

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA
TESE DEFENDIDA POR Mary Lucy Sant'Ana
Ana E APROVADA PELA
COMISSÃO JULGADORA EM 22.02.02
Ana Cervigni Guerra
ORIENTADOR

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Uma Proposta para Qualidade de Software
Através da Aplicação Integrada do
Modelo CMM[®] e da Norma NBR 13596

Autora: **Mary Lucy Sant'Ana**

Orientadora: **Ana Cervigni Guerra**

02/2002

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

**Uma Proposta para Qualidade de Software
Através da Aplicação Integrada do
Modelo CMM[®] e da Norma NBR 13596**

Autora: Mary Lucy Sant'Ana

Orientadora: Ana Cervigni Guerra

Curso: Engenharia Mecânica – Mestrado Profissional

Área de Concentração: Gestão da Qualidade Total

Trabalho Final de Mestrado Profissional apresentado à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Mestre Profissional em Engenharia Mecânica / Gestão da Qualidade Total.

Campinas, 2002

S.P. – Brasil

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Sa59p Sant'Ana, Mary Lucy
 Uma proposta para qualidade de software através da
 aplicação integrada do modelo CMM® e da Norma NBR
 13596 / Mary Lucy Sant'Ana.--Campinas, SP: [s.n.],
 2002.

 Orientador: Ana Cervigni Guerra.
 Dissertação (mestrado) - Universidade Estadual de
 Campinas, Faculdade de Engenharia Mecânica.

 1. Software - Desenvolvimento. 2. Normas técnicas.
 3. Modelo (SC). 4. Tecnologia – Avaliação. 5. Gestão da
 qualidade total. I. Guerra, Ana Cervigni. II.
 Universidade Estadual de Campinas. Faculdade de
 Engenharia Mecânica. III. Título.

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**

Trabalho Final de Mestrado Profissional

**Uma Proposta para Qualidade de Software
Através da Aplicação Integrada do
Modelo CMM[®] e da Norma NBR 13596**

Autora: Mary Lucy Sant'Ana

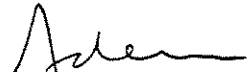
Orientadora: Ana Cervigni Guerra



Prof. Dra. Ana Cervigni Guerra, Presidente
CenPRA



Prof. Dr. Antônio Batocchio
UNICAMP



Prof. Dr. Ademir J. Petenatte
UNICAMP

Campinas, 22 de fevereiro de 2002

00239352

Dedicatória:

Dedico este trabalho ao meu querido esposo, meus filhos maravilhosos e meus amados pais:

A você Libanio, que sempre me apoiou disponibilizando todos os recursos necessários, abdicando-se de minha companhia e me impulsionando para jamais desistir.

Giovanni e Brunna contribuíram muito na elaboração de cada uma das linhas desta dissertação. A pergunta freqüente, sobre o número de páginas que ainda precisavam ser escritas, me lembrava quanta energia eles também colocavam neste trabalho. Meus queridos filhos, espero que este trabalho demonstre a vocês que todos somos capazes de realizar, basta querermos e colocarmos o esforço que for necessário.

Aos meus pais Silvio e Anésia, que já devo a vida, dedico este trabalho como um retorno simbólico por tudo que têm me oferecido: amor, educação, cultura, exemplo e filosofia de vida.

Sem vocês, o todo não faria sentido.

Agradecimentos

Este trabalho não poderia ter sido realizado sem o apoio do CenPRA – Centro de Pesquisas Renato Archer que proporcionou a oportunidade de aprofundar meus conhecimentos teóricos e práticos nas áreas de software e qualidade.

Diversas pessoas ajudaram diretamente na elaboração da dissertação, às quais presto minha homenagem:

À minha, sempre prestativa, orientadora Ana Cervigini Guerra, por suas orientações objetivas e carinhosas, além de sua dedicação extra, serei sempre grata.

Aos colegas do CenPRA e da Motorola, que me impulsionaram com suas palavras de incentivo e colaboraram com informações técnicas.

À professora Silvia O. S. da Silva, o meu muito obrigada por sua dedicação e orientações da parte ortográfica e gramatical deste trabalho.

*"... e se não houver frutos valeu a beleza das flores; se não
houver flores valeu a sombra das folhas e se não houver
folhas valeu a intenção da semente."*

Henfil

Resumo

SANT'ANA, Mary Lucy, *Uma Proposta para Qualidade de Software Através da Aplicação Integrada do Modelo CMM® e da Norma NBR 13596*, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2002. 124 p. Trabalho Final de Mestrado Profissional.

A área de Tecnologia da Informação cresce de forma substancial no Brasil e no mundo. O software exerce um papel fundamental nesta área e sua comercialização vai além dos limites territoriais dos países que os desenvolvem. Diante de sua crescente complexidade e abrangência de atuação, além da relevância financeira para seu país de origem, a qualidade de software é uma preocupação que tem mobilizado esforços tanto governamentais como de entidades privadas. As normas e modelos usados pelas organizações de software são sempre focados na qualidade do processo ou na qualidade do produto de software. O objetivo desta dissertação é apresentar uma proposta de integração das duas abordagens: aplicação do modelo de processo CMM® e a Norma NBR 13596, sobre características de qualidade de produto de software. Procurou-se analisar atividades, metas e produtos de trabalho das áreas de processo, propostas pelo CMM®, verificando instâncias onde as definições, diretrizes e características de qualidade de produto de software, abordadas na Norma NBR 13596, podem ser utilizadas. As contribuições deste trabalho abrangem a divulgação dos esforços que estão sendo realizados na busca pela qualidade de software, motivar as práticas de Engenharia de Software, proporcionar uma visão integradora das abordagens de qualidade de processo e produto, além de demonstrar uma possibilidade prática de integração.

Palavras Chave

Qualidade, Software, Processo, Produto, SPI, CMM®, NBR 13596, ISO/IEC 9126, ISO/IEC 15504

Abstract

SANT'ANA, Mary Lucy, *A Proposal for Software Quality Through the Integrated Application of CMM® Model and the NBR 13596 Norm*, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2002. 124 p. Trabalho Final de Mestrado Profissional.

The Information Technology area grows of substantial form in Brazil and worldwide. Software exerts a basic paper in this area and its commercialization goes beyond the territorial limits of countries that develop them. Ahead of its increasing complexity and action, beyond the financial relevance for its country origin, the quality of software is a concern that has mobilized governmental efforts and private entities. The norms and models used by software organizations always are focus in the process quality or the software product quality. The objective of this investigation is to present an alternative of integration of two focuses: the application of the CMM® process model and the NBR 13596 Norm of software product quality characteristics. The work consists in to analyze activities, goals and work products of the CMM® process areas verifying it instances where the definitions, direction lines and software product quality characteristics of NBR 13596 can be used. The contributions of this work enclose the spreading of the efforts that are being carried through in the software quality, to motivate the use of Software Engineering providing a integrated vision of process and product quality boarding, besides demonstrating a practical possibility of integration.

Key Words

Quality, Software, Process, Product, SPI, CMM®, NBR 13596, ISO/IEC 9126, ISO/IEC 15504

Índice

LISTA DE FIGURAS	IV
LISTA DE TABELAS.....	V
NOMENCLATURA.....	VI
1 INTRODUÇÃO	1
1.1 PANORAMA TECNOLÓGICO	1
1.2 OBJETIVOS	3
1.3 ESCOPO E LIMITAÇÕES.....	4
1.4 ESTRUTURA DA DISSERTAÇÃO.....	7
2 QUALIDADE.....	9
2.1 QUALIDADE DE PRODUTO	9
2.2 QUALIDADE DE PRODUTO DE SOFTWARE.....	12
2.2.1 Aspectos Gerais sobre Software	12
2.2.2 Software e suas Características como Produto.....	15
2.2.3 Iniciativas para a Qualidade de Produto de Software	17
2.3 QUALIDADE DE PROCESSO DE SOFTWARE.....	19
2.3.1 Aspectos Gerais sobre Processo de Software.....	19
2.3.2 Iniciativas para a Qualidade de Processo de Software.....	21
3 SW-CMM®	28
3.1 INTRODUÇÃO.....	28
3.2 RELAÇÃO ENTRE CMM® E TQM (CMM, 2000)	32
3.3 ESTRUTURA.....	34
3.4 NÍVEIS DE MATURIDADE.....	35
3.5 ÁREAS CHAVE DE PROCESSO	39

3.6	CARACTERÍSTICAS COMUM.....	41
4	NBR 13596	44
4.1	INTRODUÇÃO.....	44
4.2	RELAÇÃO ENTRE NBR 13596 E TQM.....	45
4.3	DIRETRIZES PARA USO DA NORMA NBR 13596.....	45
4.4	CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE.....	46
4.5	MODELO DE QUALIDADE DE PRODUTO DE SOFTWARE.....	47
5	INTEGRANDO CMM® E NBR 13596	51
5.1	INTRODUÇÃO.....	51
5.2	ANÁLISE DO NÍVEL 2: REPETÍVEL.....	53
5.3	ANÁLISE DO NÍVEL 3: DEFINIDO	64
5.4	ANÁLISE DO NÍVEL 4: GERENCIADO	75
5.5	ANÁLISE DO NÍVEL 5: OTIMIZANDO.....	78
5.6	RESUMO DAS CONSIDERAÇÕES	83
5.6.1	<i>Nível 2: Repetível</i>	<i>83</i>
5.6.2	<i>Nível 3: Definido</i>	<i>91</i>
5.6.3	<i>Nível 4: Gerenciado</i>	<i>95</i>
5.6.4	<i>Nível 5: Otimizando.....</i>	<i>97</i>
5.7	COMENTÁRIOS GERAIS.....	99
6	CONCLUSÃO	101
6.1	CONSIDERAÇÕES.....	101
6.2	TEMAS PARA TRABALHOS FUTUROS	102
	REFERÊNCIAS BIBLIOGRÁFICAS	104
	ANEXO: ATIVIDADES DEFINIDAS PELO CMM®	111
	GERENCIAMENTO DE REQUISITOS.....	111
	PLANEJAMENTO DE PROJETO DE SOFTWARE.....	111
	ACOMPANHAMENTO DE PROJETO DE SOFTWARE.....	112
	GERENCIAMENTO DE SUBCONTRATO DE SOFTWARE.....	113
	GARANTIA DA QUALIDADE DE SOFTWARE.....	115
	GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE.....	115
	FOCO NO PROCESSO DA ORGANIZAÇÃO	116
	DEFINIÇÃO DO PROCESSO DA ORGANIZAÇÃO	117
	PROGRAMA DE TREINAMENTO.....	117
	GERENCIAMENTO INTEGRADO DE SOFTWARE.....	118

ENGENHARIA DE PRODUTO DE SOFTWARE.....	119
COORDENAÇÃO ENTRE GRUPOS.....	120
REVISÃO DE PARES	120
GERENCIAMENTO QUANTITATIVO DE PROCESSO	121
GERENCIAMENTO DA QUALIDADE DE SOFTWARE.....	121
PREVENÇÃO DE DEFEITOS	122
GERENCIAMENTO DE MUDANÇA TECNOLÓGICA	123
GERENCIAMENTO DE MUDANÇA DE PROCESSO.....	123

Lista de Figuras

FIGURA 1.1	CONHECIMENTO DO MODELO CMM® (WEBER, 2001).....	5
FIGURA 2.1	DEFINIÇÃO DE PROCESSO DE SOFTWARE	19
FIGURA 2.2	SITUAÇÃO ATUAL DA MAIORIA DAS ORGANIZAÇÕES DE SOFTWARE (MAGNANI, 1998).....	20
FIGURA 2.3	RELAÇÃO DA ISO/IEC 15504 COM OS MODELOS DE PROCESSO.....	23
FIGURA 2.4	NÍVEIS DE CAPACIDADE E PROCESSOS	24
FIGURA 3.1	NÍVEIS DE MATURIDADE DO CMM® VERSÃO 1.1 (CMM, 2000).....	31
FIGURA 3.2	SISTEMA ORGANIZACIONAL SEGUNDO A VISÃO DO TQM (LAWRENCE, 2001).....	32
FIGURA 3.3	APLICANDO TQM A SOFTWARE (LAWRENCE, 2001).....	34
FIGURA 3.4	ESTRUTURA DO CMM® (CMM, 2000).....	35
FIGURA 3.5	PORCENTAGEM DE ORGANIZAÇÕES POR NÍVEL DE MATURIDADE (SEMA, 2002)	37
FIGURA 3.6	KPA POR NÍVEL DE MATURIDADE.....	40
FIGURA 3.7	CARACTERÍSTICAS COMUM.....	42
FIGURA 4.1	VALOR MEDIDO E NÍVEL DE PONTUAÇÃO (NBR 13596, 1996).....	49
FIGURA 5.1	CICLO PDCA	78

Lista de Tabelas

TABELA 2.1	PESQUISA DE UTILIZAÇÃO DE MÉTODOS DE ENGENHARIA DE SOFTWARE (MCT, 2001).....	16
TABELA 2.2	CATEGORIAS E PROCESSOS DA FUTURA NORMA ISO/IEC 15504 (WEBER, 2001).....	25
TABELA 2.3	CATEGORIAS E ÁREAS DE PROCESSO DO MODELO CMMI SM (SOUZA, 2001)	27
TABELA 4.1	CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE (NBR 13596, 1996).....	46
TABELA 4.2	SUBCARACTERÍSTICAS DE QUALIDADE DE SOFTWARE (TSUKUMO, 1997).....	48
TABELA 5.1	OBJETIVO DAS ATIVIDADES ATRAVÉS DAS VISÕES CMM® E CMM®+NBR13596.....	52
TABELA 5.2	INSTÂNCIAS PARA USO DA NBR 13596 E VALOR AGREGADO ÀS KPA DO NÍVEL 2	84
TABELA 5.3	INSTÂNCIAS PARA USO DA NBR 13596 E VALOR AGREGADO ÀS KPA DO NÍVEL 3	92
TABELA 5.4	INSTÂNCIAS PARA USO DA NBR 13596 E VALOR AGREGADO ÀS KPA DO NÍVEL 4	95
TABELA 5.5	INSTÂNCIAS PARA USO DA NBR 13596 E VALOR AGREGADO ÀS KPA DO NÍVEL 5	97

Nomenclatura

Superescritos

® “A letra ‘R’ dentro de um círculo é um símbolo significando marca registrada federal. O símbolo de marca registrada federal é usado para indicar que o produto, ou serviço, já foi registrado nos órgãos competentes nacionais de registro de marcas e patentes. Mesmo que a solicitação para registro esteja pendente, o símbolo não pode ser usado antes que a marca esteja realmente registrada. O símbolo de registro é normalmente superescrito do lado direito da última letra da marca (APOLLO, 2001)”.

SM “Os símbolos ‘TM’ - *Trademark* e ‘SM’ - *Service Mark* normalmente indicam que o
e direito da marca está sendo reivindicado. Estes símbolos são muito usados pela parte
TM interessada antes do registro federal. Não há regulamentação federal governando o uso
destes símbolos ou suas designações, no entanto, seu uso pode ser governado por leis
locais, estaduais ou internacionais. O símbolo ‘TM’ ou ‘SM’ é normalmente superescrito
do lado direito da última letra da marca. Estes símbolos normalmente indicam marcas
que identificam o produto no mercado (APOLLO, 2001)”.

.....

Abreviações

GCS	Gerenciamento de Configuração de Software
GQS	Garantia da Qualidade de Software
KPA	<i>Key Process Area</i> ou, em português, Área Chave de Processo
PA	<i>Process Area</i> ou, em português, Área de Processo
TI	Tecnologia da Informação

.....

Siglas

ABNT	“Fundada em 1940, a ABNT – Associação Brasileira de Normas Técnicas – é o órgão responsável pela normalização técnica no país, fornecendo a base necessária ao desenvolvimento tecnológico brasileiro. Sediada na cidade de São Paulo/ Brasil, a ABNT é uma entidade privada, sem fins lucrativos, reconhecida como Fórum Nacional de Normalização – ÚNICO – através da Resolução n.º 07 do CONMETRO, de 24.08.1992. É membro fundador da ISO - <i>International Organization for Standardization</i> , da COPANT - Comissão Pan-americana de Normas Técnicas e da AMN - Associação Mercosul de Normalização (ABNT, 2001)”.
BOOTSTRAP	“Modelo europeu para avaliação de processo de software (HAASE, 1994)”.

CenPRA

Conforme Decreto 4.043 de 04 de dezembro de 2001, o Presidente da República decretou a criação do CenPRA - Centro de Pesquisas Renato Archer, para o qual foram transferidas a estrutura organizacional e as atividades de pesquisa e desenvolvimento da Diretoria de Tecnologia da Informação da Autarquia ITI - Instituto Nacional de Tecnologia da Informação (antiga Fundação CTI - Centro Tecnológico para Informática). Ligado diretamente ao MCT - Ministério da Ciência e Tecnologia e criado em 1982 com a finalidade de desenvolver e implementar pesquisas científicas e tecnológicas no setor de informática, o CenPRA contribui ativamente com o setor acadêmico e industrial, na medida em que promove a evolução das tecnologias da informação, mantendo-se no estado da arte em diversos segmentos tecnológicos chaves, abrangendo basicamente os setores de componentes, sistemas e software. Sua sede está localizada na cidade de Campinas, Estado de São Paulo – Brasil. (CenPRA, 2002)

CMM

“Sigla utilizada para o modelo de processo de software, desenvolvido pelo SEI, denominado *Capability Maturity Model for Software* (SEI, 2001)”.

CMMI

“Sigla utilizada para o modelo de processo de software, desenvolvido pelo SEI, denominado *Capability Maturity Model Integration* (SEI, 2001)”.

DTR

Sigla utilizada junto ao número de um relatório técnico ISO quando este está sendo criado. DTR - *Draft Technical Report* significa que o rascunho do relatório técnico se tornou oficial.

IEC

“Fundada em 1906, a IEC - *International Electrotechnical Commission* é uma organização global que prepara e publica normas internacionais relacionadas com todas as tecnologias elétricas, eletrônicas e afins. A IEC, sediada em Gênova na Suíça, foi fundada como resultado de uma resolução do *International Electrical Congress* que ocorreu em 1904 na cidade de St. Louis, EUA. São mais de 60 países membros incluindo todas as grandes

nações do mundo e um crescente número de países industrializados (IEC, 2001)”.

IEEE

“O IEEE – *Institute of Electrical and Electronics Engineers*, sediado em NJ/EUA, nasceu em 1963 como fruto da fusão do AIEE - *American Institute of Electrical Engineers* e do IRE - *Institute of Radio Engineers*, que datam de 1884. O IEEE auxilia na prosperidade global promovendo a engenharia do processo de criação, desenvolvimento, integração, compartilhamento e aplicação de conhecimento sobre tecnologia elétrica, da informação e ciências para o benefício da humanidade... (IEEE, 2001)”.

ISO

“A ISO - *International Organization for Standardization* é uma federação internacional de organismos nacionais de padronização composta por aproximadamente 140 países, sendo 1 organismo de cada país. A ISO é um organismo não governamental estabelecido em 1947 e sediado em Gênova na Suíça. Sua missão é promover o desenvolvimento de padrões e atividades afins ao redor do mundo, com a visão de facilitar a troca de experiências e o desenvolvimento corporativo de atividades na esfera intelectual, científica, tecnológica e econômica. Os resultados do trabalho da ISO são consensos internacionais publicados como Normas Internacionais. (ISO, 2001)”.

MEDE-PROS®

Método de Avaliação da Qualidade de Produto de Software desenvolvido e registrado em 1996 pelo CenPRA – Centro de Pesquisas Renato Archer. (MEDE-PROS, 1996)

NBR

“Norma técnica elaborada pela ABNT, em conformidade com os procedimentos fixados para o Sistema Nacional de Metrologia, Normalização e Qualidade Industrial, pela lei 5.966, de 16 de dezembro de 1973 (FERREIRA, 1994)”.

PDCA	Sigla do ciclo <i>Plan, Do, Check e Act</i> , que em português significa Planejar, Fazer, Avaliar e Agir.
PDTR	Sigla utilizada junto ao número de um relatório técnico ISO quando este está sendo criado. PDTR – <i>Proposed Draft Technical Report</i> significa que o primeiro documento foi aceito como uma proposta de rascunho de um relatório técnico ISO.
SEI	“O SEI - <i>Software Engineering Institute</i> é um centro de pesquisas e desenvolvimento federal fundado pelo Departamento de Defesa dos EUA através do OUSD(AT&L)- <i>Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics</i> ... O propósito do SEI, em poucas palavras, é aprimorar a prática da engenharia de software (SEI, 2001)”.
SEMA	“O grupo SEMA - <i>Software Engineering Measurement and Analysis</i> é uma iniciativa do SEI focado em medições de software e pesquisas empíricas que aceleram a identificação e adoção de melhorias em praticas de engenharia de software (SEMA, 2002)”.
SEPG	“Sigla utilizada pelo SEI para <i>Software Engineering Process Group</i> (SEI, 2001)”
SPICE	“SPICE - <i>Software Process Improvement and Capability dEtermination</i> é a maior iniciativa internacional para prover suporte ao desenvolvimento de uma norma ISO de avaliação de processo de software. (SPICE, 2001)”.
TQM	Sigla utilizada para <i>Total Quality Management</i> ou, em português, Gerenciamento da Qualidade Total.

TR Sigla utilizada junto ao número de um relatório técnico ISO quanto este está sendo criado. TR – *Technical Report* significa que o relatório técnico ISO foi aprovado.

TRILLIUM “Modelo canadense para avaliação de processo de software, baseado no CMM[®], versão 1.1 e voltado para organizações de telecomunicações (UHCL, 2001)”.

Definições

Avaliação da Qualidade “Exame sistemático para determinar até que ponto uma entidade é capaz de atender os requisitos especificados (NBR ISO 8402, 1993)”.

Características de Qualidade de Software “Conjunto de atributos de um produto de software, através do qual sua qualidade é descrita e avaliada. Uma característica de qualidade de software pode ser detalhada em múltiplos níveis de subcaracterísticas (NBR 13596, 1996)”.

Entidade de Software “Por entidade de software entende-se o conjunto completo, ou um item deste conjunto, de programas de computador, procedimentos, documentação associada e dados designados (LAWRENCE, 2001)”.

Engenharia de Software “...disciplina que pode ser vista, de forma objetiva, como o estabelecimento e o uso dos princípios básicos da engenharia com a finalidade de desenvolver software de maneira sistemática e econômica, resultando em um produto confiável e eficiente (PRESSMAN, 1995)”.

Gerenciamento de Projeto de Software	“Identificar, estabelecer, coordenar e acompanhar as atividades, tarefas e recursos necessários para o desenvolvimento e manutenção de software (ISO/IEC TR 15504-5, 1998)”.
Implantar	“Introduzir, inaugurar, estabelecer, inserir (FERREIRA, 1994)”.
Implementar	“Executar um plano, programa ou projeto (FERREIRA, 1994)”.
Institucionalizar	“Tornar próprio de uma instituição. Ato de instituir, criação, estabelecimento (FERREIRA, 1994)”.
Necessidades Explícitas	Requisitos, condições e objetivos propostos, formalmente pelo consumidor, para um produto ou serviço.
Necessidades Implícitas	Requisitos, condições e objetivos, assumidos pelo consumidor como inerentes, do produto ou serviço nem sempre propostos formalmente.
Norma	“Aquilo que se estabelece como base ou medida para a realização ou a avaliação de alguma coisa. Princípio, preceito, regra, lei. Modelo, padrão (FERREIRA, 1994)”.
Norma Técnica	“Documento técnico que fixa padrões reguladores visando garantir a qualidade do produto industrial, a racionalização da produção, transporte e consumo de bens, a segurança das pessoas, a uniformidade dos meios de expressão e comunicação, etc. (FERREIRA, 1994)”.
Normalização	“Atividade que estabelece prescrições, relativas a problemas existentes ou potenciais, destinadas à utilização comum e repetitiva com vistas à obtenção do grau ótimo de ordem em um

dado contexto (ABNT, 2001)”.

Organização

“Uma unidade de uma empresa ou qualquer outra entidade que tenha projetos a serem gerenciados. Uma característica que delimita uma entidade como sendo uma organização é o fato de todos os projetos compartilharem das mesmas políticas e da mesma alta gerência (CMM, 2000)”.

Organização de Software

Toda e qualquer empresa que tenha desenvolvimento e manutenção de software como uma de suas atividades, não sendo, necessariamente, sua atividade fim.

Processo de Software

“Conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter software e seus produtos associados (CMM, 2000)”.

Produto de Software

“Entidade de software disponível para liberação a um cliente ou usuário (NBR 13596, 1996)”.

Produto de Software Intermediário

Entidade de software ainda não disponível para liberação a um cliente ou usuário.

Produto de Trabalho de Software

“Qualquer entidade criada como parte da definição, manutenção ou uso de um processo de software, incluindo descrições do processo, planos, procedimentos, programas de computador e documentos associados, que podem ou não ser liberados para o cliente ou usuário (LAWRENCE, 2001)”.

Projeto de Software

“Envolve tipicamente análise, especificação, projeto (*design*), desenvolvimento, teste e/ou manutenção dos componentes de software e da documentação associada (CMM, 2000)”.

Qualidade

“Totalidade das características de uma entidade, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas (NBR ISO 8402, 1993)”.

**Qualidade de
Software**

Totalidade das características de software, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas. Onde software representa os elementos: ‘processo e produto de software’.

.....

Capítulo 1

1 Introdução

1.1 Panorama Tecnológico

Nos últimos anos tem ocorrido um crescimento sem precedentes na área de TI - Tecnologia da Informação. Com a globalização e a *internet*, facilitando ainda mais a abertura de mercado, a produção de software é escoada independentemente das fronteiras nacionais.

No Brasil, este crescimento tem colocado as organizações nacionais de software em concorrência com organizações já bem estabelecidas tanto no mercado interno como no mercado externo.

Dados publicados no livro 'Qualidade e Produtividade em Software' (WEBER, 2001), em sua 4ª edição renovada, confirmam o crescimento do setor de software no Brasil. A taxa média anual de crescimento de receita na década de 1990 foi de 19%. Como os valores reais ainda não eram conhecidos, quando da publicação do livro, o mercado projetado para o ano 2000 era de R\$ 5,9 bilhões, provenientes da comercialização de software no mercado interno, acrescido de US\$ 1,2 bilhões resultantes da remessa em direitos autorais devido à importação de software. Outro dado importante do livro, é o salto de 42% para 51%, ao longo do período 1991/99, na participação de mercado dos produtos de software e serviços técnicos de informática em relação ao setor de TI como um todo.

Diante deste quadro, as organizações nacionais de software devem repensar as premissas fundamentais que norteiam uma disputa de mercado e enfrentar o grande desafio que é oferecer preços competitivos, menores prazos de desenvolvimento, menos defeitos e maior satisfação dos clientes e usuários. Estes são os requisitos do mercado para um software de qualidade.

Além disso, as organizações têm se deparado com projetos de software cada vez maiores, mais complexos e de grande impacto na sociedade. De acordo com (AZEVEDO, 1998), “O software faz parte do dia-a-dia de toda a sociedade. Ele transfere fundos entre instituições financeiras, pilota aviões, controla equipamentos em centros médicos, diverte as crianças, torna possível pesquisas científicas de grande complexidade aritmética e muito mais. O grande problema é que, em geral, a qualidade do software não é satisfatória por apresentar grande quantidade de defeitos e pela insatisfação de seus clientes e usuários”.

A comunidade científica, o governo e as próprias organizações, em resposta a estas necessidades, estão buscando, cada vez mais, a qualidade de software através de abordagens voltadas tanto para o produto final quanto para o processo de desenvolvimento e manutenção de software.

No entanto, o software continua com sua qualidade comprometida. Um processo de qualidade não garante a produção de um produto de software de qualidade. Percebe-se, neste ponto, uma lacuna nos esforços que vêm sendo realizados na busca pela qualidade de software. O processo, que irá resultar no produto de software, concentra seus esforços na busca pela qualidade do ‘modo de produção e manutenção’ do software, enquanto que a qualidade do produto de software é focada com mais intensidade apenas quando ele já está pronto, através da avaliação de sua qualidade.

As iniciativas pela busca da qualidade de software descritas acima, denominadas abordagem de processo e abordagem de produto respectivamente, são de grande valor e tratam seus objetivos de forma exemplar mas, agem de forma isolada.

A qualidade do produto de software precisa fazer parte, de maneira mais intensa e formal, das preocupações do processo de desenvolvimento e manutenção. As características de qualidade de um produto de software precisam ser alocadas e verificadas também nos produtos de software intermediários ao longo do processo e não apenas no produto já acabado. Este procedimento permite que desvios no produto sejam detectados durante seu processo de desenvolvimento e manutenção, promovendo ações necessárias ao processo para direcioná-lo à produção de um produto de software de qualidade.

1.2 Objetivos

Esta dissertação irá apresentar uma proposta de integração das abordagens de qualidade de software: processo e produto. A integração é realizada através da utilização conjunta do modelo CMM® - *Capability Maturity Model for Software* (CMM, 2000), que foca a qualidade do processo de software, e da Norma brasileira (NBR 13596, 1996), que aborda as características de qualidade de um produto de software.

O trabalho desta dissertação se concentra na identificação de algumas instâncias do modelo de processo CMM® onde os conceitos da norma brasileira NBR 13596 podem ser considerados e utilizados. Para tanto, análises são realizadas em atividades, metas e produtos de trabalho das áreas chave do processo de software, recomendadas pelo CMM® versão 1.1.

A contribuição para a qualidade de software, promovida por esta dissertação, vem no sentido de proporcionar uma alternativa para que o desenvolvimento e manutenção, numa organização que busca a melhoria de seu processo de software, resulte num produto de software também de qualidade.

Além de expor uma alternativa de integrar esforços de melhoria de processo e de produto de software, outros objetivos desta dissertação são: proporcionar uma visão integradora sobre qualidade de software; divulgar o estado da arte na busca pela qualidade de software no Brasil e no mundo; motivar a prática da 'Engenharia de Software' e o uso de normas e modelos existentes na área de qualidade de software.

1.3 Escopo e Limitações

Conforme mencionado anteriormente, o trabalho desta dissertação, de concatenar os esforços de melhoria da qualidade de processo e produto de software, é baseado na versão 1.1 do modelo CMM® (CMM, 2000) e na Norma (NBR 13596, 1996).

A escolha da Norma NBR 13596, versão brasileira da (ISO/IEC 9126, 1991), levou em consideração sua popularidade. O livro ‘Qualidade de Software – Teoria e Prática’ (ROCHA, 2001), relata que, no ano de 1997, 26% das empresas participantes da pesquisa conheciam a Norma ISO/IEC 9126 sendo que 7% a utilizava sistematicamente ou estavam começando a usá-la. Já em 1999, esta porcentagem de conhecimento elevou-se para 36%.

No que diz respeito ao modelo de processo, será visto em capítulo posterior que a versão 1.1 do CMM® está sendo substituído pelo CMMISM - *Capability Maturity Model Integration* (MASTER, 2000), que é um modelo aderente à futura Norma ISO/IEC 15504. Esta Norma será referência internacional para processo de software e também terá seu próprio exemplo de modelo de processo, apresentada na (ISO/IEC TR 15504-2, 1998).

O fato do CMM® estar disponível para uso desde 1993 faz com que ele seja bem mais conhecido e usado do que o CMMISM, disponível apenas no ano 2000. Este foi um fator determinante para a escolha do CMM®, e não do CMMISM no trabalho desenvolvido nesta dissertação; pois, uma aplicação prática a curto prazo da proposta aqui apresentada, em futuros trabalhos, teria mais campo de atuação.

Da mesma forma, o pouco conhecimento e uso do modelo ISO/IEC 15504-2 foi também o motivo de não considerá-lo neste trabalho. De acordo com o levantamento publicado no livro ‘Qualidade e Produtividade em Software’ (WEBER, 2001), o conhecimento e uso do CMM®, por parte das empresas consultadas, passou de 3% em 1995 para 10% em 1999, conforme mostra a Figura 1.1. Enquanto que, segundo esta mesma publicação, o conhecimento e uso do modelo de processo ISO/IEC 15504-2 passou de 1% para apenas 4% no mesmo período.

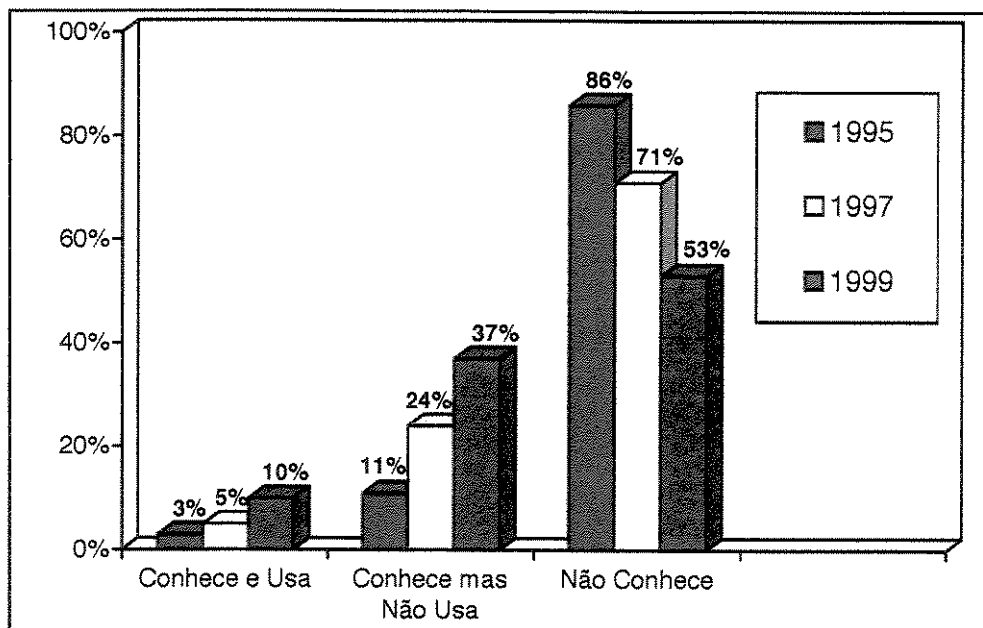


FIGURA 1.1 CONHECIMENTO DO MODELO CMM® (WEBER, 2001)

Além destes fatores, o CMM® e o CMMISM possuem essencialmente o mesmo conteúdo tornando-se relativamente simples uma migração do modelo CMM® para o CMMISM garantindo, desta forma, a aderência à futura Norma ISO/IEC 15504. Na seção 2.3.2, 'Iniciativas para a Qualidade de Processo de Software', os modelos CMMISM e ISO/IEC 15504-2 serão apresentados.

Esta dissertação faz uma análise de algumas metas, atividades e produtos de trabalho das áreas de processo, sugeridas pelo CMM® no desenvolvimento e manutenção de software, podendo existir outras instâncias e outras áreas de processo passíveis de aplicação da Norma NBR 13596.

Alguns fatores que limitam uma análise exaustiva das possibilidades de aplicação da Norma NBR 13596 são os seguintes:

- Existem diversos modelos de processo e normas com conjunto de áreas de processo distintas como, por exemplo, o CMMISM e a ISO/IEC 15504-2;

- O próprio modelo CMM[®] não aborda todas as áreas do processo de software mas apenas as que foram julgadas como ‘fundamentais’ e;
- O modelo CMM[®] sugere 316 práticas chave para o processo de software, sendo 150 delas atividades voltadas diretamente para a implantação das áreas chave de processo, o que é um número bastante elevado para ser abordado na sua totalidade.

O modelo CMM[®] não aborda a forma de implementação das práticas chave sugeridas e, portanto, diferentes formas são adotadas pelas organizações de software. Estas variações de implementação não são abordadas nesta dissertação.

Uma aplicação prática da proposta desta dissertação não foi o foco do trabalho e, desta forma, métricas não são apresentadas para verificar os resultados possíveis.

Um capítulo exclusivo é dedicado ao tema ‘Qualidade’ mas, neste momento é fundamental que o termo ‘Qualidade de Software’ seja definido e compreendido dentro do escopo desta dissertação.

A Norma (NBR ISO 8402, 1993) define ‘Qualidade’ como sendo a “Totalidade das características de uma entidade, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas”. A Norma (NBR 13596, 1996) substituiu o termo ‘entidade’, da definição acima, por ‘produto de software’ e, desta forma, define ‘Qualidade de Software’ como a “Totalidade das características de um produto de software, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas”.

A (NBR 13596, 1996) define ‘produto de software’ como sendo a “Entidade de software disponível para liberação a um cliente ou usuário”; que frequentemente é chamado de produto final de software; e, ‘Software’ como “Programas, procedimentos, regras e qualquer documentação associada pertinente à operação de um sistema computacional”.

Esta dissertação, baseada nas informações acima, reconhece a definição de ‘Qualidade’, fornecida pela Norma (NBR ISO 8402, 1993) e expande o entendimento, que a Norma (NBR

13596, 1996) tem, do termo ‘entidade’, assumindo o termo ‘software’ como sendo a composição ‘processo e produto de software’. E, por ‘processo’ acata-se a definição do SEI - *Software Engineering Institute* (SEI, 2001) que o descreve como o “Conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter software e seus produtos associados (CMM, 2000)”.

Desta forma, ‘Qualidade de Software’ considerada nesta dissertação significa a ‘Totalidade das características de software, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas’, onde por ‘software’ entende-se ‘processo e produto de software’.

1.4 Estrutura da Dissertação

Inicialmente neste capítulo, através de uma descrição do panorama histórico da área de TI, a situação da qualidade de software no Brasil foi apresentada para se compreender a necessidade dos esforços que vêm sendo empregados na sua melhoria. Em seguida, foram abordados os objetivos, escopo e limitações da dissertação que é focada no modelo de processo CMM® v.1.1 e na Norma NBR 13596 de produto de software.

No Capítulo 2 são abordados os temas: ‘Qualidade de Produto’, de forma geral; ‘Qualidade de Produto de Software’ e; ‘Qualidade de Processo de Software’. Os dois últimos temas são desenvolvidos através da apresentação de seus aspectos gerais e principais iniciativas na busca pela qualidade. Além disso, para o tema ‘Qualidade de Produto de Software’ é realizada uma análise das particularidades de um software quando comparado a um produto manufaturado.

Em seguida, são realizadas revisões da literatura sobre o modelo de processo de software CMM® e a Norma NBR 13596, que aborda as características de qualidade de produto de software, nos Capítulos 3 e 4 respectivamente.

O Capítulo 5 é dedicado a desenvolver o foco principal do trabalho, apresentando a proposta de integração das abordagens de qualidade de software: processo e produto. A seção 5.6

resume as sugestões, presentes nas seções anteriores deste capítulo, proporcionando uma leitura rápida da proposta desenvolvida nesta dissertação.

Finalizando, o Capítulo 6 apresenta a conclusão e oportunidades para novos trabalhos que podem ser realizados de forma a dar continuidade nos esforços para a melhoria da qualidade de software.

Capítulo 2

2 Qualidade

2.1 Qualidade de Produto

Existem várias propostas de definição para ‘qualidade de produto’. Entre elas (CROSBY, 1979) afirma que: “Qualidade é conformidade com os requisitos”. Além disso, ele prossegue dizendo que é preciso definir qualidade desta forma principalmente porque qualidade deve ser gerenciada.

Para entender o significado de qualidade em termos práticos é necessário conhecer os cinco principais erros cometidos por gerentes e como estes devem ser tratados que, de acordo com (CROSBY, 1979), são:

- E1 Qualidade significa ‘ótimo’, ou ‘luxo’, ou ‘brilhante’, ou ‘de grande valor’.
- S1 A palavra ‘qualidade’ muitas vezes é usada em expressões do tipo: ‘boa qualidade’, ‘má qualidade’ e até ‘qualidade de vida’. Mas, cada um que a ouve assume um significado para o que seja ‘qualidade de vida’, independentemente do que a pessoa que diz realmente quer dizer com isso. É preciso definir ‘qualidade de vida’ em termos específicos, tais como: renda familiar; saúde; controle de poluição; programas políticos e quaisquer outros itens que possam ser medidos.
- E2 Qualidade é intangível e portanto não mensurável.

- S2 Na verdade, qualidade é precisamente mensurável através da mais antiga e respeitada métrica, o dinheiro. Ignorar este fato tem levado gerentes a perder muito dinheiro. A qualidade é medida pelo ‘custo da qualidade’, que é a despesa, ou custo da ‘não conformidade’, que é o custo de fazer coisas erradas.
- E3 As desculpas dos gerentes para não fazer nada, em relação a qualidade de seus produtos, é que ‘seu negócio é diferente’ e que a ‘ciência’ da qualidade não os ajudaria a fazer o que já fazem, de forma ainda melhor.
- S3 Eles ainda não compreenderam o real significado de ‘qualidade’ e continuam acreditando que ela significa ‘luxo’. Nestes casos é importante explicar o real significado de ‘qualidade’ e que é sempre mais barato fazer certo na primeira vez.
- E4 Os problemas de qualidade são originados por trabalhadores, principalmente aqueles que trabalham diretamente na área de produção.
- S4 Os funcionários da produção de uma fábrica são acusados de provocar os problemas. Na realidade eles pouco contribuem para a prevenção, ou não, de defeitos, pois todo planejamento e criação foram definidos previamente e eles são apenas seus executores.
- E5 Qualidade é responsabilidade do departamento da qualidade.
- S5 O departamento da qualidade deve: medir a conformidade de acordo com o que foi previamente determinado; reportar os resultados destas medidas de forma clara e objetiva; liderar uma atitude positiva da empresa na busca da melhoria da qualidade; prover e capacitar os funcionários com ferramentas que podem auxiliar na melhoria da qualidade. Porém, o departamento da qualidade ‘não deve’ executar o trabalho pois, caso contrário, a empresa nunca mudará sua conduta.

Resumidamente, (CROSBY, 1979) afirma que ‘qualidade’ é conformidade com requisitos e, que estes devem estar definidos para permitir que sejam gerenciados com o uso de medidas, de forma a reduzir o retrabalho e aumentar a produtividade. Crosby afirma também que a Melhoria

da Qualidade deve estar focada nos processos e não nas pessoas e, certamente, é responsabilidade de todos.

De acordo com (JURAN, 1995), a satisfação com o produto está relacionada com o seu desempenho e, a ausência de insatisfação com o produto está relacionada com a ausência de deficiência. Portanto, a satisfação com o produto é alcançada quando as necessidades do cliente são supridas e o produto não apresenta defeitos. Os requisitos, pregados por Crosby, representam as necessidades explícitas dos clientes, e devem procurar cobrir a maior parte das necessidades dos clientes em relação ao produto.

Sobre as necessidades do cliente, (AZEVEDO, 1998) afirma que “pode haver necessidades do cliente que não estão na especificação dos requisitos, mas que também são de fundamental importância: estas são as necessidades implícitas definidas por Juran”.

Necessidades explícitas nada mais são do que a definição dos requisitos, as condições e os objetivos propostos para o produto. Já as necessidades implícitas focam a visão subjetiva do consumidor/usuário quanto ao produto, tais como: as necessidades razoáveis; implicações estéticas; itens de segurança, entre outras. As necessidades explícitas para um carro popular pode ser, por exemplo: ser econômico, fazendo 20 km com 1 litro de combustível, e custar no máximo R\$ 14.000,00; e, como necessidades implícitas: ser confortável; ter um *design* sofisticado; possuir itens de segurança e seu motor jamais falhar.

Desta forma, o melhor a fazer, de acordo com (CROSBY, 1979), é definir muito bem os requisitos de um produto e gerenciar seu processo para que, considerando Juran (JURAN, 1995), as necessidades implícitas e os defeitos possam ser reduzidos ao mínimo. Tudo isso sem deixar de monitorar a satisfação do cliente para com o produto, o que irá orientar os esforços na busca pela qualidade.

Nesta busca pela qualidade, as organizações procuram se basear em conceitos e diretrizes reconhecidos internacionalmente como as normas, que são elaboradas e revisadas por órgãos responsáveis por normalizações técnicas, tais como: ISO – *International Organization for*

Standardization (ISO, 2001), IEC – *International Electrotechnical Commission* (IEC, 2001) e ABNT - Associação Brasileira de Normas Técnicas (ABNT, 2001).

Como já mencionado no capítulo anterior, a Norma (NBR ISO 8402, 1993), sobre terminologias de gestão e garantia da qualidade, define ‘qualidade’ como sendo: “A totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas.”

De acordo com (AZEVEDO, 1998), esta definição é abrangente o suficiente para incorporar toda a visão de qualidade, basicamente relacionada a produtos manufaturados, transmitida por Crosby e Juran, e que pode também orientar as atividades para obtenção e avaliação da qualidade de software. O termo ‘totalidade das características’ indica que tanto aspectos funcionais quanto os não funcionais devem ser considerados. E, o termo ‘satisfazer necessidades implícitas e explícitas’ ressalta a importância da conformidade tanto com os requisitos, necessidades explícitas, como com a satisfação daquelas necessidades que não necessariamente estão descritas nos requisitos, as necessidades implícitas mencionadas acima.

Na próxima seção, o software será abordado quanto aos seus aspectos gerais e suas características como produto; além de apresentar as iniciativas e abordagens na busca por sua qualidade.

2.2 Qualidade de Produto de Software

2.2.1 Aspectos Gerais sobre Software

O software está presente nas mais diversas áreas das quais, algumas vezes, simplesmente são usados, outras vezes, são a ferramenta de trabalho de muitas pessoas e ainda, em outras tantas, vidas são colocadas sob sua influência. Algumas destas áreas são: Educação, Entretenimento, Transporte, Comunicação, Sistema Financeiro, Meio Ambiente, Indústria, Comércio, Medicina, Pesquisa e muitas outras de igual ou maior importância.

Um pouco da história do desenvolvimento de software, contada por (AZEVEDO, 1998), relata os problemas e a busca pela qualidade dos produtos:

Inicialmente, nos anos 50, a grande maioria dos software eram específicos, com distribuição limitada e orientados a *batch*, ou seja executados em lotes. As exceções a este quadro eram o sistema interativo da *American Airlines* e o sistema em tempo real do Departamento de Defesa americano, em função da necessidade que estes sistemas tinham de prover informações para possibilitar a tomada de decisões no exato momento em que determinadas situações ocorriam.

Nos anos 60 surgiu a demanda por novas aplicações e com níveis de sofisticação maiores. Foi neste período que apareceram os software multiusuários; processamento em tempo real; banco de dados e os pacotes de software, que são software que atendem a uma gama de usuários com necessidades semelhantes. Com toda esta demanda, era natural o surgimento de empresas de desenvolvimento de software, chamadas *Software-House*.

Problemas começaram à partir deste momento. Quando era necessária uma adaptação num software que foi comprado, ou quando era detectada uma falha, muitas linhas de código precisavam ser introduzidas, revisadas, corrigidas e testadas. O tempo, envolvido nestas atividades, começou a crescer de forma descontrolada e sua eficácia não podia ser garantida.

Todos estes fatos provocaram, segundo (ROCHA, 2001), a chamada Crise do Software e, para combatê-la, foi intensificado o desenvolvimento de métodos, ferramentas e procedimentos para o desenvolvimento de software, que são integradas na disciplina Engenharia de Software.

A Engenharia de Software, segundo (PRESSMAN, 1995), "...é uma disciplina que pode ser vista, de forma objetiva, como o estabelecimento e o uso dos princípios básicos da engenharia com a finalidade de desenvolver software de maneira sistemática e econômica, resultando em um produto confiável e eficiente." Ou seja, a Engenharia de Software é uma disciplina que auxilia na melhoria da qualidade de software.

Nos anos 70, os computadores pessoais chamados PC, com seus preços acessíveis, tornaram-se uma grande comodidade aos usuários de informática, que começaram a comprar software em grande escala. Inicia-se, também nesta década, o desenvolvimento de software para serem acoplados em automóveis, aparelhos de microondas e outros equipamentos.

Em meados dos anos 80, novas tecnologias tomam o lugar das convencionais. Neste período presencia-se a inteligência artificial, com os sistemas especialistas, saindo dos laboratórios; além da tecnologia de programação Orientada a Objetos, que parecia ser um caminho para a solução dos problemas de manutenção.

Atualmente, apesar de todos os esforços, muitos problemas continuam ocorrendo. Isso acontece principalmente devido a alguns aspectos bastante conhecidos pelas organizações:

- A habilidade de construir software não acompanha o potencial dos novos hardware, que é cada vez mais elevado;
- A capacidade de manter os programas existentes está ameaçada por projetos mal planejados e recursos inadequados e;
- Os projetos de software estão ficando cada vez mais complexos.

Com isso a quantidade de defeitos e a insatisfação dos usuários são crescentes. A visibilidade destes fatos ficam por conta dos acidentes aéreos, que não conseguem a comprovação de falha humana como causa; o controle, ou falta dele, de plantas petrolíferas e químicas; a complexidade das redes elétricas e de telecomunicações; além de inúmeros outros exemplos que poderiam ser citados.

A importância e os problemas relacionados a software deixam claro a relevância da busca pela melhoria de sua qualidade. Na próxima seção, o software será analisado em suas características particulares e suas diferenças em relação ao produto manufaturado.

2.2.2 Software e suas Características como Produto

Existem diferenças importantes entre produtos de software e produtos manufaturados que não podem deixar de ser notadas. As características inerentes à essência do software e diferenças em relação aos produtos manufaturados, segundo (CAPOVILLA, 1999) e (AGUAYO, 2001), são:

- **Complexidade:** Normalmente, um produto de software tem muitas regras a serem cumpridas; muitas linhas de código a serem implementadas; e, freqüentemente, diversos desenvolvedores envolvidos que não só tem idéias diferentes, e algumas vezes divergentes, mas que podem levar à mesma solução;
- **Invisibilidade e Intangibilidade:** O software é invisível para o usuário. O que se vê são as conseqüências da execução do software, diferentemente de um produto manufaturado. Os próprios desenvolvedores necessitam utilizar modelos para representar o sistema de software. Esta intangibilidade causa grandes dificuldades de comunicação tanto entre os elementos da equipe de desenvolvimento como entre a equipe e o cliente podendo acarretar problemas no produto de software;
- **Conformidade e Modificabilidade:** O software é a interface entre diversas entidades do meio no qual será utilizado: equipamentos, outros software, usuários e cultura organizacional entre outras. Frequentes adaptações são realizadas no software para adequá-lo a estas entidades, por ser o componente mais maleável e adaptável do sistema;
- **Produção sob Medida:** Para software não existe produção em série: cada usuário é um cliente, que usa o software à sua maneira, com ênfase em partes diferentes;
- Não se desgasta com o uso e não tem prazo de validade;
- O custo final do software é basicamente o custo do projeto e do desenvolvimento;
- Software é o único produto que quando apresenta erro o cliente paga para corrigir.

Diante destas diferenças, é difícil imaginar, de forma direta, o aproveitamento, no desenvolvimento de software, de toda a experiência e maturidade existentes no processo de fabricação de um produto manufaturado. Mas, na seção ‘Aspectos Gerais sobre Software’ deste

capítulo, é citado por (PRESSMAN, 1995), que a Engenharia de Software pode ser vista como sendo “o estabelecimento e uso dos princípios básicos da engenharia com a finalidade de desenvolver software de maneira sistemática e econômica, resultando em um produto confiável e eficiente”.

As organizações de software, que se preocupam com a qualidade, vêm utilizando a Engenharia de Software. A cada dois anos, a Secretaria de Política de Informática e Automação do Ministério da Ciência e Tecnologia conduz pesquisas diretas com organizações de software, visando acompanhar a evolução da gestão da qualidade neste setor. Conforme a Tabela 2.1, reproduzida a partir dos dados das três últimas edições desta pesquisa, é possível constatar a utilização de métodos de Engenharia de Software, o que indica a credibilidade em seu potencial.

TABELA 2.1 PESQUISA DE UTILIZAÇÃO DE MÉTODOS DE ENGENHARIA DE SOFTWARE (MCT, 2001)

MÉTODOS DE ENGENHARIA DE SOFTWARE	RESULTADOS (Nº de Empresas / %)		
	1995 (445 empresas)	1997 (589 empresas)	1999 (426 empresas)
Empresas que adotam métodos de prevenção de defeitos			
Auditorias	...	102 / 17%	88 / 21%
Gerência de Configuração	...	40 / 7%	63 / 15%
<i>Joint Application Design</i>	...	46 / 8%	36 / 9%
Medições da Qualidade (Métricas)	45 / 10%	48 / 8%	52 / 12%
Prototipação	207 / 46%	259 / 44%	187 / 44%
Reuso	166 / 37%	110 / 19%	104 / 24%
Verificação independente	...	81 / 14%	155 / 36%
Empresas que adotam métodos de detecção/remoção de defeitos			
Inspeções formais	45 / 10%	100 / 17%	87 / 20%
Revisões estruturadas	...	113 / 19%	66 / 16%
Testes de aceitação	212 / 48%	278 / 47%	205 / 48%
Testes de sistema	277 / 62%	392 / 67%	199 / 47%
Testes de unidade	105 / 24%	137 / 23%	130 / 31%
Validação	...	250 / 42%	192 / 45%
Empresas que adotam outras práticas de Engenharia de Software			
Gestão de mudança	...	32 / 5%	31 / 7%
Métodos estruturados	...	210 / 36%	146 / 34%
Métodos orientados a objetos	193 / 43%	216 / 37%	186 / 44%
Projetos interface com o usuário	...	207 / 35%	215 / 51%

Obs.: O símbolo ‘...’ indica dado não disponível

Apesar de todos os esforços da Engenharia de Software, os problemas de qualidade nos produtos persistem. Na próxima seção serão analisadas outras iniciativas que vêm sendo realizadas na busca pela qualidade do produto software.

2.2.3 Iniciativas para a Qualidade de Produto de Software

Diante do que foi apresentado nas seções anteriores, o produto de software, a algum tempo, necessita e busca por soluções para a melhoria na sua qualidade; tanto em função de seu grau de importância e integração na sociedade, quanto pelas falhas frequentes e com possibilidade de consequências desastrosas. Tudo isso, sem deixar de mencionar também, o alto custo e elevado tempo de desenvolvimento e manutenção.

Sobre qualidade de produto de software, (ROCHA, 2001) afirma que “Qualidade de software pode ser vista como um conjunto de características que devem ser alcançadas, em um determinado grau, para que o produto atenda às necessidades de seus usuários. É por meio desse conjunto de características que a qualidade de um produto de software pode ser descrita e avaliada.”

Analisando esta afirmação, percebe-se a necessidade de respostas para algumas questões básicas sobre como atingir a qualidade num produto de software a ser desenvolvido:

- A determinação do conjunto de características que atende as necessidades de seus usuários.
- A forma de avaliar se estas características foram alcançadas num grau que satisfaça seus usuários.

Diversas iniciativas surgiram para buscar as respostas destas questões. De acordo com o livro ‘Qualidade de Software – Teoria e Prática’ (ROCHA, 2001), diversos modelos foram desenvolvidos, detalhando as características de qualidade de um produto de software em subcaracterísticas e, estes em atributos. Alguns destes trabalhos são encontrados nos livros: ‘*Factors in software quality*’ de McCALL, editado em 1977, ‘*Characteristics of software quality*’

editado em 1978 por B. Boehm e '*Cornering the chimera*' de R. G. Dromey, 1996. No Brasil estas questões foram inclusive assunto de tese de doutorado, como a de A. R. C. da Rocha defendida em 1983 com o título 'Um modelo para avaliação da qualidade de especificações'.

O Subcomitê de Software - SC7 do Comitê Técnico Conjunto - JTC1 da ISO (ISO, 2001) e IEC (IEC, 2001) vem trabalhando na elaboração de normas e relatórios técnicos que permitam especificar e avaliar a qualidade de produtos de software, consolidando as diferentes visões de qualidade em uma norma internacional.

Para auxiliar no processo em si de avaliação da qualidade de produtos de software, a ISO e IEC estabeleceram o seguinte conjunto de normas: (ISO/IEC 14598-1, 1999) com uma visão geral do processo de avaliação; (ISO/IEC 14598-2, 2000) sobre o planejamento e gestão do processo; (ISO/IEC 14598-3, 2000) sobre o processo de avaliação para desenvolvedores; (ISO/IEC 14598-4, 1999) sobre processo para adquirentes; (ISO/IEC 14598-5, 1998) sobre processo para avaliadores e; (ISO/IEC 14598-6, 2001) sobre documentação de módulos de avaliação.

A norma brasileira (NBR 13596, 1996) é uma tradução da Norma (ISO/IEC 9126, 1991) contendo as características de qualidade de um produto de software e as diretrizes de utilização destas características em uma avaliação. Diante do objetivo e da abrangência desta dissertação, abordados no capítulo introdutório, será utilizada apenas a Norma NBR 13596 durante o processo de software. Esta Norma será analisada com mais detalhes no Capítulo 4.

Atualmente, o conjunto de normas ISO, que aborda avaliação da qualidade de produto de software, é composto por um conjunto de documentos: (ISO/IEC FDIS 9126-1, 2001) sobre o modelo de qualidade, (ISO/IEC 9126-2, 2001) sobre métricas externas, (ISO/IEC 9126-3, 2001) sobre métricas internas e (ISO/IEC 9126-4, 2001) sobre métricas da qualidade em uso.

Para se obter um produto software de qualidade é preciso verificar o seu processo de desenvolvimento. Na próxima seção serão abordados aspectos gerais de um processo de software, iniciativas e modelos para sua qualidade.

2.3 Qualidade de Processo de Software

2.3.1 Aspectos Gerais sobre Processo de Software

O motivador desta dissertação é a busca pela ‘Qualidade de Software’, ou seja, qualidade tanto do produto como do processo de software. Um conceito bastante difundido atualmente é que se deve atuar na qualidade do processo para se atingir a qualidade do produto de software.

A Figura 2.1 representa a definição de ‘Processo de Software’, encontrada em (CMM, 2000) e considerada nesta dissertação, que é “...um conjunto de atividades, métodos, práticas e transformações que as pessoas usam para desenvolver e manter software e seus produtos associados”.

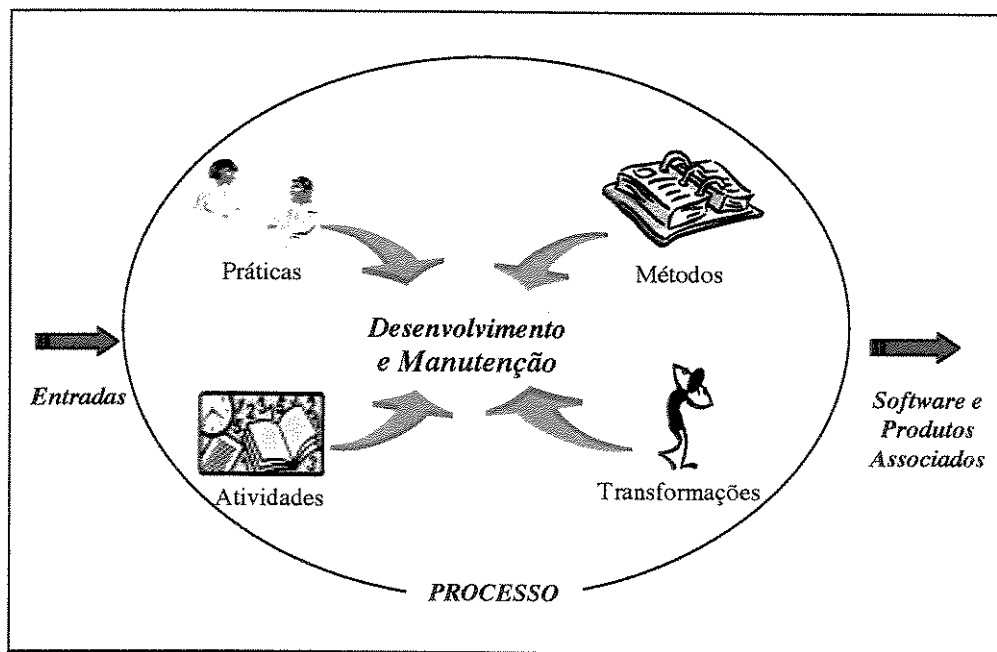


FIGURA 2.1 DEFINIÇÃO DE PROCESSO DE SOFTWARE

Esta definição relaciona processo a ações e acontecimentos, realizados por pessoas, para o desenvolvimento, em um sentido mais amplo do termo e não somente na construção propriamente dita do software, e para a manutenção tanto do software como de outros produtos

associados a ele como, por exemplo, os manuais de usuário. Tanto o produto de software como o produto manufaturado têm como requisito, para um processo de qualidade, a necessidade de ser uma atividade sistemática e passível de repetição, independentemente de quem a execute.

Como representa a Figura 2.2, a situação atual da maioria das organizações de software é complicada. Existe um grande acúmulo de trabalho, o que acarreta no abandono de planos e procedimentos, quando estes estão definidos, e as consequências disso são produtos de software que às vezes funcionam mas os prazos e custos são freqüentemente maiores do que o previsto.

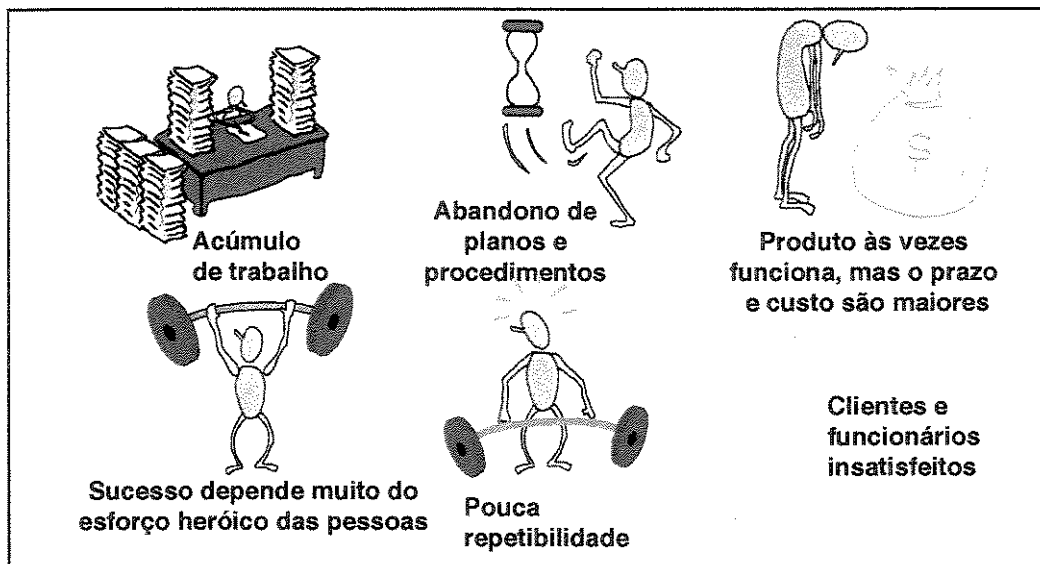


FIGURA 2.2 SITUAÇÃO ATUAL DA MAIORIA DAS ORGANIZAÇÕES DE SOFTWARE (MAGNANI, 1998)

As organizações percebem esta situação caótica mas encontram dificuldades em mudar este quadro em função de uma série de fatores, apontados por (SOUZA, 2001) como sendo, entre outros:

- As organizações estão reagindo a crises constantes;
- A ausência de guias, da mesma forma que existe nas demais engenharias, de métodos testados e comprovados de como desenvolver e fazer manutenção de software. Boas práticas estão sendo utilizadas mas são pouco disseminadas;

- A maioria dos problemas nas organizações de software é gerencial e não técnico. A tecnologia e a capacitação dos profissionais, que estas organizações já possuem, poderia gerar melhor qualidade;
- A alta velocidade de mudanças tecnológicas envolvidas, como o próprio hardware;
- O baixo estímulo à participação do usuário no processo de software, o que conduz a uma especificação incompleta de suas necessidades;
- A crescente demanda por novos software e pela manutenção dos já desenvolvidos, como a própria Figura 2.2 representa;
- A falta de conhecimento de como mudar esta realidade e dificuldade em aplicar os princípios da qualidade à software.

Apesar da demanda por qualidade, os avanços da engenharia de software, comentada na seção ‘Qualidade de Produto de Software’, não consegue acompanhar e solucionar os problemas, sendo necessário outros esforços neste sentido. Este será o assunto da próxima seção.

Tendo como referência a Figura 2.2, fica clara a dependência existente entre a qualidade do produto de software e seu processo. Desta forma, qualquer esforço no sentido de melhorar a qualidade de produto de software deve estar engajado na melhoria da qualidade de processo de software da organização.

2.3.2 Iniciativas para a Qualidade de Processo de Software

Na década de 90 houve uma grande preocupação com a modelagem e melhoria no processo para a produção de software de qualidade, dentro do prazo e com orçamentos confiáveis.

Algumas normas e modelos desenvolvidos na busca pela qualidade de processo de software são: (NBR ISO 9000-3, 1993) que é um guia de aplicação da NBR 19001 para o desenvolvimento, fornecimento e manutenção de software; (NBR ISO/IEC 12207, 1998) que define os processos de ciclo de vida do software; a futura Norma ISO/IEC 15504 sobre avaliação de processo de software, onde um modelo para avaliação é descrito no documento (ISO/IEC TR 15504-5, 1998) e, um modelo referência de processo é apresentado em (ISO/IEC TR 15504-2,

1998); o CMMISM (MASTER, 2000) desenvolvido nos EUA, que irá substituir em breve o CMM[®] (CMM, 2000); o BOOTSTRAP, modelo de processo europeu (HAASE, 1994) e; o TRILLIUM, modelo desenvolvido no Canadá (UHCL, 2001).

Em função da importância e relação com o objetivo desta dissertação, apenas a futura Norma ISO/IEC 15504 e o modelo CMMISM terão um breve resumo a seguir. O CMM[®] terá um capítulo dedicado a ele, por ser o modelo de processo a ser utilizado neste trabalho de dissertação.

A melhoria da qualidade de processo de software deve ser o dia-a-dia de uma organização de software. Conforme afirma (ROCHA, 2001), com o sucesso desta empreitada, a previsão para os anos 2000 é um esforço concentrado na redução substancial dos prazos para a entrega de produtos pelas organizações. A pressão dos concorrentes será intensa. Organizações que sejam capazes de integrar, harmonizar e acelerar seus processos de desenvolvimento e manutenção de software terão a primazia do mercado.

ISO/IEC 15504

O projeto SPICE – *Software Process Improvement and Capability dEtermination* (SPICE, 2001), iniciado em janeiro de 1993, surgiu da necessidade de um padrão internacional para processo de software uma vez que ocorria a proliferação de modelos de processo tais como CMM[®], Bootstrap e Trillium. A abordagem do SPICE para atingir a Qualidade de Software é focada no Processo de Software.

A organização do projeto SPICE ocorre através de uma estrutura própria com 5 Centros Técnicos compostos por: Canadá/América Latina, Europa/África do Sul, EUA, Pacífico Norte e Pacífico Sul. O SPICE, para obter informações da aplicação de seu modelo de processo, incentiva sua utilização experimental e a coleta de dados através dos chamados *trials*. Os *trials* tem uma estrutura similar ao SPICE com coordenações Nacionais.

No Brasil, a coordenação dos *trials* está a cargo do CenPRA – Centro de Pesquisas Renato Archer (CenPRA, 2002), órgão do Governo Federal ligado diretamente ao MCT - Ministério da Ciência e Tecnologia (MCT, 2001), localizado em Campinas, Estado de São Paulo.

Em 1994, 35 avaliações já tinham sido realizadas, utilizando a versão inicial do modelo de processo proposto, com participação no chamado *Trials-Fase 1*. Em 1996 foi fechada a versão PDTR – *Proposed Draft Technical Report* e; em 1997, a versão DTR - *Draft Technical Report*, que foi utilizada nos *Trials-Fase 2* com 70 avaliações realizadas. A versão TR – *Technical Report*, denominada ISO/IEC TR 15504: *Information Technology – Software Process Assessment*, foi lançada em 1998 dando início aos *Trials-Fase 3* com, aproximadamente, 100 avaliações realizadas, sendo 3 delas no Brasil. Esta é a versão que vem sendo utilizada e existe a previsão de transformação desta TR na Norma ISO/IEC 15504 no ano de 2002 (SOUZA, 2001).

A relação da futura Norma ISO/IEC 15504 com os modelos de processo pode ser verificada na Figura 2.3, onde está representada sua paridade com as demais normas e sua predominância em relação aos modelos, que devem ser compatíveis aos requisitos desta Norma.

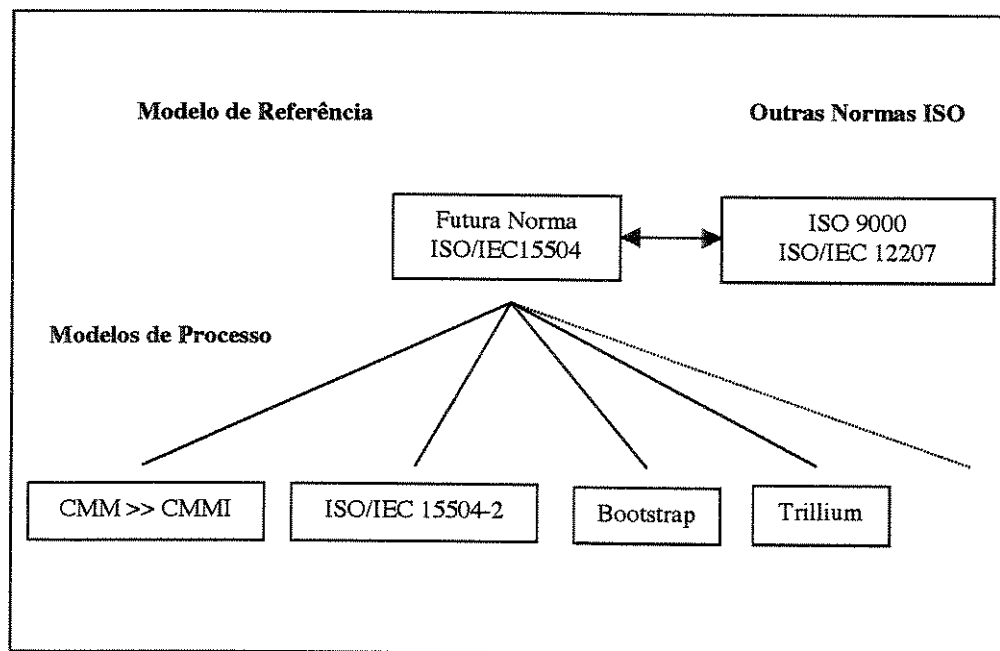


FIGURA 2.3 RELAÇÃO DA ISO/IEC 15504 COM OS MODELOS DE PROCESSO

A própria futura Norma ISO/IEC 15504 sugere um modelo de processo, que é a ISO/IEC 15504-2. O modelo CMMISM, que substituirá o CMM[®], também é um dos modelos já compatíveis com esta futura Norma.

A arquitetura proposta na futura Norma ISO/IEC 15504 é a bidimensional, sendo uma para os processos fundamentais em evidência e a outra para o nível de capacidade destes processos, conforme exemplifica a Figura 2.4. Os níveis de capacidade em que os processos são classificados são: 0 – Incompleto, 1 – Executado, 2 – Gerenciado, 3 – Estabelecido, 4 – Previsível e 5 – Otimizando.

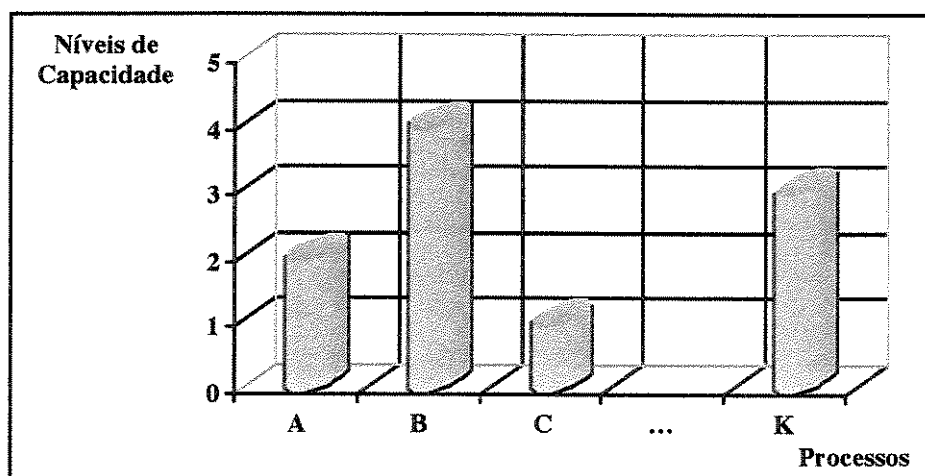


FIGURA 2.4 NÍVEIS DE CAPACIDADE E PROCESSOS

A avaliação e melhoria dos processos fundamentais são orientadas pelos objetivos de negócio da organização e envolve o planejar, gerenciar, executar, controlar e melhorar os aspectos de software quanto à adquirir, fornecer, desenvolver, operar, efetuar manutenção e fornecer suporte.

O modelo de referência da futura Norma ISO/IEC 15504, estabelece um conjunto universal de 40 processos fundamentais para a engenharia de software divididos em categorias, conforme Tabela 2.2, e um roteiro racional para a determinação da capacidade, através de uma avaliação, e também para a melhoria de cada processo.

TABELA 2.2 CATEGORIAS E PROCESSOS DA FUTURA NORMA ISO/IEC 15504 (WEBER, 2001)

CATEGORIAS	SIGLAS	PROCESSOS FUNDAMENTAIS
Cliente-Fornecedor	CUS.1	Aquisição
	CUS.1.1	Preparação da Aquisição
	CUS.1.2	Seleção de Fornecedor
	CUS.1.3	Acompanhamento do Fornecimento
	CUS.1.4	Aceitação pelo Cliente
	CUS.2	Fornecimento
	CUS.3	Elicitação de Requisitos
	CUS.4	Operação
	CUS.4.1	Uso Operacional
	CUS.4.2	Suporte ao Cliente
Engenharia	ENG.1	Desenvolvimento
	ENG.1.1	Análise de Requisitos e Projeto de Sistema
	ENG.1.2	Análise de Requisitos de Software
	ENG.1.3	Projeto de Software
	ENG.1.4	Construção de Software
	ENG.1.5	Integração de Software
	ENG.1.6	Teste de Software
	ENG.1.7	Integração e Teste de Sistema
Gerência	ENG.2	Manutenção de Sistema e Software
	MAN.1	Gerência
	MAN.2	Gerência de Projeto
	MAN.3	Gerência da Qualidade
Organização	MAN.4	Gerência de Riscos
	ORG.1	Alinhamento Organizacional
	ORG.2	Melhoria
	ORG.2.1	Estabelecimento do Processo
	ORG.2.2	Avaliação de Processo
	ORG.2.3	Melhoria de Processo
	ORG.3	Gerência de Recursos Humanos
	ORG.4	Infra-estrutura
	ORG.5	Medição
Suporte	ORG.6	Reuso
	SUP.1	Documentação
	SUP.2	Gerência de Configuração
	SUP.3	Garantia da Qualidade
	SUP.4	Verificação
	SUP.5	Validação
	SUP.6	Revisões Conjuntas
	SUP.7	Auditorias
	SUP.8	Resolução de Problema

Modelo CMMISM

CMMISM significa *Capability Maturity Model Integration* e é a evolução e integração de três diferentes modelos desenvolvidos pelo SEI - *Software Engineering Institute* (SEI, 2001) que são (SALVIANO, 2001):

- SW-CMM[®] ou CMM[®] – *Capability Maturity Model for Software*, que será visto no Capítulo 3.
- SE-CMM[®] - *Systems Engineering Capability Maturity Model*. Descreve os elementos essenciais dos processos de uma organização que devem existir para garantir uma boa prática de engenharia de sistemas.
- IPD-CMM[®] – *Integrated Product Development*. Provê um guia sistemático para o desenvolvimento de produtos ao longo do ciclo de vida, para melhor satisfazer as necessidades dos clientes.

De acordo com (MASTER, 2000), a versão 1.0 do CMMISM, patrocinada pelo U.S. *Department of Defense* – DoD e pela *National Defense Industrial Association* – NDIA, foi lançada em agosto de 2000. Seu desenvolvimento foi uma integração da indústria, governo americano e do SEI. É intenção do SEI abandonar, em 2003, os modelos precursores citados. As metas para o desenvolvimento do CMMISM eram, além de integrarem os modelos acima: eliminar inconsistências; reduzir o custo de implementação do modelo; melhorar o entendimento do modelo e; assegurar a consistência com a futura Norma ISO/IEC 15504.

Com o modelo CMMISM, a organização pode promover a melhoria do processo de software através do aprimoramento da capacidade dos processos ou da maturidade da organização. O CMMISM suporta estas duas abordagens através, respectivamente, das representações ‘contínua’, da mesma forma que a ISO/IEC 15504, e ‘por estágio’, da mesma forma que o CMM[®].

A representação ‘por estágio’ classifica a organização em 5 ‘níveis de maturidade’ quanto ao seu processo de software, como será visto no Capítulo 3, enquanto que a representação ‘contínua’ classifica os processos, e não a organização, em 6 ‘níveis de capacidade’ que são: 0 –

Incompleto, 1 - Realizado, 2 – Gerenciado, 3 – Definido, 4 – Gerenciado Quantitativamente, 5 – Otimizando.

As principais diferenças entre os modelos CMM® e CMMISM são:

- O CMM® estabelece 18 Áreas Chave de Processo, ou KPA - *Key Process Areas* (sigla adotada nesta dissertação), e a representação é ‘por estágio’.
- Já o CMMISM estabelece 22 Áreas de Processo, ou *Process Areas* (PA), que são classificadas em 4 categorias, listadas na Tabela 2.3, e a organização pode escolher entre a representação ‘por estágio’, como CMM®, ou a ‘contínua’.

TABELA 2.3 CATEGORIAS E ÁREAS DE PROCESSO DO MODELO CMMISM (SOUZA, 2001)

CATEGORIAS	ÁREAS DE PROCESSO – PA
Gerenciamento de Processo	Foco no Processo Organizacional Definição do Processo Organizacional Treinamento Organizacional Desempenho do Processo Organizacional Inovação e Melhoria Organizacional
Gerenciamento de Projeto	Planejamento de Projeto Acompanhamento e Controle de Projeto Gerenciamento de Aceite do Fornecedor Gerenciamento Integrado de Projeto Gerenciamento de Risco Gerenciamento Quantitativo de Projeto
Engenharia	Gerenciamento de Requisitos Desenvolvimento de Requisitos Soluções Técnicas Integração de Produto Verificação Validação
Suporte	Gerenciamento de Configuração Garantia da Qualidade de Processo e de Produto Medições e Análise Análise de Causa e Resolução Análise de Decisão e Resolução

A principal contribuição do modelo CMMISM é a liberdade que se adquire, com a representação ‘contínua’, de escolher quais processos devem ser trabalhados naquele momento, respeitando as necessidades prioritárias e particularidades existentes em cada organização.

Capítulo 3

3 SW-CMM®

3.1 Introdução

O contexto histórico do desenvolvimento do modelo de processo SM-CMM® – *Capability Maturity Model for Software*, ou simplesmente CMM®, (CMM, 2000) é bastante importante para compreender seu objetivo e os conceitos de qualidade que estão por traz de sua concepção e estrutura. Esta história é relatada abaixo, de acordo com (SOUZA, 2001).

O governo federal dos EUA, devido a complexidade e grau de confiança necessário em seus software, tinha, e ainda tem, a necessidade de conhecer a capacidade de seus fornecedores quanto ao desenvolvimento de software. Diante deste panorama, o governo federal solicitou ao SEI o desenvolvimento de um método apropriado para este fim.

“O SEI é um centro de pesquisa e desenvolvimento, mantido com verbas do governo americano, está instalado na universidade *Carnegie Mellon*, é gerenciado pelo DARPA – *Defense Advanced Research Projects Agency* e é administrado pelo ESC – *Electronic Systems Center*. Foi estabelecido em 1984 com a missão de prover liderança para o avanço do estado da prática da engenharia de software, melhorando a qualidade dos sistemas que dependem de software (SEI, 2001).”

Em Novembro de 1986, o SEI dá início ao trabalho solicitado pelo governo norte-americano. Em Setembro de 1987, uma versão resumida da estrutura de maturidade de processo

de software é publicada pela IEEE - *Institute of Electrical and Electronics Engineers* (IEEE, 2001) no artigo '*Characterizing the Software Process: A Maturity Framework*' e em 1989 no livro '*Managing the Software Process*'; ambos de autoria de Watts Humphrey.

A primeira versão do CMM® foi publicada em 1991 no relatório técnico CMU/SEI-91-TR-25 '*Key Practices of the Capability Maturity Model for Software v1.0*' e, a versão atualmente em uso foi publicada em 10 de fevereiro de 1993 no relatório técnico CMU/SEI-93-TR-24 '*Key Practices of the Capability Maturity Model for Software v1.1*'. A versão 2.0 do CMM® estava em elaboração e com publicação prevista para o final de 1997, o que foi adiado pelo SEI para dar prioridade ao projeto de integração de diversos modelos desenvolvidos por ele. Este modelo integrador é denominado CMMISM, que foi abordado no capítulo anterior.

O SEI, ao desenvolver o CMM®, teve a preocupação de prover um modelo baseado em práticas reais; que concentrasse as melhores práticas utilizadas; fosse documentado e de domínio público. O CMM® deveria também refletir as necessidades de melhoria e avaliação do processo de software ou, avaliação da capacidade das organizações no desenvolvimento de software.

A versão 1.1 do CMM®, segundo (CMM, 2000), teve como principais subsídios para sua construção:

- Estudo de organizações não exclusivamente desenvolvedoras de software;
- Aplicação, e observação, de avaliações de processo de software e de avaliações de capacidade de software;
- Análise de solicitações, realizadas pela comunidade, de mudanças no modelo;
- Participação em encontros e *workshops* com representantes da indústria e do governo;
- Retorno de análises realizadas no uso do modelo, por representantes da indústria e do governo.

Alguns importantes conceitos utilizados pelo CMM®, que são definidos em (PAULK, 1999), estão descritos abaixo para uma melhor compreensão deste modelo:

- ‘Capacidade’ do Processo de Software: Define os limites de resultados esperados quando se segue o processo de software em uso. A capacidade de software de uma organização prove uma predição do resultado mais provável do próximo projeto de software que a organização vai executar.
- ‘Desempenho’ do Processo de Software: Representa os resultados atuais conseguidos com a uso do processo de software. Ou seja, o desempenho do processo de software foca os resultados alcançados, enquanto que a capacidade do processo de software foca os resultados esperados.
- ‘Maturidade’ do Processo de Software: Significa o quanto um processo específico está explicitamente definido, gerenciado, medido, controlado e efetivo. Maturidade indica o potencial para crescimento em capacidade, a excelência do processo de software da organização e a consistência com que este processo é aplicado em projetos por toda a organização.

O CMM® descreve os elementos chave de um processo de software eficiente e eficaz, além de indicar uma maneira evolutiva de se transformar um processo de software imaturo, onde se trabalha de forma *ad hoc*, em um processo de software maduro, disciplinado. Desta forma, este modelo pode ser utilizado tanto para definir o nível de maturidade de uma organização, quanto ao seu processo de software, como também para orientar um trabalho de melhoria de processo.

São definidos cinco níveis de maturidade, conforme Figura 3.1, que são: 1 - Inicial, 2 – Repetível, 3 – Definido, 4 – Gerenciado e 5 – Otimizando. Cada um destes níveis atua como fundação do nível seguinte e indica a capacidade do processo de software da organização.

O CMM® abrange práticas de planejamento, gerenciamento, desenvolvimento e manutenção de software. Quando estas práticas são realizadas, a organização aumenta sua possibilidade de atingir metas de custo, cronograma, eficiência e qualidade do produto final (CMM, 2000). Ou seja, uma organização de software ‘imaturo’ pode se transformar em uma organização ‘madura’ através da execução das práticas deste modelo.

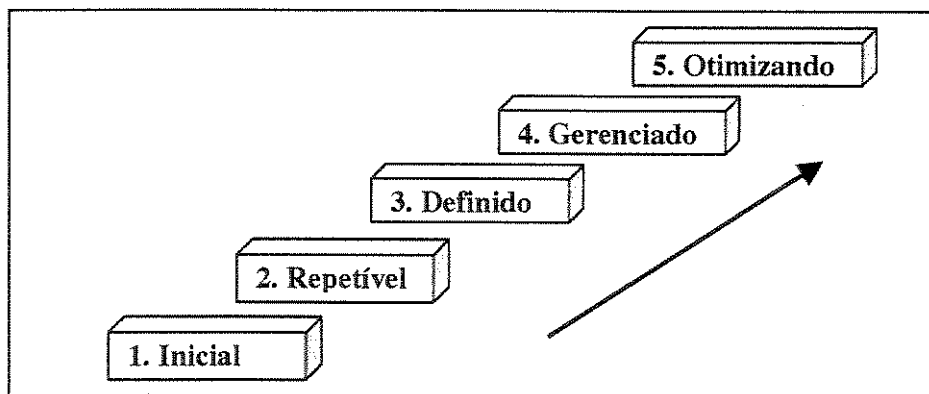


FIGURA 3.1 NÍVEIS DE MATURIDADE DO CMM® VERSÃO 1.1 (CMM, 2000)

Segundo (PAULK, 1999), em uma organização de software ‘imatura’, o processo geralmente é improvisado e, mesmo quando o processo é especificado, nem sempre é seguido. Esta organização é reativa e os gerentes estão normalmente focados em resolver crises constantes. Quando prazos rígidos são impostos para a entrega do software, normalmente as funcionalidades, assim como a qualidade do produto como um todo, ficam comprometidas.

Por outro lado, uma organização ‘madura’ possui seu processo de software bem definido e institucionalizado por toda a organização. O processo é atualizado, quando necessário, e melhorias são desenvolvidas através de testes piloto e/ou análises da relação custo x benefício.

Os gerentes da organização monitoram a qualidade dos produtos de software e dos processos que os geraram, através de bases quantitativas. O planejamento do projeto, incluindo cronograma e custos, é baseado em dados históricos e, portanto, são realistas.

Quanto aos conceitos de qualidade, que estão por traz da concepção deste modelo de processo, de acordo com o livro oficial do SEI (CMM, 2000), o CMM® é uma aplicação em software dos conceitos de gerenciamento de processos do TQM – *Total Quality Management*, conforme será visto nesta próxima seção.

3.2 Relação entre CMM® e TQM (CMM, 2000)

Conforme ilustra a Figura 3.2, TQM – *Total Quality Management* pode ser definido como sendo a aplicação de recursos humanos e métodos quantitativos para melhorar produtos e serviços, que servem como insumo, ou entradas, para uma organização, além de aprimorar todos os seus processos. O objetivo do TQM é prover as necessidades dos clientes, agora e no futuro.

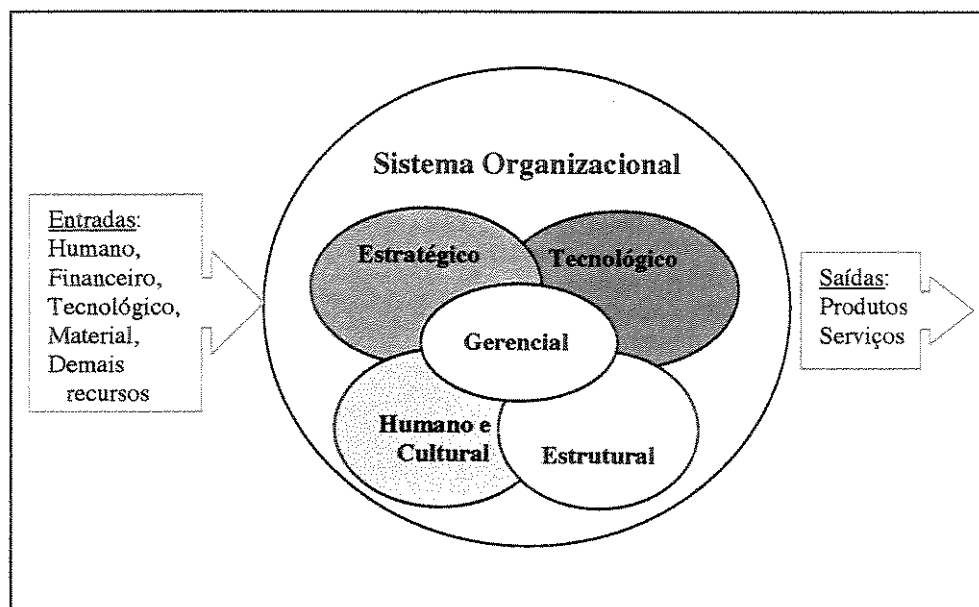


FIGURA 3.2 SISTEMA ORGANIZACIONAL SEGUNDO A VISÃO DO TQM (LAWRENCE, 2001)

O TQM é uma síntese dos conceitos de Edwards Deming (DEMING, 1990), Joseph Juran (JURAN, 1995), Philip Crosby (CROSBY, 1979) e outros. Os pontos comuns dos movimentos da qualidade são:

- A melhoria está focada nos processos, não nas pessoas;
- A qualidade deve ser medida;
- Melhoria da qualidade requer investimentos, recompensa e incentivos;
- Melhoria da qualidade é um processo contínuo e;
- Se o desconforto não for grande o suficiente, mudanças não ocorrerão.

Medir a qualidade de software é uma tarefa difícil mas avanços foram feitos nesta área. O CMM® é baseado nos princípios de qualidade, e portanto a medição é considerada em sua estrutura. Os princípios de controle estatístico da qualidade de Walter Shewart, presentes em seu livro *'Economic Control of Quality of Manufactured Product'* publicado em 1931, que foram largamente desenvolvidos e demonstrados com sucesso nos trabalhos de Edwards Deming e Joseph Juran, têm sido adaptados pelo SEI numa estrutura que estabelece fundamentos de engenharia e gerenciamento de projeto, com o objetivo de proporcionar um controle quantitativo do processo de software, o que é a base para a melhoria contínua de processo.

Philip Crosby, inspirado nestes princípios, descreve em seu livro *'Quality Is Free'* (CROSBY, 1979) cinco estágios evolutivos para adoção de práticas de qualidade no caminho da busca por um gerenciamento maduro da qualidade. Esta estrutura foi adaptada para processo de software por Ron Radice e seus colegas em trabalho supervisionado por Watts Humphrey na empresa IBM.

Humphrey, em 1986 ao ingressar no SEI, utilizou este trabalho para desenvolver o CMM®. Refinou os conceitos de níveis de maturidade e desenvolveu os fundamentos que hoje são largamente utilizados nas organizações de software. A Figura 3.3 representa, de forma simples, a relação existente na aplicação do TQM numa organização e no seu processo de software, através do CMM®.

A estratégia do CMM® é focar num conjunto limitado de atividades do processo de software e trabalhar diretamente nele, possibilitando conquistas estagiadas, mas duradouras, na capacidade do processo de software da organização. Inicialmente, através de uma avaliação realizada na organização, é determinada a maturidade do processo de software corrente da organização, classificando-a em um dos 5 Níveis de Maturidade predefinidos pelo CMM®, que serão detalhados mais a frente.

Além de classificar o Nível atual de Maturidade, outro objetivo da avaliação é prover subsídios para a melhoria do processo de software da organização identificando pontos críticos que estão comprometendo a qualidade do produto de software e seu processo como um todo.

Com estes pontos críticos evidenciados, a organização pode traçar um plano de ação para a melhoria e institucionalização das novas práticas de seu processo de software, possibilitando que o Nível de Maturidade imediatamente seguinte, ao que ela foi classificada, seja atingido. Este ciclo é continuamente realizado, promovendo a melhoria contínua no processo de software da organização. É importante notar que não basta implementar as melhorias necessárias no processo de software, igualmente importante é institucionalizar as novas práticas, ou seja, incorporá-las no dia-a-dia da organização.

Para um completo entendimento do CMM[®], sua estrutura é apresentada a seguir formando a base para o detalhamento de suas entidades, que serão o assunto das seções posteriores.

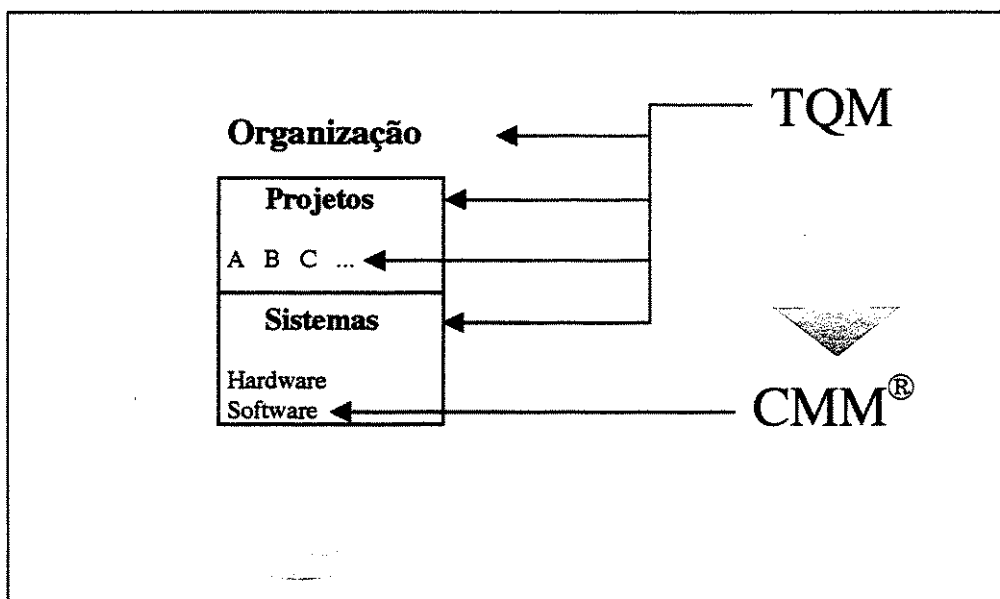


FIGURA 3.3 APLICANDO TQM A SOFTWARE (LAWRENCE, 2001)

3.3 Estrutura

Para classificar a maturidade do processo de software de uma organização, o modelo CMM[®] define os 5 Níveis de Maturidade apresentados na introdução deste Capítulo. Para cada Nível de Maturidade, exceto o Nível 1, são estabelecidas Áreas Chave de Processo, ou KPA -

Key Process Area. E, para cada KPA, o modelo sugere um conjunto de Práticas Chave que estão organizadas por Características Comum.

Como se pode verificar no parágrafo anterior, o modelo CMM[®] estabelece as entidades: Níveis de Maturidade, KPA, Práticas Chave e Características Comuns. Uma forma gráfica do relacionamento destas entidades pode ser visualizada na Figura 3.4. Em seguida, cada uma destas entidades será abordada separadamente para exploração dos seus conceitos, segundo o modelo.

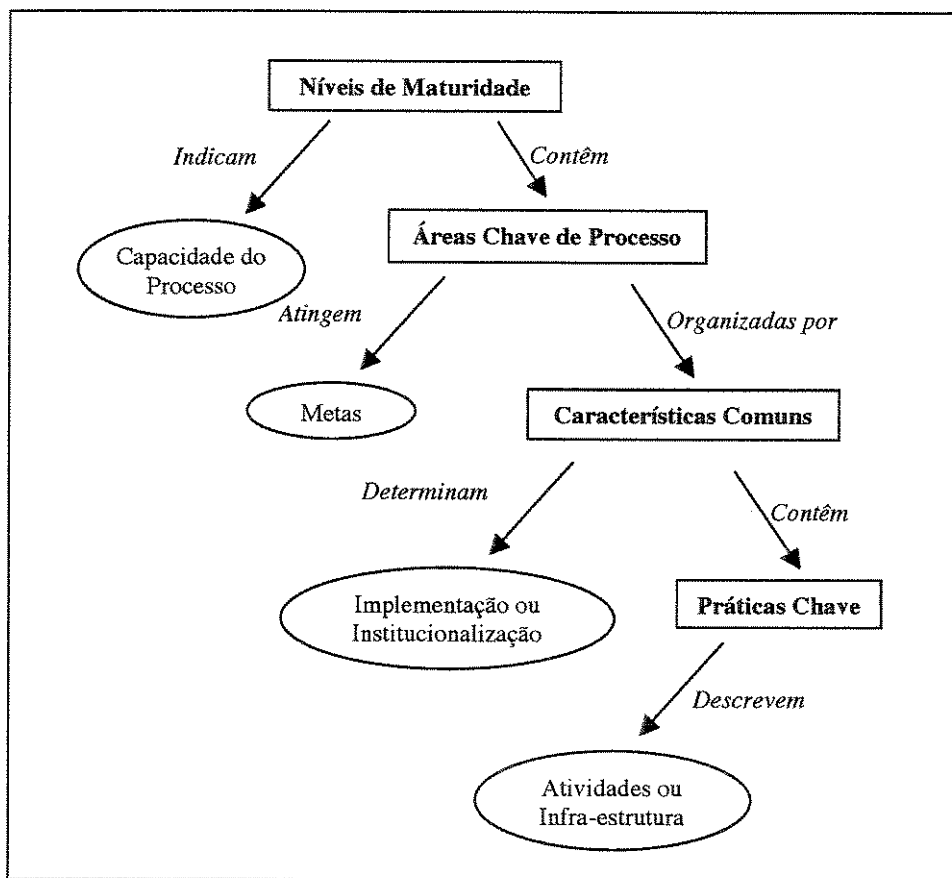


FIGURA 3.4 ESTRUTURA DO CMM[®] (CMM, 2000)

3.4 Níveis de Maturidade

Os Níveis de Maturidade são estágios do processo de software que podem ser comparados a platôs que a organização vai conquistando ao longo de uma jornada na busca pela maturidade

de seu processo de software. É importante notar que cada platô conquistado é a base para se atingir o platô seguinte.

Para determinar o Nível de Maturidade de uma organização é necessário uma avaliação reconhecida pelo SEI (SEI, 2001). Semestralmente, um documento público é disponibilizado pelo SEMA - *Software Engineering Measurement and Analysis* (SEMA, 2002) com dados sobre as avaliações CMM® realizadas no mundo todo e que foram reportadas ao SEI. De 1987 a junho de 2001, 1970 avaliações foram reportadas ao SEI, em 1506 organizações distribuídas por 43 países, incluindo EUA e Brasil. O nome das organizações não são divulgadas por uma questão de sigilo.

Um dos resultados interessantes destes documentos é a figura que apresenta a porcentagem de organizações classificadas em cada Nível de Maturidade em avaliações realizadas e reportadas ao SEI, no período de 1987 até o momento da publicação do documento. A Figura 3.5 é apresentada no último documento publicado, de agosto de 2001, e fornece uma idéia da evolução do processo de software das organizações ao redor do mundo. A base de organizações avaliadas em cada um dos períodos é citada na própria figura.

Segundo (PAULK, 1999), os Níveis de Maturidade possuem as seguintes características:

Nível 1: Inicial

O processo de software é *ad hoc* e, possivelmente, caótico. Mesmo quando alguns poucos processos estão definidos, o sucesso da organização continua dependente mesmo é dos esforços e heroísmo das pessoas.

Algumas vezes a organização desenvolve software que funciona, mas na maioria das vezes seu custo é muito além do previsto, assim como o prazo de entrega. A organização tipicamente não possui um processo estável para desenvolver e manter software e, são constantes as crises que precisa enfrentar. Durante as crises, os projetos tipicamente abandonam os procedimentos e vão direto para a codificação e testes.

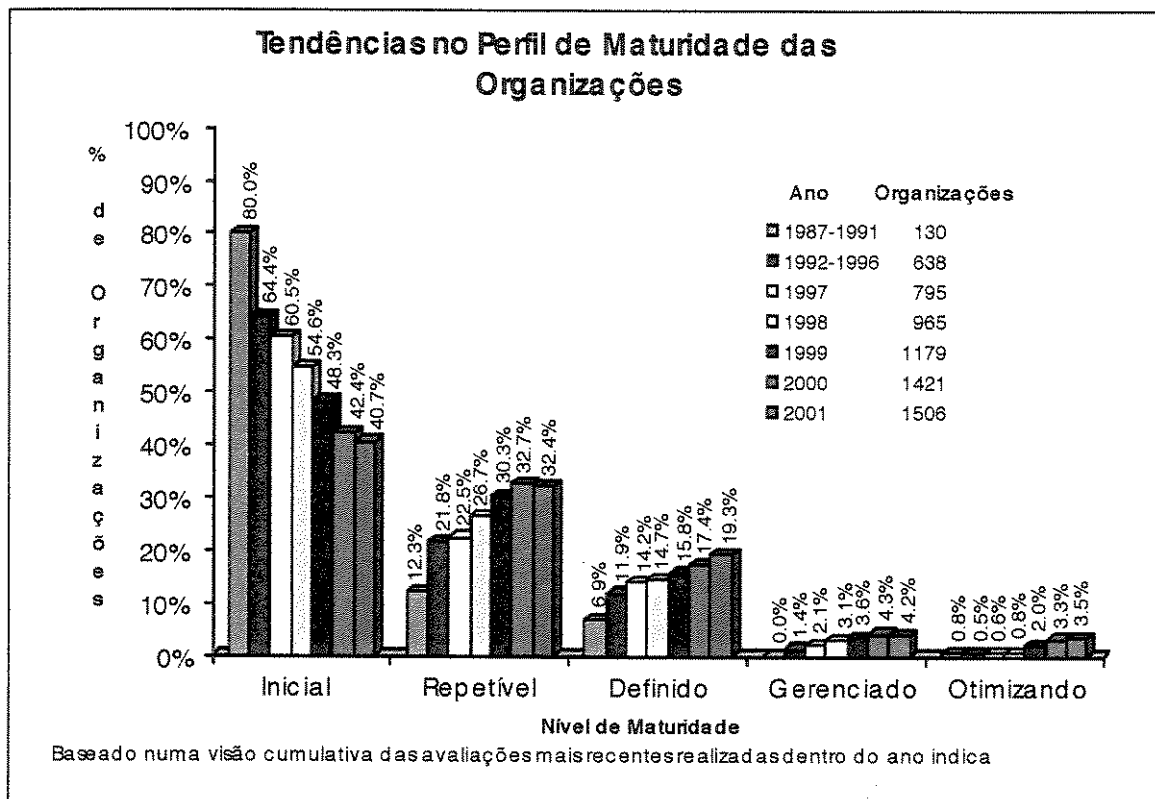


FIGURA 3.5 PORCENTAGEM DE ORGANIZAÇÕES POR NÍVEL DE MATURIDADE (SEMA, 2002)

Nível 2: Repetível

Os processos básicos de gerenciamentos estão estabelecidos para controlar custos, cronograma e funcionalidades. O planejamento do projeto baseia-se em resultados de projetos anteriores bem sucedidos e nos requisitos do projeto em questão.

Os problemas quanto ao não cumprimento de custos, cronograma e funcionalidades são identificados e trabalhados somente quando de sua ocorrência.

Os requisitos do software e os diversos produtos de trabalho do processo de software são identificados no planejamento e, os momentos da verificação da integridade de cada um deles são controlados durante a execução do projeto. Os projetos podem diferir entre si quanto à forma de trabalhar. O requisito para uma organização ser Nível 2 de Maturidade é que hajam políticas que

orientem os projetos no estabelecimento de gerenciamento de projetos apropriados às suas necessidades.

Nível 3: Definido

A organização já possui um processo de software padrão estabelecido e sendo utilizado efetivamente, por todos os projetos no desenvolvimento e manutenção de software. As atividades dos processos de gerenciamento e engenharia de software estão documentadas, padronizadas e incorporadas no processo de software.

O corpo técnico e gerencial é devidamente treinado para exercer suas funções através de um processo de treinamento definido para toda a organização. Há um grupo responsável pelas atividades do processo de software da organização que é o grupo do processo de engenharia de software, conhecido como SEPG - *Software Engineering Process Group*.

Características importantes de uma organização Nível 3 de Maturidade são: o processo de software está bem definido e, a gerência tem uma excelente visão do progresso técnico de todos os projetos em andamento.

Nível 4: Gerenciado

Medidas detalhadas são coletadas tanto do processo quanto da qualidade do produto de software. O processo e o produto de software são controlados quantitativamente em todos os projetos da organização. Tanto a produtividade quanto a qualidade de atividades importantes do processo de software são medidas e uma base de dados da organização é utilizada para armazenar e analisar estes dados.

Os projetos, numa organização Nível 4, mantêm o controle de seus produtos e processos trabalhando com as variações de desempenho para mantê-los dentro de limites aceitáveis. Neste Nível de maturidade é possível dizer que as organizações produzem produtos de software de qualidade com uma alta probabilidade de acerto.

Nível 5: Otimizando

Melhorias contínuas são realizadas no processo de software a partir de dados coletados do próprio processo e também de projetos pilotos realizados para incorporar novas idéias e novas tecnologias.

Uma organização Nível 5 tem a cultura, já incorporada, de identificar oportunidades de melhoria de forma pró-ativa com o objetivo de prevenir a ocorrência de defeitos. Para incluir uma nova tecnologia, ou realizar mudanças no processo de software padrão da organização, são realizadas análises de custo x benefícios a partir de dados coletados da eficiência do atual processo de software padrão da organização.

As equipes de projeto de software analisam os defeitos, que eventualmente ocorram, para determinar suas causas. Os processos são avaliados para prevenir novas ocorrências de tipos já conhecidos de defeitos e, as lições aprendidas são disseminadas para os outros projetos da organização.

A capacidade do processo de software de uma organização Nível 5 pode ser caracterizada como uma melhoria contínua. Estas organizações estão sempre procurando maneiras de melhorar ainda mais a capacidade de seus processos através da introdução de novas tecnologias ou da melhoria de seu processo de software.

3.5 Áreas Chave de Processo

As Áreas Chave de Processo, ou simplesmente KPA como são conhecidas, representam áreas da organização de software, identificadas pelo modelo CMM®, que possuem importância fundamental no estabelecimento de um processo de software de qualidade.

Cada Nível de Maturidade, exceto o Nível 1, possui um conjunto de KPA preestabelecido. Conforme visto na seção anterior, no Nível 1 de Maturidade, o processo de software da

organização é caótico e, portanto, não há como esperar que alguma área esteja bem estabelecida. É por este motivo que nenhuma KPA é abordada, pelo modelo, neste Nível de Maturidade.

O modelo CMM® define 18 KPA do Nível 2 ao Nível 5 de Maturidade. Não é objetivo do modelo descrever exaustivamente todas as áreas envolvidas no desenvolvimento e manutenção de software. Ele somente trabalha com as áreas que foram consideradas ‘chave’ para a obtenção de cada um dos níveis de maturidade. A Figura 3.6 apresenta as KPA definidas para cada Nível de Maturidade.

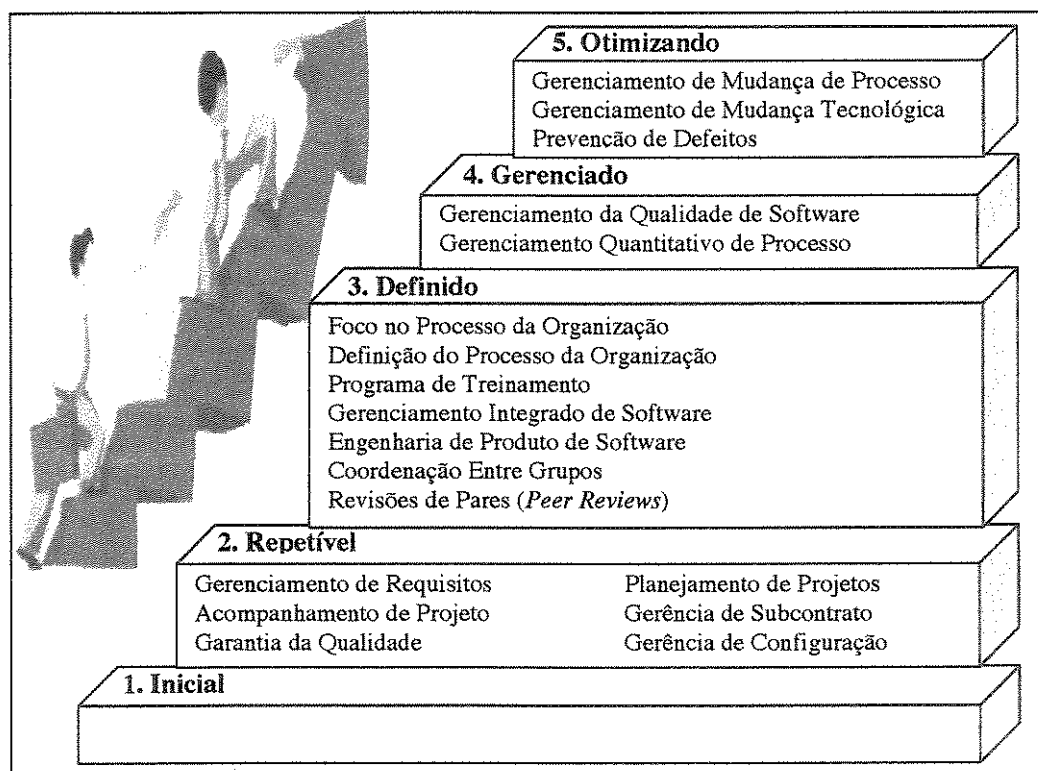


FIGURA 3.6 KPA POR NÍVEL DE MATURIDADE

Cada KPA identifica um conjunto de atividades correlatas que, ao serem executadas, atingem um conjunto de metas que foram estabelecidas pelo modelo CMM®. Uma KPA é considerada como tendo sido satisfeita, segundo o modelo, quando todas as suas metas foram atingidas.

Uma organização é classificada, através de uma avaliação, como sendo de um determinado Nível de Maturidade quando todas as metas das KPA, daquele Nível e dos Níveis anteriores, são cumpridas. Por exemplo, para uma organização ser considerada Nível 3 de Maturidade, ela deve cumprir as metas de todas as KPA do Nível 3 e do Nível 2.

3.6 Características Comum

Práticas Chave, segundo o CMM[®], são as ações que garantem que o processo está definido, documentado, entendido por todos os envolvidos e está sendo executado efetivamente. Em toda KPA, as Práticas Chave são agrupadas de acordo com o foco da ação proposta. O CMM[®] estabelece 5 focos, chamados Características Comum, que são:

Responsabilidades: Descreve as ações, que devem ser tomadas pela organização, para garantir que o processo está estabelecido e será utilizado. As Práticas Chave normalmente envolvem o desenvolvimento de uma política organizacional e o comprometimento da alta gerência.

Habilidades: Descreve as pré condições que devem existir no projeto, ou na organização, para implementar a competência necessária para a execução o processo de software. As Práticas Chave, que envolvem esta característica, normalmente são ações relacionadas com recursos, treinamentos, estrutura organizacional e ferramentas.

Atividades: Descreve as regras e procedimentos necessários para implementar a respectiva KPA. As Práticas Chave envolvem o estabelecimento de planos e procedimentos para realizar o trabalho, acompanhá-lo e tomar ações corretivas quando necessário.

Medições e Análises: Descreve a necessidade de medir o processo e analisar estes dados. As Práticas Chave incluem exemplos de medições que podem ser realizadas para verificar o estágio da atividade, e se a mesma é eficiente.

Verificações: Descreve os passos para garantir que as atividades estão sendo realizadas de acordo com o processo estabelecido. As Práticas Chave normalmente descrevem revisões e auditorias que devem ser realizadas pela gerência e garantia da qualidade de software.

Segundo (CMM, 2000), a execução das ações propostas pelas Práticas Chave indica o quanto as KPA estão estabelecidas, quão repetível são seus procedimentos e se suas práticas já fazem parte da rotina da organização. Ou seja, as Práticas Chave tratam não só de ações referentes a institucionalização, concentradas nas Características Comum: ‘Responsabilidades’, ‘Habilidades’, ‘Medições e Análise’ e, ‘Verificações’, mas também de ações relacionadas a implementação do processo, concentrada na Característica Comum ‘Atividade’. A Figura 3.7 exemplifica como as Características Comum se relacionam para proporcionar o cumprimento das metas das diversas KPA.

Este trabalho de dissertação se propõe a apresentar uma alternativa de integração das abordagens de qualidade para processo e produto de software, com o objetivo de maximizar a possibilidade da produção de um produto de software de qualidade. Para tanto, a análise de utilização da Norma NBR 13596 está concentrada nas Metas das KPA, nas práticas chave e produtos de trabalho das ‘Atividades’, ou seja, relacionadas à implementação das KPA. Isto porque, as ‘Atividades’ concentram ações diretamente relacionadas com o processo de desenvolvimento e manutenção de software e totalizam 150 das 316 Práticas Chave abordadas pelo CMM®, o que as tornam representativas.

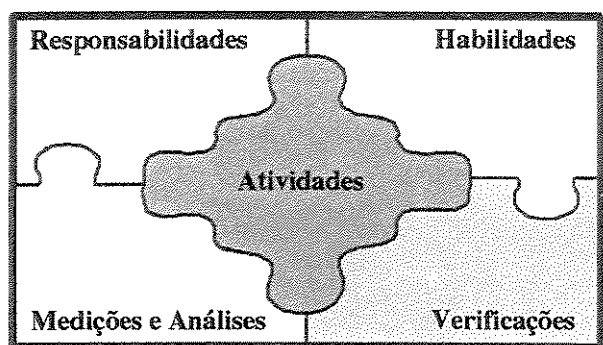


FIGURA 3.7 CARACTERÍSTICAS COMUM

Devido ao grande número de práticas chave das 'Atividades', elas são apresentadas no Anexo. No próximo capítulo será abordada a Norma NBR 13596, sobre as características de qualidade de um produto de software.

Capítulo 4

4 NBR 13596

4.1 Introdução

O trabalho de elaboração da Norma NBR 13596 envolveu em torno de duas dezenas de profissionais especializados de empresas, universidades e centros de pesquisa, que já aplicavam os conceitos em suas atividades profissionais antes mesmo do texto final ser aprovado como projeto de norma (NBR 13596, 1996).

Foi no período de preparação desta Norma que ocorreu o lançamento da SOFTEX 2000 - Programa Nacional de Software para Exportação, com o objetivo de transformar o Brasil em um país exportador de software. Para tanto, a indústria nacional de software deve estar preparada para criar e oferecer, ao mercado, produtos de software com níveis de qualidade internacional.

Um dos motivos que levou a Comissão de Estudo de Qualidade de Software a elaborar a Norma (NBR 13596, 1996) através da tradução do texto base da Norma ISO/IEC 9126 '*Information technology – Software product quality*' (ISO/IEC 9126, 1991), e não o de criar uma nova norma, é que a integração crescente da economia mundial está uniformizando os conceitos de qualidade, exigindo a utilização de textos normativos comuns a todos os países.

4.2 Relação entre NBR 13596 e TQM

Relembrando o objetivo do TQM, abordado na seção ‘Relação entre CMM® e TQM’ do Capítulo ‘SW-CMM®’, que é o de prover as necessidades dos clientes agora e no futuro (CMM, 2000) e; com a definição da Norma NBR 13596 sobre Qualidade de Software como sendo a “Totalidade das características de um produto de software, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas (NBR 13596, 1996)” é possível visualizar a relação existente entre o TQM e a NBR 13596.

Por um lado o CMM®, a partir dos conceitos do TQM, procura garantir a qualidade do insumo, ou entradas, e do processo de software; enquanto que por outro lado a NBR 13596 está preocupada em garantir que as necessidades dos clientes, em relação ao produto de software, sejam providas.

A Norma NBR 13596 define quais são as características que um produto de software deve ter, e fornece um modelo para ser utilizado numa avaliação de verificação da presença destas características. Isto significa que a partir de uma entidade de software, disponível a um usuário, é possível utilizar a Norma para verificar a qualidade desta entidade.

Neste capítulo, a Norma NBR 13596 será apresentada através de sua própria estrutura, ou seja, serão apresentadas suas diretrizes de uso; as características de qualidade de produto de software e; o modelo sugerido para avaliação.

4.3 Diretrizes para Uso da Norma NBR 13596

A Norma NBR 13596 pode ser aplicada nos seguintes momentos:

- Definição dos requisitos de qualidade de um produto de software;
- Avaliação da especificação de software para verificar se ele irá satisfazer aos requisitos de qualidade durante o desenvolvimento;

- Descrição de particularidades e atributos do software implementado como, por exemplo, em manuais de usuário;
- Avaliação do software desenvolvido, antes da entrega e;
- Avaliação do software desenvolvido, antes da aceitação.

A seguir são apresentadas as características de qualidade que um software deve ter, e que devem ser consideradas em cada um dos momentos citados acima.

4.4 Características de Qualidade de Software

As seis características de qualidade de software, segundo a Norma NBR 13596, estão transcritas na Tabela 4.1. Uma preocupação da Norma foi definir características com um mínimo de sobreposição de conceitos entre elas.

TABELA 4.1 CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE (NBR 13596, 1996)

CARACTERÍSTICAS	DEFINIÇÕES
Funcionalidade	Conjunto de atributos que evidencie a existência de um conjunto de funções e suas propriedades especificadas. As funções são as que satisfazem as necessidades explícitas ou implícitas.
Confiabilidade	Conjunto de atributos que evidencie a capacidade do software de manter seu nível de desempenho sob condições estabelecidas, durante um período de tempo definido.
Usabilidade	Conjunto de atributos que evidencie o esforço necessário para se poder utilizar o software; bem como, o julgamento individual desse uso por um conjunto explícito ou implícito de usuários. Por usuários entende-se usuários de software interativo, ou seja: operadores; usuário final e usuários indiretos, que estão sob influência ou dependência do uso do software.
Eficiência	Conjunto de atributos que evidencie o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas.
Manutenibilidade	Conjunto de atributos que evidencie o esforço necessário para fazer modificações especificadas no software.
Portabilidade	Conjunto de atributos que evidencie a capacidade do software de ser transferido de um ambiente para o outro.

Os atributos, mencionados na definição de ‘funcionalidade’, caracterizam ‘o que’ o software faz para satisfazer as necessidades implícitas e explícitas do usuário; enquanto que, os

atributos, mencionados nas demais características de qualidade, caracterizam principalmente ‘quando’ e ‘como’ o software faz para satisfazer estas necessidades.

Além de definir as características de qualidade de software e orientar quando devem ser utilizadas, a NBR 13596 se preocupou também em apresentar um modelo de qualidade de produto de software, que será o assunto da próxima seção.

4.5 Modelo de Qualidade de Produto de Software

O modelo de qualidade de produto de software da Norma NBR 13596 foi definido para apoiar o objetivo de aplicar as seis características de qualidade numa avaliação de produto de software. Numa avaliação, é realizada uma comparação do valor medido com um dado modelo pré estabelecido. A utilização do modelo apresentado pela NBR 13596 não é obrigatório; outros modelos podem ser utilizados, mas o importante é perceber a necessidade de se seguir um modelo numa avaliação. O modelo apresentado pela NBR 13596 é baseado na definição das características de qualidade de software através de subcaracterísticas. Na Tabela 4.2, são apresentadas, para cada característica, quais são as subcaracterísticas e suas definições.

Para realizar uma avaliação, não basta ter as características de qualidade de produto de software definidas através de suas subcaracterísticas, ainda é preciso estabelecer quais serão as métricas, os níveis de pontuação e os critérios de aceitação. Critério de Aceitação é a pré determinação dos níveis de pontuações considerados satisfatórios, ou não satisfatórios, para um atributo de uma entidade. De acordo com a Norma (ISO/IEC 14598-1, 1999), Métrica e Nível de Pontuação são definidos como:

- **Métrica:** Um método e uma ‘escala’ de ‘medição’. Por ‘escala’ entende-se: um conjunto de valores com propriedades definidas e; por ‘medição’: a determinação de um valor (que pode ser um número ou uma categoria) para um atributo de uma entidade.
- **Nível de Pontuação:** Pontuações de uma escala ordinal que é utilizada para categorizar uma escala de medição.

TABELA 4.2 SUBCARACTERÍSTICAS DE QUALIDADE DE SOFTWARE (TSUKUMO, 1997)

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	DESCRIÇÕES
Funcionalidade	Adequação	Presença de um conjunto de funções e sua apropriação para as tarefas especificadas.
	Acurácia	Geração de resultados ou efeitos corretos.
	Interoperabilidade	Capacidade de interagir com outros sistemas especificados.
	Conformidade	Estar de acordo com normas, convenções e regulamentações.
	Segurança de Acesso	Capacidade de evitar acesso não autorizado a programas e dados.
Confiabilidade	Maturidade	Frequência de falhas.
	Tolerância a falhas	Capacidade de manter o nível de desempenho em caso de falha ou violação nas interfaces.
	Recuperabilidade	Capacidade de restabelecer seu desempenho e restaurar dados após falha.
Usabilidade	Inteligibilidade	Atributos do software que evidenciam o esforço do usuário para reconhecer o conceito lógico e sua aplicabilidade.
	Apreensibilidade	Atributos do software que evidenciam o esforço do usuário para aprender sua aplicação.
	Operacionalidade	Atributos do software que evidenciam o esforço do usuário para sua operação e controle da sua operação.
Eficiência	Comportamento em relação ao tempo	Tempo de resposta, de processamento e velocidade na execução de funções.
	Comportamento em relação aos recursos	Quantidade de recursos utilizados.
Manutenibilidade	Analisabilidade	Esforço necessário para diagnosticar deficiência e causas de falhas.
	Modificabilidade	Esforço necessário para realizar modificações e remoção de defeitos.
	Estabilidade	Ausência de riscos de efeitos inesperados ocasionados por modificações.
	Testabilidade	Facilidade de ser testado.
Portabilidade	Adaptabilidade	Capacidade de ser adaptado a ambientes diferentes.
	Capacidade para ser instalado	Esforço necessário para a instalação.
	Conformidade	Consonância com padrões ou convenções de portabilidade.
	Capacidade para substituir	Capacidade e esforço necessário para substituir outro software.

A NBR 13596 lembra a necessidade do uso de métricas e níveis de pontuação, além de mostrar a relação existente entre eles, conforme Figura 4.1. No entanto, a Norma não os define pois, estes podem variar de acordo com a organização e/ou de acordo com a aplicação.

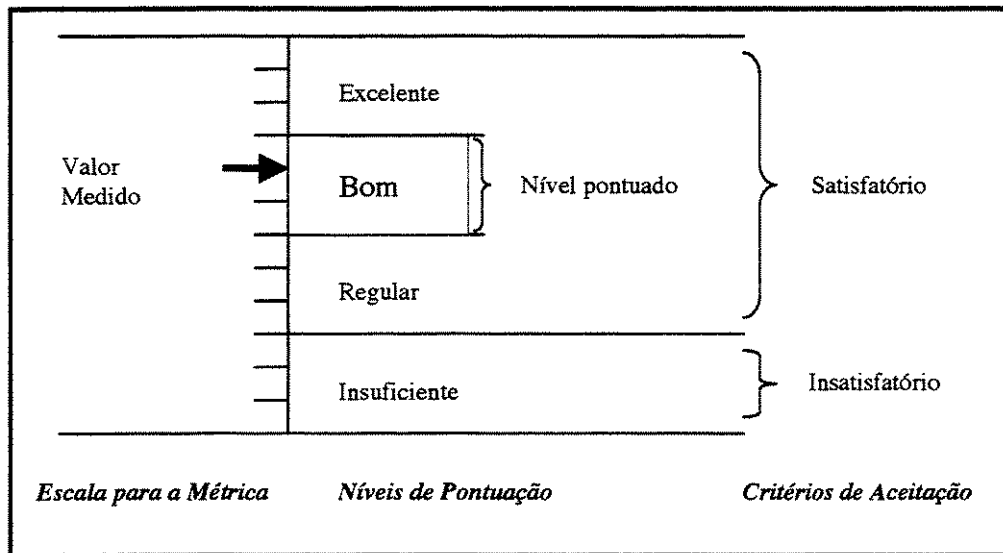


FIGURA 4.1 VALOR MEDIDO E NÍVEL DE PONTUAÇÃO (NBR 13596, 1996)

Outros fatores a serem considerados numa avaliação são: o 'grau de importância' de cada característica de qualidade e; sob qual 'ponto de vista' o produto de software está sendo analisado. Estes fatores podem ser considerados bastante subjetivos mas é possível, numa avaliação comparativa de diferentes aplicações de software, ponderar as características de qualidade através de um sistema de pesos proporcionais a sua importância na área de aplicação do software.

O 'grau de importância' de cada característica de qualidade varia de software para software. Alguns exemplos são: a 'Confiabilidade' é mais importante para software de um sistema de missão crítica; a 'Eficiência' é mais importante para software de sistema em tempo real e; a 'Usabilidade' é mais importante para software interativo.

A Norma aponta três ‘pontos de vista’ a serem considerados numa avaliação da qualidade de software: do usuário; da equipe de desenvolvimento e; do gerente. Cada um deles tem foco de interesse próprio:

- O usuário está interessado no uso do software e não nos seus aspectos internos. Ele quer saber se as funções requeridas estão disponíveis no software, quão confiável, eficiente e fácil de usar é o software.
- A equipe de desenvolvimento está interessada tanto na qualidade dos produtos intermediários como na qualidade do produto final. O processo de desenvolvimento requer que o usuário e a equipe de desenvolvimento utilizem as mesmas características de qualidade de software, uma vez que elas se aplicam tanto para os requisitos como para a aceitação.
- A visão do gerente é comercial. Ele está interessado na qualidade de forma geral e não em características específicas de qualidade. Sua responsabilidade é otimizar a qualidade dentro das limitações de custo, recursos humanos e prazos.

Resumidamente, a Norma NBR 13596 apresenta um modelo para avaliação da qualidade de produto de software, baseado no desmembramento das 6 características de qualidade em 18 subcaracterísticas e; além disso, alerta a necessidade de se definir, caso a caso, as métricas, níveis de pontuação, critérios de aceitação, grau de importância de cada característica e ponto de vista a ser considerado na avaliação. Esta abertura permite a aplicação deste modelo nas diversas organizações de software, em diferentes necessidades.

A versatilidade de suprir diferentes necessidades é que permite a aplicação da NBR 13596 em diversos momentos do processo de software de uma organização. A seguir será desenvolvida a proposta desta dissertação, apontando alguns destes momentos de aplicação.

Capítulo 5

5 Integrando CMM® e NBR 13596

5.1 Introdução

Neste Capítulo será desenvolvida a proposta desta dissertação que é apresentar uma alternativa de integração das abordagens de qualidade de software: processo e produto. Serão analisadas e apresentadas diversas instâncias no processo de software de uma organização, que faz uso da versão 1.1 do modelo CMM®, onde pode ser considerada a utilização da Norma brasileira NBR 13596.

As possibilidades de aplicação da Norma NBR 13596 foram baseadas nos ‘conceitos’, ‘diretrizes de uso’, ‘características’ e ‘subcaracterísticas’ de qualidade de produto de software, estabelecidos por esta Norma. A estratégia adotada foi analisar e apontar oportunidades de aplicação da NBR 13596 nas diversas ‘Metas’, ‘Práticas Chave’ e ‘Produtos de Trabalho’ das ‘Atividades’ das 18 KPA definidas pelo modelo de processo CMM®.

As ‘Atividades’, sugeridas pelo CMM®, são de grande importância na análise desta dissertação porque, de acordo com (CMM, 2000), descrevem ‘o que’ deve ser feito para se estabelecer a capacidade do processo de software. Conforme Capítulo 3, ‘capacidade do processo de software’ define os limites de resultados esperados quando se segue o processo de software em uso.

Ao introduzir a Norma NBR 13596 nas ‘práticas chave’ das ‘Atividades’, pretende-se expandir seu objetivo para que possam definir também ‘o que’ deve ser feito para estabelecer a ‘capacidade do produto de software’. O termo ‘capacidade de produto’, extrapolando a definição de ‘capacidade de processo’, define os limites dos resultados esperados quando se executa o produto de software. A Tabela 5.1 resume estas considerações.

TABELA 5.1 OBJETIVO DAS ATIVIDADES ATRAVÉS DAS VISÕES CMM® E CMM®+NBR13596

	CMM®	CMM® + NBR 13596
‘O que’ deve ser feito para estabelecer a ...	Capacidade do Processo de Software	Capacidade do Processo e do Produto de Software
‘Foco’ na previsibilidade dos resultados do ...	Processo	Processo e Produto.

As seções seguintes apresentam, por Nível de Maturidade, as análises realizadas em cada uma das KPA, quanto a aplicação da Norma NBR 13596. Desta forma, organizações que se encontram em qualquer um dos Níveis de Maturidade do CMM®, ou seguem modelos de processo semelhantes a este, podem fazer uso destas sugestões e verificar os benefícios de se focar na qualidade de ‘software’, ou seja, processo e produto simultaneamente. O conteúdo das seções está organizado da seguinte forma:

- Apresentação do foco principal do Nível de Maturidade, de acordo com o modelo CMM®;
- Apresentação, por KPA, das considerações de aplicação da NBR 13596 com a seguinte estrutura:
 - Propósito da KPA, definido pelo CMM®;
 - Metas da KPA, definidas pelo CMM® e;
 - Análise realizada sobre o uso da Norma NBR 13596 em instâncias do modelo CMM®.

Obs.: As referências de instâncias podem ser:

- ‘Metas’ ou ‘Produtos de Trabalho’ descritos na própria seção;
- Práticas Chave de ‘Atividades’, descritas no Anexo.

5.2 Análise do Nível 2: Repetível

O foco, atribuído pelo CMM®, às organizações de Nível 2, está no gerenciamento de projetos de software. Como resultado deste gerenciamento é possível repetir o mesmo desempenho na execução de projetos, com tarefas semelhantes a projetos realizados no passado. As expectativas da organização estão expressas em políticas organizacionais.

Sem perder este foco de vista, que é no projeto, e trazendo a preocupação quanto a qualidade de produto no mesmo patamar da preocupação, é possível fazer algumas considerações de uso da Norma NBR 13596 em KPA de Nível 2 de Maturidade contribuindo para que a qualidade dos produtos desenvolvidos em projetos realizados no passado possam ser repetidos em novos projetos. Abaixo, serão abordadas as considerações sugeridas para cada KPA.

KPA: Gerenciamento de Requisitos

Como propósito da KPA de Gerenciamento de Requisitos, o CMM® coloca a necessidade de se estabelecer um entendimento comum, entre o cliente e a equipe do projeto de software, sobre os requisitos do cliente a serem implementados no software. Para tanto duas metas devem ser atingidas:

- M1. Os requisitos do cliente devem ser documentados e gerenciados e;
- M2. Os planos, produtos e atividades devem ser consistentes com os requisitos do cliente.

Cliente, segundo o CMM®, pode ser interno ou externo, ou seja, a engenharia de sistemas da organização, marketing e o usuário. Cada um deles tem seus próprios requisitos que devem ser alocados ao software. Para atingir as metas e cumprir o propósito do Gerenciamento de Requisitos é preciso fazer a especificação de requisitos do software que é um documento que descreve o comportamento do software esperado pelo cliente.

Os requisitos podem ser subdivididos em técnicos e não técnicos. Os requisitos não técnicos são aqueles que afetam e determinam as atividades do projeto de software, como por

exemplo: produtos a serem entregues; datas negociadas e outros fatores de condições; acordos e; termos contratuais. Já os requisitos técnicos são relacionados ao software, como por exemplo, as funções a serem executadas, o desempenho esperado do software, a forma de interface com o usuário, o suporte ao usuário, entre outros.

O usuário é quem fornece os principais requisitos técnicos para a especificação de requisitos do software. E, ao executar o levantamento destes requisitos junto ao cliente, a sugestão desta dissertação é que as características de qualidade de produto de software sejam abrangidas. As características de qualidade de um produto de software, segundo a NBR 13596 são: 'Funcionalidade', 'Confiabilidade', 'Usabilidade', 'Eficiência', 'Manutenibilidade' e 'Portabilidade'. Para facilitar o entendimento do real significado destas características, e sua abrangência por parte do cliente, o trabalho de levantamento de requisitos pode ser realizado com o apoio de uma lista de verificação, construído a partir das características e subcaracterísticas do modelo de qualidade da NBR 13596.

Por exemplo, para o levantamento de requisitos relacionados à característica 'Funcionalidade', o modelo de qualidade da NBR 13596 define como uma das subcaracterísticas a 'Segurança de Acesso'. Perguntas formuladas numa lista de verificação, para cobrir esta subcaracterísticas, podem ser, entre outras que se façam necessárias: 'O software precisa estar preparado para evitar o acesso de pessoas não autorizadas a determinadas dados ou funções?', 'Quais são os dados e funções que devem ser protegidos?', 'Quais são as pessoas que podem ter acesso a estes dados e funções?', 'Quão rígido o software precisa ser no bloqueio ao acesso de pessoas não autorizadas?'.

Não se deve esquecer que o cliente é tanto interno como externo e, as perguntas a serem formuladas devem estar de acordo com o perfil e necessidades do cliente a ser entrevistado numa sessão de levantamento de requisitos. É evidente que para se obter uma especificação de requisitos de software de qualidade deve ser estabelecida uma comunicação ativa entre cliente e desenvolvedores.

Com estas atitudes, as características de qualidade do produto de software, que poderiam ser esquecidas, tanto pelo desenvolvedor quanto pelo cliente, são lembradas e estarão registradas no documento de ‘Especificação dos Requisitos de Software’, o que é fundamental pois este servirá como suporte às atividades de gerenciamento do projeto, desenvolvimento, teste do software e aceitação do produto pelo cliente.

As três atividades, propostas pelo CMM[®], são ações posteriores à ‘Especificação de Requisitos de Software’ mas totalmente dependentes deste documento pois: na Atividade n° 1, o grupo de engenharia de software revê os requisitos a serem alocados antes de incorporá-los ao projeto de software; na Atividade n° 2, o grupo de engenharia de software usa os requisitos alocados como base para os planos de software, produtos de trabalho e atividades e; na Atividade n° 3, as mudanças nos requisitos alocados devem ser revisadas e incorporadas no projeto de software.

Portanto, ao considerar o modelo da Norma NBR 13596 na elaboração do documento ‘Especificação de Requisitos de Software’, necessariamente as características de qualidade estarão sendo consideradas pelo grupo de engenharia de software em suas atividades de revisão dos requisitos a serem alocados, ou que sofreram mudanças, além de servirem como base para os planos de software, produtos de trabalho e atividades do projeto.

O propósito, de acordo com o CMM[®], da área de Gerenciamento de Requisitos, que é o de estabelecer um entendimento comum, entre o cliente e a equipe do projeto de software, sobre os requisitos do cliente a serem implementados no software, continua sendo cumprido. Mas agora, tendo estabelecido, e documentado, também as características de qualidade como requisitos do cliente a serem implementados no produto de software.

KPA: Planejamento de Projetos de Software

Segundo o modelo CMM[®], o estabelecimento de planos razoáveis para desenvolver software e para gerenciar projetos de software é o propósito desta KPA. As metas a serem atingidas de forma a cumprir este propósito são:

- M1. Ter, de forma documentada, as estimativas necessárias para o gerenciamento do projeto de software;
- M2. Ter, de forma documentada, as atividades do projeto de software e os compromissos necessários e;
- M3. As pessoas e grupos envolvidos no projeto de software estão comprometidos com os compromissos do projeto de software.

O objetivo de qualquer 'plano' é prover uma base para iniciar e gerenciar um trabalho. Quando se fala em projetos de software é possível encontrar diferentes nomes para o plano elaborado para gerenciá-los: 'Plano de Software', que será o utilizado nesta dissertação; 'Plano de Gerenciamento de Projeto de Software'; 'Plano de Projeto de Software'; 'Plano de Gerenciamento de Projeto' e; 'Plano de Gerenciamento da Engenharia de Software', entre outros. O importante não é o nome dado, mas sim o que deve ser especificado nele.

O 'Plano de Software' deve ser documentado e conter informações tais como: o ciclo de vida definido para o software; a lista de produtos a serem desenvolvidos; o cronograma de execução; os riscos do projeto; estimativas de esforço, custo, recursos e outras informações relevantes ao desenvolvimento do software e ao gerenciamento do projeto.

Quando se fala em 'Plano de Software', está sendo referenciado, na verdade, um conjunto de planos que podem estar combinados em um único documento, ou podem estar separados em um conjunto de documentos, tais como: 'Plano de Desenvolvimento de Software'; 'Plano de Garantia da Qualidade'; 'Plano de Gerência de Configuração'; 'Plano de Gerência de Riscos'; 'Plano de Testes de Software'; e 'Plano de Treinamento'.

Para se cumprir planos é necessário haver o compromisso dos envolvidos. Compromisso é um pacto que é livremente assumido, visível e esperado de ser mantido pelas pessoas envolvidas. Desta forma, para que as pessoas estejam comprometidas com o projeto de software é fundamental que o 'Plano de Software' seja o mais realista possível.

É importante ressaltar que todo plano deve ser baseado em estimativas. Para se fazer estimativas de tamanho, esforço, custo, cronograma e recursos computacionais necessários ao projeto devem ser considerados os dados históricos, quando existirem.

Como exemplo de produtos de trabalho o CMM® cita: software operacional e de suporte; produtos de trabalho 'a serem' entregues ao cliente e produtos que 'não serão' entregues; produtos de trabalho que 'são' software e que 'não são' software, como por exemplo: a documentação e os produtos de software intermediários. Algumas atividades de projeto de software, exemplificadas pelo CMM®, são as atividades para desenvolvimento, verificação e validação dos produtos de trabalho.

Portanto, atividades de verificação e validação de um software, assim como estimativas de esforços para executá-las, devem estar descritas e documentadas no 'Plano de Software' tanto para produtos de trabalho intermediários quanto para o produto final. O que este trabalho de dissertação recomenda é que na descrição destas atividades no 'Plano de Software' seja referenciado o uso do modelo de qualidade da Norma NBR 13596 para uma avaliação da implementação das características de qualidade de produto de software, que foram documentadas na 'Especificação de Requisitos de Software', mencionada na KPA de Gerenciamento de Requisitos. Ao se estimar o esforço necessário para a execução destas avaliações é possível racionalizar sobre quais os momentos e em quais produtos de trabalho estas atividades devem ser executadas.

O estabelecimento de planos razoáveis para gerenciar projetos de software e desenvolver software, que apresentem características de qualidade implementadas, passa a ser o novo propósito desta KPA, sem causar prejuízo aos objetivos do CMM®.

KPA: Acompanhamento de Projetos de Software

O propósito é prover visibilidade adequada do progresso atual do projeto de software, de forma a possibilitar que a gerência tome medidas efetivas quando o desempenho se desvia

significativamente do planejado. Três metas devem ser atingidas para que este propósito seja cumprido:

- M1. O desempenho e os resultados atuais do projeto são acompanhados e revisados em relação ao que consta nos planos;
- M2. Ações corretivas são realizadas e gerenciadas quando o desempenho e resultados reais desviam significativamente dos planos e;
- M3. Alterações necessárias nos comprometimentos têm a concordância dos grupos e pessoas envolvidas.

As atividades desta KPA reforçam a necessidade de acompanhamento do progresso do projeto de software, tendo como referência os planos com suas especificações, tais como: tamanho de produtos de trabalho, esforços e custos do projeto, recursos computacionais; cronograma e riscos. Especificamente na Atividade nº 9, é reforçada a necessidade de se acompanhar as atividades técnicas da engenharia de software e tomar ações corretivas, quando necessárias. Uma orientação desta dissertação, que é sugestão do próprio CMM® na Atividade nº 9, é que qualquer problema que tenha sido identificado nos produtos de trabalho do software deve ser reportado e documentado. Esta ação pode contribuir na formação de dados históricos, que devem ser considerados em novos projetos de software.

De acordo com a Metas de nº 1 e de nº 2 e a Atividade nº 9 do CMM®, as atividades de verificação e validação dos produtos de software intermediários e final, que estão descritas e documentadas no ‘Plano de Software’, devem ser acompanhadas e tomadas as ações corretivas necessárias, caso os resultados obtidos desviem do que foi planejado.

Seguindo a recomendação feita nesta dissertação para a KPA de Planejamento de Projetos de Software, as atividades de verificação e validação dos produtos de software intermediários e final devem incluir avaliações, utilizando o modelo de qualidade da Norma NBR 13596, para que a implementação das características de qualidade de produto de software sejam verificadas.

Os esforços necessários na execução destas avaliações também devem ser acompanhados e, ações corretivas são tomadas caso estes sejam significativamente diferentes do planejado. É recomendável que estes ajustes não sejam apenas em relação ao dimensionamento dos esforços necessários, mas também quanto aos momentos em que as avaliações devem ocorrer e em quais produtos de trabalho.

Os resultados das avaliações em produtos de software intermediários, durante o processo de software, irá permitir a tomada de ações que se façam necessárias para orientar a implementação de características de qualidade que estejam sendo negligenciadas. Este esforço, durante o processo de software, tem por objetivo diminuir o retrabalho, que é necessário quando falhas na implementação são detectadas apenas quando o produto de software está sendo avaliado para o aceite do cliente, ou quando o produto já está em uso.

A visibilidade do progresso atual do projeto é acrescida de informações do progresso da implementação das características de qualidade nos produtos de software intermediários. Com informações do progresso do projeto e do produto, a gerência poderá tomar medidas efetivas, em relação aos dois aspectos, quando estes se desviam significativamente do planejado.

KPA: Gerenciamento de Subcontrato de Software

Subcontrato de software ocorre quando o desenvolvimento de um software, ou parte dele, é realizado por outra organização, através de um contrato. Nesta situação, a organização contratante precisa realizar uma série de atividades que vai desde a seleção do fornecedor até o recebimento do software adquirido. A KPA responsável por este conjunto de atividades é o Gerenciamento de Subcontrato de Software.

O propósito formal do CMM® para esta KPA é selecionar fornecedores qualificados de software e gerenciá-los eficazmente. As metas a serem atingidas para que este propósito seja cumprido são:

- M1. O contratante seleciona fornecedores qualificados de software;

- M2. Os compromettimentos são estabelecidos entre contratante e fornecedor;
- M3. Contratante e fornecedor mantêm uma comunicação ativa e;
- M4. O contratante acompanha os resultados e desempenho do fornecedor, baseando-se no que foi acordado entre ambos.

Na seleção e gerenciamento de subcontratos de software, o contratante deve realizar diversas atividades adicionais ao gerenciamento de um projeto comum. Ele deverá especificar, entre outras coisas, os critérios para seleção e avaliação de fornecedores; o trabalho a ser desenvolvido; os produtos a serem entregues; critérios de aceitação dos produtos de software; padrões e procedimentos a serem seguidos pelo fornecedor.

Ao selecionar o fornecedor, sua qualificação pode depender de fatores tais como a capacidade do processo de software; a experiência em engenharia de software e o domínio de conhecimento da organização na aplicação a ser desenvolvida.

O gerenciamento do subcontrato de software é baseado no documento de contrato firmado e no documento 'Plano de Software', que é elaborado pelo fornecedor e aprovado pelo contratante. Ao gerenciar o subcontrato, o contratante deve assegurar que o fornecedor está seguindo o 'Plano de Software' aprovado, os padrões e os procedimentos estabelecidos.

Conforme visto na KPA de Planejamento de Projeto de Software, atividades de verificação e validação de um software, assim como estimativas de esforços para executá-las, devem estar descritas e documentadas no 'Plano de Software' tanto para produtos de trabalho intermediários quanto para o produto final.

Assim como sugerido na KPA de Planejamento de Projeto, cabe aqui recomendar que na descrição destas atividades, no 'Plano de Software', seja referenciado o uso do modelo de qualidade da Norma NBR 13596 para a realização de avaliações das características de qualidade do produto de software, que devem estar documentadas na 'Especificação de Requisitos de Software', conforme sugestão apresentada na KPA de Gerenciamento de Requisitos.

Além disso, a Atividade nº 9 coloca a necessidade do contratante realizar revisões formais, em momentos significativos do projeto, para que sejam verificados os resultados atingidos e o trabalho realizado pela engenharia de software do fornecedor e, estas revisões devem seguir um procedimento documentado, que inclua a Norma NBR 13596 como ferramenta de avaliação. Na execução desta Atividade, o modelo de qualidade da NBR 13596 pode ser usado em avaliações de produtos de software intermediários, uma vez que foi referenciado no 'Plano de Software' e no procedimento documentado.

Outra atividade fundamental desta KPA, é a de nº 12 sobre a condução de testes de aceitação como parte do processo de entrega do produto de software. Inicialmente, a Norma NBR 13596 poderá ser usada no estabelecimento dos critérios de aceitação, onde as características e subcaracterísticas de qualidade de produto de software deverão ser estabelecidas como requisitos. A avaliação do produto de software, utilizando a Norma NBR 13596, como teste de aceitação a ser realizado pelo contratante, deve estar registrada no procedimento documentado a ser seguido na execução da Atividade nº 12. E finalmente, o contratante irá utilizar o modelo de qualidade desta Norma durante os testes de aceitação do produto de software, como mais um parâmetro de aceite, ou não, do produto.

Com estas sugestões, a KPA de Gerenciamento de Subcontrato de Software será capaz de selecionar fornecedores qualificados e gerenciar eficazmente o processo e a qualidade do produto de software a ser entregue.

KPA: Garantia da Qualidade de Software

O propósito da GQS - Garantia da Qualidade de Software é prover um gerenciamento tanto do processo, que está sendo usado no projeto de software, como também dos produtos de software que estão sendo desenvolvidos. As metas a serem atingidas são:

- M1. As atividades da GQS são planejadas;
- M2. A conformidade dos produtos de software e processos quanto aos padrões, procedimentos e requisitos aplicáveis é verificada objetivamente;

- M3. Os grupos e pessoas envolvidas são informados das atividades e resultados da GQS e;
- M4. Questões de não conformidade que não podem ser resolvidas no projeto são tratadas com as gerências superiores.

As metas nº 1 e nº 2, desta KPA, atentam para o fato de que deve ser verificado objetivamente a aderência dos produtos de software a padrões e requisitos e, que toda atividade da Garantia da Qualidade de Software deve ser planejada. Portanto deve-se planejar e, efetivamente, realizar atividades de verificação da aderência dos produtos de software com padrões e requisitos. Conforme mencionado nas 'Diretrizes para o Uso das Características de Qualidade', da Norma NBR 13596, ela pode ser usada na avaliação de especificação de software para verificar se ele irá satisfazer aos requisitos de qualidade durante o desenvolvimento.

O valor agregado pela GQS é que ela provê uma visão independente do que está ocorrendo com as atividades do projeto, com os processos e com os produtos. Ela atua como 'olhos e ouvidos' do gerenciamento e é por isso que normalmente este é um grupo independente.

A Atividade nº 2 diz que as atividades do grupo de GQS são realizadas de acordo com o 'Plano de GQS' pré estabelecido. Este Plano deve garantir a participação do grupo de GQS no estabelecimento do 'Plano de Software', dos padrões e procedimentos que sejam apropriados para uso no projeto. A participação do grupo de GQS deve se dar desde o início do projeto.

Assim, uma contribuição do grupo de GQS, que esta dissertação recomenda, é o de garantir a indicação da Norma NBR 13596 como modelo padrão de qualidade de produto de software, a ser seguido pelo projeto; além de garantir a inclusão no 'Plano de Software' de atividades relacionadas à avaliações de qualidade de produtos de software intermediários e final.

Ainda na descrição da Atividade nº 2, o CMM® coloca que o grupo de GQS deve realizar avaliações tanto no software quanto nos demais produtos derivados, como é o caso de sua documentação que inclui o Manual do Usuário. Desta forma, avaliações a serem realizadas pelo grupo de GQS, tanto de software como de documentações do usuário, podem seguir o modelo de qualidade sugerido pela Norma NBR 13596.

E por fim, mas não menos importante, a Atividade nº 5 relata a necessidade do grupo de GQS realizar, antes da entrega ao cliente, auditorias nos produtos para verificar sua aderência ao que foi especificado. Esta atividade vai de encontro com a diretriz de uso na NBR 13596 que diz ser recomendado seu uso na avaliação do software, antes da entrega.

A visibilidade fornecida pelo grupo de GQS, com estas sugestões, enfatiza a necessidade de acompanhamento da implementação das características de qualidade no produto de software em desenvolvimento.

KPA: Gerenciamento de Configuração de Software

O propósito do GCS - Gerenciamento de Configuração de Software é estabelecer e manter a integridade de determinados produtos do projeto, durante o ciclo de vida do software. Suas metas são:

- M1. As atividade de GCS são planejadas;
- M2. Os produtos de trabalho a serem colocados sob gerência de configuração, chamados *baselines*, são identificados, controlados e disponibilizados;
- M3. As mudanças realizadas nas *baselines* são controladas e;
- M4. Grupos e pessoas do projeto de software envolvidas com as *baselines* são informados de seus estados e conteúdos.

A GCS coordena o acesso aos produtos de trabalho do software e as mudanças nestes produtos. Como fica evidenciado nas metas acima, o GCS atua em produtos de trabalho identificados como *baselines*. *Baselines* são produtos ou especificações que: devem ser formalmente revisados; os grupos e pessoas, que atuam no projeto, devem estar cientes e de acordo com as mudanças efetuadas; servem de base para trabalhos futuros e; só podem ser mudados através de um procedimento formal de controle de mudanças.

Alguns produtos de trabalho não necessitam da formalidade do gerenciamento de configuração, mas precisam ter alguma forma de controle de versão e de mudanças. Nas

atividades da KPA de Gerenciamento de Configuração de Software encontramos o termo ‘gerenciado e controlado’ que significa justamente esta forma menos formal de controle.

Seja através de um controle menos formal ou através do cumprimento da Meta 2, esta KPA contribui para que a característica de qualidade de software ‘Manutenibilidade’, descrita na Norma NBR 13596, esteja presente nos produtos de trabalho a serem entregues. Como visto no Capítulo 4, esta característica contribui para minimizar o esforço necessário na realização de modificações e remoção de defeitos em um produto de software.

Num processo de software, onde o propósito da KPA de GCS foi cumprido, tem-se garantido no mínimo um controle de versão do produto desenvolvido. O controle de versão dos produtos de trabalho de um dado processo de software é fundamental para facilitar a realização de modificações, ou remoção de defeitos, num produto de software já entregue ao cliente.

5.3 Análise do Nível 3: Definido

No Nível 3, a ênfase está na organização. As melhores práticas dos projetos são disseminadas para a organização como um todo, e os processos são desenvolvidos de forma apropriada. A organização prove suporte aos projetos estabelecendo treinamentos, processos e medições.

Uma organização de software Nível 3 CMM[®] possui suas atividades de gerenciamento e de engenharia documentadas, padronizadas e integradas. Todos os projetos usam uma versão apropriada e documentada do processo de software da organização.

Preservando o foco existente neste Nível, que resumidamente está no estabelecimento de um processo de software padrão para a organização como um todo, é possível fazer algumas considerações de uso da Norma NBR 13596. Estas considerações têm por objetivo maximizar a possibilidade de que o processo padrão estabelecido gere produtos de software de qualidade.

A seguir, as considerações sugeridas estão detalhadas por KPA.

KPA: Foco no Processo da Organização

Esta KPA tem por propósito estabelecer uma responsabilidade organizacional em atividades que melhoram a capacidade do processo de software da organização. As metas a serem atingidas para o cumprimento deste propósito são:

- M1. O desenvolvimento do processo de software da organização e atividades de melhoria são coordenadas para a organização como um todo;
- M2. As melhores práticas e necessidades de melhoria, detectadas nos processos de software em uso, são identificadas a partir do processo padrão;
- M3. O desenvolvimento do processo de software da organização e as atividades de melhoria são planejadas.

As atividades necessárias para atingir as metas acima envolvem desenvolver e manter um entendimento dos processos de software utilizados nos projetos da organização, além de coordenar atividades de avaliação, desenvolvimento, manutenção e melhoria destes processos. O grupo de pessoas responsável pelas atividades que irão prover o foco no processo da organização normalmente recebe a sigla SEPG, proveniente do nome em inglês *Software Engineering Process Group*.

Ao analisar esta KPA de forma genérica, a tendência é não encontrar aplicação prática da Norma NBR 13596. Mas procurando compreender as Práticas Chave das Atividades desta KPA, é possível verificar que a Atividade nº 5 descreve a necessidade de monitorar, avaliar e, quando apropriado, transferir para todas as outras áreas da organização, novos processos, métodos e ferramentas que estão em uso na organização de forma restrita ou experimental.

Como ferramenta, por exemplo, pode-se ter um software. Uma avaliação de qualidade pode ser efetuada nesta ferramenta para averiguar se ela possui as características de qualidade descritas na Norma NBR 13596. Isto porque uma organização, que trabalha para garantir a qualidade do processo e dos produtos de software que desenvolve e mantém, deve também se preocupar e garantir que as ferramentas, utilizadas nos processos da organização, sejam de qualidade.

Esta ação não só preserva o propósito desta KPA como reforça a responsabilidade organizacional nas atividades que melhoram a capacidade do processo de software da organização considerando, inclusive, a qualidade das ferramentas de software utilizadas neste processo.

KPA: Definição do Processo da Organização

O propósito desta KPA é o desenvolvimento e a manutenção de um conjunto de recursos, a serem utilizados no processo de software, que irá melhorar o desempenho do processo e prover a base cumulativa para benefícios a longo prazo. A diferença entre as KPA Foco no Processo da Organização e Definição do Processo da Organização é que a primeira está centrada em 'Quem' será o responsável por atividades que melhoram a capacidade do processo de software da organização, enquanto que a segunda KPA se preocupa com 'O Que' deve ser feito para isto.

As metas a serem atingidas pela KPA Definição do Processo da Organização são:

- M1. Um processo de software padrão para a organização é desenvolvido e mantido;
- M2. Informações relacionadas ao uso de padrões e ao processo de software da organização são coletadas, revisadas e disponibilizadas.

As atividades de desenvolvimento e manutenção tanto do processo de software da organização quanto dos recursos relacionados fazem parte dos objetivos desta KPA. O conjunto de recursos relacionados ao processo de software da organização incluem: padrões do processo de software; guias e critérios para uso dos padrões do processo de software; descrições dos ciclos de vida de software que podem ser usados pela organização; banco de dados dos processos e seus produtos de software resultantes e; uma biblioteca dos documentos relacionados com o processo de software.

A Atividade nº 5, que coloca a necessidade do estabelecimento de um banco de dados da organização com informações dos processos e produtos de software resultantes, aborda também

alguns requisitos importantes para a base de dados, tais como: restrição de acesso às informações, integridade e acurácia dos dados armazenados.

Sendo este banco de dados um software, a KPA Definição do Processo da Organização pode verificar o cumprimento destes requisitos através de uma avaliação utilizando a Norma NBR 13596. Esta norma é recomendada, particularmente neste caso, justamente por considerar as subcaracterísticas 'Acurácia' e 'Segurança de Acesso' em seu modelo de qualidade de produto de software.

A preocupação com os dados armazenados no banco de dados são justificadas pelo fato de que elas formam a base cumulativa para que benefícios no processo de software da organização sejam realizados a longo prazo.

Exemplos, fornecidos pelo CMM[®], de informações a serem armazenadas neste banco de dados são: estimativas e resultados reais de tamanho, esforço e custo do software resultante do processo; dados sobre produtividade do processo; resultados de testes e medidas sobre a qualidade do processo e do produto de software resultante.

Uma forma de obter medidas sobre a qualidade dos produtos de software resultantes do processo de software da organização, é comparando estes produtos com um dado modelo pré estabelecido. A NBR 13596 propõe um modelo, baseado na definição das características de qualidade de software através de subcaracterísticas, que pode ser utilizado. Desta forma, dados sobre a qualidade de produtos de software poderão ser obtidos para serem armazenados no banco de dados.

Os dois momentos de aplicação da Norma NBR 13596, nesta KPA, dizem respeito ao banco de dados do processo de software. Num primeiro momento procurando garantir sua qualidade, enquanto software, e num segundo momento coletando dados da qualidade dos produtos de software, que devem ser armazenados nele. Portanto, o propósito desta KPA fica inalterado, sendo a Norma NBR 13596 apenas uma ferramenta a ser utilizada para auxiliar no desenvolvimento de uma base cumulativa para benefícios do processo de software, a longo prazo.

KPA: Programa de Treinamento

O propósito do Programa de Treinamento é desenvolver as habilidades e conhecimentos dos indivíduos para que eles possam desenvolver suas atividades dentro da organização, de forma efetiva e eficiente. As metas desta KPA são:

- M1. As atividades de treinamento são planejadas;
- M2. São fornecidos treinamentos para desenvolver as habilidades e conhecimentos necessários no desempenho de funções técnicas e gerenciais em software;
- M3. Os integrantes do grupo de engenharia de software, e dos demais grupos envolvidos com processo de software, recebem o treinamento necessário para desenvolverem suas atividades.

Um dos tópicos que não pode faltar em qualquer programa de treinamento de uma organização de software é o de 'Qualidade de Software'. A Meta 3 coloca que indivíduos da organização, envolvidos com processo de software, devem receber treinamento necessário para que possam desenvolver suas atividades e, a Norma NBR 13596 deve ser de conhecimento de qualquer pessoa que desenvolve ou mantém software.

Assim, nesta KPA cabe a sugestão de incluir a Norma 13596 no Programa de Treinamento de qualquer organização que pretende desenvolver software de qualidade. Pessoas envolvidas com processo de software precisam conhecer quais são as características e subcaracterísticas de qualidade que um produto de software deve ter, para poder implementá-las.

O propósito desta KPA estará sendo enriquecida com a treinamento dos indivíduos também em qualidade de produto de software, auxiliando a organização no desenvolvimento de produtos de software.

KPA: Gerenciamento Integrado de Software

O propósito desta KPA é o de integrar, de forma coerente, as atividades de engenharia e gerenciamento de um projeto, em um processo de software, definido a partir do processo do software padrão da organização. As metas desta KPA são duas:

- M1. O processo de software definido para cada projeto é uma versão derivada do processo de software padrão da organização;
- M2. Um projeto é planejado e gerenciado de acordo com o processo de software definido para ele.

No Nível 3 do CMM[®], um projeto define seu processo de software a partir do processo padrão da organização mas sem deixar de considerar suas necessidades particulares, lições aprendidas e dados de projetos anteriores. Além disso, o projeto retorna para a organização dados e documentos apropriados para serem considerados por outros projetos.

O 'Plano de Software' do projeto é construído agora baseado no processo de software definido para o projeto. Desta forma é possível ver claramente que o Gerenciamento Integrado de Software é a evolução das KPA Planejamento de Projetos de Software e Acompanhamento de Projetos de Software do Nível 2 de maturidade.

A Atividade n° 10 coloca a necessidade de se identificar, avaliar, documentar e gerenciar os riscos do projeto de software de acordo com um procedimento documentado. É necessário estabelecer um 'Plano de Gerenciamento de Riscos' para que os riscos e a forma de gerenciá-los sejam documentados.

Um risco que deve ser gerenciado, segundo o CMM[®], é a possibilidade significativa de falha no projeto em atingir os objetivos: de cronograma; custo; uso dos recursos computacionais críticos e; implementação de funcionalidades no software em desenvolvimento.

A 'Funcionalidade' é uma das características de qualidade de software considerada na Norma NBR 13596. As demais características de qualidade de software desta Norma, conforme já visto em capítulo específico, são: 'Confiabilidade', 'Usabilidade', 'Eficiência', 'Manutenibilidade' e 'Portabilidade'. Assim, a proposta é generalizar a preocupação explicitada pelo CMM[®], em relação à Funcionalidade, para as demais características de qualidade de produto de software. Com isso, o 'Plano de Gerenciamento de Riscos' deve prever uma forma de medir as características de qualidade nos produtos de software intermediários e como lidar com desvios detectados na implementação destas características.

Desta forma, fica inalterado o propósito de integração de atividades de engenharia e gerenciamento do projeto de software, reforçando apenas a necessidade da gerência de riscos quanto a implementação de todas as características de qualidade no produto de software em desenvolvimento pelo projeto.

KPA: Engenharia de Produto de Software

A Engenharia de Produto de Software deve executar, de forma eficiente e eficaz, um processo de engenharia bem definido, que integre todas as atividades do desenvolvimento de produtos de software consistentes e sem erros. Suas metas são:

- M1. As tarefas de engenharia de software são definidas, integradas e realizadas de forma consistente para produzir o software;
- M2. Os produtos de trabalho de software são mantidos consistentes entre si.

As tarefas da Engenharia de Produto de Software são de analisar e implementar os requisitos do sistema alocados ao software durante as fases de *design* (ou arquitetura), projeto, codificação, integração dos componentes e testes de software. Os testes de software são importantes para verificar se os requisitos especificados foram, de fato, implementados.

A Atividade n° 2, desta KPA, diz que os requisitos do software são implementados, mantidos, documentados e verificados através de análises sistemáticas dos requisitos alocados de

acordo com o processo de software definido para o projeto. Os requisitos de software mencionado cobre as funções do software, seu desempenho, a interface tanto do software como do hardware, e outros componentes do sistema.

Para a verificação da implementação dos requisitos, a Norma NBR 13596 provê uma aplicação imediata.

A característica ‘Funcionalidade’, da Norma NBR 13596, pode prover informações sobre a implementação das funções do software através das suas subcaracterísticas tais como a ‘Adequação’, que irá investigar a presença de um conjunto de funções e sua apropriação para as tarefas especificadas; a ‘Conformidade’, que estará preocupada com o fato das funções estarem sendo implementadas de acordo com normas, convenções e regulamentações; e a ‘Interoperabilidade’, que se volta a verificar a capacidade do software de interagir com outros sistemas especificados.

A característica ‘Eficiência’ também deve contribuir na verificação dos requisitos do software, quanto ao seu desempenho, pois ela está voltada a análise do tempo de resposta de processamento e velocidade na execução das funções através da sua subcaracterística ‘Comportamento em Relação ao Tempo’.

Outras características e subcaracterísticas, descritas na Norma NBR 13596, certamente poderão contribuir nas tarefas alocadas à Engenharia de Produto de Software. Isso porque de acordo com seu objetivo, “Essa Norma é dirigida àqueles envolvidos com aquisição, desenvolvimento, uso, suporte, manutenção ou auditoria de software.” ... “A definição das características e o modelo do processo de avaliação de qualidade correspondente, nesta Norma, são aplicáveis na especificação dos requisitos de produtos de software e na avaliação da sua qualidade ao longo do seu ciclo de vida.”

Outras Atividades desta KPA podem fazer uso do modelo de qualidade da Norma NBR 13596 mas, a Atividade nº 2 já demonstra a capacidade que esta Norma tem, de auxiliar a Engenharia de Produto de Software a executar, de forma consistente, o seu propósito.

KPA: Coordenação entre Grupos

O propósito da Coordenação entre Grupos é estabelecer os meios para que o grupo de engenharia de software participe ativamente com outros grupos de engenharia para que o projeto possa, de forma eficaz e eficiente, melhor satisfazer às necessidades do cliente. As metas a serem cumpridas para que este propósito seja atingido são:

- M1. Os requisitos do cliente são acordados por todos os grupos envolvidos;
- M2. As responsabilidades entre os grupos de engenharia são acordadas por todos os grupos envolvidos;
- M3. Os grupos de engenharia identificam, acompanham e resolvem discordância entre grupos.

Num projeto de software, o grupo de engenharia de software interage ativamente com diversos outros grupos interdisciplinares, tais como: engenharia de sistemas, marketing, treinamento, gerenciamento de subcontrato e documentação. A coordenação entre grupos é o primeiro passo para manter em curso um projeto em que tantas engenharias devem trabalhar juntas.

As melhores práticas para se cumprir as metas são relacionadas a: interações disciplinadas e; coordenações, dos grupos de engenharia do projeto, para se cumprir os requisitos do sistema, os objetivos e os planos. Os compromissos entre grupos são documentados e acordados entre as partes. A coordenação entre grupos pode ser desde uma a duas reuniões entre os grupos do projeto até a formação de uma equipe dedicada a esta tarefa.

A Atividade nº 1 coloca a necessidade da participação de todos os grupo de engenharia, além do grupo de engenharia de software, no estabelecimento dos requisitos do sistema junto ao cliente e/ou usuário final. Além disso, estes grupos devem documentar os critérios de aceitação de cada produto a ser entregue ao cliente. Normalmente a responsabilidade pelos requisitos do sistema é do grupo de engenharia de software, mas é esperado que os outros grupos de engenharia tenham um envolvimento significativo nesta tarefa.

Nesta Atividade, a Norma NBR 13596 estará presente como diretiva para a 'Especificação de Requisitos de Software' junto ao cliente e/ou usuário, da mesma forma como se faz presente na KPA Gerenciamento de Requisitos do Nível 2 de Maturidade. Além disso, é fundamental acrescentar a necessidade do cumprimento das características de qualidade como critério de aceitação do produto de software a ser entregue ao cliente; mesmo que algumas destas características tenham sido negligenciadas pelo cliente, durante o levantamento dos requisitos do software.

Portanto, a proposta nesta KPA é prover, aos grupos de engenharia, um meio para que as necessidades dos clientes sejam satisfeitas; ou seja, o uso das características de qualidade de um produto de software, definidas na Norma NBR 13596, como referência para o levantamento dos requisitos e critérios de aceitação do produto de software.

KPA: Revisões de Pares

Remover defeitos de produtos de trabalho de software, o mais cedo possível e de modo eficiente, é o propósito desta KPA. Para tanto, duas metas devem ser cumpridas:

- M1. As revisões são planejadas;
- M2. Defeitos nos produtos de trabalho de software são identificados e removidos.

Como efeito colateral das Revisões de Pares obtém-se um melhor entendimento dos produtos de trabalho e de defeitos que possam ser evitados. As atividades de Revisões de Pares envolve identificar produtos de trabalho de software que passarão por revisões de pares e, execução de exames metódicos nos produtos identificados para detectar defeitos e áreas onde mudanças são necessárias.

Na Atividade nº 1, descrita pelo CMM[®], as revisões de pares são planejadas identificando-se os produtos de trabalho e o cronograma para a execução das revisões. Os produtos de trabalho de software, que serão entregues ao cliente, podem passar por revisões de pares e, neste caso, devem constar no plano.

A execução das revisões de pares, de acordo com a Atividade nº 2, deve ser de acordo com um procedimento documentado. O que é sugerido, por esta dissertação, é referenciar no procedimento o uso de 'Listas de Verificação' como forma de se executar revisões de maneira consistente. 'Listas de Verificação' podem ser definidas para produtos de trabalho específicos e, também para revisões de pares com objetivos específicos. Alguns objetivos que podem nortear o desenvolvimento de uma 'Lista de Verificação' são a necessidade de averiguar o cumprimento de padrões ou procedimentos; a acurácia; regras de construção e; manutenibilidade.

Assim, uma 'Lista de Verificação' também pode ser desenvolvida e utilizada em revisões de pares para averiguar a implementação das características de qualidade de produto de software intermediários, ou no produto final. No Brasil já existem instituições especializadas em avaliações da qualidade de produto de software que utilizam uma Lista de Verificação baseada na NBR 13596, como é o caso do (CenPRA, 2002) que desenvolveu e utiliza o método MEDE-PROS® - Método de Avaliação da Qualidade de Produto de Software (MEDE-PROS, 1996).

O CenPRA vem utilizando a seis anos o MEDE-PROS® na prestação de serviços de avaliação da qualidade de produtos de software para organizações, clientes ou usuários de software e entidades que promovem a qualidade de software no Brasil através de premiações e financiamentos. O Laboratório de Avaliação da Qualidade de Produto de Software do CenPRA, que realiza as avaliações utilizando o MEDE-PROS®, também se preocupa com a qualidade na prestação deste serviço e, por isso trabalhou no sentido de se estruturar para atender aos requisitos da Norma ISO 9002:1994 (SANT'ANA, 2000) com enfoque sistêmico (SILVEIRA, 2001).

A utilização da NBR 13596 é bastante útil para se atingir o propósito desta KPA uma vez que auxilia na detecção de problemas em produtos de software intermediários, considerando um padrão internacionalmente reconhecido. A remoção de defeitos, ainda na fase de desenvolvimento do produto de software, minimiza os custos de retrabalho, que são bastante elevados quando o produto já está totalmente pronto; e promove a satisfação das necessidades do cliente.

5.4 Análise do Nível 4: Gerenciado

No Nível 4 são feitas medições detalhadas do processo e da qualidade do produto de software. Estas medições permitem que tanto o processo como o produto sejam controlados e compreendidos quantitativamente.

Considerar o modelo de qualidade da norma NBR 13596 como ferramenta a ser utilizada nas KPA de Nível 4 do CMM® é facilitar a implementação de medições detalhadas sobre a qualidade do produto de software.

KPA: Gerenciamento Quantitativo de Processo

O propósito desta KPA, segundo o CMM®, é controlar quantitativamente a execução do processo de um projeto de software. Para tanto as seguintes metas são estabelecidas:

- M1. As atividades de gerenciamento quantitativo de processo são planejadas;
- M2. O desempenho do processo de software definido para o projeto é controlado quantitativamente;
- M3. A capacidade do processo de software padrão da organização é conhecida de forma quantitativa.

A Atividade nº 4 tem por objetivo coletar dados para possibilitar o controle quantitativo do processo de software de um dado projeto. De acordo com o CMM®, estas medições devem cobrir as propriedades das KPA e seus principais produtos de trabalho. A coleta dos dados deve ser realizada de acordo com um procedimento documentado. Este procedimento determina que: os dados a serem coletados sejam especificados; suas definições sejam precisas e; que sejam definidas a intenção de uso, a análise dos dados a serem realizados e, os momentos do processo em que os dados serão coletados.

O CMM® fornece também alguns exemplos de dados a serem medidos, tais como: dados de produtividade; eficiência das revisões de pares; número de defeitos encontrados nos requisitos do

software e; a gravidade destes defeitos. Apesar do foco desta KPA ser no processo, fica evidenciado através destes exemplos que não apenas dados do processo mas também dados de produtos de software precisam ser coletados e analisados para que o processo possa ser gerenciado.

Portanto, uma composição básica para o uso da Norma NBR 13596 é formada na Atividade nº 4. Ou seja, medições devem ser realizadas, em determinados momentos do processo, para cobrir propriedades dos principais produtos de trabalho de software, como o número e a gravidade de defeitos relacionados aos requisitos destes produtos.

Como já abordado nesta dissertação, as características de qualidade de produto de software, definidas na NBR 13596, precisam ser contempladas na ‘Especificação de Requisitos de Software’. Esta Norma, além das características de qualidade, sugere um modelo que permite avaliar o produto de software através de subcaracterísticas, também estabelecidas por ela. Esta é uma forma de medir possíveis defeitos relacionados aos requisitos de determinados produtos de trabalho, de um dado processo de software, que se deseja controlar quantitativamente.

KPA: Gerenciamento da Qualidade de Software

Desenvolver um entendimento quantitativo da qualidade dos produtos de software dos projetos e, atingir metas específicas de qualidade, são os propósitos da KPA de Gerenciamento da Qualidade de Software. As metas estabelecidas pelo CMM[®], para que estes propósitos sejam atingido, são:

- M1. As atividades do gerenciamento da qualidade de software de um dado projeto são planejadas;
- M2. Metas mensuráveis para a qualidade do produto de software e suas prioridades são definidas;
- M3. O progresso atual, quanto ao cumprimento das metas de qualidade para os produtos de software, são quantificados e gerenciados.

O foco desta KPA está no produto de software e, suas atividades envolvem definir metas de qualidade para os produtos e estabelecer planos para atingir estas metas. Além disso, o CMM® coloca a necessidade de monitorar e ajustar ‘Planos de Software’, produtos de trabalho de software, atividades e, metas de qualidade para que as necessidades dos clientes e/ou usuários sejam satisfeitas. Nas organizações Nível 2 de CMM®, o foco da qualidade é ‘conformidade com requisitos’, já no Nível 4 a ênfase está no entendimento das necessidades do cliente e/ou usuário.

A Atividade nº 1 desta KPA estabelece a necessidade de desenvolver o ‘Plano de Qualidade de Software’ que deve conter as metas de qualidade do produto de forma quantitativa. O CMM®, neste ponto, alerta mais uma vez que é preciso ter bem entendida a necessidade do cliente e/ou usuário para que as metas de qualidade do produto possam ser estabelecidas. Este Plano será a base para o gerenciamento da qualidade de software e, portanto, deve estabelecer os momentos do processo onde a qualidade de software será medida e como será medida. É neste momento que a NBR 13596 pode ser citada como referência a ser utilizada para medir a qualidade do software.

As palavras literais da Atividade nº 3, desta KPA, que determinam a necessidade de se definir, monitorar e revisar, durante todo o ciclo de vida do produto, as metas de qualidade quantitativas do produto de software são: *“Characteristics of product quality that describe how well the software product will perform or how well it can be developed and maintained are identified (CMM, 2000)”*.

Examinando a Norma NBR 13596 temos, também literalmente, descrito em seu objetivo: “A definição das características e o modelo do processo de avaliação de qualidade correspondente, nesta Norma, são aplicáveis na especificação dos requisitos de produtos de software e na avaliação da sua qualidade ao longo do seu ciclo de vida (NBR 13596, 1996)”.

De forma direta, a aplicação da NBR 13596 é cabível nesta KPA. É verdade também que esta Norma não fornece métricas nem métodos para a medição e julgamento da qualidade do produto de software. O método pode ser desenvolvido pela própria organização, ou esta avaliação pode ser realizada em organizações especializadas, como é o caso do (CenPRA, 2002) no Brasil, já mencionado anteriormente.

A utilização da Norma NBR 13596 contribui com o desenvolvimento de um entendimento quantitativo da qualidade dos produto de software.

5.5 Análise do Nível 5: Otimizando

‘Otimizando’ é o nome dado pelo CMM® ao Nível 5 de Maturidade que representa a Melhoria Contínua onde mudanças são realizadas de forma controlada gerando uma maior capacidade do processo de software da organização.

Portanto, a Melhoria Contínua é o foco do CMM® para organizações de Nível 5 e a chave para o sucesso é realizar mudanças de forma disciplinada. Através das atividades previstas pelo CMM®, e analisando a Figura 5.1, percebe-se que o ciclo PDCA – *Plan, Do, Check e Act* é utilizado de forma implícita, como ferramenta da qualidade para disciplinar a realização de mudanças nas organizações de software.

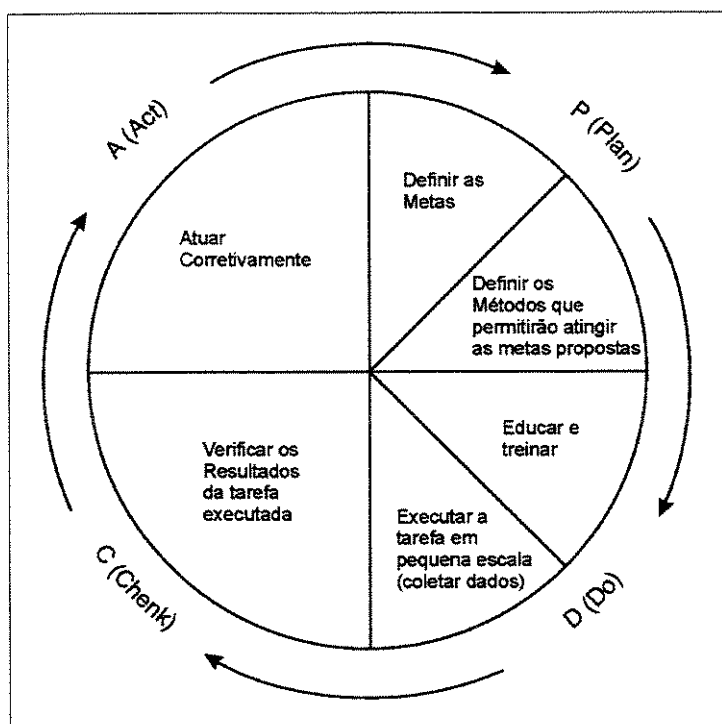


FIGURA 5.1 CICLO PDCA

Utilizando o ciclo PDCA na interpretação das atividades previstas pelo CMM[®], verifica-se que a melhoria contínua é prevista e apoiada nas seguintes ações recomendadas: Em *'Plan'*, designar uma equipe para atuar nas atividades de melhoria, planejar o que pode ser melhorado e avaliar o propósito das possíveis melhorias. Em *'Do'*, projetos pilotos são conduzidos utilizando-se a melhoria proposta para que, em *'Check'*, os resultados do projeto piloto sejam comparados com os resultados já existentes da execução sem a melhoria e, uma decisão seja tomada. A incorporação, ou não, da melhoria na organização é executada, em *'Act'*, atualizando-se os procedimentos, treinando os envolvidos além de outras atividades necessárias para que a melhoria seja institucionalizada na organização.

A melhoria contínua do processo é viabilizada através de retornos quantitativos do próprio processo e de testes que são realizados com novas idéias e novas tecnologias. No Nível 4, o processo de software é compreendido de forma quantitativa e, no Nível 5, a melhoria contínua deste processo, fundamenta as condições necessárias para que sua capacidade seja sempre melhorada. A Norma NBR 13596 é introduzida neste momento para auxiliar as KPA nesta difícil missão.

KPA: Prevenção de Defeitos

O propósito da Prevenção de Defeitos, dentro da abordagem do CMM[®], significa identificar as causas de defeitos, que ocorreram no passado, e realizar ações que previnam que os mesmos voltem a ocorrer. Para tanto, três metas são necessárias:

- M1. Atividades para prevenção de defeitos são planejadas;
- M2. Causas comuns de defeitos são procuradas e identificadas;
- M3. As causas comuns de defeitos são priorizadas e sistematicamente eliminadas.

Os defeitos identificados podem ser no projeto em andamento ou em outros projetos já finalizados e, as ações de prevenção devem ser tomadas tanto em nível de projeto como no nível da organização.

Para atingir este propósito, todo projeto de software deve elaborar um ‘Plano de Atividades para Prevenção de Defeitos’. Este plano irá cobrir, entre outras coisas, as atividades a serem executadas. Uma destas atividades, conforme descreve a Atividade nº 2 desta KPA, é a reunião entre as equipes envolvidas num projeto de software que está iniciando. Esta reunião tem por objetivo preparar as equipes nas tarefas a serem executadas tanto no projeto de software como também nas atividades previstas de prevenção de defeitos.

Nesta preparação das equipes, as metas de qualidade para os produtos de software do projeto são abordadas. Assim, estas mesmas metas de qualidade do produto de software, que são definidas no ‘Plano de Qualidade de Software’ comentada na KPA de Gerenciamento da Qualidade de Software do Nível 4, serão monitoradas na KPA de Prevenção de Defeitos.

Conforme descrito na Atividade nº 3, nas reuniões de análises causais, os desvios ou defeitos encontrados, por exemplo, nas metas de qualidade de produto de software, são analisados para se determinar as possíveis causas. Após identificadas as causas, a Atividade nº 4 irá definir ações a serem tomadas para prevenir que estes defeitos voltem a ocorrer tanto no projeto em andamento como nos futuros projetos da organização.

De forma indireta, a Norma NBR 13596 está envolvida pois, dentro da proposta desta dissertação, ela foi considerada na definição das metas de qualidade do produto de software, conforme descrito na KPA de Gerenciamento da Qualidade de Software, e estará sendo utilizada aqui quando se analisa problemas em atingi-las.

KPA: Gerenciamento de Mudança Tecnológica

O Gerenciamento de Mudança Tecnológica tem como propósito identificar novas tecnologias (ferramentas, métodos e processos) e incorporá-las na organização de uma maneira ordenada. As metas para cumprir este propósito são:

M1. A incorporação de mudanças tecnológicas deve ser planejada;

- M2. Novas tecnologias devem ser avaliadas para determinar seus efeitos na qualidade e produtividade;
- M3. As novas tecnologias apropriadas são adotadas como práticas normais em toda a organização.

As organizações Nível 5 de Maturidade, segundo o CMM[®], mantêm-se atentas às inovações tecnológicas relacionadas a software e, sistematicamente, às experimentam e avaliam para selecionar aquelas apropriadas para melhorar a qualidade de seus software e a produtividade de suas atividades. Mudanças tecnológicas nestas organizações já são institucionalizadas, ou seja, faz parte de sua cultura.

Conforme consta no propósito desta KPA, por tecnologia entende-se ferramentas, métodos e processos. Portanto, novos software são adquiridos como ferramentas a serem utilizadas no processo de software da organização. Estas ferramentas precisam ser avaliadas para verificar se são apropriadas para melhorar a qualidade dos produtos de software, desenvolvidos pela organização, e para aumentar a produtividade das atividades em que estarão envolvidas.

O que esta dissertação certamente sugere é que todo software, a ser adquirido pela organização, seja avaliado para verificar a sua qualidade. A NBR 13596 provê o modelo de qualidade a ser utilizado assegurando uma avaliação fundamentada nas características de qualidade de produto de software de reconhecimento internacional.

A Atividade n^o 1, desta KPA, determina que o gerenciamento de mudança tecnológica deve ser conduzida por um plano definido e mantido pela organização. O CMM[®] também reforça a necessidade de se constar neste plano, como novas tecnologias serão avaliadas pela organização. Assim, a Norma NBR 13596 poderá constar como modelo de referência para avaliação da qualidade de um produto de software que esteja em estudo pela organização para ser adquirido.

O propósito de identificar novas tecnologias e incorporá-las na organização de uma maneira ordenada fica enriquecida com esta sugestão através da diminuição da possibilidade de produtos de software sem qualidade sejam adquiridos como ferramentas.

KPA: Gerenciamento de Mudança de Processo

O propósito da KPA de Gerenciamento de Mudança de Processo é melhorar continuamente os processos de software da organização, com o objetivo de melhorar a qualidade do software, aumentar a produtividade e, minimizar o tempo de desenvolvimento.

- M1. A melhoria contínua de processo é planejada;
- M2. A participação nas atividades de melhoria do processo de software da organização é um dever de todos na organização;
- M3. O processo de software padrão da organização e os processos de software definidos para projetos são melhorados continuamente.

A melhoria contínua ainda é o foco do CMM® e a chave para o sucesso é realizar mudanças de forma disciplinada e, o ciclo PDCA também pode ser identificado nas atividades propostas para esta KPA.

Para executar a fase de *'Check'* do PDCA, o Gerenciamento de Mudança de Processo armazena os dados relacionados às atividades de melhoria de processo, que irão ser utilizados na comparação com os resultados já existentes da execução do processo sem a melhoria realizada. Os resultados já existentes do processo estão armazenados no Banco de Dados, estabelecido na Atividade nº 5 da KPA de Definição do Processo da Organização, e que também será utilizado para armazenar os dados relacionados às atividades de melhoria de processo.

A sugestão desta dissertação na KPA de Definição do Processo da Organização foi de utilizar a Norma NBR 13596 para avaliar este Banco de Dados no sentido de verificar o cumprimento de requisitos quanto a algumas características de qualidade, fundamentais para a integridade dos dados armazenados, que são a *'Acurácia'* e a *'Segurança de Acesso'*.

Assim a KPA de Gerenciamento de Mudança de Processo, em sua Atividade nº 9 de registrar e ter acesso a registros sobre o processo de software e sobre a melhoria de processo de

software, está sendo beneficiada de forma indireta com o uso da Norma NBR 13596 na garantia da integridade dos registros do banco de dados utilizado.

5.6 Resumo das Considerações

Nesta seção, para cada Nível de Maturidade, é realizado um resumo de todas as análises, e apresentado no seguinte formato:

- Tabela Resumo do Nível de Maturidade onde, para cada KPA são apresentados: propósito definido pelo CMM®; novo propósito com o valor agregado pelas considerações desta dissertação, ou seja, CMM® + NBR 13596 e; referências das instâncias do modelo CMM® onde as considerações de aplicação da NBR 13596 foram propostas.
- Para cada referência de instância, citada nas KPA, são transcritas:
 - sua descrição pelo modelo CMM® e;
 - as considerações propostas por esta dissertação.

Conforme já mencionado na seção 5.1, as referências de instâncias podem ser:

- ‘Metas’ ou ‘Produtos de Trabalho’ descritos, por Nível de Maturidade, nas seções 5.2, 5.3, 5.4 e 5.5 ou;
- Práticas Chave de ‘Atividades’ descritas, também por Nível de Maturidade, no Anexo.

5.6.1 Nível 2: Repetível

A Tabela 5.2 resume onde considerações de uso da NBR 13596 podem ser feitas e o que vem a acrescentar nos propósitos de cada KPA do Nível 2.

TABELA 5.2 INSTÂNCIAS PARA USO DA NBR 13596 E VALOR AGREGADO ÀS KPA DO NÍVEL 2

KPA	PROPÓSITOS		INSTÂNCIAS
	CMM [®]	CMM [®] + NBR 13596	
Gerenciamento de Requisitos	Entendimento comum, entre cliente e equipe do projeto de software, sobre os requisitos do cliente a serem implementados no software.	Entendimento comum, entre cliente e equipe do projeto de software, sobre os requisitos do cliente, <u>incluindo as características de qualidade</u> , a serem implementados no software.	Produto de Trabalho: 'Especificação de Requisitos de Software'; Atividade n° 1; Atividade n° 2; Atividade n° 3.
Planejamento de Projetos de Software	Estabelecimento de planos razoáveis para desenvolver software e para gerenciar projetos de software.	Estabelecimento de planos razoáveis para desenvolver software, <u>que apresentem características de qualidade implementadas</u> , e para gerenciar projetos de software.	Produto de Trabalho: 'Plano de Software'; Meta 2.
Acompanhamento de Projetos de Software	Visibilidade adequada do progresso atual do projeto de software, possibilitando medidas efetivas da gerência quando o desempenho se desvia significativamente do planejado.	Visibilidade adequada do progresso atual do projeto <u>e do produto</u> de software, possibilitando medidas efetivas da gerência quando <u>seus desempenhos</u> se desviam significativamente do planejado.	Meta 1; Meta 2; Atividade n° 9.
Gerenciamento de Subcontrato de Software	Selecionar fornecedores qualificados de software e gerenciá-los eficazmente.	Selecionar fornecedores qualificados e gerenciar eficazmente <u>o processo e a qualidade do produto de software a ser entregue</u> .	Produtos de Trabalho: 'Especificação de Requisitos de Software' e; 'Plano de Software'; Atividade n° 9; Atividade n° 12.
Garantia da Qualidade de Software	Prover visibilidade quanto ao processo, em uso no projeto, e aos produtos, em desenvolvimento.	Prover visibilidade quanto ao processo, em uso no projeto, e <u>à qualidade</u> dos produtos, em desenvolvimento.	Meta 1; Meta 2; Atividade n° 2; Atividade n° 5.
Gerenciamento de Configuração de Software	Estabelecer e manter a integridade de determinados produtos do projeto, durante o ciclo de vida do software.	Estabelecer e manter a integridade de determinados produtos do projeto, estando atento também para a presença da característica de qualidade 'Manutenibilidade' no produto em desenvolvimento, facilitando o rastreamento de problemas, durante o ciclo de vida do software.	Meta 2.

KPA Gerenciamento de Requisitos

Produto de Trabalho: 'Especificação de Requisitos de Software'

- Descrição do CMM®: Especificação documentada descrevendo o comportamento do software esperado pelo cliente.
- Considerações Propostas: Garantir a presença das características de qualidade de produto de software, definidas na NBR 13596, no documento 'Especificação de Requisitos de Software', através do uso de uma lista de verificação para o levantamento, junto ao cliente, dos requisitos relacionados às características de qualidade.

Atividades n^{os} 1, 2 e 3

- Descrições do CMM®:
 - Atividade n° 1. O grupo de engenharia de software revê os requisitos a serem alocados antes de incorporá-los ao projeto de software.
 - Atividade n° 2. O grupo de engenharia de software usa os requisitos alocados como base para os planos de software, produtos de trabalho e atividades.
 - Atividade n° 3. As mudanças nos requisitos alocados são revisadas e incorporadas no projeto de software.
- Considerações Propostas: Ao considerar o modelo da Norma NBR 13596 na 'Especificação de Requisitos de Software', necessariamente as características de qualidade estarão sendo consideradas pelo grupo de engenharia de software em suas atividades de revisão dos requisitos a serem alocados, ou que sofreram mudanças, além de servirem como base para os planos de software, produtos de trabalho e atividades do projeto.

KPA Planejamento de Projetos de Software

Produto de Trabalho: 'Plano de Software'

- Descrição do CMM®: Plano documentado com informações, tais como: o ciclo de vida definido para o software; a lista de produtos a serem desenvolvidos; o cronograma de execução; os riscos do projeto; estimativas de esforço, custo, recursos; além de outras informações relevantes ao desenvolvimento do software e gerenciamento do projeto.
- Considerações Propostas:
 - Estabelecer, no documento 'Plano de Software', o uso da Norma NBR 13596 para avaliar a implementação das características de qualidade de produto de software em produtos de trabalho do projeto de software.
 - Fazer constar no 'Plano de Software' os produtos de trabalho que deverão passar por avaliações de qualidade de produto de software; assim como cronograma, riscos, estimativas de esforço, custo e recursos necessários para estas avaliações.

Meta 2

- Descrição do CMM®: Ter, de forma documentada, as atividades do projeto de software e os compromissos necessários.
- Considerações Propostas: Estabelecer, no documento 'Plano de Software', atividades de avaliações da qualidade de produto de software em produtos intermediários pertinentes e no produto de software final.

KPA Acompanhamento de Projetos de Software

Meta 1

- Descrição do CMM®: O desempenho e os resultados atuais do projeto são acompanhados e revisados em relação ao que consta nos planos.

- Considerações Propostas:

Quanto aos 'resultados atuais': As atividades de verificação e validação dos produtos de software intermediários e final, que constam no 'Plano de Software', devem incluir avaliações utilizando a Norma NBR 13596, para que a implementação das características de qualidade de produto de software esperada, naquele produto, seja comparada com a real, constatada na avaliação.

Quanto ao 'desempenho': A adequação dos produtos de software intermediários, os momentos e esforços necessários para a execução das avaliações da qualidade de produto de software também devem ser acompanhados.

Meta 2

- Descrição do CMM®: Ações corretivas são realizadas e gerenciadas quando o desempenho e resultados reais desviam significativamente dos planos.
- Considerações Propostas:

Quanto aos 'resultados atuais': Os resultados das avaliações em produtos de software intermediários, durante o processo de software, irá permitir a tomada de ações, que se façam necessárias para orientar a implementação das características de qualidade que estejam desviando de seus requisitos.

Quanto ao 'desempenho': Ações corretivas devem ser tomadas, caso os esforços necessários na execução das avaliações, das características de qualidade de produto de software, sejam significativamente diferentes do planejado. Recomenda-se também que estes ajustes não sejam apenas quanto ao dimensionamento dos esforços necessários, mas também quanto aos momentos em que as avaliações devem ocorrer e em quais produtos de trabalho.

Atividade n° 9

- Descrição do CMM®: As atividades técnicas de engenharia de software são acompanhadas e ações corretivas são tomadas quando necessário.

- Considerações Propostas: De acordo com a própria orientação do CMM® para esta atividade, qualquer problema que tenha sido identificado nos produtos de trabalho de software deve ser reportado e documentado, contribuindo na formação de dados históricos a serem considerados em novos projetos.

KPA Gerenciamento de Subcontrato de Software

Produto de Trabalho: 'Especificação de Requisitos de Software'

- Descrição do CMM®: Especificação documentada descrevendo o comportamento do software esperado pelo cliente.
- Considerações Propostas: Garantir a presença das características de qualidade do produto de software no documento 'Especificação de Requisitos de Software', conforme sugestão presente na KPA de Gerenciamento de Requisitos.

Produto de Trabalho: 'Plano de Software'

- Descrição do CMM®: Plano documentado com informações tais como: o ciclo de vida definido para o software; a lista de produtos a serem desenvolvidos; o cronograma de execução; os riscos do projeto; estimativas de esforço, custo, recursos; além de outras informações relevantes ao desenvolvimento do software e gerenciamento do projeto.
- Considerações Propostas:

Estabelecer, no documento 'Plano de Software', o uso da Norma NBR 13596 para avaliar a implementação das características de qualidade em produtos de software intermediários e no produto final.

Fazer constar no 'Plano de Software', revisões formais do fornecedor, que incluirão avaliações de qualidade em produtos de trabalho, que também deverão ser identificados. Além disso, deverão constar o cronograma destas avaliações; riscos; estimativas de esforço; custo e recursos necessários.

Atividade nº 9

- Descrição do CMM®: Em momentos significativos do projeto e de acordo com um procedimento documentado, revisões formais são conduzidas, pela engenharia de software do fornecedor, para verificar o trabalho realizado e os resultados atingidos.
- Considerações Propostas: Na execução desta Atividade, o modelo de qualidade da Norma NBR 13596 será usado para verificar os resultados atingidos quanto à implementação das características de qualidade de produtos de software intermediários, que foram identificados como relevantes.

Atividade nº 12

- Descrição do CMM®: O contratante conduz testes de aceitação, como parte do processo de entrega dos produtos de software pelo fornecedor, de acordo com procedimentos documentados.
- Considerações Propostas: O contratante conduz uma avaliação do produto de software, utilizando a Norma NBR 13596, como parte do processo de entrega do produto de software. Para isso, o contratante utiliza um procedimento documentado que também especifica a necessidade desta avaliação, com a NBR 13596, como parte do processo de entrega do produto de software.

KPA Garantia da Qualidade de Software

Meta 1

- Descrição do CMM®: As atividades da GQS são planejadas.
- Considerações Propostas: Incluir atividades de verificação da aderência do produto de software final, e de produtos intermediários relevantes, ao modelo padrão de qualidade de produto de software fornecido pela Norma NBR 13596.

Meta 2

- Descrição do CMM®: A conformidade dos produtos de software e processos quanto aos padrões, procedimentos e requisitos aplicáveis é verificada objetivamente.
- Considerações Propostas: Avaliações de conformidade, do software e da documentação do usuário, com o modelo de qualidade sugerido pela Norma NBR 13596, que é um padrão aplicável.

Atividade nº 2

- Descrição do CMM®: As atividades são realizadas de acordo com o plano de GQA.
- Considerações Propostas: O 'Plano de GQS' deve garantir a participação do grupo de GQS no estabelecimento do 'Plano de Software', dos padrões e procedimentos que sejam apropriados para uso no projeto. Desta forma, o grupo de GQA pode vir a garantir o uso da Norma NBR 13596, como padrão de qualidade de produto de software a ser seguido.

Atividade nº 5

- Descrição do CMM®: O grupo de GQA faz a auditoria em produtos de trabalho de software, para verificar sua aderência ao que foi especificado.
- Considerações Propostas: O grupo de GQS realiza auditorias em produtos de trabalho de software para verificar também sua aderência ao modelo de qualidade da NBR 13596.

KPA Gerenciamento de Configuração de Software

Meta 2

- Descrição do CMM®: Os produtos de trabalho a serem colocados sob gerência de configuração, chamados *baselines*, são identificados, controlados e disponibilizados.

- Considerações Propostas: O cumprimento desta meta contribui para que a característica de qualidade de produto de software 'Manutenibilidade', descrita na Norma NBR 13596, esteja presente nos produtos de software a serem entregues ao usuário. Isto porque, o controle de versões facilita a realização de modificações e remoção de defeitos num produto de software.

5.6.2 Nível 3: Definido

Da mesma forma que no Nível 2, a Tabela 5.3 apresenta o resumo das considerações de uso da NBR 13596, no Nível 3, e seus benefícios agregados aos propósitos de cada KPA.

KPA Foco no Processo da Organização

Atividade nº 5

- Descrição do CMM®: Novos processos, métodos e ferramentas, de uso limitado na organização, são monitorados, avaliados e, quando apropriado, transferidos para outras partes da organização.
- Considerações Propostas: Quando a ferramenta em uso experimental é um software, sua qualidade deve ser avaliada, utilizando a Norma NBR 13596, para averiguar a presença das características de qualidade. Esta avaliação pode ser um dos critérios para verificar se a ferramenta é apropriada, ou não, para ser utilizada pela organização.

KPA Definição do Processo da Organização

Atividade nº 5

- Descrição do CMM®: Um banco de dados do processo de software da organização é estabelecido e mantido.

TABELA 5.3 INSTÂNCIAS PARA USO DA NBR 13596 E VALOR AGREGADO ÀS KPA DO NÍVEL 3

KPA	PROPÓSITOS		INSTÂNCIAS
	CMM [®]	CMM [®] + NBR 13596	
Foco no Processo da Organização	Estabelecer uma responsabilidade organizacional em atividades que melhoram a capacidade do processo de software da organização.	Estabelecer uma responsabilidade organizacional em atividades que melhoram a capacidade do processo <u>e do produto</u> de software da organização.	Atividade nº 5.
Definição do Processo da Organização	Desenvolver e manter um conjunto de recursos, a serem utilizados no processo de software, que melhora o desempenho do processo e prove uma base cumulativa para benefícios a longo prazo.	Desenvolver e manter um conjunto de recursos <u>de qualidade</u> , a serem utilizados no processo de software, que melhora o desempenho do processo e prove uma base cumulativa para benefícios a longo prazo.	Atividade nº 5: • Banco de Dados; • Dados do Banco de Dados.
Programa de Treinamento	Desenvolver as habilidades e conhecimentos dos indivíduos para que eles possam desenvolver suas atividades de forma efetiva e eficiente.	Desenvolver as habilidades e conhecimentos dos indivíduos, <u>inclusive sobre qualidade de software</u> , para que eles possam desenvolver suas atividades de forma efetiva e eficiente.	Meta 3.
Gerenciamento Integrado de Software	Integrar, a partir do processo de software padrão da organização, as atividades de engenharia e gerenciamento de projeto em um processo de software, de forma coerente.	Integrar, a partir do processo de software padrão da organização, as atividades de engenharia e gerenciamento de projeto em um processo de software, de forma coerente.	Produto de Trabalho: 'Plano de Gerenciamento de Riscos'; Atividade nº 10.
Engenharia de Produto de Software	Executar, eficientemente e eficazmente, um processo de engenharia bem definido, que integre todas as atividades do desenvolvimento de produtos de software consistentes e sem erros.	Executar, eficientemente e eficazmente, um processo de engenharia bem definido, que integre todas as atividades no desenvolvimento de produtos de software <u>de qualidade</u> .	Atividade nº 2.
Coordenação entre Grupos	Estabelecer os meios para que o grupo de engenharia de software participe ativamente com outros grupos de engenharia para que o projeto possa, de forma eficaz e eficiente, melhor satisfazer às necessidades do cliente.	Estabelecer os meios, <u>incluindo o modelo de qualidade da NBR 13596</u> , para que o grupo de engenharia de software participe ativamente com outros grupos de engenharia para que o projeto possa, de forma eficaz e eficiente, melhor satisfazer às necessidades do cliente.	Atividade nº 1.
Revisões de Pares	Remover defeitos de produtos de trabalho de software o mais cedo possível e de modo eficiente.	Remover defeitos, <u>inclusive quanto às características de qualidade</u> , de produtos de trabalho de software o mais cedo possível e de modo eficiente.	Atividade nº 2.

- **Considerações Propostas:**

Banco de Dados: Verificar o cumprimento dos requisitos, definidos pelo CMM[®], de restrição de acesso às informações, integridade e acurácia dos dados armazenados, através de uma avaliação utilizando a Norma NBR 13596.

Dados a serem armazenados no Banco de Dados: Utilizar a Norma NBR 13596 para medir a qualidade do produto de software resultante do processo da organização, que deve ser armazenada no Banco de Dados.

KPA Programa de Treinamento

Meta 3

- **Descrição do CMM[®]:** Os integrantes do grupo de engenharia de software, e dos demais grupos envolvidos com processo de software, recebem o treinamento necessário para desenvolverem suas atividades.
- **Considerações Propostas:** Incluir a Norma 13596 no Programa de Treinamento dos integrantes do grupo de engenharia de software e dos demais grupos envolvidos com processo de software.

KPA Gerenciamento Integrado de Software

Produto de Trabalho: 'Plano de Gerenciamento de Riscos'

- **Descrição do CMM[®]:** Documento formal que identifica, avalia, documenta e gerencia os riscos do projeto de software. Um dos riscos a serem previstos é o de falha na implementação de funcionalidades no produto de software.
- **Considerações Propostas:** Documento formal que identifica, avalia, documenta e gerencia os riscos do projeto de software. Um dos riscos a serem previstos é o de falha na implementação de todas as características de qualidade de produto de software, definidos na Norma NBR 13596.

Atividade nº 10

- Descrição do CMM®: Os riscos do projeto de software são identificados, avaliados, documentados e gerenciados de acordo com um procedimento documentado.
- Considerações Propostas: O procedimento documentado deve prever a necessidade de avaliações em produtos de software intermediários; para que riscos relacionados à implementação das características de qualidade sejam gerenciadas ao longo do processo.

KPA Engenharia de Produto de Software

Atividade nº 2

- Descrição do CMM®: Os requisitos de software são implementados, mantidos, documentados, e verificados por uma análise sistemática dos requisitos alocados de acordo com o processo de software definido para o projeto.
- Considerações Propostas: Utilização da Norma NBR 13596 para se verificar a implementação dos requisitos de qualidade alocados ao software.

KPA Coordenação entre Grupos

Atividade nº 1

- Descrição do CMM®: Tanto o grupo de engenharia de software como os demais grupos de engenharia mantêm contato com os clientes e usuários finais, quando apropriado, para estabelecer os requisitos do sistema. Entre outras atividades, estes grupos devem: participar do estabelecimento dos requisitos do sistema junto ao cliente e/ou usuário final e; documentar os critérios de aceitação de cada produto a ser entregue ao cliente.
- Considerações Propostas: O grupo de engenharia de software e os demais grupos de engenharia devem participar da 'Especificação de Requisitos de Software' e da

definição dos critérios de aceitação. As características de qualidade de produto de software devem ser consideradas na especificação dos requisitos e, sua implementação considerada como um dos critérios de aceitação do produto pelo cliente. Desta forma todos ficam comprometidos com a qualidade do produto de software.

KPA Revisões de Pares

Atividade nº 2

- Descrição do CMM®: As revisões de pares são realizadas de acordo com um procedimento documentado.
- Considerações Propostas: Referenciar 'Listas de Verificação', baseadas na NBR 13596, para serem utilizadas em Revisões de Pares especificadas para averiguar a implementação das características de qualidade de produto de software.

5.6.3 Nível 4: Gerenciado

A Tabela 5.4 apresenta o resumo das instâncias de uso da NBR 13596 nas KPA do Nível 4 de Maturidade e os benefícios agregados aos seus propósitos.

TABELA 5.4 INSTÂNCIAS PARA USO DA NBR 13596 E VALOR AGREGADO ÀS KPA DO NÍVEL 4

KPA	PROPÓSITOS		INSTÂNCIAS
	CMM®	CMM® + NBR 13596	
Gerenciamento Quantitativo de Processo	Controlar quantitativamente a execução do processo de um projeto de software.	Controlar quantitativamente a execução do processo de um projeto de software.	Atividade nº 4.
Gerenciamento da Qualidade de Software	Desenvolver um entendimento quantitativo da qualidade dos produtos de software dos projetos e, atingir metas específicas de qualidade.	Desenvolver um entendimento quantitativo da qualidade dos projetos e dos produtos de software, <u>considerando características de qualidade de software reconhecidas internacionalmente</u> , e atingir metas específicas de qualidade.	Atividade nº 1; Atividade nº 3.

KPA Gerenciamento Quantitativo de Processo

Atividade nº 4

- Descrição do CMM®: Os dados, usados para controlar quantitativamente o processo de software definido para o projeto, são coletados de acordo com um procedimento documentado.
- Considerações Propostas: Dados sobre o número de defeitos relacionados aos requisitos de um produto de software, que são usados para controlar quantitativamente o processo de software definido para o projeto, são coletados através de uma avaliação utilizando-se a Norma NBR 13596.

KPA Gerenciamento da Qualidade de Software

Atividade nº 1

- Descrição do CMM®: O ‘Plano de Qualidade de Software’ do projeto é desenvolvido e mantido de acordo com um procedimento documentado.
- Considerações Propostas: O ‘Plano de Qualidade de Software’ inclui quais produtos de software devem ter sua qualidade gerenciada; em quais momento do processo e; a forma de medir esta qualidade. Recomenda-se que o uso da Norma NBR 13596 como forma de medição da qualidade de produto de software.

Atividade nº 3

- Descrição do CMM®: As metas quantitativas de qualidade dos produtos de software do projeto são definidas, monitoradas e revisadas ao longo do ciclo de vida do software.
- Considerações Propostas: Conforme a própria norma NBR 13596, “A definição das características e o modelo do processo de avaliação de qualidade correspondente,

desta Norma, são aplicáveis na avaliação da sua qualidade ao longo do seu ciclo de vida (NBR 13596, 1996)”.

5.6.4 Nível 5: Otimizando

A Tabela 5.5 apresenta o resumo das considerações de uso da NBR 13596 no Nível 5 do CMM®, e quais os benefícios agregados aos propósitos das KPA.

TABELA 5.5 INSTÂNCIAS PARA USO DA NBR 13596 E VALOR AGREGADO ÀS KPA DO NÍVEL 5

KPA	PROPÓSITOS		INSTÂNCIAS
	CMM®	CMM® + NBR 13596	
Prevenção de Defeitos	Identificar as causas de defeitos e prevenir suas recorrências.	Identificar as causas de defeitos, <u>inclusive quanto ao cumprimento das metas relacionadas às características de qualidade do produto de software</u> , e prevenir suas recorrências.	Atividade nº 2; Atividade nº 3; Atividade nº 4.
Gerenciamento de Mudança Tecnológica	Identificar novas tecnologias e incorporá-las na organização de uma maneira ordenada.	Identificar novas tecnologias, <u>que sejam de qualidade</u> , e incorporá-las na organização de uma maneira ordenada.	Atividade nº 1.
Gerenciamento de Mudança de Processo	Melhorar continuamente os processos de software da organização com o objetivo de melhorar a qualidade do software, aumentar a produtividade e minimizar o tempo de desenvolvimento de produto.	Melhorar continuamente os processos de software da organização com o objetivo de melhorar a qualidade do software, aumentar a produtividade e minimizar o tempo de desenvolvimento de produto.	Atividade nº 9.

KPA Prevenção de Defeitos

Atividade nº 2

- Descrição do CMM®: No início dos esforços para se realizar uma determinada tarefa de um projeto de software, a equipe encarregada desta tarefa participa de uma reunião

para se preparar em relação às atividades relacionadas com a tarefa e com a prevenção de defeitos.

- Considerações Propostas: Na preparação das equipes, as metas de qualidade para os produtos de software do projeto, com base nas características definidas pela Norma NBR 13596, são abordadas. Assim, estas metas de qualidade do produto de software que foram definidas no 'Plano de Qualidade de Software' serão monitoradas na KPA de Prevenção de Defeitos.

Atividade nº 3

- Descrição do CMM®: Reuniões para análises de causas são conduzidas de acordo com um procedimento documentado.
- Considerações Propostas: Nas reuniões de análises causais, os desvios ou defeitos encontrados nas metas para as características de qualidade de produto de software são analisados para se determinar as possíveis causas.

Atividade nº 4

- Descrição do CMM®: Todas as equipes designadas para coordenar as atividades de prevenção de defeitos se reúnem periodicamente para rever e coordenar implementações de ações propostas nas reuniões de análises de causas.
- Considerações Propostas: Ao se detectar as causas para os desvios, ou defeitos, encontrados nas metas para as características de qualidade de produto de software, ações são definidas e coordenadas para prevenir novas ocorrências no projeto em andamento e nos demais projetos da organização.

KPA Gerenciamento de Mudança Tecnológica

Atividade nº 1

- Descrição do CMM®: A organização desenvolve e mantém um plano para o gerenciamento de mudança tecnológica.
- Considerações Propostas: O 'Plano de Gerenciamento de Mudança Tecnológica' deve prever a avaliação de todo software a ser adquirido pela organização, utilizando-se a Norma NBR 13596 para verificar as características de qualidade de produto de software.

KPA Gerenciamento de Mudança de Processo

Atividade nº 9

- Descrição do CMM®: São mantidos registros das atividades de melhoria de processo de software.
- Considerações Propostas: Benefício indireto por utilizar um Banco de Dados que teve as características de qualidade de software verificadas através do uso na Norma NBR 13596, na KPA de Definição do Processo da Organização.

5.7 Comentários Gerais

Para o sucesso dos esforços de um trabalho de melhoria de qualidade, qualquer que seja o objeto, é fundamental o seu alinhamento com os negócios da organização e com qualquer outro programa de melhoria que a organização possa estar trabalhando. A ausência do alinhamento do trabalho de melhoria de qualidade com o negócio da organização afeta o comprometimento da alta gerência, que não irá prover o suporte necessário para a execução das atividades planejadas. Este é um ponto fundamental que não deve ser subjugado quando está se pensando em qualquer esforço para promoção da qualidade.

Independentemente do negócio da organização, a satisfação dos clientes é o fator decisivo de sua permanência no mercado. A meta do TQM de satisfação do cliente também é a meta do CMM[®] já na sua concepção em trabalhos supervisionados por Watts Humphrey, conforme visto na seção ‘Relação do TQM com o CMM[®]’ do Capítulo 3. O fator que desencadeou o desenvolvimento do modelo CMM[®] pelo SEI foi exatamente a busca do Departamento de Defesa dos EUA a fornecedores que satisfizessem às suas necessidades, provendo software de qualidade.

O CMM[®] é uma aplicação em software dos conceitos de gerenciamento de processos do TQM, e sua estratégia é focar um conjunto limitado de atividades do processo de software e trabalhar diretamente nele, possibilitando conquistas estagiadas, mas duradouras, na capacidade do processo de software da organização. A melhoria do processo de software de uma organização, promovida pelo modelo CMM[®], ocorre através de ciclos contínuos onde:

- Pontos Críticos, que comprometam a ‘qualidade do produto e do processo de software’, são identificados em avaliações realizadas na organização;
- O resultado destas avaliações é a classificação do Nível de Maturidade da organização e, um plano de melhoria pode ser elaborado para solucionar os pontos críticos detectados visando atingir o próximo Nível de Maturidade;
- Executa-se o plano de melhoria;
- Institucionaliza-se as novas práticas e;
- Uma nova avaliação é realizada.

A Norma NBR 13596 foi desenvolvida para ser aplicada na especificação dos requisitos de produtos de software e na avaliação de sua qualidade ao longo do seu ciclo de vida. Ela é dirigida àqueles envolvidos com aquisição, desenvolvimento, uso, suporte, manutenção ou auditoria de software, o que viabilizou as análises e considerações realizadas no trabalho desta dissertação.

Através dos ciclos contínuos para a melhoria do processo de software de uma organização, o CMM[®] está contribuindo para que as necessidades do cliente de software sejam satisfeitas. E, a introdução da Norma NBR 13596 no processo de software agrega valor a este processo promovendo a qualidade do software que, nesta dissertação, abrange processo e produto.

Capítulo 6

6 Conclusão

O trabalho desta dissertação evidenciou o potencial existente no uso da Norma NBR 13596 durante um processo de software baseado no modelo CMM®. A proposta de apresentar uma alternativa de integração das abordagens de qualidade de software: processo e produto, é atingida com a análise de diversas instâncias do processo de software com potencial de utilização dos conceitos da Norma brasileira sobre características de qualidade de produto de software.

A seguir, as considerações gerais deste trabalho são comentadas, e na seção 6.2 são abordadas algumas sugestões para trabalhos futuros, que proporcionam continuidade nos esforços para a melhoria da qualidade de software.

6.1 Considerações

No modelo CMM®, a organização deve trabalhar em parceria com o cliente entendendo suas necessidades, e promovendo um desenvolvimento ou manutenção de produtos de software, que estabelece os requisitos do cliente de forma documentada, e garante a alocação destes requisitos aos componentes do software.

De acordo com a Norma NBR 13596, para que se possa gerenciar a qualidade de software, é muito importante a tecnologia para especificar e avaliar de forma objetiva e quantitativa, tanto a ‘qualidade do produto de software’ quanto a ‘qualidade do processo de seu desenvolvimento e manutenção’.

Portanto, o CMM® e a Norma NBR 13596 reconhecem o fato de que a qualidade de um software é altamente influenciada pela qualidade do seu processo e, qualquer iniciativa no sentido de melhorar a qualidade de produto de software, deve estar engajada na melhoria da qualidade de processo de software da organização. Assim, é possível dizer que qualidade de software é a integração qualidade de processo e qualidade de produto.

Isto vêm de encontro com a proposta desta dissertação de integração das abordagens de qualidade de software: processo e produto, em benefício do próprio software. Com esta iniciativa, o foco estará na ‘qualidade do software’ e não só no processo, ou só no produto final. Ou seja, deve-se executar as melhores práticas no processo para desenvolver e manter um produto que possua as características de qualidade reconhecidas internacionalmente.

Através de uma análise superficial do modelo de processo CMM®, poderia ter sido considerado agir apenas em instâncias das KPA do Nível 4, denominada ‘Gerenciado’, pois conforme descrito no Capítulo 3, suas características básicas são as únicas a explicitarem o controle da qualidade do produto de software. Mas, como demonstrado no capítulo anterior, nos níveis de maturidade anteriores é possível, com o uso da NBR 13596, trabalhar nas ações que irão prover os resultados a serem gerenciados quantitativamente no Nível 4 e; no Nível 5 de Maturidade é possível promover uma melhoria contínua destes resultados.

Todas as 18 KPA distribuídas nos 4 Níveis de Maturidade do modelo de processo CMM®, possuem instâncias apontadas no trabalho desta dissertação, com potencial oportunidade de envolver e utilizar os conceitos da Norma NBR 13596. Isto reflete que, independente do Nível de Maturidade da organização, a oportunidade de utilizar a Norma NBR 13596 no processo de software é sempre possível, aplicável e recomendada.

6.2 Temas para Trabalhos Futuros

A aplicação prática da proposta apresentada nesta dissertação, em uma organização que esteja implantando melhoria de processo através do modelo CMM®, pode ser vista como um

trabalho futuro. Com a utilização de métricas pertinentes, este trabalho viria demonstrar a aplicabilidade desta proposta e seus resultados práticos.

A proposta apresentada nesta dissertação foi baseada no modelo de processo CMM[®], devido sua significativa disseminação e utilização ao redor do mundo, no momento de desenvolvimento do trabalho. No entanto, com algumas adaptações, esta proposta pode ser aplicada a outros modelos de processo. Novos trabalhos podem ser realizados desenvolvendo esta mesma proposta para outros modelos de processo como por exemplo: o CMMI e a futura Norma ISO/IEC 15504, em função da tendência de institucionalização destes modelos.

Qualquer esforço no sentido de promover a qualidade de software será bem aceita pela comunidade brasileira de TI, que não mede esforços para elevar a qualidade de nosso software de forma a torná-lo cada vez mais competitivo, tanto no mercado interno quanto no mercado externo.

Referências Bibliográficas

ABNT. **Associação Brasileira de Normas Técnicas**. Disponível em: <<http://www.abnt.org.br>>. Acesso em: 16 outubro 2001.

AGUAYO, M. T. V.; AZEVEDO, G. D. F. *Conceptos y Directrices para Evaluar y Mejorar la Calidad de Productos de Software*. Curso do CenPRA – Centro de Pesquisas Renato Archer. Ministrado no Intec - *Corporación de Investigación Tecnológica de Chile*. 17-18 abril 2001. Santiago. 213 p.

APOLLO. *Apollo Marks Inventory. Trademarks: Definitions*. Disponível em: <<http://www.apollogrp.com/trademarks/definitions.htm>>. Acesso em 21 novembro 2001.

AZEVEDO, G. D. F.; SOUZA, E. P. de. **Aplicação de Modelos de Qualidade à Especificação de Requisitos**. Curso do CenPRA – Centro de Pesquisas Renato Archer. Ministrado no CITS – Conferência Internacional de Tecnologia de Software: Qualidade de Software, 9, 1998, Curitiba. 150 p.

CAPOVILLA, I. G. G. **Elementos Intrínsecos do Software e sua Influência na Qualidade do Processo de Desenvolvimento**. 1999. 108 f.. Dissertação (Mestrado em Qualidade) – IMECC - Instituto de Matemática, Estatística e Computação Científica, UNICAMP - Universidade Estadual de Campinas, Campinas.

CenPRA. **Centro de Pesquisas Renato Archer**. Disponível em: <<http://www.cenpra.gov.br>>. Acesso em 18 janeiro 2002.

CMM. CARNEGIE MELLON UNIVERSITY. SEI - Software Engineering Institute. *The Capability Maturity Model: Guidelines for Improving the Software Process*. 14. ed. EUA: Addison Wesley Longman, 2000. 441 p.

CROSBY, P. B. *Quality Is Free*. 1. ed. USA: McGraw-Hill, 1979. 309 p.

DEMING, W. E. **Qualidade: A Revolução da Administração**. Título do original em inglês *Out of the Crisis*. Tradução de Clave Comunicações e Recursos Humanos SC Ltda. 1. ed. Rio de Janeiro: Marques-Saraiva, 1990. 367 p.

FERREIRA, A. B. de H. **Dicionário Aurélio Eletrônico**. v. 1.4. Rio de Janeiro: Nova Fronteira, 1994. CD-ROM.

HAASE, V.; et al. *Bootstrap: Fine-Tuning Process Assessment*. **IEEE Software**, USA, p. 25-35, julho 1994.

IEC. *International Electrotechnical Commission*. Disponível em: <<http://www.iec.ch>>. Acesso em: 16 outubro 2001.

IEEE. *Institute of Electrical and Electronics, Inc.* Disponível em: <www.ieee.org>. Acesso em 4 agosto 2001.

ISO. *International Organization for Standardization*. Disponível em: <<http://www.iso.ch>>. Acesso em: 16 outubro 2001.

ISO/IEC 9126. *INTERNATIONAL ORGANIZATION FOR STANDARLIZATION. Information technology – Software product evaluation: Quality characteristics and guidelines for their use*. Genebra, 1991. 13 p.

ISO/IEC 9126-2. *INTERNATIONAL ORGANIZATION FOR STANDARLIZATION. Information technology – Software product quality – Part 2: External metrics*. Genebra, 2001. 100 p.

ISO/IEC 9126-3. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information technology – Software product quality – Part 3: Internal metrics*. Geneva, 2001. 62 p.

ISO/IEC 9126-4. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information technology – Software product quality – Part 4: Quality in use metrics*. Geneva, 2001. 62 p.

ISO/IEC 14598-1. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information technology – Software product evaluation – Part 1: General overview*. Geneva, 1999. 20 p.

ISO/IEC 14598-2. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information technology – Software product evaluation – Part 2: Planning and management*. Geneva, 2000. 20 p.

ISO/IEC 14598-3. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information technology – Software product evaluation – Part 3: Process for developers*. Geneva, 2000. 16 p.

ISO/IEC 14598-4. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information technology – Software product evaluation – Part 4: Process for acquirers*. Geneva, 1999. 43 p.

ISO/IEC 14598-5. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information technology – Software product evaluation – Part 5: Process for evaluators*. Geneva, 1998. 50 p.

ISO/IEC 14598-6. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Information technology – Software product evaluation – Part 6: Documentation of evaluation modules*. Geneva, 2001. 42 p.

ISO/IEC FDIS 9126-1. *INTERNATIONAL ORGANIZATION FOR STANDARDIZATION.*

Information technology – Software product quality – Part 1: Quality model. Genebra, 2001. 25 p.

ISO/IEC TR 15504-2. *INTERNATIONAL ORGANIZATION FOR STANDARDIZATION.*

Information technology – Software process assessment – Part 2: A reference model for processes and process capability. Genebra, 1998. 39 p.

ISO/IEC TR 15504-5. *INTERNATIONAL ORGANIZATION FOR STANDARDIZATION.*

Information technology – Software process assessment – Part 5: An assessment model and indicator guidance. Genebra, 1998. 123 p.

JURAN, J. M. **Planejando para a Qualidade.** Título do original em inglês *Juran on Planning for Quality*. Tradução de João Mário Csillag e Cláudio Csillag. 3. ed. São Paulo: Pioneira, 1995. 394 p.

LAWRENCE, Jackie L. *Introduction to the Capability Maturity Model.* Curso oficial do SEI - *Software Engineering Institute*. Patrocinado pelo DoD – *U.S. Department of Defense*. Oferecido por *Integrated System Diagnostics Incorporated*. 25-27 de abril de 2001. São Paulo. 330 f.

MAGNANI, G. **Melhoria de Processo: Visão Geral e Estudos de Caso.** Tutorial oferecido na Semana de Engenharia de Software, 3, 12 agosto 1998, São Paulo.

MASTER, S. *An Overview of Capability Maturity Model Integration (CMMISM): version 1.0.* In: SIMPROS 2000 – SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE, 2, 2000, São Paulo. **Anais do SIMPROS 2000.** São Paulo: SENAC, 19 setembro 2000. Anexo: 82 p.

MCT. Ministério da Ciência e Tecnologia. Secretaria de Política de Informática e Automação. **Qualidade e Produtividade no Setor de Software Brasileiro**. Disponível em: <http://www.mct.gov.br/sepin/Dsi/qualidad/qualidad.htm>. Acesso em: 17 setembro 2001.

MEDE-PROS. CenPRA – Centro de Pesquisas Renato Archer. LAPS - Laboratório de Tecnologia de Avaliação de Qualidade de Produto de Software. LAQS – Laboratório de Avaliação de Qualidade de Software. **MEDE-PROS® - Método de Avaliação da Qualidade de Produto de Software**. versão 1.0. Campinas, 1996.

NBR 13596. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Tecnologia de informação - Avaliação de produto de software: Características de qualidade e diretrizes para o seu uso**. Rio de Janeiro, 1996. 10 p.

NBR ISO 8402. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Gestão da qualidade e garantia da qualidade: Terminologia**. Rio de Janeiro, 1993. 14 p.

NBR ISO 9000-3. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Gestão da Qualidade e Garantia da Qualidade – Parte 3: Diretrizes para a aplicação da NBR 19001 ao desenvolvimento, fornecimento e manutenção de software**. Rio de Janeiro, 1993. 14 p.

NBR ISO/IEC 12207. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Tecnologia da informação – Processos de ciclo de vida de software**. Rio de Janeiro, 1998. 35 p.

PAULK, M. C.; et al. *The Capability Maturity Model for Software*. Pittsburgh: SEI - Software Engineering Institute, 1999. 26 p.

PRESSMAN, R. S. **Engenharia de Software**. Tradução: José Carlos Barbosa dos Santos. Revisão técnica: Maldonado J. C.; Masiero P. C.; Sanches R. São Paulo: Makron Books, 1995. 1056 p.

ROCHA, A. R. C.; MALDONADO, J.C.; WEBER, K.C. **Qualidade de Software – Teoria e Prática**. 1. ed. São Paulo: Prentice Hall, 2001. 303 p.

SALVIANO, C. F.; SANT'ANA, M. L.; SOUZA, E. P. de. **Planejamento e Acompanhamento de Projetos**. Curso do CenPRA – Centro de Pesquisas Renato Archer. Ministrado na PETROBRAS. 6-7 agosto 2001. Salvador. 74 p.

SANT'ANA, M. L.; et al. Implantação do Sistema da Qualidade no Laboratório de Avaliação da Qualidade de Produto de Software da Fundação CTI. In: SIMPROS 2000 - SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE, 2, 2000, São Paulo. **Anais do SIMPROS 2000**. São Paulo: SENAC, 19 setembro 2000. 142 p., pp.117-130.

SEI. *Software Engineering Institute*. Disponível em: <<http://www.sei.cmu.edu>>. Acesso em 13 novembro 2001.

SEMA. *Software Engineering Measurement and Analysis*. Disponível em <<http://www.sei.cmu.edu/sema>>. Acesso em 23 janeiro 2002.

SILVEIRA, M. A.; SANT'ANA, M. L. **Implantação da ISO 9002:1994 com Enfoque Sistêmico no ITI/LAQS – Laboratório de Avaliação da Qualidade de Produto de Software**. Revista IMES, ano XVIII, (51), pp. 38-48, 2001.

SOUZA, E. P. de.; SALVIANO, C. F.; SANT'ANA, M. L. Introdução aos Modelos de Processo de Software: ISO/IEC 15504 (SPICE) e CMM/CMMI. In: SIMPROS 2001 - SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE, 3, 2001, São Paulo. **Anais do SIMPROS 2001**. São Paulo: SENAC, 2001. Anexo I: 194 p., pp. 1-53.

SPICE. *Software Process Improvement and Capability dEtermination*. Disponível em <<http://www.sqi.gu.edu.au/spice>>. Acesso em 10 outubro 2001.

TSUKUMO, A. N.; et al. Qualidade de Software: Visões de Produto e Processo de Software. In: CITS - CONFERÊNCIA INTERNACIONAL DE TECNOLOGIA DE SOFTWARE, 8, 1997, Curitiba. Anais da VIII CITS. Curitiba: Gráfica e Editora Linarth Ltda., junho 1997. 268. **Anexo T2: Qualidade de Software: As Normas ISO/IEC de Produto e Processo e sua Aplicação.** 17 p.

UHCL. *University of Houston Clear Lake. The Trillium Model.* Documento em pdf cedido pela Bell Canadá. Disponível em: <<http://www2.umassd.edu/swpi/BellCanada/trillium-html/trillium.html>>. Acesso em 13 novembro 2001.

WEBER, K. C.; ROCHA, A. R. C.; NASCIMENTO, C. J. **Qualidade e Produtividade em Software.** 4. ed. renovada. São Paulo: Makron Books, 2001. 188 p.

Anexo: Atividades definidas pelo CMM®

A seguir as ‘Atividades’ são listadas por KPA de cada Nível de Maturidade, obedecendo a mesma numeração utilizada no livro oficial do SEI ‘*The Capability Maturity Model*’ (CMM, 2000). Desta forma, a localização da ‘Atividade’, referenciada numericamente no Capítulo 5, pode ser realizada de forma direta.

Nível 2: Repetível

Gerenciamento de Requisitos

1. O grupo de engenharia de software revê os requisitos a serem alocados antes de incorporá-los ao projeto de software.
2. O grupo de engenharia de software usa os requisitos alocados como base para os planos de software, produtos de trabalho e atividades.
3. As mudanças nos requisitos alocados são revisadas e incorporadas no projeto de software.

Planejamento de Projeto de Software

1. O grupo de engenharia de software participa da equipe de projeto.
2. O planejamento do projeto de software é iniciado nos estágios iniciais do, e em paralelo com o, planejamento do projeto como um todo.
3. O grupo de engenharia de software participa com outros grupos afetados por todo o planejamento e durante a vida (existência) do projeto.

4. As responsabilidades, em relação ao projeto de software, de indivíduos ou grupos externos a organização são revistas com a alta gerência de acordo com um procedimento documentado.
5. O ciclo de vida do software com estágios predefinidos de gerenciamento são identificados e definidos.
6. O plano de software do projeto é desenvolvido de acordo com um procedimento documentado.
7. Todos os planos do projeto são documentados.
8. Os produtos de trabalho necessários para estabelecer e manter o controle do projeto de software são identificados.
9. As estimativas do tamanho dos produtos de trabalho, ou das mudanças nestes produtos, são realizadas de acordo com procedimento documentado.
10. As estimativas de esforços e custo necessários para o projeto de software são realizadas de acordo com procedimento documentado.
11. As estimativas dos recursos computacionais críticos para o projeto de software são realizadas de acordo com procedimento documentado.
12. O cronograma do projeto de software é realizado de acordo com procedimento documentado.
13. Os riscos associados ao custo, recursos, cronograma e aspectos técnicos do projeto são identificados, avaliados e documentados.
14. Os planos para os recursos de engenharia e ferramentas de suporte necessários ao projeto de software são preparados.
15. Os dados utilizados para o planejamento de software são registrados.

Acompanhamento de Projeto de Software

1. Um plano de software documentado é usado para o acompanhamento das atividades e a comunicação do seu andamento.
2. O plano de software do projeto é revisado de acordo com um procedimento documentado.

3. As responsabilidades, em relação ao projeto de software, e as mudanças nestas responsabilidades, de indivíduos ou grupos externos a organização são revistas com a alta gerência de acordo com um procedimento documentado.
4. Mudanças em responsabilidades aprovadas, que afetam o projeto de software, são comunicadas aos membros do grupo de engenharia de software e outros grupos relacionados ao projeto de software.
5. Os tamanhos dos produtos de trabalho de software, ou das mudanças nestes produtos, são acompanhados e ações corretivas são tomadas quando necessário.
6. Os esforços e custo necessários para o projeto de software são acompanhados e ações corretivas são tomadas quando necessário.
7. Os recursos computacionais críticos para o projeto são acompanhados e ações corretivas são tomadas quando necessário.
8. O cronograma do projeto é acompanhado e ações corretivas são tomadas quando necessário.
9. As atividades técnicas de engenharia de software são acompanhadas e ações corretivas são tomadas quando necessário.
10. Os riscos associados com custo, recursos, cronograma e aspectos técnicos do projeto são acompanhados.
11. Dados coletados e dados replanejados para o projeto de software são registrados.
12. O grupo de engenharia de software conduz revisões internas periódicas para acompanhar o progresso técnico, planos, desempenho e resultados em relação ao plano de software.
13. Revisões formais para verificar o trabalho realizado e os resultados do projeto de software são conduzidas em momentos significativos do projeto de acordo com um procedimento documentado.

Gerenciamento de Subcontrato de Software

1. O trabalho a ser subcontratado é definido e planejado de acordo com um procedimento documentado.

2. O fornecedor de software é selecionado de acordo com um procedimento documentado, através de uma avaliação de seu potencial para a execução do trabalho.
3. O contrato fechado entre o contratante e o fornecedor de software é usado como base para o gerenciamento do subcontrato.
4. O documento que contém o plano de software, elaborado pelo fornecedor, é revisado e aprovado pelo contratante.
5. O plano de software documentado e aprovado pelo contratante é usado para o acompanhamento das atividades de software e para a comunicação do seu andamento.
6. Mudanças no escopo do trabalho, nos termos e condições do contrato e outras responsabilidades acordadas com o fornecedor são resolvidas de acordo com um procedimento documentado.
7. O gerente de contratos da organização conduz periodicamente revisões de acompanhamento com o gerente de contratos do fornecedor.
8. Revisões técnicas periódicas são realizadas com o fornecedor.
9. Revisões formais para verificar o trabalho realizado e os resultados atingidos pela engenharia de software do fornecedor são conduzidas em momentos significativos do projeto de acordo com um procedimento documentado.
10. O grupo de garantia de qualidade de software da organização contratante monitora as atividades da garantia de qualidade de software do fornecedor de acordo com um procedimento documentado.
11. O grupo de gerência de configuração de software da organização contratante monitora as atividades da gerência de configuração de software do fornecedor de acordo com um procedimento documentado.
12. O contratante conduz testes de aceitação como parte do processo de entrega dos produtos de software pelo fornecedor de acordo com procedimentos documentados.
13. O desempenho do fornecedor é avaliado periodicamente e os resultados são revisados com o fornecedor.

Garantia da Qualidade de Software

1. Um plano de Garantia da Qualidade de Software - GQA é preparado para o projeto de software de acordo com um procedimento documentado.
2. As atividades são realizadas de acordo com o plano de GQA.
3. O grupo de GQA participa na preparação e em revisões do plano de software, padrões e procedimentos a serem usados no projeto.
4. O grupo de GQA revisa as atividades da engenharia de software para verificar se é condizente com o plano de software do projeto.
5. O grupo de GQA faz a auditoria nos produtos de trabalho de software designados para verificar sua aderência ao que foi especificado.
6. O grupo de GQA reporta periodicamente os resultados de suas atividades para o grupo de engenharia de software.
7. Desvios identificados nas atividades e nos produtos de trabalho de software são documentados e tratados de acordo com um procedimento documentado.
8. O grupo de GQA conduz periodicamente revisões de suas atividades e averiguações com seus clientes, quando apropriado.

Gerenciamento de Configuração de Software

1. Um plano de GCS é preparado para cada projeto de software de acordo com um procedimento documentado.
2. Um plano de GCA, documentado e aprovado, é usado como base para a realização das atividades de GCA.
3. Um sistema de biblioteca de gerência de configuração é estabelecido como um repositório para as *baselines* do software.
4. Os produtos de trabalho de software que serão colocados sob gerência de configuração são identificados.
5. Mudanças requeridas e problemas reportados, para todos os itens/unidades de configuração, são iniciados, registrados, revisados, aprovados e acompanhados de acordo com um procedimento documentado.

6. Mudanças nas *baselines* são controladas de acordo com procedimento documentado. Produtos da biblioteca das *baselines* são criados e suas versões são controladas de acordo com um procedimento documentado.
7. O estado dos itens/unidades de configuração é registrado de acordo com um procedimento documentado.
8. Relatórios padrões documentam as atividades de GCA e o conteúdo das *baselines* são gerados e disponibilizados para os grupos e indivíduos afetados.
9. Auditorias nas *baselines* são conduzidas de acordo com um procedimento documentado.

Nível 3: Definido

Foco no Processo da Organização

1. O processo de software é avaliado periodicamente e, planos de ação são desenvolvidos para tratar os problemas detectados na avaliação.
2. A organização desenvolve e mantém um plano para o desenvolvimento e melhoria de seu processo de software.
3. As atividades da organização e as atividades dos projetos para desenvolvimento e melhoria de seu processo de software são coordenados em toda a organização.
4. O uso dos dados provenientes do processo de software da organização é coordenado para toda a organização.
5. Novos processos, métodos, e ferramentas de uso limitado na organização são monitorados, avaliados e, quando apropriado, transferidos para outras partes da organização.
6. Treinamentos no processo de software da organização e em processos de software definidos para os projetos são coordenados por toda a organização.
7. Os grupos que fazem uso do processo de software da organização são informados sobre projetos e atividades de aprimoramento deste processo de software e qualquer outra atividades que esteja sendo realizada para a organização.

Definição do Processo da Organização

1. Um processo de software padrão da organização é desenvolvido e mantido de acordo com um procedimento documentado.
2. O processo de software padrão da organização é documentado de acordo com padrões organizacionais estabelecidos.
3. Descrições dos ciclos de vida de software aprovados para uso em projetos são documentados e mantidos.
4. Diretrizes e critérios para elaborar projetos, baseados no processo de software padrão da organização, são desenvolvidos e mantidos.
5. Um banco de dados do processo de software da organização é estabelecido e mantido.
6. Uma biblioteca de documentos relacionados ao processo de software é estabelecida e mantida.

Programa de Treinamento

1. Cada projeto de software desenvolve e mantém um plano de treinamento que especifica suas necessidades de treinamento.
2. Um plano de treinamento da organização é desenvolvido e revisado de acordo com um procedimento documentado.
3. Os treinamentos são realizados de acordo com o plano de treinamento da organização.
4. Cursos de treinamento, em nível organizacional, são desenvolvidos e mantidos de acordo com padrões organizacionais.
5. Um procedimento de concessão para treinamentos solicitados é estabelecido e usado para determinar se os indivíduos já possuem, ou não, o conhecimento e habilidade requerida para desempenhar suas funções.
6. Registros de treinamentos realizados são mantidos.

Gerenciamento Integrado de Software

1. Os processos de software definidos para os projetos são alinhados a partir do processo de software padrão da organização de acordo com um procedimento documentado.
2. O processo de software definido para o projeto é revisado de acordo com um procedimento documentado.
3. O plano de software do projeto, que descreve o uso do processo de software definido para o projeto, é desenvolvido e revisado de acordo com um procedimento documentado.
4. O projeto de software é gerenciado de acordo com o processo de software definido para aquele projeto específico.
5. O banco de dados do processo de software da organização é usado para o planejamento e estimativas do software.
6. O tamanho dos produtos de trabalho do software, ou o tamanho das mudanças nestes produtos, é gerenciado de acordo com um procedimento documentado.
7. Os esforços e custo do projeto de software são gerenciados de acordo com um procedimento documentado.
8. Os recursos computacionais críticos do projeto são gerenciados de acordo com um procedimento documentado.
9. As dependências e caminhos críticos do cronograma do projeto de software são gerenciados de acordo com um procedimento documentado.
10. Os riscos do projeto de software são identificados, avaliados, documentados e gerenciados de acordo com um procedimento documentado.
11. Revisões periódicas são realizadas no projeto de software determinando ações necessárias para que o desempenho e os resultados do projeto estejam alinhados com as necessidades atuais e projetadas, do cliente e dos usuários finais.

Engenharia de Produto de Software

1. Ferramentas e métodos de engenharia de software apropriados são integrados ao processo de software definido para o projeto.
2. Os requisitos de software são implementados, mantidos, documentados, e verificados por uma análise sistemática dos requisitos alocados de acordo com o processo de software definido para o projeto.
3. A arquitetura do software é implementada, mantida, documentada e verificada de acordo com o processo de software definido para o projeto, incorporando os requisitos do software e estabelecendo a estrutura para a codificação.
4. O código do software é desenvolvido, mantido, documentado e verificado de acordo com o processo de software definido para o projeto, de forma a implementar os requisitos e a arquitetura do software.
5. Testes de software são realizados de acordo com o processo de software definido para o projeto.
6. Testes de integração do software são planejados e realizados de acordo com o processo de software definido para o projeto.
7. Testes de sistema e de aceitação do software são planejados e realizados para demonstrar que o software satisfaz seus requisitos.
8. A documentação, que será usada para operar e realizar manutenções no software, é desenvolvida e mantida de acordo com o processo de software definido para o projeto.
9. Dados de defeitos identificados em testes e revisões de pares são coletados e analisados de acordo com o processo de software definido para o projeto.
10. Os produtos de trabalho do software são mantidos consistentes entre si, incluindo os planos de software, descrição de processo, requisitos alocados, requisitos de software, arquitetura de software, código, planos de teste e procedimentos de teste.

Coordenação entre Grupos

1. Tanto o grupo de engenharia de software como os demais grupos de engenharia mantêm contato com os clientes e usuários finais, quando apropriado, para estabelecer os requisitos do sistema.
2. Os representantes do grupo de engenharia de software trabalham com representantes dos outros grupos de engenharia para monitorar e coordenar atividades e resolver discordâncias técnicas.
3. Um plano documentado é usado para comunicar responsabilidades entre grupos, coordenar e acompanhar o trabalho realizado.
4. Dependências críticas entre grupos de engenharia são identificadas, negociadas e acompanhadas de acordo com um procedimento documentado.
5. Os produtos de trabalho que serão insumo para outros grupos de engenharia são revisados por representantes destes grupos para assegurar que os produtos de trabalho satisfazem suas necessidades.
6. Discordância entre grupos, que não puderam ser resolvidas pelos representantes dos grupos de engenharia do projeto, são tratadas de acordo com um procedimento documentado.
7. Os representantes dos grupos de engenharia do projeto conduzem periodicamente revisões técnicas e intercâmbio de informações.

Revisão de Pares

1. As revisões de pares são planejadas e os planos documentados.
2. As revisões de pares são realizadas de acordo com um procedimento documentado.
3. Os dados sobre as condutas e resultados das revisões de pares são registrados.

Nível 4: Gerenciado

Gerenciamento Quantitativo de Processo

1. Um plano de gerenciamento quantitativo de processo é desenvolvido para todo projeto de software de acordo com um procedimento documentado.
2. As atividades de gerenciamento quantitativo de processo, para um dado projeto de software, são realizadas de acordo com o plano de gerenciamento quantitativo de processo estabelecido para aquele projeto específico.
3. A estratégia de coleta de dados e as análises quantitativas a serem realizadas no projeto são definidas no processo de software do projeto.
4. Os dados usados para controlar quantitativamente o processo de software do projeto são coletados de acordo com um procedimento documentado.
5. O processo de software definido para o projeto é analisado e mantido sob controle quantitativo de acordo com um procedimento documentado.
6. São preparados e distribuídos relatórios documentando os resultados das atividades do gerenciamento quantitativo de processo do projeto de software.
7. A *baseline* da capacidade do processo de software padrão da organização é estabelecida e mantida de acordo com um procedimento documentado.

Gerenciamento da Qualidade de Software

1. O plano de qualidade de software do projeto é desenvolvido e mantido de acordo com um procedimento documentado.
2. O plano de qualidade de software do projeto é a base para as atividades de gerenciamento da qualidade de software do projeto.
3. As metas quantitativas de qualidade dos produtos de software do projeto são definidas, monitoradas e revisadas ao longo do ciclo de vida do software.
4. A qualidade dos produtos de software do projeto, em momentos pré estabelecidos, é medida, analisada e comparada com as metas quantitativas de qualidade dos produtos.

5. As metas quantitativas de qualidade de software do projeto são alocadas apropriadamente aos produtos de software subcontratados do projeto.

Nível 5: Otimizando

Prevenção de Defeitos

1. O projeto de software desenvolve e mantém um plano para atividades de prevenção de defeitos.
2. No início dos esforços para se realizar uma determinada tarefa de um projeto de software, a equipe encarregada desta tarefa participa de uma reunião ser preparada em relação às atividades relacionadas com a tarefa e com a prevenção de defeitos.
3. Reuniões para análises de causas são conduzidas de acordo com um procedimento documentado.
4. Todas as equipes designadas para coordenar as atividades de prevenção de defeitos se reúnem periodicamente para rever e coordenar implementações de ações propostas nas reuniões de análises de causas.
5. Dados relacionados com prevenção de defeitos são documentados e acompanhados pelas equipes designadas para coordenar atividades de prevenção de defeitos.
6. As revisões no processo de software padrão da organização, resultantes de ações de prevenção de defeitos, são incorporadas de acordo com um procedimento documentado.
7. As revisões em processos de software definidos para um projeto, resultantes de ações de prevenção de defeitos, são incorporadas de acordo com um procedimento documentado.
8. Os membros do grupo de engenharia de software e demais grupos envolvidos com processo de software recebem informações periódicas do estado e dos resultados das atividades de prevenção de defeitos tanto da organização como dos projetos.

Gerenciamento de Mudança Tecnológica

1. A organização desenvolve e mantém um plano para o gerenciamento de mudança tecnológica.
2. O grupo responsável pelas atividades de gerenciamento de mudança tecnológica da organização trabalha com os projetos de software para identificar áreas passíveis de mudanças tecnológicas.
3. Os gerentes de software e o corpo técnico são mantidos informados sobre as novas tecnologias.
4. O grupo responsável por gerenciar a mudança tecnológica da organização analisa sistematicamente o processo de software padrão da organização para identificar áreas que precisam, ou poderiam se beneficiar, de novas tecnologias.
5. Novas tecnologias são selecionadas e adquiridas para a organização e para projetos de software de acordo com um procedimento documentado.
6. Antes de introduzir uma nova tecnologia como prática normal dentro da organização, projetos pilotos são conduzidos quando apropriado.
7. As novas tecnologias apropriadas são incorporadas ao processo de software padrão da organização de acordo com um procedimento documentado.
8. Novas tecnologias apropriadas são incorporadas ao processo de software de um projeto de acordo com um procedimento documentado.

Gerenciamento de Mudança de Processo

1. Um programa de melhoria de processo de software é estabelecido para encorajar os membros da organização a melhorar os processos da organização.
2. O grupo responsável pelas atividades do processo de software da organização coordena as atividades de melhoria do processo de software.
3. A organização desenvolve e mantém um plano para a melhoria do processo de software de acordo com um procedimento documentado.
4. As atividades de melhoria do processo de software são realizadas de acordo com o plano de melhoria do processo de software.

5. Propostas de melhoria no processo de software são tratadas de acordo com um procedimento documentado.
6. Os membros da organização participam ativamente em grupos para desenvolver melhorias no processo de software em áreas específicas de processo.
7. Onde for apropriado, a melhoria no processo de software é introduzida como piloto para determinar seus benefícios e eficiência antes de ser implantada como prática normal da organização.
8. Quando se toma a decisão de implantar a melhoria no processo de software como uma prática normal da organização, esta é implementada de acordo com um procedimento documentado.
9. São mantidos registros das atividades de melhoria de processo de software.
10. Em momentos apropriados, os gerentes de software e o pessoal técnico recebem retorno do estágio e resultados das atividades de melhoria de processo de software.