria." Mais de 40% disseram que a melhoria de processo tem sido superada pelos eventos e crises, e que outras coisas têm levado prioridade. Quase 75% disseram que "a melhoria de processo tem frequentemente sofrido devido às limitações de tempo e recursos"; mais de 75% disseram que a melhoria de processo levou muito mais tempo do que esperavam; mais de 33% disseram que tem custado mais do que esperavam.

Tais dificuldades frequentemente afligem organizações quando elas tentam alcançar metas desafiadoras. Claramente, entretanto, há uma necessidade de se opor a expectativas não realistas sobre melhoria de processo em algumas organizações de *software*.

4.2.3 Sobre os Fatores de Sucesso de Implementação do CMM

Os respondentes responderam várias questões sobre as características de suas organizações que estão relacionadas ao aumento de sucesso que eles atribuíram a seus esforços de melhoria de processo. Assim, gerentes podem adotar muitas ações baseadas nesses resultados:

- Podem monitorar ativamente o progresso das melhorias de processo em suas organizações de *software*, formular metas de melhoria de processo, e trabalhar para assegurar que recursos adequados sejam investidos em seus esforços.
- Podem ter algum controle sobre os meios, na medida em que seus esforços de melhoria são apoiados e recompensados. A melhoria de processo não é alguma coisa a ser realizada em um tempo sobressalente, após o trabalho "real" ter sido concluído. Pessoas envolvidas na melhoria do processo deveriam ser bem respeitadas em suas organizações.

Os dados obtidos sugerem:

- Um número de fatores que podem criar dificuldades para alcançar a melhoria de processo. Entre eles, aspectos da cultura da organização. Quando os respondentes disseram que identificaram excessiva política organizacional, eles também informaram menor sucesso no tratamento das

descobertas e recomendações apontadas em suas avaliações. Resultados similares existem quando há cinismo e desânimo deixados de falhas anteriores ou quando o pessoal técnico tende a sentir que o SPI entra no modo do trabalho "real".

- Meios em que a comunidade de pesquisa e desenvolvimento pode contribuir à vista para a melhoria de processo de *software*. Aqueles respondentes da pesquisa que disseram que as recomendações levantadas por suas avaliações eram também ambiciosas, são também menos prováveis para relatar esforços prósperos de melhoria que seguem as avaliações. Resultados similares existem quando os respondentes são questionados sobre a necessidade de maior direção, aconselhamento, e assistência em implementar as melhorias sugeridas pelas avaliações.

4.3 A Experiência da Raytheon Electronic Systems na Melhoria do Processo de *Software* (HALEY *et al.*, 1995)

A coleção e análise de métricas instituídas como parte da Iniciativa da Engenharia de *Software* tem capacitado a *Raytheon Electronic Systems* (RES) a monitorar o impacto dessa iniciativa. Em particular, o impacto tem sido avaliado nas seguintes áreas: custo da qualidade, produtividade de *software*, índice de desempenho de custo, qualidade de produto global, benefício a outras organizações, e benefícios ao pessoal.

4.3.1 Custo da Qualidade

A aplicação inicial do modelo de custo de qualidade usou seis projetos contínuos grandes, principalmente porque os seis projetos empregaram 80 a 90% dos engenheiros de *software*.

Os dados combinados mostram que o custo médio de retrabalho ou não-conformidade tem diminuído seguindo o princípio da iniciativa, como ilustrado pela Figura 4.3. Nos dois anos anteriores à iniciativa, os custos de retrabalho tinham calculado a média aproximada de 41% dos custos de projeto. Nos dois anos seguintes, aqueles valores tinham caído, aproximadamente, 20% e a tendência era continuar baixando.

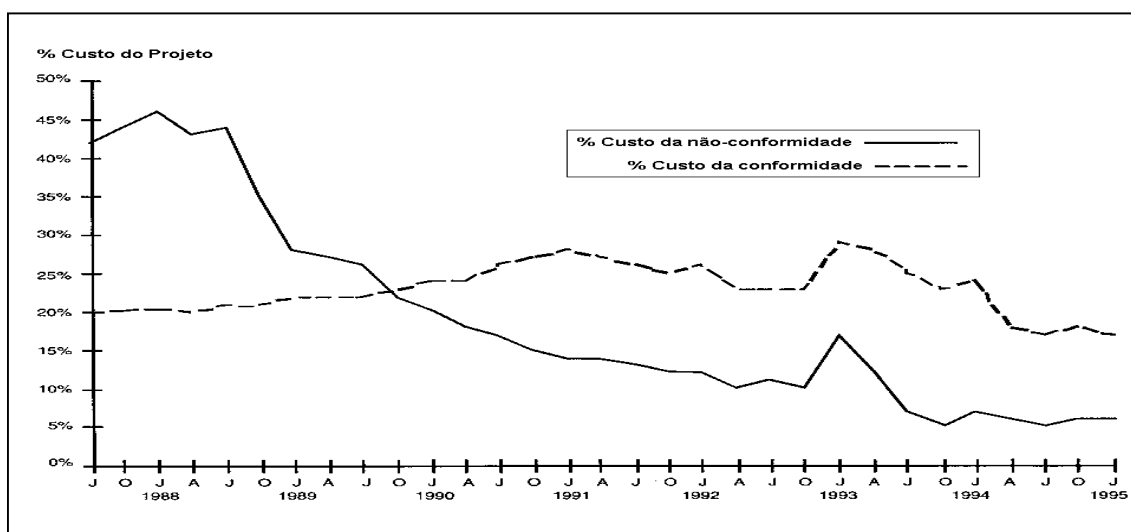


FIGURA 4.3 - CUSTO DA QUALIDADE VERSUS TEMPO
FONTE: HALEY, T. *et al.* (1995, p. 50).

Custos de avaliação subiam quando revisões informais eram substituídas por inspeções formais, e custos de prevenção subiam quando treinamentos de inspeção eram instituídos. Também, custos de retrabalho associados com defeitos achados durante projeto subiam de, aproximadamente, 0,75% para em termo de 2% do custo do projeto e aqueles

associados com defeitos achados durante codificação subiram de, aproximadamente, 2,5% para em termo de 4% do custo do projeto.

A maior redução nos custos de retrabalho eram aquelas associadas com problemas de código-fonte encontrados durante a integração, que caíram aproximadamente 20% de seu valor original. O segundo grande contribuinte para a redução do retrabalho era o custo de re-teste que, aproximadamente, caiu para a metade de seu valor inicial. Isto claramente indica que os custos adicionais de atividades de inspeções formais e o treinamento que deve precedê-lo são justificados basicamente por encontrar problemas no início do processo, resultando numa integração mais eficiente.

Esta análise inicial do custo da qualidade era realmente uma experiência de aprendizagem. Não era fácil naqueles muitos projetos pessoas terem que ser desviadas para estas tarefas de "não-projeto". Também não era barato, pois custava, aproximadamente, \$25.000 a mais.

Como base para a pré-melhoria dos custos de retrabalho, usaram o valor médio dos projetos (41%) na época em que começou a iniciativa (agosto de 1988). Então, calcularam as economias de retrabalho por projeto, por mês, conforme as diferenças entre o projeto atual e a base (41%). Somando isto sobre a amostragem de projetos para o período de um ano, rendeu uma economia total de \$4.480.000,00.

Durante o ano de 1990, os projetos experimentais tinham empregado 58% do total disponível da força de trabalho do SEL (*Software Engineering Laboratory*). Assumindo todos aqueles projetos beneficiados pelos investimentos da melhoria de processo, ratearam o total do investimento

(\$1.000.000,00) para os projetos experimentais, rendendo um investimento de \$580.000,00. O resultado sobre o investimento era, então, de 7.7 para 1 (\$4.480.000 / \$580.000).

Conforme a análise era atualizada (anualmente em 1991 e 1992, e semestralmente depois disso), novos projetos eram adicionados à base de dados e novas percepções eram adquiridas. Economias projetadas que tinham sido previstas antes no desenvolvimento estavam, de fato, ocorrendo. Dois dos seis projetos experimentais completados desenvolveram *software* durante 1991 com reservas significativas intactas. Ambos os projetos eram grandes, com custos, só de *software*, na faixa de \$17.000.000,00, e ambos completados um pouco à frente do prazo. Um estava 4% abaixo do orçamento e o segundo, 6%. Quando o último projeto foi entregue ao cliente, a Raytheon recebeu um prêmio de incentivo de programa de \$9.600.000, que não é incluído em qualquer dos cálculos acima do retorno sobre o investimento.

4.3.2 Produtividade de *Software*

A empresa continuou a atualizar a importância da produtividade média conforme projetos eram adicionados à seu banco de dados; agora tinham dados sobre 24 projetos, nem todos completos. Os últimos resultados estão refletidos na Figura 4.6, que mostra um aumento da produtividade média de aproximadamente 170% sobre o período da iniciativa. A Figura 4.4 não inclui contas múltiplas de *software* capturado em múltiplas versões para outros clientes - todos os programas são uma simples versão do sistema.

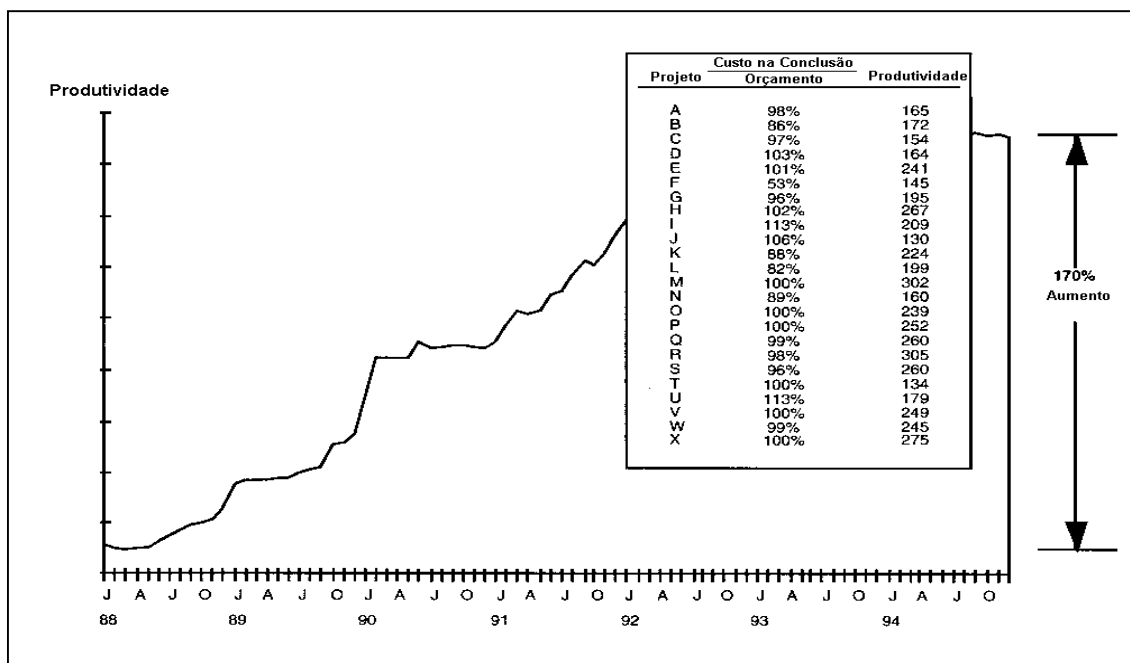


FIGURA 4.4 - AUMENTO DA PRODUTIVIDADE DE SOFTWARE VERSUS TEMPO
FONTE: HALEY, T. *et al.* (1995, p. 53).

Comparar estes 24 projetos com o passar do tempo é um modo válido para avaliar a melhoria de produtividade porque, embora todo projeto seja único, eles são todos do mesmo tipo (tempo real embutido) e com um alcance de tamanho razoável, de 70.000 a 500.000 DSI's (*Delivered Source Instructions* - Linhas de Código-Fonte Distribuídas). Assim, se nem a natureza da aplicação nem o método de medida não muda neste tempo, é razoável creditar a melhoria ao que mudou, isto é, o processo. Os cálculos de produtividade incluem engenharia (planejamento de *software*, código, teste de unidade e integração do *software*), preparação e geração de documento (SDD - *software design document*, IDD - *interface design document*, pastas de desenvolvimento de *software*), prototipagem, gerenciamento de configuração de *software*, gerenciamento de *software*, administração, e resolução de problemas de *software* relatados. Os cálculos de produtividade não incluem análise de exigências de *software* ou teste de qualificação formal

4.3.3 Índice de Desempenho de Custo

Outra preocupação era se a RES estava desempenhando sobre os projetos. O assunto era tratado coletando dados nos custos orçados dos projetos (previstos) e o custo atual na conclusão (CAC - *Cost At Completion*). Esta proporção do índice de desempenho de custo (CAC/Orçamento) para cada projeto era então usada para computar a média mensal da importância (novamente usando a mesma abordagem quanto ao custo de qualidade) para render um diagrama desta medida de tempo variável. Os resultados eram encorajadores, mostrando que o índice de desempenho de custo era dramaticamente melhorado sobre os 40% excedidos da faixa anterior para o começo da Iniciativa, até a faixa de 3% para 1991 (quando atingiram o Nível 3 do SEI/CMM) e continuando pelo tempo presente, como mostra a Figura 4.5.

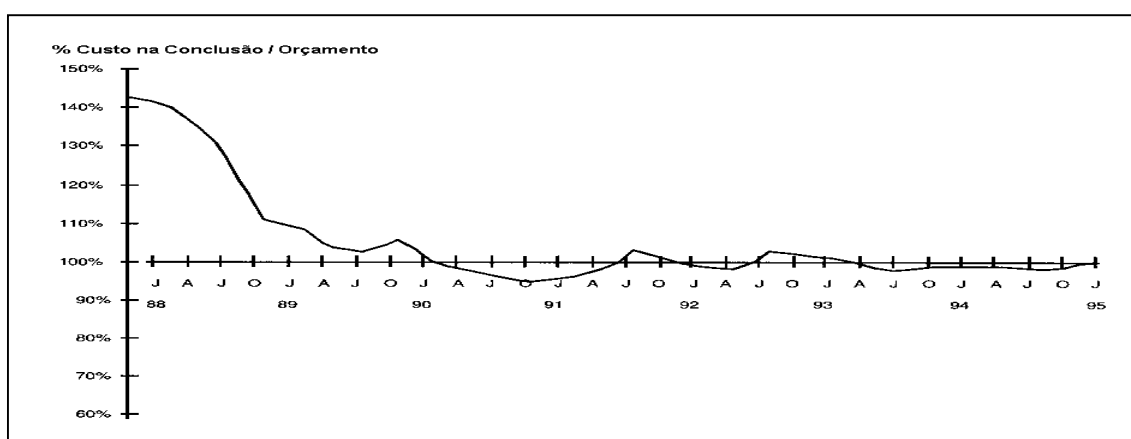


FIGURA 4.5 - ALCANÇANDO A PREVISIBILIDADE DE PROJETO
 FONTE: HALEY, T. *et al.* (1995, p. 55).

4.3.4 Qualidade Global do Produto

A medida primária usada para avaliar a qualidade global do produto é a densidade de defeito no produto de *software* final. Mede-se este fator em

"número de relatórios de problemas de *software* (STR - *Software Trouble Reports*) por milhares de linhas de código-fonte distribuídas (STR/KDSI)" sobre uma base de projeto individual. A densidade de defeitos do projeto são então combinadas para computar a média da importância mensal rendendo desse modo um diagrama de tempo variável da medida da qualidade global do produto. Conforme mostrado na Figura 4.6, os dados coletados sobre o período da iniciativa representam uma melhoria de uma média de 17,2 STRs/KDSI para o nível atual de 4,0 STRs/KDSI.

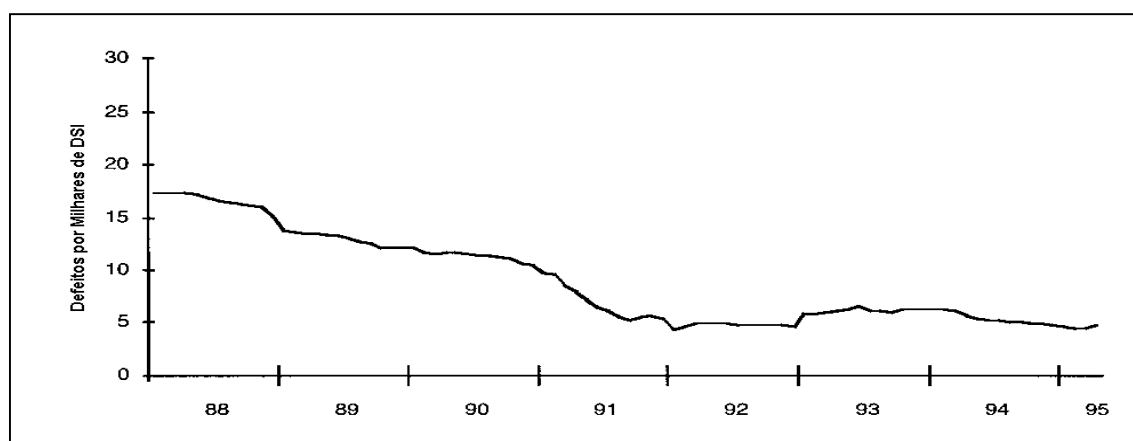


FIGURA 4.6 - DENSIDADE DE DEFEITO VERSUS TEMPO

FONTE: HALEY, T. *et al.* (1995, p. 56).

O processo na SEL (*Software Engineering Laboratory*) engloba 1.200 pessoas. O SEPG (*Software Engineering Process Group*) da RES tem suportado sua grande comunidade de *software* numa variedade de modos. Materiais de treinamento têm sido compartilhados com múltiplas entidades da Raytheon (*Computer Sciences Raytheon* e *Raytheon Canada*) e com clientes da Raytheon, incluindo o programa PRISM da *Air Force Electronic Systems Center* e as autoridades da aviação da Noruega e Alemanha. Foi utilizado o processo da empresa para avaliar e fazer *benchmarking* de processos de subcontratados em potencial. Fluiu-se esse processo aos subcontratados, e em

algumas instâncias, a equipe da RES foi colocada em equipes de contratados devido à sua iniciativa e habilidade para migrar o processo aos principais membros da equipe e outros.

A experiência com o SEPG tem sido compartilhada com a grande comunidade de *software* e filiados do SEI num número de foros, incluindo *boards* de aconselhamentos, *workshops*, *briefings*, e grupos de correspondência.

4.3.5 Pessoal

Quanto às características de projeto que cuidadosamente foram trilhadas para avaliar o sucesso da melhoria de processo (custo de retrabalho, índice de desempenho de custo, e qualidade global do produto), a equipe da RES vê menos tangível mas igualmente importantes os resultados ocorrendo em áreas que afetam seu pessoal.

As métricas discutidas anteriormente quantificam o excelente desempenho de seus engenheiros de *software* que resulta a estes, a satisfação de trabalho e aumento na carreira que vem com próspero desempenho sobre programas dentro da RES. O real desafio está no gerenciamento que fornece o apoio adequado.

4.3.6 Um Sumário das Atividades

A *Raytheon* começou suas atividades de melhoria de processo de *software* em 1988, direcionando-se a melhorar na previsibilidade do custo e prazo dos componentes de *software* de suas principais áreas empresariais. A

RES escolheu o caminho da melhoria de processo, guiada pelo CMM, porque (1) ele faz sentido e (2) seus clientes suportaram esta abordagem. Esta escolha tem provado porque esta empresa tem tornado suas atividades de desenvolvimento de *software* previsíveis, mais produtivas, e capazes de produzir produtos de alta qualidade. Junto com isso, tem se tornado mais competitiva e conquistado negócios adicionais.

A iniciativa foi organizada dentro de um comitê executivo responsável por dirigir e observar, e dentro de quatro Grupos de Trabalho SEPG - cada um responsável por uma área maior na iniciativa. O Grupo de Política e Procedimentos inicialmente capturou e documentou as melhores práticas, tanto que elas puderam ser aplicadas em todos os projetos. O Grupo de Treinamento elevou a importância de treinamento de caótico a um total compromisso da organização de *software* para assegurar que cada projeto tenha seus engenheiros totalmente treinados antes de começar o trabalho. O Grupo de Ferramentas e Métodos desenvolveu as tecnologias (ferramentas CASE - *Computer Aided Software Engineering*, estações de trabalho) e os métodos (Ada, orientação à objeto). O Grupo de Banco de Dados de Processo desenvolveu o processo e controle estatístico de processo e métricas de qualidade para avaliar o desempenho dos projetos e processos. Estes grupos de trabalho confeccionaram o processo a ser efetivo dentro da cultura da RES.

Hoje, as demandas de negócio estão mudando com a aceleração das tecnologias de hardware e *software*. Por causa dessas circunstâncias, o processo de *software* dessa organização mudou, tanto que continuam a distribuir soluções efetivas dentro do contexto da Iniciativa de Engenharia de

Software. Estão adotando a tecnologia e processos desenvolvidos por seu laboratório RAPID (*Raytheon Advanced Prototyping, Integration, and Demonstration*) e institucionalizando-os dentro da estrutura de organização da Iniciativa. Os processos para avaliação efetiva de prototipação, sistemas de *software*, uso de técnicas de orientação à objetos, e reuso de domínio específico, estão ficando tão padronizados quanto inspeções de código dentro da RES.

4.4 O Uso do Modelo CMM e ISO 12207 na Celepar (MACHADO *et al.*, 1997)

Este estudo de caso relata a experiência da Celepar - Companhia de Informática do Estado do Paraná, órgão responsável por alavancar o processo de informatização no âmbito dos serviços públicos do Estado do Paraná.

Em decorrência deste papel, a Celepar necessita dominar, com destreza, o processo de produção de *software*. Assim, na busca deste diferencial, vários esforços vêm sendo desenvolvidos, principalmente com foco no estabelecimento de um processo metodológico de produção de *software*. A definição deste processo tem apresentado diversas dificuldades, mas duas merecem uma ênfase especial. A primeira dificuldade diz respeito ao adequado entendimento de quais devem ser os componentes deste processo metodológico. É comum a abordagem do processo metodológico com uma ênfase maior na questão de engenharia do produto, relegando outros aspectos como, por exemplo, a contratação e o acompanhamento de projetos de *software*. A segunda dificuldade refere-se ao entendimento de qual deve ser o procedimento adotado na implementação deste processo. Devem ser

observados, principalmente, aspectos ligados à possibilidade de assimilação, pela Celepar, dos diversos componentes identificados como necessários na composição de um modelo metodológico de desenvolvimento de *software*. Neste contexto foi identificada a oportunidade de se utilizar o modelo CMM.

4.4.1 O Contexto da Celepar

A Celepar possui algumas características relevantes para a definição de um processo metodológico de desenvolvimento de *software*. Dentre estas, cabe destacar o elevado volume de atividades de manutenção, a recente prática de contratação de terceiros para atividades de desenvolvimento e manutenção de *software*, a utilização de diversas tecnologias e a existência de equipes de projetos e clientes com diferentes aspectos culturais. Todos estes fatores somam-se ao porte da sua equipe de desenvolvimento, que conta com aproximadamente 120 analistas de informática (analistas de sistemas e programadores).

Este cenário traz variáveis que são determinantes na definição e consolidação de um processo metodológico. Esta consolidação vem sendo buscada a partir das experiências, observações e vivências, levando ao desenvolvimento de processo metodológico flexível, adaptável e abrangente, com ênfase em aspectos de organização e gestão do processo de desenvolvimento de *software*.

4.4.2 Análise da MDS frente ao Modelo CMM

À partir de modelo definido de desenvolvimento de serviços, da Celepar, foi realizada uma análise, não formal, da MDS (Modelo de Desenvolvimento de Serviços) frente ao modelo CMM, que indicou estarem centrando esforços em áreas-chave pertencentes ao nível 2 (Disciplinado). Portanto, a análise está focada em relação a este nível, conforme descrito a seguir.

Gerência de Requisitos: A MDS atende parcialmente a esta área-chave, através do Roteiro de Projeto Preliminar (RPRE) e do Roteiro de Acompanhamento (RPAC), estabelecendo um procedimento de acompanhamento com o cliente. Entretanto, não existem procedimentos definidos formalmente para o gerenciamento de mudanças destes requisitos.

Planejamento do Projeto de Software: A MDS atende totalmente a esta área-chave, através do Roteiro de Projeto Preliminar (RPRE).

Supervisão e Acompanhamento do Projeto de Software: A MDS atende totalmente a esta área-chave, através do Roteiro de Acompanhamento de Projetos (RPAC).

Garantia de Qualidade de Software: A MDS prevê alguns procedimentos em nível de verificação no Roteiro de Revisão (RREV). Outros procedimentos deste roteiro superam o previsto para esta área-chave.

A MDS não contempla formalmente as áreas-chave de Gerência de Subcontratação de Software e Gestão de Configuração de Software.

4.4.3 Proposta de Evolução da MDS

O modelo CMM auxiliou na definição de "como" podem ser implantados os diversos componentes do processo, respeitando a possibilidade de assimilação desta questão pela Celepar, de acordo com seu nível de maturidade. Esta visão é complementar e possibilitada pela norma ISO 12207, uma vez que alguns processos nela referenciados possuem uma abrangência que não seria suportada pela Celepar num primeiro momento. Assim, definiu-se como objetivo da empresa a consolidação do processo metodológico para as áreas-chave do nível 2 do modelo CMM.

Deste estudo foram extraídos importantes subsídios para um redirecionamento dos esforços metodológicos, cabendo destacar:

- Necessidade de alguns ajustes no Roteiro de Projeto Preliminar (RPRE), deixando mais claros alguns aspectos de especificação de requisitos de sistema;
- Necessidade de revisão do Roteiro de Revisão (RREV), entendendo que este possui características do nível 3 do modelo CMM, sendo incompatível com atual realidade;
- Necessidade de consolidação de um processo de Garantia da Qualidade de *Software* compatível com o nível 2 do modelo CMM e balizado pela norma ISO 12207;
- Necessidade de definição de um roteiro de Subcontratação de *Software* balizado pelo modelo CMM e pelo processo de Aquisição da norma ISO 12207; e

- Necessidade de definição de um projeto de Gerência de Configuração, balizado pelo modelo CMM e pela norma 12207-2 - Gerência de Configuração.

4.4.4 A Conclusão da Celepar

A utilização do CMM como modelo de referência de evolução da MDS-Celepar mostrou-se bastante eficiente por auxiliar na verificação dos processos a serem implementados na empresa, isto é, na sequência de sua implementação e na amplitude destes processos. Os processos que, pelo CMM, eram pertencentes ao nível 3 e que estavam definidos na MDS, eram difíceis de serem assimilados pelos técnicos, ocasionando uma utilização não efetiva destes processos. Os processos que eram pertencentes ao nível 2 eram aqueles que os técnicos entendiam facilmente e eram utilizados pela empresa de forma mais homogênea.

A utilização simultânea da norma ISO 12207 e do modelo CMM parecia bastante favorável, uma vez que a norma apresenta uma boa referência para a consolidação do modelo metodológico e do modelo CMM mostra uma excelente referência para o direcionamento dos esforços necessários a esta consolidação.

A adoção desta norma e modelo trouxe para a Celepar um plano de melhoria e evolução do processo metodológico, associado a uma linguagem comum no tratamento deste processo. São sólidas referências, pois decorrem de esforços internacionais na busca de um patamar de excelência na produção de *software*.

4.5 Uma Experiência com a Norma ISO 15504 (SALVIANO *et al.*, 1999)

Este estudo de caso relata a experiência prática de implementação das fases iniciais da melhoria de processo de *software*, baseada na futura Norma ISO 15504 (SPICE). Estas fases incluem o início dos trabalhos, a avaliação de processos e o planejamento da melhoria. Tal experiência foi realizada em uma organização de *software* brasileira, pela Fundação Centro Tecnológico para Informática (CTI).

A trajetória desta empresa é típica das empresas nacionais de sucesso no Brasil: inicialmente uma empresa pequena, com um único produto e um estilo informal de trabalho que se mostrou eficiente para o seu crescimento. Com este crescimento, a empresa passou a comercializar múltiplos produtos, precisando então de um estilo mais sistematizado para continuar a desenvolver e manter produtos de qualidade com custos e prazos compatíveis com a demanda do mercado. A abordagem de melhoria de processo é uma forma eficiente e já experimentada para guiar esta evolução.

4.5.1 Projeto de Melhoria dos Processos de *Software* da Empresa

O processo de melhoria escolhido para este projeto foi a ISO 15504-5 (SPICE). O objetivo principal da melhoria de processo desta empresa é continuar a evolução de uma empresa pequena, para se tornar uma empresa média. O objetivo principal deste projeto de melhoria foi a construção de um plano para a melhoria do processo de *software* da organização. Além deste objetivo principal, outros objetivos também foram considerados: avaliar as

práticas correntes da empresa com relação a um modelo de processo; e capacitar os seus funcionários em melhoria de processos de *software*.

4.5.2 Execução da Avaliação dos Processos e Planejamento da Melhoria

Cada fase executada no Projeto foi precedida pela elaboração de um plano de trabalho e a preparação do material necessário. Cada plano de trabalho incluiu a agenda da fase, com a relação das atividades a serem executadas. Cada atividade teve uma descrição, o número de horas previsto, o período e horário a ser realizado, as pessoas envolvidas e a infra-estrutura necessária.

A primeira fase teve como resultado principal o Plano da Avaliação das Práticas Correntes. A abrangência desta avaliação é caracterizada pelos processos a serem avaliados, as instâncias de processo e os níveis de capacidade. A escolha dos processos foi diretamente influenciada pelos seguintes subsídios:

- Posicionamento dos processos da ISO 15504 em um quadro de riscos (baixo, médio e alto), com duas dimensões: importância do processo para a organização e probabilidade da execução deficiente do mesmo.
- Mapeamento das respostas aos processos da ISO 15504, obtidas a partir de questionários que visavam opiniões sobre vários aspectos da organização, incluindo pontos fortes e fracos, experiências bem e mal sucedidas, sugestões para melhoria, ameaças e oportunidades.
- Experiências das outras organizações.
- Restrições da avaliação.

Com estes subsídios, foram analisados os dados coletados e escolhidos cinco processos da ISO 15504, conforme mostra a Tabela 4.2.

TABELA 4.2 PROCESSOS SELECIONADOS DA ISO 15504

FONTE: SALVIANO, C.F. *et al.*, (1999), adaptado pelo autor.

Processos	Descrição
Suporte ao Cliente	Estabelecer e manter um nível aceitável de serviços ao cliente/usuário que apoie o uso efetivo do produto de <i>software</i> .
Garantia da Qualidade	Garantir que os processos e produtos de trabalho de um processo ou projeto satisfaçam seus requisitos específicos e sejam consistentes com seus planos estabelecidos.
Gerenciamento de Projeto	Identificar, coordenar e, acompanhar atividades, tarefas e recursos necessários para um projeto produzir um produto ou serviço que satisfaça os requisitos.
Alinhamento Organizacional	Garantir que os indivíduos da organização compartilhem de uma mesma visão, cultura e o entendimento dos objetivos de negócio para que possam executar suas funções de modo mais efetivo.
Estabelecimento de Processo	Estabelecer um conjunto de processos organizacionais para todos os processos do ciclo de vida de <i>software</i> que se aplicam às atividades de negócio.

O resultado da avaliação dos processos da organização forneceu subsídio para corroborar a escolha dos processos da ISO 15504, principalmente pelos critérios de probabilidade de execução deficiente e dos relacionamentos com pontos fracos da organização.

A segunda fase teve como objetivo principal a avaliação das práticas correntes da organização, focando a avaliação de exemplos de execuções de processos selecionados em relação aos níveis de capacidade selecionados do modelo de processo.

A terceira fase teve como objetivo principal a elaboração de um Plano de Melhoria, que é o resultado principal da fase.

A quarta fase teve como objetivo a realização de um *workshop* interno, para a definição final do Plano de Melhoria, com base nos comentários produzidos na revisão.

Segundo SALVIANO *et al.* (1999), o resultado principal, que é a melhoria da organização, ainda não havia sido alcançado, porque depende da implementação do plano. Haviam, porém, indicadores que sinalizavam a qualidade do projeto realizado até o momento, incluindo a conclusão do trabalho, respostas a questionários sobre o trabalho na organização e revisões do trabalho.

4.6 Considerações Finais

Cinco casos foram analisados, sendo três ocorridos entre EUA e Canadá, e dois ocorridos no Brasil. Os três primeiros casos apresentaram um bom grau de detalhamento, o que não aconteceu nos dois últimos casos, ocorridos no Brasil. Os três primeiros casos apresentaram informações referentes a resultados de implementações de modelos e, os dois últimos casos, referentes a diagnóstico. No entanto, nenhum deles apresentou um descritivo sobre as implementações de modelos de SPI.

Considerando que este trabalho visa contribuir com o meio acadêmico e profissional brasileiro, seria importante que se tivesse casos ocorridos no Brasil com um grau de detalhamento maior do que o que foi apresentado neste capítulo. No entanto, isso não foi possível devido ao fato de as empresas envolvidas considerarem essas informações de teor estratégico e, portanto, confidenciais (sigilosas).

Estas empresas, em sua totalidade, e até mesmo algumas instituições como o CTI - Centro Tecnológico para Informática, e os departamentos de Engenharia de Produção e Ciência da Computação - USP, integram um grupo denominado SPIN (*Software Process Improvement Network* - Grupo de Melhoria do Processo de *Software*), que reúnem-se periodicamente para discutir diversos aspectos de melhoria, como estes apresentados neste capítulo, e experiências e soluções relacionadas a melhoria do desenvolvimento de *softwares*. Este grupo envolve, ainda, organizações como NEC do Brasil, Xerox do Brasil, Credicard, CityBank e Unibanco, entre outras, todas em fase de implementação do modelo CMM, que cedem seus espaços para estas reuniões. O autor desse trabalho, mesmo sendo membro deste grupo, encontrou dificuldades no levantamento de estudos de casos ocorridos no Brasil. O custo de implementação de modelos de SPI, foi uma das informações, considerada das mais importantes, em que não se obteve êxito em seu levantamento.

Como ilustração, a Xerox do Brasil implementa, atualmente, as práticas do Nível 5 do CMM, enquanto que a NEC do Brasil está em fase de conclusão do Nível 3 do CMM.

5 ANÁLISE DOS CASOS ESTUDADOS

Uma vez perfilados cinco casos de implementações de modelos de SPI, este capítulo visa traçar uma análise desses casos, destacando aspectos de melhoria pertinentes a essas implementações.

Inicialmente, serão destacados os principais aspectos de melhoria observados nos cinco casos. Em seguida, serão destacados alguns desses aspectos que possuem relação, também observadas nos casos. Finalmente, uma análise de outras possíveis relações entre esses aspectos, é feita nesse trabalho.

5.1 Comparação entre os Casos Estudados

Considerando que os casos estudados no capítulo anterior avaliaram aspectos de melhoria sob diferentes visões, a Tabela 5.1 sistematiza esses aspectos relativos a cada caso. São eles: Retrabalho, Reuso de Código, "Turnover", Satisfação do Cliente, Densidade de Defeito, Riscos do SPI, Cumprimento de Prazos e Custos, Produtividade, Qualidade do Produto, Moral do Pessoal, Conformidade do Produto, Alinhamento Organizacional e, Qualidade do Processo.

É importante destacar que os aspectos Qualidade do Produto e Qualidade do Processo são considerados, neste trabalho, como aspectos que possuem implícitos outros aspectos, isto é, englobam outros aspectos de melhoria e, portanto, são aspectos resultantes. No entanto, são analisados

separadamente, na tabela a seguir, por terem sido avaliados dessa forma em alguns dos casos estudados.

Os casos 1 e 2 relatam os resultados de pesquisas realizadas com empresas que passaram pela avaliação de modelos de SPI. O caso 3 relata a experiência de uma única empresa na implementação de um modelo de SPI e seus resultados. Por esse motivo, os aspectos de melhoria descritos para os Casos 1, 2 e 3, ocorridos no exterior, são mais detalhados por analisarem os resultados da implementação de modelos.

O caso 4, ocorrido no Brasil, descreve um diagnóstico para implementação de um modelo de SPI e da norma ISO 12207, analisando pontos em comum entre o modelo metodológico já existente e os novos modelos a serem implementados. Contudo, a empresa não esclarece detalhes das melhorias obtidas, não propiciando segurança a uma análise mais consistente.

No caso 5 a empresa pesquisada solicitou sigilo de suas informações por considerá-las estratégicas, restringindo-se a descrever um diagnóstico para implementação do modelo.

O caso 1, de acordo com o relatado, explicitamente, obteve melhoria nos aspectos Retrabalho, Reuso de Código, *Turnover*, Densidade de Defeito, Riscos do SPI, Cumprimento de Prazos e Custos, Produtividade e, Moral do Pessoal.

O caso 2, explicitamente, obteve melhoria nos aspectos Retrabalho, Satisfação do Cliente, Cumprimento de Prazos e Custos, Qualidade do Produto, Moral do Pessoal, Alinhamento Organizacional.

O caso 3, explicitamente, aponta melhorias nos aspectos Reuso de Código, Densidade de Defeito, Cumprimento de Prazos e Custos, Produtividade, Alinhamento Organizacional, Qualidade do Processo.

Finalmente, o caso 4 aponta a intenção de melhoria no aspecto Alinhamento Organizacional, através do uso de um modelo de SPI.

O autor deste trabalho considera que aspectos como Satisfação do Cliente, Cumprimento de Custos e Prazos, Densidade de Defeito e Conformidade do Produto, são tidos como metas primárias na busca da qualidade de software e, portanto, mesmo não tendo sido abordados "explicitamente" por alguns dos casos, considera que tenha havido algum esforço na busca da melhoria para estes aspectos.

TABELA 5.1 ASPECTOS DE MELHORIA VERSUS ESTUDO DE CASOS

Estudo de Casos X Aspectos de Melhoria	Caso 1: modelo não informado	Caso 2: CMM	Caso 3: CMM	Caso 4: ISO 12207 e CMM	Caso 5: ISO 15504
Retrabalho	A redução do retrabalho ocasionou uma redução nos custos de desenvolvimento	-	A redução do custo de retrabalho caiu de 41% para 20%, após a implementação do modelo	-	-
Reuso de Código	O reuso de códigos ocasionou a redução de retrabalho, desenvolvimento, esforço e, conseqüentemente, de custos. Melhoria da produtividade, qualidade do produto e satisfação do cliente. O código reusado apresentou menos defeito/KLOC do que novos códigos	As organizações pesquisadas são questionadas quanto ao reuso, no entanto, este caso não aponta evoluções ou involuções para este item	A criação de biblioteca de códigos reduziu os Prazos de entrega de produtos	-	-

TABELA 5.1 ASPECTOS DE MELHORIA VERSUS ESTUDO DE CASOS (CONTINUAÇÃO)

Estudo de Casos X Aspectos de Melhoria	Caso 1: modelo não informado	Caso 2: CMM	Caso 3: CMM	Caso 4: ISO 12207 e CMM	Caso 5: ISO 15504
Turnover	A redução do <i>turnover</i> e retreinamento puderam pagar sozinhos os custos de melhoria.	O <i>turnover</i> de gerenciamento sênior representa uma barreira para a melhoria, e o <i>turnover</i> de gerenciamento médio e pessoal técnico não parece ser importante.	-	-	Aumento no quadro de desenvolvedores, mas sem relato de melhora ou piora no processo.
Satisfação do Cliente	Dificuldade em estimar esta melhoria	Há uma inexplicável queda deste item no nível 2. Ainda assim, a pesquisa relata um melhor desempenho para este item no nível seguinte.	Mudança do processo em função de exigências do cliente.	-	Adoção do processo de Suporte ao Cliente, da ISO/IEC 15504.
Densidade de Defeito	Redução de 80%, ocasionou a redução de custo e tempo para encontrar defeitos.	Item não abordado nesta pesquisa.	Redução na densidade de defeitos do <i>software</i> .	-	Implícito no item de qualidade do produto.

TABELA 5.1 ASPECTOS DE MELHORIA VERSUS ESTUDO DE CASOS (CONTINUAÇÃO)

Estudo de Casos X Aspectos de Melhoria	Caso 1: modelo não informado	Caso 2: CMM	Caso 3: CMM	Caso 4: ISO 12207 e CMM	Caso 5: ISO 15504
Riscos do SPI	Probabilidade de risco baixa, tendo reduzido em 91%. Riscos são associados a perda de empregados-chave, alto <i>turnover</i> de empregados, e entrega atrasada de produtos (ver Tabela 5.2)	A disposição da gerência para aceitar riscos parece não Ter importância, porém, este item é considerado como fator de sucesso, por 54% dos envolvidos no processo de SPI.	Uma das atividades críticas do modelo é identificar riscos e desenvolver uma estratégia de mitigação para cada um deles.	A escolha dos processos foi influenciada pela probabilidade (baixo, médio e alto) de execução deficiente dos mesmos.	Item implícito no Plano de Melhoria da empresa.
Cumprimento de Prazos e Custos	Redução de: Prazos - 37%. Custos de desenvolvimento -73%. Custos de manutenção - 95%.	Há uma progressão na melhora destes itens, conforme avançam os níveis.	Conclusão de projetos antes do término do prazo. Redução nos custos, principalmente, em função da redução de retrabalho.	Item integrante no modelo metodológico da empresa.	Item integrante do Plano de Melhoria da empresa.
Produtividade	Aumento da produtividade da equipe de desenvolvimento em 222%.	-	Aumento de 170% no período de 7 anos, com 24 projetos envolvidos.	-	-

TABELA 5.1 ASPECTOS DE MELHORIA VERSUS ESTUDO DE CASOS (CONTINUAÇÃO)

Estudo de Casos X Aspectos de Melhoria	Caso 1: modelo não informado	Caso 2: CMM	Caso 3: CMM	Caso 4: ISO 12207 e CMM	Caso 5: ISO 15504
Qualidade do Produto	Item implícito nos aspectos de melhoria referentes a produtos de <i>software</i> .	Os resultados da pesquisa comprovam que quanto maior o nível, maior a Qualidade do produto.	Item integrante dos demais aspectos de melhoria.	-	Adoção do processo de Garantia da Qualidade, da ISO/IEC 15504.
Moral do Pessoal	Devido a redução de <i>turnover</i> , houve um aumento no moral e confiança dos desenvolvedores.	Pequeno progresso na melhora para este item, conforme avançam os níveis.	-	-	-
Conformidade do Produto	Item implícito na pesquisa. Não abordado especificamente.		Estabelecimento de métodos de verificação que garantam que o <i>software</i> satisfaça toda e cada exigência do projeto e do cliente.		Adoção do processo de Gerenciamento de Projeto, da ISO/IEC 15504.

TABELA 5.1 ASPECTOS DE MELHORIA VERSUS ESTUDO DE CASOS (CONTINUAÇÃO)

Estudo de Casos X Aspectos de Melhoria	Caso 1: modelo não informado	Caso 2: CMM	Caso 3: CMM	Caso 4: ISO 12207 e CMM	Caso 5: ISO 15504
Alinhamento Organizacional	-	A cultura organizacional está fortemente relacionada ao sucesso da melhoria de processo. O excesso de políticas tem sido visto como prejudicial.	A implementação do modelo apoiou-se na cultura da organização, que era favorável. Resistências foram vencidas através de <i>workshops</i> e treinamentos.	Dificuldade de assimilar processos do nível 3. Processos do nível 2 são facilmente assimilados. A definição de processos pela norma 12207 foi bem aceita.	Adoção do processo Alinhamento Organizacional, da ISO/IEC 15504
Qualidade do Processo	Item implícito na pesquisa. Não abordado especificamente.		O uso do novo processo estabelecido no nível 3 resultou numa redução de prazos	A empresa compatibiliza-se com a ISO 12207. Consolidação do processo metodológico da empresa para as KPA's do nível 2.	Criação dos Grupos de Garantia da Qualidade (GGQ) e Estabelecimento e Melhoria do Processo (GEP).

5.2 Considerações a respeito do uso de modelos de SPI

Considerando a análise efetuada na seção anterior é possível identificar algumas relações entre os aspectos de melhoria, resultantes, portanto, dos casos estudados, como mostra a Tabela 5.2, a seguir, onde procurou-se identificar as evidências de cada caso. Essas evidências ou aspectos têm relação com as etapas do CMM e foram, alguns explicitamente e outros implicitamente, apontados pela pesquisa do Ministério da Ciência e Tecnologia, no Capítulo 1.

A Tabela 5.2 visa determinar dentre os aspectos de melhoria abordados, quais apresentam-se como possível "causa" e quais apresentam-se como possível "efeito", considerando somente as relações indicadas nos casos estudados.

Com relação ao aspecto Qualidade do Processo, o mesmo não foi incluído pois está relacionado com todos os aspectos de melhoria, o mesmo valendo para o aspecto Qualidade do Produto, relacionado diretamente com os aspectos Reuso de Código, Densidade de Defeito e Conformidade do Produto, por considerá-lo implícito nestes aspectos.

O aspecto Alinhamento Organizacional não foi incluído na Tabela 5.2, por não relacionar-se explicitamente a nenhum outro aspecto de melhoria. No entanto, nota-se que este aspecto é um forte fator de sucesso para a melhoria do processo de *software*, pois os indivíduos da organização compartilham de uma mesma visão, cultura e entendimento dos objetivos de negócio, executando melhor suas funções. Deve-se considerar, também, que este aspecto pode ser afetado pela variação de desempenho de outros aspectos, como o "Turnover".

TABELA 5.2 RELAÇÃO ENTRE ASPECTOS DE MELHORIA APONTADOS NO ESTUDO DE CASOS

Causa X Efeito	Retrabalho	Reuso	Turnover	Densidade de Defeito	Cumprimento de Prazos e Custos
Retrabalho		O reuso de códigos resultou em redução de retrabalho		Redução de defeitos ocasionou redução no retrabalho	
Cumprimento de Prazos e Custos	Redução dos custos de desenvolvimento	Redução de prazos de entrega de produtos e custos de desenvolvimento	Redução de custos de contratação e treinamento	Redução de custo e tempo para encontrar defeitos	
Moral do Pessoal			A redução de "turnover" ocasionou um aumento no moral e confiança dos desenvolvedores		
Produtividade		Melhoria da produtividade			
Conformidade do Produto		Melhoria da Qualidade do Produto		Melhoria da Qualidade do Produto	
Riscos do SPI			Riscos são associados a perda de empregados-chave e alto "turnover"		Riscos são associados a atrasos na entrega do produto

Sem a necessidade de um estudo mais detalhado dos casos e relações apresentadas neste capítulo, é possível verificar que o aumento de melhoria é praticamente inevitável quando implementados modelos de melhoria do processo de *software* (SPI).

Analisando mais detalhadamente os aspectos de melhoria é possível verificar que a variação de seus desempenhos influenciam o desempenho de outros aspectos, conforme observado na Tabela 5.2.

Basta observar que a redução do *Retrabalho* ocasionou a redução nos custos de desenvolvimento, influenciando o aspecto *Cumprimento de Prazos e Custos*.

A prática do *Reuso* influenciou os aspectos *Retrabalho*, *Cumprimento de Prazos e Custos*, *Produtividade* e *Conformidade do Produto*, pois ocasionou a redução de retrabalho, dos prazos de entrega do *software* e dos custos de desenvolvimento. Ocasinou, também, uma melhoria da produtividade e da qualidade do produto.

A redução do *Turnover* ocasionou redução nos custos de contratação e treinamento, influenciando, assim, o aspecto *Cumprimento de Prazos e Custos*. Ocasinou, também, um aumento no moral e confiança dos desenvolvedores, influenciando o aspecto *Moral do Pessoal*. O alto *Turnover* pode representar *Riscos ao SPI*.

A *Densidade de Defeito* influenciou os aspectos *Retrabalho*, *Cumprimento de Prazos e Custos* e *Conformidade do Produto*, por ocasionar uma redução no retrabalho e no custo e tempo para encontrar defeitos. Essa redução ocasionou, também, uma melhora da qualidade do produto.

Finalmente, o não *Cumprimento de Prazos e Custos* representa *Riscos ao SPI*. Assim, conforme verificado anteriormente, a variação do desempenho dos aspectos de melhoria influenciam o desempenho de outros aspectos.

Conforme descrito anteriormente, a Tabela 5.2 visa confrontar os aspectos de melhoria que se relacionam nos casos estudados. No entanto, vale uma análise mais detalhada, onde serão consideradas as relações passíveis de acontecer, conforme mostrado na Tabela 5.3. Ou seja, a Tabela 5.3 apenas mostrará outras relações entre aspectos de melhoria, ainda não apontadas até o momento.

Vale acrescentar que as relações entre os aspectos apresentados na Tabela 5.3 são, também, o resultado de uma simulação realizada com alunos do terceiro ano técnico de informática, da Organização Einstein de Ensino, na disciplina Gestão & Qualidade, ministrada pelo autor desse trabalho. A simulação consistia em apontar o máximo de relações possíveis entre aspectos de melhoria, justificando-os. Relações consistentes foram filtradas e anexadas a este estudo.

TABELA 5.3 OUTRAS RELAÇÕES ENTRE OS ASPECTOS DE MELHORIA

Causa X Efeito	Retrabalho	Reuso de Código	Turnover	Cumprimento de Custos e Prazos	Moral do Pessoal	Densidade de Defeito	Conformidade do Produto	Alinhamento Organizacional
Retrabalho						X	X	
Reuso de Código								X
Satisfação do Cliente		X		X			X	
Riscos do SPI	X	X			X	X		X
Produtividade	X		X		X			X
Densidade de Defeito		X			X			
Conformidade do Produto						X		
Alinhamento Organizacional			X		X			

Baseado na Tabela 5.3, é possível descrever, além das relações já descritas, outras relações:

- O Retrabalho representa Risco ao SPI, e pode afetar a produtividade.
- O Reuso de códigos pode reduzir a probabilidade de riscos ao SPI e a densidade de defeito, e contribui para a satisfação do cliente.
- O alto *Turnover* pode reduzir a produtividade e, ainda, afetar o alinhamento organizacional.
- O Cumprimento de Prazos representa a satisfação do cliente.
- O baixo Moral do Pessoal pode representar risco ao SPI, pois compromete a produtividade e o alinhamento organizacional e, conseqüentemente, pode aumentar a densidade de defeito.
- O aumento na Densidade de Defeito, compromete a conformidade do produto e ocasiona o retrabalho, representando risco ao SPI.
- A não Conformidade do Produto ocasiona o retrabalho e a insatisfação do cliente.
- O Alinhamento Organizacional pode institucionalizar o reuso de códigos e melhorar a produtividade e o processo. No entanto, problemas neste aspecto podem representar riscos ao SPI.

É difícil afirmar que a implementação de um modelo de SPI em empresas nacionais venha a ter o mesmo êxito que empresas estrangeiras, pelo fato de não apontarem, em seus relatos, suas evoluções ou involuções em seus processos de *software*. No entanto, desconsiderando o contexto cultural das empresas nacionais, a implementação de modelos de SPI tende a ter resultados tão satisfatórios quanto de empresas estrangeiras.

Os resultados apontados pelos casos 1, 2 e 3, sugerem uma idéia dos resultados que poderão vir a ocorrer com as empresas nacionais. Mesmo que insucessos aconteçam, estes servirão como contribuição para futuras implementações em outras empresas nacionais, ocasionando, talvez, uma adaptação desses modelos para nosso contexto cultural e profissional.

5.3 Considerações Finais

Um aspecto que poderia ser considerado adicionalmente, é com relação ao pessoal envolvido na implementação do modelo de desenvolvimento de *software*. Neste sentido, TEIXEIRA (2000) destaca em seu trabalho, as principais atividades, referentes ao CMM, para o sucesso de implementação de um modelo de desenvolvimento, onde diversos cuidados deveriam ser considerados para o sucesso da implementação, como por exemplo, comunicação, treinamento, criação de grupos, compensação, formação de líderes, etc.

É importante destacar que apesar da dificuldade de não se ter disponível mais casos que abordassem implementações de modelos de SPI, ainda assim, estabeleceu-se relações entre os aspectos mais importantes identificados nos casos, os quais permitiram por sua vez assimilá-los com as etapas do CMM.

6 CONCLUSÃO

Foram apontados inicialmente, neste trabalho, os aspectos de melhoria da indústria nacional de *software*, onde observou-se que a qualidade de *software* caminha a curtos passos no Brasil.

Baseado neste contexto, o Capítulo 2 destacou ferramentas gerenciais e operacionais utilizadas para a garantia da qualidade de *software*, algumas delas utilizadas por outros processos de produção, como o TQM, o QFD e as Sete Ferramentas da Qualidade. Apresentou, também, duas das principais normas ISO relacionadas à qualidade do processo de *software*, assim como, conceitos referentes ao assunto. O objetivo do Capítulo 2 foi o de apresentar um texto que pudesse embasar os capítulos seguintes, para um melhor entendimento.

O Capítulo 3 destacou três modelos de melhoria do processo de *software*, com o objetivo de apresentar um texto que pudesse melhorar o entendimento dos casos apresentados no capítulo seguinte. A análise comparativa realizada entre os modelos descritos neste capítulo e as normas ISO, descritas no Capítulo 2, possibilitou que fossem observadas as características positivas e negativas de cada um deles, tornando possível a melhor escolha, a quem possa interessar, de modelos e normas ligados à qualidade do processo de *software*.

Nos capítulos 4 e 5, respectivamente, foram descritos e analisados cinco casos de utilização de modelos de melhoria de processos de *software*, sendo que três deles, desenvolvidos nos EUA, enfocam mais os benefícios

dessa utilização, enquanto que os outros dois estão mais voltados para análise da situação do processo de desenvolvimento de *software* nas empresas onde foram implementados.

Neste ponto, é importante frisar que um dos objetivos deste trabalho é contribuir com o meio acadêmico e profissional brasileiro ao que se refere à qualidade de *software* e, para tanto, seria importante que houvessem, para estudo, casos ocorridos no Brasil com um grau de detalhamento maior do que o que foi apresentado no Capítulo 4.

Conforme descrito anteriormente, o levantamento dessas informações tornou-se dificultoso devido ao fato de empresas envolvidas considerarem essas informações sigilosas. Até mesmo instituições envolvidas com o assunto dificultaram o acesso a tais informações por razões obscuras. Apesar de o autor deste trabalho ser membro do SPIN - grupo formado por estudiosos e usuários de métodos, ferramentas e modelos de SPI, onde tais empresas e instituições também são membros, houve inúmeras dificuldades no levantamento de informações referentes a casos ocorridos no Brasil.

No Capítulo 5, foram destacados os aspectos considerados críticos de sucesso na implementação desses modelos a partir da comparação dos casos estudados. Neste capítulo, foram apontadas as relações causa-efeito ocorridas nos casos, entre os aspectos de melhoria apontados. Foram, também, consideradas relações causa-efeito passíveis de ocorrerem entre os mesmos aspectos. Isso permitiu identificar todas as relações possíveis, facilitando a inserção desses modelos no processo de desenvolvimento de *software* de empresas interessadas em melhorá-lo.

O que se pôde concluir das relações entre os referidos aspectos, é que a variação de seus desempenhos influenciam o desempenho de outros aspectos. Tal variação representa evoluções ou involuções num processo de *software*. Considerando os resultados apontados, no Capítulo 5, entre estes aspectos e, desconsiderando o contexto cultural, o que se conclui, é que a implementação de modelos de SPI tende a ter resultados satisfatórios tanto em empresas nacionais quanto estrangeiras.

Os aspectos apresentados no Capítulo 4 e comentados anteriormente neste capítulo, direta ou indiretamente, convergem para a racionalização de prazos e custos. O reconhecimento da melhor prática (Nível 2) leva ao reconhecimento do trabalho, respeita o lado artesanal na elaboração de *software* ao mesmo tempo que procura na sequência padronizá-los permitindo o estabelecimento de "blocos de construção" os quais servirão para a elaboração de novos *softwares*. Esse reuso de "blocos" devidamente documentados terão como novidade a cada elaboração a necessidade de desenvolvimento de novos "blocos" e/ou integração dos mesmos. Um bom acompanhamento na densidade de defeitos deve permitir um melhor gerenciamento no processo de desenvolvimento de *software*. Em outras palavras, uma mensuração tanto qualitativa quanto quantitativa. Uma forma de se acompanhar este aspecto é através do número de defeitos/KLOC (milhares de linhas de código-fonte). Na medida em que há um melhor acompanhamento no desenvolvimento, os prazos tendem a ser mais respeitados, com isso, respeitando um dos fatores críticos mais importantes na elaboração de *softwares*.

Sugere-se para estudos futuros:

- a implementação de um modelo de maturidade em uma organização ou instituição nacional, de forma mais abrangente que os casos apresentados.
- o uso de normas internacionais/métodos em conjunto com modelos de maturidade como apoio na implementação dos processos de *software*.
- a análise detalhada de outros modelos de maturidade que possam se enquadrar em nosso contexto cultural e profissional.

REFERÊNCIAS BIBLIOGRÁFICAS

- AKAO, Y. *Quality function deployment: integrating customer requirements into product design*. Cambridge: Productivity Press, 1990.
- AZUMA, M. *Software products evaluation system: quality models, metrics and processes - International Standards and Japanese Practices*. Information and Software Technology, Tokyo: Waseda University, p. 145-154, 1996.
- BARNETT, W. D., RAJA, M. K. *Application of QFD to the software development process*. International Journal of Quality & Reliability Management. MCB University Press: v. 12, n. 6, 1995.
- BIO, S. R. *Sistemas de Informação: um enfoque gerencial*. São Paulo: Ed. Atlas, 1985, págs. 114-125, 137-142.
- BRASSARD, M. *Qualidade: ferramentas para uma melhoria contínua - "The Memory Jogger"*. Rio de Janeiro: Qualitymark, 1985.
- CARNEGIE MELLON UNIVERSITY/Software Engineering Institute. *The Capability Maturity Model: Guidelines for improving the software process*, EUA, 1997.
- CHRISTEL, M.G. *et al.* Issues in Requirements Elicitation. Technical Report CMU/SEI-92-TR-12, Pittsburgh: Software Engineering Institute - Carnegie Mellon University, p. 25, 37, 54-56, set., 1992.
- CORTADA, J. W. & QUINTELLA, H.L.M.M. *TQM: Gerência da Qualidade Total*. São Paulo: Makron Books, 1994.
- DENTON, D.K. *Qualidade em serviços: o atendimento ao cliente como fator de vantagem competitiva*. São Paulo: Makron-Books, 1990, págs. 194-216.

- ELBOUSHI, M. I., SHERIF, J. S. *Object-oriented software design utilizing quality function deployment*. J. Systems Software. Elsevier Science Inc: n. 38, 1997.
- FERNANDES, A. A. *Gerência de Software Através de Métricas*. São Paulo: Ed. Atlas, 1995.
- FELICIANO NETO, A. e SHIMIZU, T. *Sistemas Flexíveis de Informações*. São Paulo: Makron Books, 1996, págs. 1-5, 9, 103, 109.
- FUNG, R. Y. K., LAW, D. S. T., IP, W. H. *Design targets determination for inter-dependent product attributes in QFD using fuzzy inference*. Integrated Manufacturing Systems. MCB University Press: v. 10, n. 6, 1999.
- GIL, A.L. *Qualidade Total em Informática*. São Paulo: Ed. Atlas, 1995.
- GRAHL, E. A. *et al.* Comparativo entre o modelo CMM-SEI e a norma ISO/IEC 12207. In: Simpósio Brasileiro de Engenharia de *Software*, 11., WQS'97 - Workshop Qualidade de *Software*, Fortaleza, 13/10/1997. **Anais**. Fortaleza, Universidade Federal do Ceará, 1997, p. 32-39.
- GOLDENSON, D.R. *et al.* After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success. Technical Report CMU/SEI-95-TR-009, Pittsburgh: *Software Engineering Institute* - Carnegie Mellon University, ago, 1995.
- HALEY, T. *et al.* Raytheon Electronic Systems Experience in *Software Process Improvement*. Technical Report CMU/SEI-95-TR-017, Pittsburgh: *Software Engineering Institute* - Carnegie Mellon University, nov, 1995.
- Indicadores da Qualidade e Produtividade em Software*.
<http://www.mct.gov.br/Temas/info/dsi/qualidad/indic.htm>, 07/09/2000.

- ISO/IEC 12119, International Standard. Information Technology - Software packages - Quality requirements and testing; Oct / 1994.
- ISO/IEC 12207-1, Software life-cycle process; 1994.
- ISO/IEC 14598-1, International Standard. Information Technology - Software product evaluation - Part 1: General Overview; Oct / 1996.
- ISO/IEC 14598-2, International Standard. Information Technology - Software product evaluation - Part 2: Planning and Management; Dec / 1996.
- ISO/IEC 14598-3, International Standard. Information Technology - Software product evaluation - Part 3: Process for developers; Jul / 1996.
- ISO/IEC 14598-4, International Standard. Information Technology - Software product evaluation - Part 4: Process for acquirers; Sep / 1996.
- ISO/IEC 14598-5, International Standard. Information Technology - Software product evaluation - Part 5: Process for evaluators; May / 1996.
- ISO/IEC 14598-6, International Standard. Information Technology - Software product evaluation - Part 6: Evaluation modules; Aug / 1996.
- ISO/IEC 9126-1, International Standard. Information Technology - Software quality characteristics and metrics - Part 1: Quality characteristics and sub-characteristics; Jan / 1997.
- ISO/IEC 9126-2, International Standard. Information Technology - Software quality characteristics and metrics - Part 2: External metrics; Jan / 1997.
- ISO/IEC 9126-3, International Standard. Information Technology - Software quality characteristics and metrics - Part 3: Internal metrics; Oct / 1996.
- JURAN, J.M. & GRAYNA, F.M. *Controle da Qualidade: Conceitos, Políticas e Filosofia da Qualidade*. 4^a Edição. São Paulo: Makron, McGraw-Hill, 1991.

- KALARGEROS, N., GAO, J. X. *QFD: focusing on its simplification and easy computerization using fuzzy logic principles*. International Journal of Vehicle Design. Elsevier Engineering Information Inc: v. 19, n. 3, 1998.
- KAPLAN, D.I. *Qualidade total na prestação de serviços - como aprimorar as práticas gerenciais adotando a melhoria contínua*. São Paulo: Nobel, 1996, págs. 7-10.
- KOPLER, P. *Administração de marketing: análise, planejamento, implementação e controle*. 5ª Edição. São Paulo: Atlas, 1998.
- MACHADO, C.A.F. *et al.* Utilização da norma ISO/IEC 12207 e do modelo CMM-SEI na CELEPAR. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 11., WQS'97 - WORKSHOP QUALIDADE DE SOFTWARE, Fortaleza, 13/10/1997. **Anais**. Fortaleza, Universidade Federal do Ceará, 1997b. p. 40-48.
- MCGIBBON, T. *et al.* A Business Case for *Software* Process Improvement Revised. An Updated DACS State-of-the-Art Report, Nova York: DoD Data & Analysis Center for *Software*, 30 de set, 1999.
- MIGUEL, P.A.C. *Qualidade: Enfoques e Ferramentas*. Aranda Editora Técnica: São Paulo, 2000.
- NBR ISO 9000-3 - "Diretrizes para a aplicação da NBR 19001 ao desenvolvimento, fornecimento e manutenção de *software*", Rio de Janeiro, Brasil, 1993.
- NBR ISO 9001 - "Sistemas da qualidade - Modelo para garantia da qualidade em projeto, desenvolvimento, produção, instalação e serviços associados", Rio de Janeiro, Brasil, 1994.

OLIVEIRA, S. T. *Ferramentas para o aprimoramento da qualidade*. 2ª ed. São Paulo: Ed. Pioneira, 1996.

PESSÔA, M.S.P. *et al.* *Qualidade de Software*. In: SIMPÓSIO NIPO-BRASILEIRO DE CIÊNCIA E TECNOLOGIA, São Paulo, 13/08/1995. **Anais**. São Paulo, Academia de Ciências do Estado de São Paulo / Sociedade Brasileira de Pesquisadores Nikkeis, 1995. p. 280-290.

Qualidade no Setor de *Software* Brasileiro N.1 (1995). Brasília: Ministério da Ciência e Tecnologia. Secretaria de Política de Informática e Automação, 1995.

Qualidade no Setor de *Software* Brasileiro N.2 (1997). Brasília: Ministério da Ciência e Tecnologia. Secretaria de Política de Informática e Automação, 1997.

Qualidade no Setor de *Software* Brasileiro N.3 (2000). Brasília: Ministério da Ciência e Tecnologia. Secretaria de Política de Informática e Automação, 2000.

Qualidade de Software. <http://www.barreto.com.br/qualidade>, 05/08/1998.

SALVIANO, C. F. *et al.* *Experiência de Avaliação de Processos e Planejamento da Melhoria Utilizando a Futura Norma ISO/IEC 15504 (SPICE)*. In: Simpósio Brasileiro de Engenharia de *Software*, 13., WQS'99 - Workshop Qualidade de *Software*, Florianópolis, SC, 13-15/10/1999. *Anais*. Florianópolis, 1999, p. 1-10.

SPICE - Software Process Improvement Capability Determination. <http://www.sqi.gu.edu.au/spice>, 20/05/1998.

TEIXEIRA, R.F. *Interpretação do P-CMM como facilitador da melhoria de processo de software*. São Carlos: USP, 2000. 85p. Dissertação (Mestrado) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2000.

The Capability Maturity Model for Software, version 1.1.

<http://www.sei.cmu.edu>, 16/09/1998.

Trillium Model Description. <http://ricis.cl.uh.edu>, 15/08/1997.

TSUKUMO, A.N. *et al.* Modelos de Processo de *Software*: Visão Global e Análise Comparativa. CTI – Centro de Tecnologia para Informática, Campinas, 1996. **Mimeo**.

TSUKUMO, A.N. *et al.* Qualidade de *Software*: visões de produto e processo de *software*. In.: II ESCOLA REGIONAL DE INFORMÁTICA - SBC REGIONAL SÃO PAULO, 03/06/1997, Piracicaba. **Anais**. Piracicaba: Gráfica UNIMEP, Universidade Metodista de Piracicaba, 1997. 189 p. p. 173-189.

BIBLIOGRAFIA CONSULTADA

CABRAL FILHO, C.A.. Pesquisa: Tendências do Mercado Brasileiro de Informática - 1996. MIS, São Paulo, Set/Out, 1996.

CHIAVENATO, I. *Teoria Geral da Administração*. 5ª ed. São Paulo: Makron-Books, 1997, pág. 301 e seg.

CHINELATO FILHO, J. A Arte de Organizar para Informatizar. 2ª ed. Rio de Janeiro: Ed. LTC, 1994.

CHINELATO FILHO, J. *O&M Integrado à Informática*. 7ª ed. Rio de Janeiro: Ed. LTC, 1997.

Crítérios de Excelência: O estado da arte da gestão para a excelência do desempenho. São Paulo: FPNQ - Fundação para o Prêmio Nacional da Qualidade, 1999.

DAHLBERG, T.& JARVINEN, J. Challenges to IS quality. *Information and Software Technology*, Helsinki, Finlândia, 1997.

Estratégias do Projeto SOFTEX 2000.
http://www.upf.tche.br/computacao/trab_96-2/estrategias.html,
25/10/1997.

EUREKA, W.E. & RYAN, N.E. *QFD: Perspectivas Gerenciais do Desdobramento da Função Qualidade*. Rio de Janeiro: Qualitymark, 1992.

FERNANDES, A.A. & COSTA NETO, P.L.O. O significado do TQM e modelos de implementação. Gestão & Produção, UFSCar, vol.3, n.2, p.173-187, ago, 1996.

ISHIKAWA, K. *Controle da qualidade total à maneira japonesa*. Rio de Janeiro: Campus, 1993.

- JORGENSEN, M. *Software quality measurement*. Advances in Engineering Software, Oslo, p. 907-912, 1999.
- JURAN, J. M. *Juran na liderança pela qualidade: um guia para executivos*. São Paulo: Pioneira/IMAM, 1990.
- MASTERS, S. *et al. CMM Appraisal Framework, Version 1.0*, CMU/SEI-95-TR-001, Fevereiro 1995.
- NAKAGAWA, M. *Gestão Estratégica de Custos: Conceito, Sistemas e Implementação*. São Paulo: Atlas, 1993.
- NAKAGAWA, M. *ABC – Custo baseado em atividade*. São Paulo: Atlas, 1994.
- PESSÔA, M. e SPINOLA, M. *Qualidade de Software: Trabalhar Melhor*. Boletim Fundação Vanzolini, São Paulo, 13 p., Jul/Ago, 1995.
- PESSÔA, M. e SPINOLA, M. *Qualidade de Software: Qual é a sua maturidade?* Boletim Fundação Vanzolini, São Paulo, 15 p., Nov/Dez, 1995.
- PESSÔA, M. e SPINOLA, M. *Qualidade de Software: Que modelo seguir?*. Boletim Fundação Vanzolini, São Paulo, Mar/Abr, 1997.
- PRAHALAD, C. K. e HAMEL, G. *The Core Competence of the Corporation*, Harvard Business Review, pp. 79-91, Mai-Jun, 1990.
- ROBLES JR., A. *Custos da Qualidade: Uma Estratégia para a Competição Global*. São Paulo: Atlas, 1994.
- SOFTEX 2000 – Programa Nacional de *Software* para Exportação. <http://www-cite.cnpq.br/softex/o-que-eh-softex.html>, 25/10/1997.
- Uma revolução da ordem. Revista INFO EXAME, São Paulo, págs. 68-70, ago., 1998.
- XIMENES, F.B. Assim caminha a computação. Revista INFO EXAME, São Paulo, pg. 82-84, ago., 1998.

WALKER, A.J. Improving the quality of ISO 9001 audits in the field of *software*.
Information and Software Technology, Johannesburg: University of the
Witwatersrand, p. 865-869, 1998.

ANEXOS

ANEXO 1 - Indicadores da Qualidade e Produtividade em *Software*

As Tabelas seguintes apontam, segundo o MCT, os principais indicadores de qualidade e produtividade em *software* coletados nas pesquisas de 1995, 1997 e 1999, assim como, metas estabelecidas para 2001.

TABELA 1.1 - CONSCIENTIZAÇÃO E MOTIVAÇÃO

FONTE: Indicadores de Qualidade e Produtividade, MCT (1995, 1997, 1999), adaptado pelo autor.

Indicadores Q&P	1995	1997		1999			2001	
		Previsto 1995	Realizado 1997	Previsto 1995	Previsto 1997	Realizado 1999	Previsto 1997	Previsto 1999
Número de projetos aprovados no SSQP/SW-PBQP por ano	37	50	56	60	80	79	100	100
Percentual de projetos realizados no SSQP/SW-PBQP sobre o total de projetos aprovados no ano	81%	85%	79%	90%	Acompanhamento			

TABELA 1.2 - ARTICULAÇÃO INSTITUCIONAL

FONTE: Indicadores de Qualidade e Produtividade, MCT (1995, 1997, 1999), adaptado pelo autor.

Indicadores Q&P	1995	1997		1999			2001	
		Previsto 1995	Realizado 1997	Previsto 1995	Previsto 1997	Realizado 1999	Previsto 1997	Previsto 1999
Percentual de financiamentos a empresas de <i>software</i> sobre o valor total dos financiamentos Fonte: FINEP Fonte: BNDES	2% 0%	3% ...	3% 0%	4% ...	Acompanhamento	5% 0%	Acompanhamento	
Percentual de financiamentos em qualidade e produtividade em <i>software</i> sobre o valor total dos financiamentos para qualidade e produtividade Fonte: FINEP Fonte: BNDES	0% ...	2% ...	0% ...	4% ...		0% ...		

TABELA 1.3 - MÉTODOS DE GESTÃO

FONTE: Indicadores de Qualidade e Produtividade, MCT (1995, 1997, 1999), adaptado pelo autor.

Indicadores Q&P	1995	1997		1999			2001	
		Previsto 1995	Realizado 1997	Previsto 1995	Previsto 1997	Realizado 1999	Previsto 1997	Previsto 1999
Percentual de empresas com programa da Qualidade total, sistema da Qualidade ou similar implantado sobre o total de empresas	11%	20%	18%	30%	30%	26%	50%	50%
Percentual de empresas com sistema da Qualidade certificado (ISO 9001 e ISO 9002) sobre o total de empresas	2%	8%	8%	20%	20%	16%	35%	35%
Número de empresas com sistema da Qualidade certificado (ISO 9001 e ISO 9002)	8	40	45	100	120	72	200	200
Número de empresas com <i>software</i> explicitado no escopo do certificado de qualidade (ISO 9001 e ISO 9002)			16		50	35	100	100
Percentual de empresas que conhecem e usam o modelo CMM sobre o total de empresas	3%	10%	5%	20%	10%	10%	20%	20%
Número de empresas com modelo CMM implantado, por nível Nível 2 Nível 3			1 -	Acompanhamento		3 2	Acompanhamento	
Percentual de empresas que fazem, de forma sistemática, medições usando a técnica de pontos por função sobre o total de empresas			4%		Acompanhamento	19%		
Percentual de empresas que mantém, de forma sistemática, contabilidade de custos da qualidade sobre o total de empresas	4%		6%		Acompanhamento	13%		

TABELA 1.4 - SERVIÇOS TECNOLÓGICOS

FONTE: Indicadores de Qualidade e Produtividade, MCT (1995, 1997, 1999), adaptado pelo autor.

Indicadores Q&P	1995	1997		1999			2001	
		Previsto 1995	Realizado 1997	Previsto 1995	Previsto 1997	Realizado 1999	Previsto 1997	Previsto 1999
Número de laboratórios de ensaios de <i>software</i> no País	1	1	5	2	7	5	9	9
Número de produtos de <i>software</i> com qualidade avaliada por terceiros, Segundo a ISO/IEC 9126	25	60	148	100	210	13	280	280
Comissões de estudos da ABNT envolvidas com engenharia de <i>software</i> e com qualidade de processos e produtos de <i>software</i>	4		12	Acompanhamento		11	Acompanhamento	

TABELA 1.5 - RECURSOS HUMANOS

FONTE: Indicadores de Qualidade e Produtividade, MCT (1995, 1997, 1999), adaptado pelo autor.

Indicadores Q&P	1995	1997		1999			2001	
		Previsto 1995	Realizado 1997	Previsto 1995	Previsto 1997	Realizado 1999	Previsto 1997	Previsto 1999
Número de mestres e doutores em empresas que atuam no segmento de <i>software</i>	1.010	1.200	877	1.500	1.000	1.264	1.100	1.100
Número de profissionais certificados em Qualidade em empresas que atuam no segmento de <i>software</i>	390	500	366	700	500	823	700	700
Percentual dos investimentos anuais em treinamento para melhoria da qualidade sobre a comercialização bruta proveniente de <i>software</i>	3%	3,5%	2,5%	4%	3%	3%	3,5%	3,5%
Percentual dos investimentos anuais em treinamento para engenharia / tecnologia de <i>software</i> sobre a comercialização bruta proveniente de <i>software</i>	3%	3,5%	5%	4%	6%	6%	7%	7%

TABELA 1.6 - TECNOLOGIA DE SOFTWARE

FONTE: Indicadores de Qualidade e Produtividade, MCT (1995, 1997, 1999), adaptado pelo autor.

Indicadores Q&P	1995	Realizado em 1997	1999	2001
			Previsto em 1997 e Realizado em 1999	Previsto em 1997 e 1999
Percentual de empresas que adotam métodos de prevenção de defeitos sobre o total de empresas Gestão de configuração Gestão de mudança <i>Joint Application Design – JAD</i> Medições da qualidade (Métricas) Métodos estruturados Métodos orientados a objetos Projetos de interface com o usuário Prototipação Reuso 10% ... 43% ... 46% 37%	7% 5% 8% 8% 36% 37% 35% 44% 19%	Acompanhamento	
Percentual de empresas que adotam métodos de detecção/remoção de defeitos sobre o total de empresas Auditorias Inspeções formais Revisões estruturadas Testes de aceitação Testes de sistema Testes de unidade Validação Verificação independente	... 10% ... 48% 62% 24%	17% 17% 19% 47% 67% 23% 42% 14%	Acompanhamento	
Percentual de empresas que adotam ferramentas avançadas de engenharia de <i>software</i> sobre o total de empresas CASE Gerador de código-fonte Gerador de telas Gerenciador de bibliotecas de módulos Gerenciador de configuração Gerenciador de projetos Prototipador Não utiliza ferramentas automatizadas	26% 37% 47% 20% 10% ... 17% 11%	20% 28% 35% 20% 10% 24% 15% 21%	Acompanhamento	

TABELA 1.7 - MARKETING DE SOFTWARE

FONTE: Indicadores de Qualidade e Produtividade, MCT (1995, 1997, 1999), adaptado pelo autor.

Indicadores Q&P	1995	1997		1999			2001	
		Previsto 1995	Realizado 1997	Previsto 1995	Previsto 1997	Realizado 1999	Previsto 1997	Previsto 1999
Percentual de empresas que utilizam, de forma sistemática, dados de pesquisa ou de reclamações na revisão de projetos ou na especificação de novos produtos sobre o total de empresas	41%	50%	44%	60%	50%	44%	60%	60%
Percentual de empresas que atuam no segmento de <i>software</i> e realizam, de forma sistemática, pesquisas de expectativas dos clientes sobre o total de empresas	19%	25%	21%	35%	30%	21%	35%	35%
Percentual de empresas que atuam no segmento de <i>software</i> e realizam, de forma sistemática, pesquisas de satisfação dos clientes sobre o total de empresas	19%	25%	25%	35%	35%	29%	40%	40%
Produtividade de pessoal (valor bruto proveniente da comercialização de <i>software</i> nos mercados interno e externo sobre o número de pessoas nas empresas)	US\$ 8 mil	US\$ 50 mil	US\$ 26 mil	US\$ 100 mil	US\$ 50 mil	US\$ 50 mil	US\$ 100 mil	US\$ 100 mil

ANEXO 2 - Comparativo das Normas e Modelos Apresentados

A tabela apresentada a seguir dá uma visão conjunta dos principais aspectos dos modelos e normas apresentados até o momento, neste trabalho, no que diz respeito a objetivos e formas de como atingir esses objetivos, tipo/porte de organizações alvo, composição da definição de processos, flexibilidade dos modelos, assim como outros aspectos abordados.

Tabela 2.1 - Quadro comparativo das normas e modelos apresentados

FONTES: TSUKUMO *et al* (1996), adaptado pelo autor.

Aspectos abordados	CMM	Trillium	SPICE	ISO 9000-3	ISO/IEC 12207-1
Objetivo	Determinar a capacitação da organização e apoiar a sua evolução de acordo com os níveis estabelecidos.	Melhorar continuamente o desenvolvimento de produtos e a capacidade do processo de suporte.	Conhecer e avaliar os processos da organização, determinar a capacitação e promover a melhoria.	Certificar a organização de acordo com os padrões estabelecidos em situações de contrato de fornecimento de <i>software</i> .	Estabelecer uma terminologia e um entendimento comum aos processos entre os envolvidos com <i>software</i> .
Abordagem	Avaliação dos processos e enquadramento da organização em um dos níveis de maturidade.	Avaliação do desenvolvimento de produtos e capacidade de suporte de fornecedores de tecnologia básica de informação ou telecomunicações.	Avaliação dos processos da organização em relação a níveis de capacitação.	Verificação de conformidade de processos a padrões documentados.	Definição dos processos para aquisição, fornecimento, desenvolvimento, operação e manutenção de <i>software</i> .
Organizações Alvo	Organizações que necessitam de comprovação formal de sua capacidade.	Organizações que utilizam sistemas de <i>software</i> embutido e que necessitam de comprovação formal de sua capacidade.	Organizações em geral.	Organizações que necessitam de uma certificação.	Organizações em geral.
Definição de Processos	Estabelece 18 áreas de processos organizados em 5 níveis crescentes de maturidade.	Estabelece 8 áreas de capacidade e 27 áreas de processos organizados em 5 níveis crescentes de maturidade.	Estabelece 35 processos organizados em 5 categorias.	Não estabelece processos, mas atividades a serem cumpridas, com visão de estrutura, ciclo de vida e suporte	Estabelece 17 processos organizados em 3 categorias.

Tabela 2.1 – QUADRO COMPARATIVO DAS NORMAS E MODELOS APRESENTADOS (CONTINUAÇÃO)

FONTES: TSUKUMO *et al* (1996), adaptado pelo autor.

Aspectos abordados	CMM	Trillium	SPICE	ISO 9000-3	ISO/IEC 12207-1
Flexibilidade nos aspectos definidos pelo modelo	Níveis e área chave de processo são a base do modelo e não podem ser alterados.	Níveis, áreas de capacidade e "roadmaps" são a base do modelo e não podem ser alterados.	Permite a definição de perfis de processo e práticas de acordo com os objetivos da organização.	Não admite adaptação nos aspectos abordados.	Classificação de processos pode ser utilizada conforme os objetivos da organização.
Instrumento de Avaliação	Questionário.	Questionário.	Fornecer orientações para montar questionário.	Lista de verificação.	Não se aplica.
Inspiração e Influência	Princípios de Shewart, Deming, Juran, Crosby.	CMM, ISO 9001, ISO 9000-3, Bellcore, Malcolm Baldrige, IEEE, IEC300.	TQM, PDCA, CMM, STD, Trillium, Malcolm Baldrige, Bootstrap.	Normas militares americanas, canadenses, Sistemas de Qualidade do UK.	TQM, PDCA.
Aspectos positivos	Estabelecimento de diretrizes para melhoria contínua; Difusão extensa nos EUA.	Flexibilização e expansão do modelo CMM; Incorpora, também, práticas de outros critérios e Normas Internacionais.	Norma internacional em elaboração; Expansão e flexibilização dos modelos citados	Norma Internacional; Difusão extensa; Reconhecimento do valor da certificação.	Norma Internacional; Definição de uma taxonomia para processos útil para qualquer organização.
Limitações	Pouca consideração à diversidade das organizações; Dificuldade de aplicação em pequenas organizações; Falta abordagem de produto.	Dificuldade de aplicação em pequenas organizações.	Dificuldade de aplicação devido à grande quantidade de informações; Falta abordagem de produto.	Risco de se colocar a Certificação como objetivo principal; Ausência de apoio à melhoria contínua.	Apenas uma definição de taxonomia de processos.

