



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-10424-TDI/921

**LEVANTAMENTO DA QUALIDADE DO PROCESSO DE
SOFTWARE COM FOCO EM PEQUENAS ORGANIZAÇÕES**

Jaciara Silva Carosia

Dissertação de Mestrado do Curso da Pós-Graduação em Computação Aplicada,
orientada pelo Dr. Tatu Nakanishi, aprovada em 30 de setembro de 2003.

681.3.06

CAROSIA, J. S.

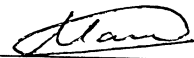
Levantamento da qualidade do processo de software com foco em pequenas organizações / J. S. Carosia. – São José dos Campos: INPE, 2003.

158p. – (INPE-10424-TDI/921).

1.Engenharia de software. 2.Melhoria. 3.Processos. 4.Qualidade. 5.Avaliações. I.Título.

Aprovada pela Banca Examinadora em
cumprimento a requisito exigido para a
obtenção do Título de **Mestre** em
Computação Aplicada.

Dr. Maurício Gonçalves Vieira Ferreira



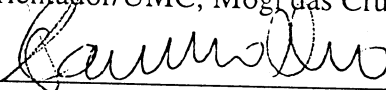
Presidente/INPE, SJCampos-SP

Dr. Tatu Nakanishi



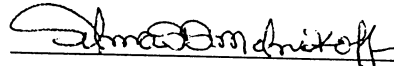
Orientador/UMC, Mogi das Cruzes-SP

Dr. Solon Venâncio de Carvalho



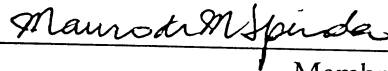
Membro da Banca/INPE, SJCampos-SP

Dr^a Selma Shin Shimizu Melnikoff



Membro da Banca
Convidada POLI/USP, São Paulo-SP

Dr. Mauro de Mesquita Spinola



Membro da Banca
Convidado USP, São Paulo-SP

Candidata: Jaciara Silva Carosia

São José dos Campos, 30 de setembro de 2003.

*“Para que haja luz não bastará
temer a presença da sombra. É
preciso acendê-la”.*

André Luiz
Livro "Estude e Viva"
Psicografado por Chico Xavier

A meus avós – Pedro e Lula –
pelo apoio e incentivo
em todos os momentos de minha vida,
mas que infelizmente não puderam
presenciar a conclusão de mais esta etapa.
Dedico a vocês onde quer que estejam.

AGRADECIMENTOS

A Deus.

A meus pais, Mauro e Regina, pelos incansáveis esforços por minha formação, pelo apoio incondicional e incentivo.

A meu namorado e grande amigo, Fabrício, pela grande paciência e companheirismo em todos os momentos.

A meu irmão, Mauro Junior, também pela paciência nos momentos mais estressantes.

A meu orientador, Prof. Dr. Tatuó, que com sua dedicação e confiança é o grande responsável pela viabilização deste trabalho.

Às amigadas que conquistei no decorrer deste trabalho.

Aos profissionais da área que colaboraram com suas experiências.

RESUMO

A sociedade passa por severas transformações. A capacidade de inovar é o diferencial para acompanhar o ritmo acelerado destas mudanças. Muitas organizações de software apostam na qualidade como alternativa de inovação. Melhorar a qualidade do processo de desenvolvimento de software é um dos grandes objetivos da engenharia de software. A avaliação é a primeira etapa para a melhoria do processo. O sucesso de um programa de melhoria depende grandemente da confiabilidade dos dados adquiridos nesta etapa. Uma avaliação eficiente permite a elaboração de um plano de melhoria compatível com a realidade da organização. Este trabalho apresenta uma estratégia para levantamento da qualidade do processo, através de uma mini-avaliação baseada no modelo CMMI. O foco específico deste trabalho é organizações de pequeno porte que estão dando seus primeiros passos em direção a melhoria de seu processo de desenvolvimento.

SOFTWARE PROCESS QUALITY ASSESSMENT FOCUSED ON SMALL ORGANIZATIONS

ABSTRACT

Society goes through strict changes. The capacity of innovation is what makes the difference in order to follow the accelerate rhythm of these changes. Many software organizations invest on this quality as their innovation alternative. Improving the quality of the development process is one of the software engineering biggest goals. Assessment is the first step in order to improve the process. The success of an improvement program depends widely on the reliability of the acquired data on this stage. An efficient assessment allows the elaboration of a plan in compliance with the organization reality. This work presents a strategy to assessment the quality of the process, through a mini-assessment based on the model CMMI. The specific focus of this work is the small organizations that are walking the first steps towards the improvement of their development process.

SUMÁRIO

	Pág.
LISTA DE FIGURAS	
LISTA DE TABELAS	
CAPÍTULO 1 – INTRODUÇÃO.....	19
CAPÍTULO 2 – QUALIDADE.....	23
2.1 - Origem da Qualidade.....	23
2.2 - Definições.....	23
2.3 - Qualidade de Software.....	27
2.4 - Qualidade do Produto e Qualidade do Processo.....	28
2.4.1 - Qualidade do Produto.....	29
2.4.2 - Qualidade do Processo.....	31
CAPÍTULO 3 - MODELOS DE MELHORIA DO PROCESSO DE SOFTWARE.....	33
3.1 - Modelo de Maturidade da Capacidade.....	34
3.1.1 - Origem do CMM.....	34
3.1.2 - Maturidade.....	35
3.1.3 - Estrutura do CMM.....	36
3.1.4 - Implementação do CMM.....	38
3.2 - Modelo de Maturidade da Capacidade Integrado.....	39
3.2.1 - Estrutura do CMMI.....	41
3.2.2 - Componentes do Modelo.....	44
3.2.3 - Abordagem em Estágios.....	47
3.2.4 - Abordagem Contínua.....	52
3.2.5 - Perspectivas de Implementação do CMMI.....	54
CAPITULO 4 - PANORAMA ATUAL.....	57
4.1 – A Indústria do Software.....	57

4.2 - Os Processos de Software.....	59
4.3 - A Utilização do CMM.....	61
4.4 - O CMM e o Porte das Organizações.....	65
4.5 - Custos e Benefícios do CMM.....	68
4.6 – CMM em Pequenas Organizações.....	71
CAPITULO 5 – MELHORIA DO PROCESSO DE SOFTWARE.....	75
5.1 - Diretrizes Iniciais.....	75
5.2 - Escolha do Escopo da Organização.....	76
5.3 - Escolha do Modelo de Melhoria.....	78
5.4 - Escolha da Abordagem.....	78
5.5 - Escolha da Perspectiva.....	80
5.6 - Escolha da Disciplina.....	80
5.7 - Escolha das Áreas de Processo.....	81
5.8 – Importância do Gerenciamento de Projetos.....	88
CAPÍTULO 6 – LEVANTAMENTO DA QUALIDADE DO PROCESSO DE SOFTWARE.....	93
6.1 - Avaliações CMMI.....	93
6.2 - Avaliações Alternativas.....	95
6.3 - Preparação da Avaliação.....	97
6.4 - Método para Mini-avaliação.....	98
6.5 - Requisitos de Avaliações Classe C.....	100
6.6 - Interpretação dos Requisitos.....	107
6.7 - Questionário de Maturidade.....	113
6.8 - Uma Proposta de Questionário.....	115
6.8.1 - Papéis.....	116
6.8.2 - Práticas Básicas.....	119
6.8.3 - Metodologia para criação dos questionários.....	120

CAPÍTULO 7 – CONCLUSÃO	125
7.1 - Considerações sobre o Desenvolvimento.....	127
7.2 - Dificuldades Encontradas.....	129
7.3 - Considerações Finais.....	130
REFERÊNCIAS BIBLIOGRÁFICAS	133
APÊNDICE A	141
APÊNDICE B	149

LISTA DE FIGURAS

3.1 - Estrutura do CMMI em estágios.....	48
3.2 - Estrutura do CMMI contínuo.....	53
4.1 - Conhecimento do Modelo CMM, Norma ISO 12207 e SPICE.....	62
4.2 - Conhecimento no Modelo CMM (%).....	63
4.3 - Porte das empresas quanto a força de trabalho efetiva e total.....	66
6.1 - Avaliação CMMI.....	93
6.2 - Limites da avaliação.....	120
6.3 - Esquema para elaboração dos questionários.....	121

LISTA DE TABELAS

2.1 - Enfoques Antigo e Novo da Qualidade.....	24
2.2 – Características de qualidade internas e externas.....	30
2.3 - Características de qualidade em uso.....	30
4.1 - Conhecimento dos modelos CMM e SPICE – 1999.....	62
4.2 - Conhecimento dos modelos CMM e SPICE – 2001.....	63
4.3 - Conhecimento do CMM segundo o porte da organização (%).....	65
4.4 - Implantação de programas de qualidade (%).....	67
5.1 Áreas de processo do Nível 2.....	84

CAPÍTULO 1

INTRODUÇÃO

A frase de Georges Clemenceau (1841-1920), estadista e jornalista francês, *L'homme absurde est celui qui ne change jamais* (Um homem absurdo é aquele que não muda nunca), também pode ser empregada no contexto das organizações. O dicionário Aurélio define absurdo como contrário à razão, ao bom senso; que fere as regras da lógica ou as leis da razão (Ferreira, 1999). Pode-se dizer que uma organização que não se preocupa em mudar, em acompanhar a dinâmica acelerada que a sociedade imprime, também é uma organização absurda.

Não é novidade o fato de que a sociedade mundial passa por severas transformações de cunho político, econômico, cultural e tecnológico. A velocidade impressa a esta dinâmica transformadora, é algo, há apenas alguns anos atrás, praticamente impensável. Para que a velocidade das transformações não cause terríveis danos às organizações, é necessário que a inovação (Bastos, 2000) esteja cada vez mais presente como um valor importante na cultura destas organizações.

A globalização, como exemplo claro destas transformações, mudou clássicos conceitos de administração. Hoje, não é admissível permanecer indiferente a esta realidade, sob pena de colocar em risco o patrimônio da organização. Conseguir acompanhar o ritmo das mudanças é um critério indispensável para garantir competitividade. As organizações conseguem e mantêm vantagem competitiva por meio da melhoria, da inovação e do aperfeiçoamento (Porter, 1993).

Dentro do contexto destas transformações, o diferencial é a capacidade de inovar. As organizações que desenvolvem software, obviamente, são também afetadas por estas transformações. Porém, devem estar mais atentas, visto que os seus produtos - os softwares - são potenciais alvos de contínuas evoluções. Estas organizações estão

apostando na qualidade como ferramenta de inovação. No panorama atual, não é exagero afirmar que o investimento em qualidade é questão de sobrevivência.

Os esforços pela qualidade na indústria automobilística provaram que o aumento da qualidade é sempre acompanhado pela redução de custos na forma de menos re-trabalho e no menor índice de refugo. Além da maior satisfação do cliente que pode ser refletida, muitas vezes, na maior participação no mercado. Assim, o axioma que diz que investimentos em qualidade dão lucro, é plenamente aceito pela indústria de manufatura. As organizações apostam que este axioma também funciona com software.

O interesse pela qualidade já não é novidade no setor industrial, visando o aumento da produtividade e, ao mesmo tempo, a satisfação das crescentes exigências dos consumidores. No setor do software também é notável este interesse. Com as progressivas exigências do mercado, várias iniciativas têm surgido, objetivando a elaboração de normas e padrões internacionais para melhoria da qualidade do software, do ponto de vista do produto e do processo de desenvolvimento.

As crescentes demandas dos clientes e usuários, a sempre maior complexidade da tecnologia disponível e as exigências de maior confiabilidade, menor custo e prazo, e maior produtividade estão levando as organizações a se convencerem que a abordagem artesanal no desenvolvimento e manutenção do software tem os dias contados.

São vários os modelos disponíveis que auxiliam na busca pela qualidade do software, porém ainda há certa resistência por parte das organizações em implementá-los.

Um dos fatores desta resistência são certas reações defensivas (Argyris, 1992), nem sempre conscientes, de empecilho às inovações. Estas reações, como alienação e oposição, são conseqüências do receio inerente a qualquer mudança (Pereira, 1997). Este receio é uma característica do ser humano, que pode ser superada através do esclarecimento dos benefícios da mudança. A conscientização de que algumas mudanças são imprescindíveis para melhorar a produtividade e a qualidade do produto desenvolvido, é um passo decisivo para superar possíveis mitos.

Outros fatores de resistência à adoção aos modelos de melhoria são mais complexos e de difíceis soluções. Algumas organizações apresentam certas carências que não podem ser supridas apenas com esclarecimentos. Recursos administrativos, financeiros e humanos, quando ineficazes ou insuficientes, apresentam-se como uma grande barreira à busca da qualidade. A implementação destes modelos não é uma tarefa trivial, mas em pequenas organizações, esta tarefa pode ser ainda mais difícil, devido à carência de recursos.

A importância da indústria do software para a economia nacional, a relevância das pequenas organizações de software e a crescente importância da qualidade no mercado competitivo, motivou o desenvolvimento de um trabalho para auxiliar na aproximação destas organizações aos princípios de qualidade de software.

Em qualquer programa de melhoria, a primeira atividade a ser executada é a avaliação do processo atual, pois sem conhecer algo é impossível melhorá-lo. A avaliação inicial, muitas vezes, torna-se uma barreira intransponível para muitas organizações.

No contexto das pequenas organizações que desenvolvem software, este trabalho tem como objetivo esclarecer algumas questões como: As organizações de software brasileiras estão se preocupando com a melhoria do seu processo de desenvolvimento? Quais as maiores dificuldades encontradas na implementação de um programa de melhoria de processo? É possível melhorar a qualidade do processo em organizações que dispõem de poucos recursos? Quais os primeiros passos a serem dados na busca pela qualidade do processo de software? Qual modelo de melhoria utilizar? Quais as diretrizes para a elaboração de uma avaliação inicial?

Assim, o presente trabalho descreve algumas diretrizes para a criação de uma avaliação de processo inicial, visando minimizar as dificuldades que as pequenas organizações encontram em seus primeiros passos em direção a melhoria da qualidade de software. Esta proposta é baseada no CMMI (CMM Integrado), que é um modelo de melhoria de

processo, cujas características se mostram atraentes a organizações que aspiram sair do estado de imaturidade no desenvolvimento de software.

Este trabalho está organizado em 7 capítulos.

No Capítulo 2, são dadas definições de qualidade e qualidade de software, onde são mencionadas as características da qualidade do produto e qualidade de processo, com ênfase especial nesta última. A clara compreensão do termo qualidade é imprescindível para que esta possa ser buscada e assuma função estratégica na competitividade.

O Capítulo 3 é composto de uma breve apresentação do CMM (*Capability Maturity Model*) e da sua mais recente evolução, o CMMI (*Capability Maturity Model Integration*). Ambos são modelos voltados à melhoria do processo de software e foram utilizados como base no desenvolvimento deste trabalho.

No Capítulo 4, são mostrados o panorama atual do desenvolvimento de software, a aceitação e disseminação do CMM e CMMI, e as principais dificuldades que algumas organizações enfrentam na implementação de um modelo de melhoria de processo.

Nos Capítulos 5 e 6, é proposta uma estratégia para preparação e execução de uma mini-avaliação baseada no modelo CMMI.

No Capítulo 5, são feitas algumas escolhas visando a adaptação do modelo CMMI ao escopo da organização enfocada neste trabalho. Estas escolhas compreendem o primeiro passo para a elaboração de uma avaliação de processo.

No Capítulo 6, é apresentada uma estratégia para que a organização descreva seu próprio método de levantamento da qualidade do processo de software.

Finalmente, no Capítulo 7, são dadas as conclusões sobre o desenvolvimento deste trabalho.

CAPÍTULO 2

QUALIDADE

2.1 – Origem da Qualidade

O desenvolvimento contínuo da qualidade começou no Japão, mas foi inspirada em dois americanos, Edward Deming e Joseph Juran. Muitas pessoas reconheceram que após a Segunda Guerra Mundial, para o Japão se reconstruir e vir a ser uma grande nação comerciante, um intenso programa de desenvolvimento contínuo da qualidade seria necessário. Os produtos japoneses estavam longe da competitividade na qualidade e confiabilidade e eram vendidos apenas à base de baixos preços. Então, trabalhando com altos administradores no Japão, Deming e Juran, executando métodos de controle estatístico da qualidade, aplicando em todas as áreas produtivas da companhia, deram as condições para se fazer os produtos japoneses mais competitivos. As idéias difundiram-se amplamente e, com este impulso, o Japão liderou os programas de controle total da qualidade e a idéia de melhoria contínua da mesma, sendo fonte de inspiração de muitas nações (Vieira, 1995).

Muitos nipônicos consideram Deming e Juran os inspiradores do milagre industrial japonês iniciado na década de 50. Sobre suas idéias foi criada uma revolução da qualidade que restabeleceu a confiança na indústria nacional (Cardoso, 1995).

Na década de 80 o fator qualidade emergiu mundialmente como uma necessidade básica na luta pelo mercado cada vez mais competitivo (Barros, 1999).

2.2 – Definições

A definição do termo qualidade depende do âmbito em que ocorre, possuindo interpretações diversas. Por isso mesmo, a qualidade continua facilmente mal entendida com decorrências que prejudicam as organizações que desejam implementá-la. Uma

melhor compreensão do termo é fundamental para que a qualidade possa assumir função estratégica na competitividade. Na formulação de um conceito de qualidade é essencial a clareza de uma definição, mas o fundamental é que seu significado esteja perfeitamente entendido e que seja uma linguagem comum em toda a organização.

Juran (1990) admite a existência de várias definições para o termo qualidade. Segundo ele, um dos significados de qualidade é o “desempenho do produto”, este desempenho resulta das características do produto que levam à satisfação do cliente e interferem na decisão de compra. Ou ainda, qualidade é a “ausência de deficiências”, as deficiências do produto afetam os custos, por falhas no uso, repetição de trabalho e desperdício. A conveniência de unir estas duas definições levou a conhecida definição simples de qualidade como sendo a “adequação ao uso”. Apesar de ser considerada pelo próprio autor como uma definição muito resumida, este tem sido o conceito mais reconhecido para a qualidade. A abordagem de Juran possui um forte ingrediente gerencial e focaliza planejamento, fluxo organizacional, responsabilidade gerencial para qualidade e a necessidade de estabelecer metas e objetivos para melhoria (Cardoso, 1995). A gerência para a qualidade tem passado ao longo dos anos por mudanças profundas. Juran (1992) lidera uma nova fase para o conceito de qualidade, que passou dos aspectos tecnológicos da fábrica para a preocupação com a qualidade global em todos os aspectos do gerenciamento e em toda a organização. A Tabela 2.1 ilustra estas mudanças.

TABELA 2.1 - Enfoques Antigo e Novo da Qualidade.

Enfoque Antigo	Enfoque Novo
Qualidade por inspeção	Qualidade fabricada
Qualidade maior aumenta custos	Qualidade maior diminui custos
Orientada para o procedimento	Orientada para o processo
Induzida pelos departamentos	Induzida pela liderança
Atende especificações do cliente	Atende e excede expectativas do cliente
Foco no chão de fábrica	Foco holístico em toda organização
Qualidade é obrigação de outro	Qualidade é minha obrigação

FONTE: Juran (1992).

Deming (1990) também reconhece a amplitude do conceito de qualidade que só pode ser definida em termos de quem a avalia. Ele afirma que qualidade não significa luxo, mas é um grau previsível de uniformidade e dependência, a baixo custo, adequada ao mercado. Em outras palavras, qualidade é qualquer coisa que o cliente necessita ou deseja. E como as necessidades e desejos dos clientes estão sempre mudando, a solução para a definição da qualidade em termos de cliente é redefinir os requisitos constantemente. Deming enfatiza uma sistemática para solução de problemas de qualidade conhecida por Ciclo Deming ou *PDCA – Plan, Do, Check, Action*. Este método de análise sistemática é a base para o processo de melhoria contínua em todos os níveis da organização que deseja qualidade. No mundo de Deming, você nunca chega. Você sempre pode fazer melhor e nunca deve interromper o processo de aperfeiçoamento.

Para salientar a pluralidade de definições para o termo qualidade, pode-se destacar a opinião de outros consultores reconhecidos internacionalmente por seus trabalhos desenvolvendo técnicas ou teorias em qualidade e produtividade.

Crosby (1991) define qualidade como a conformidade com os requisitos e é medida pelo custo da não-qualidade. Para ele qualidade é um estado binário: ou há conformidade (qualidade) ou há não-conformidade (não-qualidade). Usando esta abordagem, Crosby desenvolveu, em 1961, o conceito de "zero defeito", enfatizando que todas as pessoas da organização são capazes de fazer o seu trabalho de maneira correta, na primeira e em todas as vezes.

Teboul (1991) associa o conceito de qualidade a três elementos básicos: o cliente, a oferta e a concorrência. Este autor afirma que qualidade é, antes de qualquer coisa, a conformidade às especificações. É também a resposta ajustada à utilização que se tem em mente, na hora da compra e também a longo prazo. Mas é também aquele “algo mais” de sedução e excelência, mais próximo do desejo do que da qualidade. Nesta definição, consolidam-se várias proposições de qualidade que, de acordo com Teboul, não são exclusivas. Para ele, ter conformidade é necessário, mas em relação às

necessidades ou a um certo uso. E esta satisfação das necessidades deve ser feita de maneira superior à da concorrência, com algo mais de sedução (Vieira, 1995).

Garvin (2002) destaca a necessidade de um melhor entendimento do termo qualidade, para que ela possa realmente assumir um papel estratégico. Ele afirma que estudiosos de filosofia, economia, marketing e gerência de operações têm visto a qualidade sob aspectos diferentes. A filosofia concretiza-se nas questões de definição; a economia, na maximização dos lucros e no equilíbrio do mercado; o *marketing*, nos elementos críticos determinantes do comportamento dos compradores e na satisfação dos clientes, e a gerência de operações, nas práticas de engenharia e no controle da produção.

Baseando-se nestes diferentes enfoques, Garvin identifica cinco abordagens principais para a definição de qualidade (Viera,1995):

- Transcendental: sinônimo de excelência nata.
- Baseada no produto: que identifica a qualidade com os atributos e/ou ingredientes de um produto.
- Baseada no cliente: que parte da premissa que a qualidade está diante dos olhos de quem observa.
- Baseada na fabricação: a qualidade é vista como conformidade às especificações.
- Baseada no valor: definindo qualidade em termos de custos e preços.

Este mesmo autor também identifica oito dimensões ou categorias de qualidade como esquema de análise da qualidade: desempenho, características, confiabilidade, conformidade, durabilidade, atendimento, estética e qualidade percebida. Garvin afirma que estas categorias são estanques e distintas entre si, o que permite que um produto ou serviço com alta qualidade em uma dimensão seja mal classificado em outra. Porém, em muitos casos, as dimensões estão inter-relacionadas, pois a melhoria de qualidade em uma dimensão, só pode ser atingida às custas de outra (Vieira,1995).

Estes são alguns dos principais conceitos associados à qualidade, descritos sob os mais diversos pontos de vista, embora não se tenha esgotado o assunto. É válido salientar a necessidade de se escolher uma definição que supra com as necessidades do contexto (ou do ambiente) onde será empregada.

2.3 - Qualidade de Software

O software ultrapassou o hardware como a chave para o sucesso de muitos sistemas baseados em computador. Seja o computador usado para dirigir um negócio, controlar um produto ou capacitar um sistema, o software é um fator que diferencia. A inteireza e a oportunidade das informações oferecidas pelo software diferenciam uma empresa de suas concorrentes. A inteligência e a função oferecidas pelo software muitas vezes diferenciam dois produtos de consumo ou indústrias idênticas.

Com os avanços da informática cada vez mais os computadores passam a integrar a rotina diária e a produção de software vem sofrendo um constante aumento. Os sistemas baseados em computador estão sendo usados nas mais diversas áreas e na maioria das situações não admitem erros. Se algum sistema de uso global deixar de funcionar, aproximadamente 40% da população mundial (Reed, 2000), sofrerá as conseqüências do problema.

Considerando a importância do software na atualidade, aliada às crescentes exigências dos clientes, tem-se verificado um notável aumento nos interesses pela qualidade do software. Melhorar a qualidade do software é o principal objetivo da engenharia de software.

O foco principal da maioria das definições de qualidade de software é o atendimento às necessidades do cliente. Muitas vezes aplica-se à qualidade de software a definição de Crosby (1991) que é a conformidade com os requisitos. Esta definição resume conceitos de várias personalidades que se destacam na engenharia de software (Kan, 1995) (Pressman, 1995).

Existem duas visões de qualidade de software, uma dos clientes e outra dos que desenvolvem o software, mas ambas concordam que o software não pode ter defeitos. O cliente avalia o software sem conhecer seus aspectos internos, está apenas interessado na facilidade do uso, no desempenho, na confiabilidade dos resultados obtidos e também no preço do software. Os que desenvolvem o software avaliam aspectos internos como taxa de defeitos, confiabilidade, facilidade de manutenção e também aspectos de conformidade em relação aos requisitos dos clientes (ISO, 1991).

Em ambas visões, os requisitos de software formam a base de onde a qualidade é avaliada. Pode-se afirmar que a falta de conformidade com os requisitos é falta de qualidade. Além dos requisitos, existem padrões específicos que definem os critérios de desenvolvimento que servem como diretriz para o software que está sendo produzido. Se os critérios não são seguidos, resulta na falta de qualidade (Pressman, 1995).

Pode-se dizer que, um software de boa qualidade produz resultados úteis e confiáveis na oportunidade certa; é ameno ao uso, é mensurável, corrigível, modificável e evolutivo; opera em máquinas e ambientes reais; foi desenvolvido de forma econômica e no prazo estipulado; e opera com economia de recursos. Qualidade de software é um conceito muito mais amplo do que o de um software correto e bem documentado, requerendo para ser conseguida, metodologias e técnicas de desenvolvimento específicas (Staa, 1987).

2.4 - Qualidade do Produto e Qualidade do Processo

Raramente a qualidade pode ser incorporada ao produto final após o processo de desenvolvimento terminado. Portanto, a qualidade do produto de software é um objetivo do processo de desenvolvimento. Assim, ao se desenvolver um produto, é importante que se tenha previamente definidas as características de qualidade que se desejam alcançar. Se o processo de desenvolvimento de software considerar as características de qualidade, há uma grande tendência de que o produto final apresente tais características.

Apesar de serem conceitos integrados, e na maioria das vezes dependentes, pode-se fazer um estudo separado dos conceitos de qualidade de produto e qualidade de processo de software.

2.4.1 - Qualidade do Produto

A qualidade do produto 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. Cada característica pode ser detalhada chegando-se a um amplo conjunto de atributos, que descrevem a qualidade do produto de software. Assim, é necessária a escolha de um modelo que organize os atributos e permita avaliar a qualidade do software (Rocha, 2001).

Vários modelos surgiram apresentando uma hierarquia de características que contribuem para a qualidade do produto. A norma ISO/IEC 9126 (1999) surgiu para consolidar as diferentes visões da qualidade de produto em um modelo como norma internacional. Este modelo está dividido em duas partes: modelo de qualidade para características externas e internas e modelo da qualidade para qualidade em uso.

O modelo de qualidade para características externas e internas classifica os atributos de qualidade em seis características, que são descritas brevemente na Tabela 2.2.

TABELA 2.2 – Características de Qualidade Internas e Externas.

Funcionalidade	Existência de um conjunto de funções, que satisfazem as necessidades explícitas e implícitas e suas propriedades específicas.
Confiabilidade	Capacidade de o software manter seu nível de desempenho, sob condições estabelecidas, por um período de tempo.
Usabilidade	Esforço necessário para usar um produto de software, bem como o julgamento individual de tal uso por um conjunto de usuários.
Eficiência	Relacionamento entre o nível de desempenho do software e a quantidade dos recursos utilizados sob as condições estabelecidas.
Manutenibilidade	Esforço necessário para fazer modificações específicas no software.
Portabilidade	Capacidade de o software ser transferido de um ambiente para outro.

FONTE: ISO/IEC 9126 (1999).

No modelo de qualidade em uso, os atributos são classificados em quatro características, que são mostradas na Tabela 2.3. A qualidade em uso é a capacidade de o produto de software permitir a determinados usuários atingir metas específicas com efetividade, produtividade, segurança e satisfação em um contexto de uso determinado.

TABELA 2.3 – Características de Qualidade em Uso.

Efetividade	Capacidade de possibilitar aos usuários atingir metas especificadas com acurácia e completeza.
Produtividade	Capacidade de possibilitar aos usuários utilizar uma quantidade adequada de recursos em relação à efetividade.
Segurança	Oferecer níveis aceitáveis de risco de dados a pessoas, negócios, software, propriedade ou ao ambiente.
Satisfação	Capacidade de satisfazer usuários.

FONTE: ISO/IEC 9126 (1999).

2.4.2 - Qualidade do Processo

Um processo de software é um conjunto de ferramentas, métodos e práticas usadas para produzir software. O processo de software é representado por um conjunto seqüencial de atividades, objetivos, transformações e eventos que integram estratégias para cumprimento da evolução de software (Pressman, 1995).

Em meados da década de 80, surgiu um movimento focalizando os processos de desenvolvimento de software. Uma das maiores evoluções no estudo da qualidade foi a verificação de que a qualidade do produto é importante, mas a qualidade do processo de produção tem uma importância ainda maior. As falhas nos processos de gerência e manutenção do desenvolvimento de software foram reconhecidas como inibidores principais no crescimento da qualidade e produtividade (Lamprech, 1997). O processo utilizado no desenvolvimento de um projeto tem grande reflexo na produtividade e na qualidade do software desenvolvido.

Após anos de estudos e promessas de solucionar problemas de tecnologia com mais tecnologia, surge um movimento de compreensão de que os problemas de desenvolvimento de software não são apenas tecnológicos, mas gerenciais e organizacionais.

O processo utilizado para desenvolver e manter o software afeta significativamente no custo, qualidade e prazo de entrega do produto. O impacto é tão significativo que a melhoria do processo de software é vista por alguns como a mais importante forma para melhorar o produto de software (Henry, 1994).

Segundo Humphrey, para que a indústria do software contribua de forma construtiva para a sociedade, é necessário aprender a entregar produtos com qualidade, no prazo estabelecido e com os custos planejados. Outras indústrias, à medida que

amadureceram, atingiram este nível de desempenho. Não há razões para que isso não seja possível para o software (Weber, 2001).

Um processo disciplinado faz com que o desenvolvimento e a manutenção sejam indistinguíveis em um ciclo de evolução do produto. Para que o desenvolvimento e a manutenção sejam feitos de maneira ordenada, as organizações devem ter seus processos como sendo passíveis de controle, melhorias e medidas. Isto requer programas voltados para o aumento da produtividade e da qualidade. Antes de qualquer tentativa de uma organização aperfeiçoar seu processo, deve-se conhecer como o software é desenvolvido. O conhecimento de onde o esforço deve ser concentrado é essencial. Sem o conhecimento do tipo de processo existente e os produtos característicos produzidos durante o desenvolvimento, não há meio de aperfeiçoar ou quantificar a qualidade do produto (Kan, 1995).

Considerar o processo de desenvolvimento uma atividade que deve ser controlada, medida e melhorada é um passo importante para torna-lo realmente efetivo. Para tanto, é indispensável que se tenha o controle do relacionamento entre as atividades, as ferramentas, o pessoal, os métodos e principalmente as características do produto de software que é o objetivo final de todo o processo (Rosa, 1997).

A importância da qualidade no setor de software é indiscutível. Assim, muitas instituições preocuparam-se em criar modelos para auxiliar na busca à tão almejada qualidade tanto do produto quanto do processo de software.

CAPÍTULO 3

MODELOS DE MELHORIA DO PROCESSO DE SOFTWARE

Como a qualidade do produto pode ser vista como uma consequência da qualidade do processo, os certificados mais valiosos são aqueles que certificam o processo de produção de um produto e não os que certificam simplesmente o produto. Os estudos sobre qualidade, mais recentemente, são voltados para o melhoramento do processo de software, pois ao garantir a qualidade do processo, já se está dando um grande passo para garantir também a qualidade do produto.

Um importante objetivo das organizações é melhorar sistematicamente a qualidade e a produtividade de seus processos e produtos. Por sistemático entende-se que a organização pode descrever explicitamente seu processo e com base em medições confiáveis, analisar sua eficácia, assim como sugerir melhorias (Pfleeger, 1994).

Com a intensa dinâmica que rege os processos de software, a definição explícita de um processo já não é uma tarefa trivial, gerencia-los, então, é ainda mais difícil. Para amenizar esta dificuldade, foi necessária a definição de modelos de apoio a melhoria da qualidade do processo de software.

Entre os modelos disponíveis, os que mais se destacam são: a norma ISO/IEC 12207 - Processos de Ciclo de Vida de Software (ISO, 1995); futura norma ISO/IEC 15504 ou projeto SPICE - *Software Process Improvement and Capability Determination* (ISO, 1997), e o CMM - *Capability Maturity Model* (Paulk, 1993).

No desenvolvimento deste trabalho, o modelo CMM, em sua mais nova evolução, o CMMI, foi utilizado como base para delinear os pontos do processo a serem avaliados. Tais pontos referem-se às características básicas de um processo qualificado.

3.1 – Modelo de Maturidade da Capacidade

Em linhas gerais, o Modelo de Maturidade da Capacidade ou CMM (*Capability Maturity Model*) é um modelo que baseia em um processo gradual, que leva as organizações a se aprimorarem continuamente, na busca de soluções próprias para os problemas existentes no desenvolvimento de software.

3.1.1 – Origem do CMM

O CMM (Paulk, 1993) foi desenvolvido pelo SEI (*Software Engineering Institute*), ligado a Universidade Carnegie Mellon e financiado pelo Departamento de Defesa Norte-Americano, com o objetivo de estabelecer um padrão de qualidade para o software desenvolvido para as Forças Armadas. Este modelo foi proposto por Watts S. Humphrey, a partir de conceitos de qualidade total estabelecido por Crosby. Segundo Tingey (1997), Crosby mostrou que a implementação de sistemas de qualidade em empresas segue um amadurecimento gradativo em patamares denominados “incerteza, despertar, esclarecimento, sabedoria e certeza”.

Em 1987, a partir do trabalho de Humphrey, o SEI lançou uma breve descrição de um ambiente de maturidade de processo de software e um questionário de maturidade para avaliar o estado corrente das práticas de software e identificar as áreas que necessitavam de melhoria. Em 1991, o SEI criou o CMM, como uma evolução deste ambiente. A partir do lançamento e divulgação da versão 1.1 do CMM, em 1993, o tema de melhoria do processo foi ganhando força. Esta força foi consequência dos resultados práticos obtidos pelas organizações que realizaram programas de melhoria com o CMM como modelo de referência.

3.1.2 - Maturidade

O SW-CMM (CMM para software, como também é conhecido o modelo), tem seu foco no processo de software por entender que a qualidade de um sistema de software é fortemente influenciado pela qualidade do processo utilizado para desenvolvê-lo e mantê-lo. Portanto, uma premissa do SW-CMM é o foco no processo da mesma forma que no produto, pois enfocando apenas o produto se perde conhecimento de como produzi-lo melhor, por não se ter planejado, conhecido e, constantemente, melhorado o processo utilizado para desenvolvê-lo.

Para um melhor entendimento do modelo é necessária a compreensão dos conceitos de: processo, capacidade e maturidade.

Segundo Humphrey (1989), um processo de software envolve métodos, ferramentas e pessoas. Para um processo funcionar satisfatoriamente ele deve possuir:

- Procedimentos e métodos que descrevam a relação entre as tarefas.
- Ferramentas e equipamentos que dêem suporte a realização das tarefas, simplificando e automatizando o trabalho.
- Pessoas com perfil adequado, treinadas nos métodos e nas ferramentas para poderem realizar as atividades adequadamente.

Este conjunto deve estar integrado harmoniosamente para funcionar de maneira eficaz (Rocha, 2001).

A capacidade do processo de software descreve o conjunto de resultados esperados que pode ser atingido quando se segue o processo de software estabelecido. A capacidade do processo de software de uma organização fornece um meio de se prever, de forma mais provável, os resultados a serem esperados no próximo projeto a ser empreendido pela organização (Gonçalves, 2001).

A maturidade do processo de software é quando um processo específico é explicitamente definido, documentado, gerenciado, medido, controlado e efetivo. A maturidade implica num potencial de crescimento da capacidade e indica tanto a riqueza do processo de software de uma organização, quanto à consistência na qual o processo é aplicado nos projetos de toda a organização (Paulk, 1993). Em outras palavras, a maturidade do processo pode ser traduzida como a possibilidade de entregar sistemas de software dentro dos prazos, utilizando os mesmos recursos, planejando e atendendo requisitos e qualidade desejados.

Quando uma organização alcança maturidade no processo de software, ela institucionaliza seus processos através de políticas, padrões e estrutura organizacional. A institucionalização implica na construção de uma infra-estrutura e uma cultura corporativa que suportem os métodos, práticas e procedimentos do processo (Paulk, 1993).

O SW-CMM proporciona às organizações um guia de como aumentar o controle sobre seu processo de desenvolvimento e manutenção de software, e de como alterar o ambiente em direção a uma cultura de excelência em engenharia de software. Concentrando-se em um conjunto limitado de atividades, as organizações podem melhorar seu processo e possibilitar um contínuo ganho no desempenho (Humphrey, 1989).

O CMM foi concebido para o desenvolvimento de grandes projetos militares, portanto, para sua aplicação em projetos menores e em outras áreas é necessário um trabalho cuidadoso de interpretação e adequação à realidade da organização (Rocha, 2001).

3.1.3 – Estrutura do CMM

O objetivo do SW-CMM é identificar forças e fraquezas de uma organização, entender as atividades necessárias para um programa de melhoria de processo, definir e melhorar os processos e classificar as empresas de software.

O SW-CMM baseia-se em pequenos passos e não em inovações revolucionárias. Com base nos estudos de amadurecimento gradativo de Crosby (Tingey, 1997), estes passos evolutivos foram organizados em cinco níveis de maturidade: **inicial**, **repetitivo**, **definido**, **gerenciado** e **otimizado**. Estes níveis representam a situação da organização quanto ao seu processo de software (Lamprech, 1997). Tais níveis definem uma escala para medir a maturidade do processo e avaliar a capacidade do mesmo. Além disso, contém diretrizes para a organização priorizar seus esforços de melhoria.

Embora não seja uma norma emitida por uma instituição internacional (como a ISO ou o IEEE), o SW-CMM tem tido uma grande aceitação mundial.

Se alguém espera que o CMM oriente a aplicação de algum método de desenvolvimento de software altamente avançado, irá se decepcionar. O CMM opera em outro nível, diz respeito a aspectos que devem ser considerados para o desenvolvimento bem-sucedido de software. É descritivo, não prescritivo (Furlan, 2001), assim, concentra-se no questionamento, não na resposta. Isto pode parecer pouco estimulante, mas era exatamente o que a indústria de software precisava para fazer um progresso real no modo de desenvolvimento de software.

Embora o CMM requeira a documentação minuciosa do processo, ele não tende a burocratização, uma vez que propõe que o processo documentado seja adaptado às características da organização e da categoria de software que desenvolve. O CMM não pretende resolver problemas, ele se propõe a ajudar pessoas e organizações a encontrar suas próprias soluções. O CMM também não é uma proposta de auto-ajuda, é, sim, um modelo estabelecido em bases racionais e que requer, para obter sucesso, apoio da gerência, um bom domínio de conhecimento, habilitação técnica, participação e esforço de todos os envolvidos (Fiorini, 1999).

3.1.4 – Implementação do CMM

A implementação do SW-CMM é cíclica, composta de fases distintas: análise da situação (avaliação), comparação com o próximo nível do modelo, planejamento das ações corretivas, execução destas ações e nova análise da situação (Pires, 1998).

A primeira medida a ser tomada para realizar um programa de melhorias sobre SW-CMM, é a realização de uma avaliação dos processos de software da organização. Uma “avaliação formal” do CMM, onde a organização é certificada em um nível de maturidade, é realizada somente por auditores credenciados pelo SEI. Ao contrário de outros programas de qualidade, como a série ISO 9000, não existem entidades certificadoras, apenas pessoas físicas certificadas pelo SEI.

O processo de avaliação é iniciado com a aplicação de um questionário. O SW-CMM é acompanhado por um questionário intitulado Questionário da Maturidade ou simplesmente MQ - *Maturity Questionnaire* (Zubrow, 1994). O objetivo do MQ é colher dados iniciais sobre a organização avaliada, isto é, trata-se de uma fonte primária de informação. A partir destes dados, novos questionários são criados, de acordo com a necessidade dos avaliadores.

De um modo geral, as avaliações são realizadas através de questionários, entrevistas e análise de documentações da organização. Todo o trabalho é realizado por uma equipe (grupo avaliador) liderada por um auditor certificado pelo SEI. O resultado final é um relatório onde são apontados os pontos fracos e fortes, bem como oportunidades de melhorias, avaliando-se o posicionamento dos processos em relação ao modelo SW-CMM e seus níveis.

A partir do resultado da avaliação, a organização deverá estabelecer equipes de melhoria de processo, que conduzirão projetos, com o objetivo de definir e implementar os processos considerados prioritários.

O SEI desenvolveu muitos métodos de avaliação baseados no CMM com foco no processo de desenvolvimento de software. Um método é um técnica ou modo de proceder para se aplicar um modelo.

A iniciativa da avaliação pode ser da própria empresa, quando esta decide iniciar um programa de melhorias em seus processos de software (avaliação interna), ou de uma empresa que deseja avaliar concorrentes para um contrato (avaliação externa).

A Avaliação da Capacidade do Software baseado no CMM, ou SCE - *Software Capability Evaluation* (Byrnes, 1996), é um método para avaliação do processo de desenvolvimento de software de um fornecedor, ou seja, para aquisição de software, para monitoramento de contratos e para incentivos. Esta avaliação é feita por um grupo externo, raramente são incluídas pessoas da organização avaliada ao grupo avaliador.

A Avaliação para Melhoria do Processo Interno baseado no CMM, ou CBA-IPI - *CMM Based Appraisal for Internal Process Improvement* (Dunaway, 1996), é um método para avaliação interna, tem a característica de uma auto-avaliação, é utilizado quando uma organização quer avaliar seu próprio processo de desenvolvimento de software.

Estes dois métodos de avaliação (SCE e CBA-IPI) são de acordo com a Estrutura de Avaliação do CMM, ou CAF - *CMM Appraisal Framework* (Marters, 1995). A CAF identifica os requisitos e características intrínsecas das avaliações baseadas no CMM, para melhorar a consistência e integridade dos métodos e seus resultados. A CAF, em sua única versão, usa o CMM v1.1 como modelo de referência.

3.2 – Modelo de Maturidade da Capacidade Integrado

O modelo SW-CMM tornou-se, ao longo de uma década de uso por várias organizações de software de diversos tamanhos e setores, o modelo mais conhecido, usado e respeitado pela comunidade de engenharia de software (Belloquim, 2002). Por ser um

modelo baseado em experiências de organizações de software, trata-se de um modelo que sugere práticas eficientes e eficazes, não sendo apenas um modelo teórico.

Desde sua primeira versão, lançada em 1991, na esteira de sucesso do SW-CMM, diversos outros modelos foram criados visando cobrir outras áreas de interesse, como por exemplo:

- SA-CMM - *Software Acquisition CMM*: usado para avaliar a maturidade de uma organização em seus processos de seleção, compra e instalação de software desenvolvido por terceiros.
- SE-CMM - *System Engineering CMM*: avalia a maturidade da organização em seus processos de engenharia de sistemas, concebidos como algo maior que o software. Um sistema inclui o software, o hardware e quaisquer outros elementos que participam do produto completo.
- IPD-CMM - *Integrated Product Development CMM*: ainda mais abrangente que o SE-CMM, inclui também outros processos necessários à produção e suporte ao produto, tais como suporte ao cliente e processos de fabricação.
- P-CMM - *People CMM*: avalia a maturidade da organização em seus processos de administração de recursos humanos no que se refere a software, recrutamento e seleção de profissionais e treinamento.

O surgimento de todos estes outros modelos acabou trazendo alguns problemas. Como nem todos os modelos usavam a mesma terminologia, um mesmo conceito podia receber nomes diferentes em cada modelo. Os modelos tinham diferentes números de níveis ou formas diferentes de avaliar o processo. Outro problema era os altos custo de treinamento, avaliação e harmonização para organizações que quisessem usar mais de um modelo. Assim, tornou-se necessária uma padronização.

Além disso, a experiência no uso de SW-CMM, durante uma década, serviu para identificar pontos em que o modelo poderia ser melhorado. O surgimento do projeto SPICE levou à necessidade de tornar o CMM compatível com a norma ISO 15504 (1997), resultado deste projeto.

Por estas razões, o SEI iniciou um projeto chamado CMMI - *CMM Integration*. Seu objetivo era gerar uma nova versão do CMM que resolvesse esses problemas. Concretamente, a primeira idéia, como o nome sugere, foi integrar os diversos CMMs em uma estrutura única, todos com a mesma terminologia, processos de avaliação e estrutura. O projeto se preocupou ainda em tornar o CMM compatível com a norma ISO 15504, de modo que avaliações em um modelo sejam reconhecidas como equivalentes aos do outro. E, naturalmente, incorporar ao CMM as sugestões de melhoria surgidas ao longo dos anos (CMMI, 2002a).

O Modelo de Maturidade da Capacidade Integrado, ou CMMI, foi resultado da integração dos modelos (CMMI, 2002a):

- SW-CMM V2C - *Capability Maturity Model for Software V2.0, draft C*.
- SECM - *EIA Interim Standard 731 – System Engineering Capability Model*.
- IPD-CMM - *Integrated Product Development Capability Maturity Model, draft 0.98*.

O CMMI é claramente uma melhoria sobre o CMM. No entanto, apesar das melhorias incorporadas ao CMMI, seu precursor - o CMM - continua sendo um modelo bastante eficiente (Furlan, 2001).

3.2.1 – Estrutura do CMMI

Há quatro áreas de interesse disponíveis para implementar o CMMI, tais áreas são chamadas de disciplinas. São elas (CMMI, 2002a):

- Engenharia de Sistemas.
- Engenharia de Software.
- Desenvolvimento de Processo e Produto Integrado.
- Contratos de Fornecedores.

A disciplina Engenharia de Sistemas compreende o desenvolvimento do sistema como um todo, podendo incluir ou não o software. Os engenheiros de sistema focalizam na

transformação das necessidades do consumidor, expectativas e restrições para o desenvolvimento dos produtos e suporte aos mesmos.

A disciplina Engenharia de Software cobre o desenvolvimento de sistemas de software. Os engenheiros de software focalizam na aplicação de uma abordagem sistemática, disciplinada e quantitativa para desenvolver, operar e manter o software.

A disciplina de Desenvolvimento de Processo e Produto Integrado, baseado no modelo é uma abordagem sistemática que busca uma oportuna colaboração dos interessados, através da vida do produto para melhor satisfazer as necessidades, expectativas e requisitos do cliente. Um processo, para suportar esta disciplina, deve estar integrado com outros processos da organização.

Quando o projeto é muito complexo, normalmente a implementação de algumas funções é terceirizada. A disciplina Contrato de Fornecedores cobre a aquisição de produtos de fornecedores, através do monitoramento das atividades dos fornecedores antes da entrega do produto.

O CMMI foi criado para integrar estas disciplinas e resolver o problema de organizações que utilizam múltiplos modelos CMM. O CMMI fornece modelos com foco em disciplinas individuais e em disciplinas combinadas. A escolha da disciplina varia de acordo com o escopo da organização na qual será implementado o modelo.

Cada uma destas disciplinas pode ser implementada utilizando duas abordagens:

- **Abordagem em estágios:** proporciona uma estratégia pré-definida para melhoria organizacional, similar ao SW-CMM existente.
- **Abordagem contínua:** proporciona flexibilidade para que as organizações escolham quais processos serão enfatizados ou priorizados para melhoria, conforme sua disponibilidade financeira para a implementação de cada processo.

Os modelos CMMI são:

- CMMI-SW - *CMMI for Software Engineering*: este modelo inclui apenas a disciplina Engenharia de software.
- CMMI-SE - *CMMI for Systems Engineering*: este modelo inclui apenas a disciplina Engenharia de Sistemas.
- CMMI-SE/SW - *CMMI for Systems Engineering and Software Engineering*: este modelo inclui as disciplinas Engenharia de Software e Engenharia de Sistemas.
- CMMI-SE/SW/IPPD - *CMMI for Systems Engineering, Software Engineering, and Integrated Product and Process Development*: Este modelo inclui as disciplinas do modelo anterior acrescido da disciplina Desenvolvimento de Processo e Produto Integrado.
- CMMI-SE/SW/IPPD/SS - *CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing*: este é o modelo mais completo e inclui todas as disciplinas (Engenharia de software, Engenharia de Sistemas, Desenvolvimento de Processo e Produto Integrado e Contratos de Fornecedores).

Cada modelo tem sua descrição em cada uma das abordagens (em estágios e contínua). A escolha do modelo e da abordagem depende da realidade e das necessidades da organização. Se uma organização preocupa-se exclusivamente com Engenharia de Software ou exclusivamente com Engenharia de Sistemas, então os modelos apropriados seriam CMMI-SW e CMMI-SE, respectivamente. No entanto, se a organização preocupa-se com ambas disciplinas o modelo CMMI-SE/SW seria o mais adequado, o que evitaria repetições e a sobrecarga administrativa, que é comum na manutenção de disciplinas isoladas. Finalmente, os modelos CMMI-SE/SW/IPPD e CMMI-SE/SW/IPPD/SS são apropriados para organizações que empreguem Desenvolvimento de Processo e Produto Integrado em suas práticas, combinado ou não com Contratos de Fornecedores.

3.2.2 – Componentes do Modelo

Os principais componentes do CMMI são: as áreas de processo, os objetivos específicos, as práticas específicas, os objetivos genéricos e as práticas genéricas.

Os modelos CMM não descrevem em detalhes todas as áreas do processo que são envolvidas no desenvolvimento e manutenção de software. Certas áreas têm sido identificadas como determinantes da capacidade do processo e melhoria do mesmo. Estas áreas de processo foram identificadas com base em anos de experiência em engenharia de software e gerenciamento dos profissionais do SEI.

Cada área de processo é um conjunto de práticas que, quando executadas corretamente, satisfaz um conjunto de objetivos considerados importantes para implementar melhorias significantes na área de processo em questão. As áreas de processo do CMMI são:

Foco no processo organizacional: o propósito desta área de processo é planejar e implementar melhorias no processo organizacional, através do entendimento dos pontos positivos e negativos dos processos da organização.

Definição de Processo Organizacional: o objetivo desta área é estabelecer e manter um conjunto de itens de processo organizacional. Tais itens incluem a descrição do processo e elementos (tarefas e atividades) do processo, descrição de modelos de ciclo de vida, guia de execução de processo, dados e documentação do processo.

Treinamento Organizacional: esta área relaciona-se com o desenvolvimento das habilidades e conhecimentos das pessoas para que elas possam desempenhar seu papel efetiva e eficientemente.

Desempenho do Processo Organizacional: o propósito desta área é estabelecer e manter um entendimento quantitativo da capacidade dos processos padrões em suportar

objetivos de qualidade e de desempenho, visando colher os dados necessários ao gerenciamento quantitativo dos projetos da organização.

Disposição e Inovação Organizacional: esta área permite a seleção e distribuição ordenada de melhorias (incrementais e inovadoras) que podem aumentar a habilidade da organização para alcançar seus objetivos de qualidade e desempenho do processo.

Planejamento de Projeto: esta área compreende o estabelecimento e manutenção de planos que definam as atividades do projeto.

Controle e Monitoração de Projeto: o propósito desta área é proporcionar um entendimento do processo utilizado em um projeto, tal que ações corretivas apropriadas possam ser tomadas quando o desempenho do projeto desvia significativamente do plano estabelecido.

Gerenciamento de Contrato de Fornecedores: esta área objetiva gerenciar a aquisição de produtos de fornecedores de forma que exista um contrato formal.

Gerenciamento de Projeto Integrado: esta área tem por finalidade estabelecer e gerenciar o projeto e o envolvimento das pessoas relevantes (indivíduos ou grupos envolvidos com o projeto, como fornecedores, clientes, usuários, e outros), de acordo com um processo definido e integrado baseado nos processos padrões da organização.

Gerenciamento de Riscos: o objetivo desta área é identificar potenciais problemas antes que eles ocorram, através do planejamento e execução de atividades específicas em situações de riscos, visando atenuar os impactos adversos que possam influenciar no alcance aos objetivos.

Gerenciamento Quantitativo do Projeto: o propósito desta área é gerenciar quantitativamente o processo definido para o projeto, visando atingir os objetivos de qualidade e de performance estabelecidos para o mesmo.

Gerenciamento de Requisitos: esta área de processo tem o propósito de gerenciar os requisitos dos produtos do projeto e seus componentes, e identificar inconsistências entre estes requisitos e os estabelecidos no plano.

Desenvolvimento de Requisitos: analisar os requisitos do cliente, do produto e dos componentes do produto, de modo que eles realmente supram as necessidades das pessoas envolvidas com o projeto.

Solução Técnica: o propósito desta área é projetar, desenvolver e implementar soluções para os requisitos, abrangendo produtos, componentes de produtos e produtos do ciclo de vida do processo, cada um individualmente ou combinados.

Integração de Produto: o objetivo é reunir todos os componentes do produto, assegurar que o produto, quando integrado, funciona bem.

Verificação: garantir que os produtos de trabalho vão ao encontro de seus requisitos especificados.

Validação: demonstrar que um produto ou componentes do produto cumpre seu uso desejado quando mantido em seu ambiente específico.

Gerenciamento da Configuração: o propósito é estabelecer e manter a integridade dos produtos de trabalho usando identificação, controle, relatório de *status* e auditoria da configuração.

Garantia de Qualidade de Produto e Processo: o objetivo desta área é garantir a entrega de produtos e serviços de alta qualidade, através da avaliação da qualidade do processo de desenvolvimento.

Medições e Análises: o objetivo desta área é desenvolver e sustentar uma capacidade de medição usada para suportar o gerenciamento das informações necessárias.

Resolução e Análise de Decisão: o propósito desta área é analisar decisões usando um processo de avaliação formal que avalia as possíveis alternativas e estabelece critérios.

Resolução e Análise das Causas: o objetivo desta área é analisar as causas dos defeitos e outros problemas e tomar atitudes para que eles não voltem a ocorrer no futuro.

Cada área de processo é composta por seus objetivos específicos. Estes objetivos descrevem o que deve ser atingido para satisfazer cada área. Cada objetivo específico, por sua vez, é composto por práticas específicas, tais práticas descrevem o que deve ser executado para atingir cada objetivo específico.

Os objetivos genéricos são assim chamados porque um mesmo objetivo é aplicado a múltiplas áreas de processo. Estes objetivos são organizados em práticas genéricas, que são atributos a serem observados que indicam se a implementação e a institucionalização de cada área de processo é efetiva, repetível e duradoura. As áreas de processo do CMMI são comuns para ambas abordagens.

3.2.3 – Abordagem em Estágios

Esta abordagem permite uma fácil migração do SW-CMM para o CMMI, por terem uma estrutura bastante semelhante.

A abordagem em estágios descreve uma seqüência de atividades de melhorias, iniciando com práticas básicas de gerenciamento e progredindo através de um caminho pré-definido de níveis de melhoria sucessivos, cada um servindo de base para o próximo nível (CMMI, 2002a). Cada nível compreende o nível de maturidade do processo de software da organização como um todo.

A abordagem em estágios organiza as áreas de processo em 5 níveis de maturidade, para dar suporte e orientar a melhoria do processo, indicando quais áreas de processo implementar para atingir cada nível.

Os níveis de maturidade são: **inicial**, **gerenciado**, **definido**, **quantitativamente gerenciado** e **otimizado**. Cada nível consiste em um conjunto pré-definido de áreas de processo. O nível de maturidade é medido pelo atendimento a todos os objetivos genéricos (agrupados em características comuns) e específicos de todas as áreas de processo do nível em questão, como ilustra a 3.1.

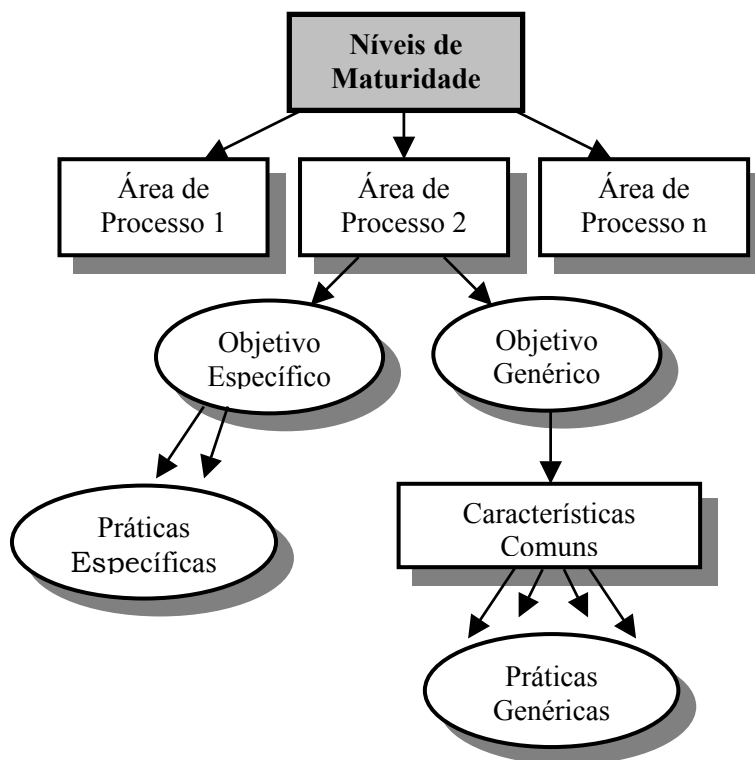


FIGURA 3.1 – Estrutura do CMMI em estágios.

FONTE: CMMI, (2002a).

Os Níveis de Maturidade da abordagem em estágios são:

Nível 1 - Inicial: O desenvolvimento de software é caracterizado como *ad-hoc* ou caótico. A organização não possui um ambiente estável, é caracterizada pela não existência de processos definidos, abandono de processos em momentos de crises e falta de comprometimento com os projetos. Os sucessos dependem de esforços individuais e

na maioria das vezes não são capazes de repetir sucessos anteriores. Os projetos destas organizações normalmente excedem custos e cronogramas.

Nível 2 - Gerenciado: Uma organização neste nível tem seus processos planejados, executados, medidos e controlados. A disciplina do processo reflete na continuidade das práticas mesmo em momentos de crise. Os projetos são executados e gerenciados de acordo com um plano documentado. Neste nível os requisitos, processos, produtos de trabalho e serviços são gerenciados. Além disso, compromissos são estabelecidos e revisados entre as pessoas interessadas quando necessário. Os produtos e serviços satisfazem seus requisitos, padrões e objetivos.

Nível 3 - Definido: Os processos das organizações neste nível são compreendidos e descritos em procedimentos, padrões, ferramentas e métodos. É estabelecido um conjunto de processos padrões da organização. Estes processos padrões são utilizados por toda a organização. Os gerentes de departamento estabelecem objetivos baseados neste conjunto de processos padrões e asseguram que estes objetivos são apropriadamente enfocados.

A diferença entre o Nível 2 e o Nível 3 é que no primeiro os padrões, descrição de processos e procedimentos podem ser bastante diferente em cada projeto. No Nível 3 os processos devem ser descritos com mais detalhes e devem ser seguidos de forma mais rigorosa.

Nível 4 - Quantitativamente Gerenciado: Os processos são controlados utilizando estatísticas e outras técnicas quantitativas. Objetivos quantitativos para qualidade e desempenho são estabelecidos e usados como critério no gerenciamento dos processos. Os objetivos quantitativos são baseados nas necessidades dos consumidores, clientes, organização e executores do processo. Medidas detalhadas do desempenho do processo são coletadas e analisadas estatisticamente. Causas de possíveis variações de desempenho do processo são identificadas e corrigidas para prevenir ocorrências futuras.

Nível 5 - Otimizado: Os processos são melhorados continuamente baseados na compreensão quantitativa das causas comuns de variação do processo. As organizações neste nível focalizam na melhoria contínua do desempenho do processo através de melhorias incrementais e inovações tecnológicas. Objetivos de melhorias quantitativas de processo são estabelecidas e continuamente revisados para refletir em mudanças nos objetivos do negócio.

Na abordagem em estágios, as áreas de processo são agrupadas em níveis de maturidade. Com exceção do Nível 1, todos os níveis do CMMI em estágios é composto por áreas de processo. A seguir são apresentadas as áreas de processo em seus respectivos níveis.

Nível 2: Gerenciado

- Gerenciamento de Requisitos.
- Planejamento de Projeto.
- Controle e Monitoração de Projeto.
- Gerenciamento de Contrato de Fornecedores.
- Medições e Análises.
- Garantia de Qualidade de Produto e Processo.
- Gerenciamento da Configuração.

Nível 3: Definido

- Desenvolvimento de Requisitos.
- Solução Técnica.
- Integração de Produto.
- Verificação.
- Validação.
- Foco no Processo Organizacional.
- Definição de Processo Organizacional.
- Treinamento Organizacional.
- Gerenciamento de Projeto Integrado.

- Gerenciamento de Risco.
- Resolução e Análise de Decisão.

Nível 4 : Quantitativamente Gerenciado

- Desempenho do Processo Organizacional.
- Gerenciamento Quantitativo do Projeto.

Nível 5: Otimização

- Disposição e Inovação Organizacional.
- Resolução e Análise Causal.

Para uma organização alcançar um nível de maturidade, deve atingir o objetivo genérico e os objetivos específicos de todas as áreas de processo do nível em questão, através da execução das práticas genéricas e específicas deste nível (CMMI, 2002a).

Por exemplo, para atingir o Nível 2 de maturidade, a organização deve implementar as sete áreas de processo do Nível 2. Para implementar cada uma destas áreas é necessário atender aos objetivos específicos de cada uma, através da execução de suas práticas específicas. Por exemplo, a área de processo Gerenciamento de Requisitos, em particular, tem apenas um objetivo específico, que é Gerenciar Requisitos e suas práticas específicas são: obter um entendimento dos requisitos, obter comprometimento com os requisitos, gerenciar as mudanças de requisitos, manter um controle bidirecional dos requisitos, identificar inconsistência entre os requisitos e os produtos de trabalho.

Além do cumprimento dos objetivos específicos, a organização que busca o Nível 2 de maturidade, deve alcançar o objetivo genérico deste nível, que é: Institucionalizar um Processo Gerenciado. Então, a organização deve executar todas as práticas genéricas deste objetivo, que são: estabelecer uma política organizacional, planejar o processo, fornecer recursos, designar responsabilidades, treinar pessoas, gerenciar configuração, identificar e envolver as pessoas interessadas pelo projeto, monitorar e controlar o processo, avaliar objetivamente o processo, revisar o s situação e comparar com o

próximo nível. Estes objetivos genéricos devem ser aplicados a cada área de processo do Nível 2. Vale lembrar que a execução destas práticas genéricas determina o quanto a implementação da área de processo é institucionalizada na organização, isto é, o quanto a implementação da área de processo é efetiva, repetível e duradoura.

Para uma organização alcançar o Nível 3 de maturidade, é preciso cumprir todas as áreas do Nível 2 e mais as do Nível 3. E assim até o nível 5. Uma organização pode estar no Nível 2 e possuir práticas de níveis superiores, mas ser apenas Nível 2 por não executar todas as áreas de processo do nível superior. Assim, nesta abordagem, a tentativa de saltar um nível é improdutiva, pois cada nível serve de fundação para o nível superior.

3.2.4 – Abordagem Contínua

A abordagem contínua utiliza 6 níveis de capacidade, sendo eles: **incompleto, executado, gerenciado, definido, quantitativamente gerenciado e otimizado**. Os níveis de capacidade traçam um caminho evolutivo de melhoria para cada área de processo.

A abordagem contínua, ao mesmo tempo em que define uma seqüência para melhoria de uma área de processo, também permite uma flexibilidade na escolha das áreas de processo a serem melhoradas. Assim, a organização pode direcionar seus esforços de melhoria nas áreas que julgar mais relevantes para o desenvolvimento como um todo.

Nesta abordagem, assim como na abordagem em estágios, cada área de processo é composta por objetivos específicos e genéricos, organizados em práticas específicas e genéricas respectivamente, com exceção do Nível 0. Cada área de processo possui seus objetivos específicos próprios, compostos por suas práticas específicas. Os objetivos genéricos e práticas genéricas são aplicados a múltiplas áreas de processo, descrevendo uma seqüência de níveis de capacidade. Cada nível possui um objetivo genérico, comum a todas as das as áreas no mesmo nível, como mostrado na Figura 3.2.

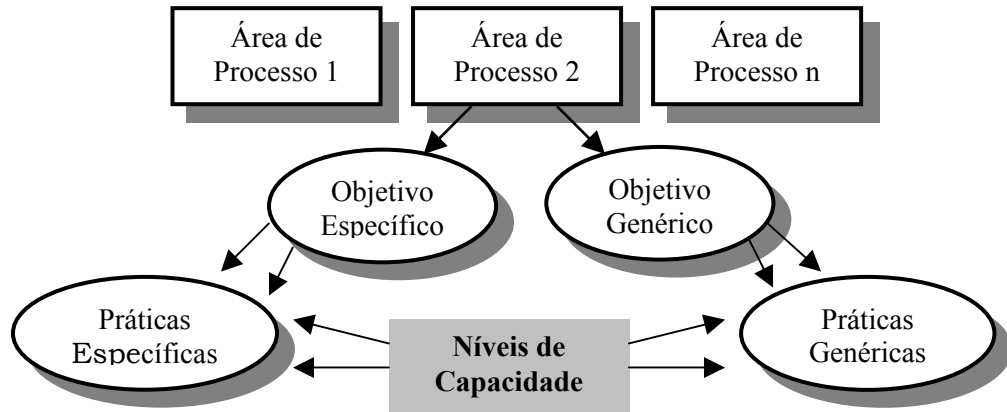


FIGURA 3.2 – Estrutura do CMMI contínuo.

FONTE: CMMI (2002b).

Em resumo, a abordagem contínua, descreve uma discreta seqüência de melhorias a ser aplicadas a cada área de processo.

Os Níveis de Capacidade da abordagem contínua são (CMMI, 2002b):

Nível 0 – Incompleto: uma área de processo é considerada incompleta quando ela não é executada ou é parcialmente executada. Isto é, um ou mais objetivos específicos da área não é executado.

Nível 1 – Executado: Uma área de processo é executada quando o objetivo genérico e todos os objetivos específicos da área são satisfeitos.

Nível 2 – Gerenciado: Uma área de processo gerenciada é uma área executada que também é planejada, está de acordo com uma política, emprega pessoas com habilidades técnicas para produzir adequadamente seus produtos, envolve as pessoas que tem interesses no projeto, é monitorado, controlado e revisado.

Nível 3 – Definido: Uma área de processo definida é uma área gerenciada e que está de acordo um conjunto de processos padrões da organização.

Nível 4 – Quantitativamente Gerenciado: Uma área de processo quantitativamente gerenciada é uma área definida que é controlada utilizando estatísticas e outras técnicas quantitativas. Os objetivos quantitativos para a qualidade e desempenho do processo são estabelecidos e usados como critério no gerenciamento do processo.

Nível 5 – Otimizado: Uma área de processo otimizada é uma área quantitativamente gerenciada que é modificada e adaptada para atingir os objetivos de negócio. Um processo otimizado focaliza em melhorias tecnológicas incrementais e inovadoras.

O nível de capacidade focaliza na crescente habilidade da organização para desenvolver, controlar e melhorar seu desempenho na área de processo em questão. Cada nível de capacidade é construído sobre o outro, seguindo uma ordem de melhoria de processo.

Nesta abordagem cada área de processo possui características relativas a mais de um nível. Assim, uma área de processo que, no modelo em estágios, pertence exclusivamente ao Nível 2, no modelo contínuo pode ter características que a coloquem em outros níveis. Em outras palavras, no modelo contínuo, cada área é classificada separadamente, de modo que a organização possa ter áreas no Nível 1, outras no Nível 2, assim por diante.

3.2.5 – Perspectivas de Implementação do CMMI

A implementação do CMMI pode ser visualizada sob duas perspectivas:

- O CMMI pode ser implementado visando uma avaliação para melhoria do processo interno, que pode ser aplicada a uma organização ou a um projeto, de forma expandida ou restrita, tendo em vista os objetivos de melhoria mais relevantes. A escolha da disciplina, das áreas de processo a serem melhoradas, do nível de maturidade/capacidade a ser alcançado deve depender das práticas

que apóiam as necessidades e objetivos de negócio da organização. A escolha das áreas a serem focalizadas deve ser cuidadosa, devido ao inter relacionamento entre os componentes do modelo. Os esforços de melhoria também devem ser analisados tendo em vista os objetivos de custo e cronograma das atividades de melhoria. Esta perspectiva tem as características de uma auto-avaliação.

- O CMMI também é implementado visando a formalização da capacidade/maturidade do processo da organização, para fins de comparação com outras organizações (em processos de terceirização) ou disseminação pública com propósito de *marketing*. Neste caso, a implementação deve obedecer de forma mais rigorosa às indicações do modelo, com o objetivo de garantir a consistência dos resultados da avaliação (CMMI, 2002b).

Os critérios de desenvolvimento, definição e utilização dos métodos de avaliação baseados no CMMI, estão descritos nos Requisitos de Avaliação para CMMI, ou ARC - *Appraisal Requirements for CMMI*, compatível com a ISO/IEC 15504. O ARC é uma evolução do CAF (*CMM Appraisal Framework*) (Marsters, 1995) que foi uma estrutura produzida originalmente para fornecer uma base comum para avaliações aplicando o CMM. O ARC define os métodos para avaliação interna (auto-avaliação) e externa (avaliação de fornecedores), incorpora as múltiplas disciplinas do CMMI e as representações em estágios e contínua (ARC, 2001).

CAPITULO 4

PANORAMA ATUAL

4.1 – A Indústria do Software

A partir das últimas décadas do século XX, o desenvolvimento dos países foi surpreendido por novos fatores e causas que passaram a determinar os novos padrões do crescimento econômico mundial. Os avanços em Tecnologias de Informação e Comunicação (TICs) tornaram possível difundir e acessar informações em uma velocidade e em uma escala nunca vista antes.

Os investimentos em TICs têm sido o componente mais dinâmico dos investimentos em negócio. Os países que obtiveram um aumento do PIB nos anos 1990, geralmente acumularam mais capital em TICs. Igualmente importantes, são os dados relativos ao acúmulo de capital em software, que entre os anos de 1995 a 1999, representou um terço de todo o capital investido em TICs. Uma explicação para a onda de investimento em software é o rápido aparecimento e difusão de novas tecnologias de propósito geral, como a Internet. O que é novo, comparado a outras tecnologias, é que a Internet provê uma infra-estrutura para novas formas de negócios eletrônicos (Araújo, 2003).

Organizações de software em todo o mundo empregam cerca de 7 milhões de técnicos e geram anualmente uma receita de US\$ 600 bilhões (Cordeiro, 1999). A indústria de software é vista atualmente como um dos segmentos mais promissores, com um enorme potencial futuro. O mercado do software tem um crescimento invejável, mantendo taxas de crescimento superiores a 15% ao ano (Araújo, 2003).

Os Estados Unidos continuam na liderança de gastos com software. Na América Latina, o Brasil é o líder de investimentos e ocupa o décimo lugar na escala mundial. Em 2001, os movimentos brasileiros no setor somaram US\$ 7,2 bilhões, valor três vezes maior que os de 1993 (WITSA, 2002). As importações brasileiras somam aproximadamente

US\$ 1 bilhão. O Programa para Promoção da Excelência do Software Brasileiro – SOFTEX, do Ministério da Ciência e Tecnologia, expandiu o desenvolvimento do software em várias partes do país e contribuiu para o crescimento das exportações de software para US\$ 100 milhões, em 2001(Araújo, 2003). Porém, o mercado local ainda detém grande parcela da produção de software nacional.

Apesar do expressivo mercado interno, o Brasil ainda não encontrou o caminho que o levasse a uma conquista significativa de outros mercados e o projetasse internacionalmente na indústria do software. Ainda são poucas as organizações brasileiras que conseguem avançar no mercado internacional. A inserção da indústria do software nacional no mercado mundial é um grande desafio para o país, e há condições básicas para enfrenta-lo, através de uma política agressiva, articulada e consistente.

Na última década, políticas governamentais e alguns programas prioritários do Ministério da Ciência e Tecnologia, para o setor de informática, têm propiciado o crescimento dos investimentos em pesquisa e desenvolvimento por parte da iniciativa privada. Isso tem atraído investimentos de grandes empresas internacionais e os produtos fabricados no país têm preços competitivos em relação aos seus similares importados. No campo tecnológico, o setor de informática é o segmento industrial que mais investe em pesquisa e desenvolvimento proporcionalmente ao seu faturamento.

Para a indústria de software nacional se destacar internacionalmente, principalmente considerando o ambiente de avanços tecnológicos muito rápidos e um processo de globalização irreversível, é necessária a articulação de alguns fatores fundamentais. Segundo a SEPIN - Secretaria de Política de Informática e Automação do Ministério da Ciência e Tecnologia, tais fatores são inovação, seletividade e qualidade. A inovação é exercitada através da pesquisa e desenvolvimento de novos produtos para nichos específicos de mercado. A seletividade é desenvolvida principalmente através da definição do que produzir no país e em que escala. A qualidade é, inquestionavelmente, um requisito indispensável para inserção neste mercado.

Um atrativo recente da indústria do software são os processos de terceirização (*outsourcing*) dos setores de Tecnologia da Informação (Araújo, 2003). O interesse em reduzir custos é um dos principais fatores da terceirização, que tem se mostrado como propulsor de crescimento para países como Índia e China, detentores de mão-de-obra imensa e barata que tem atraído o mercado norte-americano. Este é um mercado bastante promissor também para o Brasil, tanto para a expansão do software no mercado nacional, quanto para o reconhecimento internacional.

Há indústrias de software interessadas em resolver problemas de software através da normalização da qualidade. A terceirização de serviços de informática por grandes empresas públicas ou privadas, e a necessidade de expansão dos mercados nacionais e internacionais, são os dois eixos da busca pela certificação de processos do segmento. A América Latina, principalmente o Brasil, é um mercado bastante atrativo e potencial para consultorias especializadas em resolver problemas de qualidade.

Há bases de dados históricos nacionais que permitem afirmar que o Brasil tem projetos e estratégias em direção ao alcance de padrões internacionais efetivos em qualidade e produtividade no setor de software, há evidências de que a qualidade de software no país tem apresentado tendências de melhoria contínua. Programas de qualidade total, sistemas da qualidade ou similares tem apresentado um número crescente de implantação a cada ano (PBQP, 2002). Muito se tem falado da obtenção da certificação CMM como passaporte para o mercado externo, mas outro motivador da busca de certificações é a possibilidade de adotar processos que tornem o crescimento mais consistente.

4.2 – Os Processos de Software

Um projeto de software eficiente é aquele que visa a construção de um produto de software que satisfaça as necessidades do cliente e seja desenvolvido e entregue dentro do custo e prazo especificado. Em outras palavras, as três principais características de um projeto são: custo, planejamento e qualidade, onde qualidade representa quão bom o

produto é e satisfaz o cliente. Um projeto é um sucesso quando atinge estas três características - custo, planejamento e qualidade.

Porém, a indústria de software pode citar muitos exemplos de projetos que fracassaram. As principais causas destes fracassos podem ser combinadas em uma categoria chamada falha no processo. Um projeto de software falha, na maioria das muitas vezes, porque o processo seguido não é apropriado. As maiores razões para o descontrole são: objetivos pouco claros, planejamento ruim, utilização de novas tecnologias de forma impensada, falta de metodologia de gerenciamento do projeto e pessoal insuficiente (Glass, 1998). Para um projeto ter sucesso, é necessário que o processo utilizado seja apropriado. Como o processo tem grande efeito sobre a qualidade e produtividade, um modo de melhorar este par está na melhoria do processo utilizado pela organização.

Infelizmente muitos projetos de informática, quando não são cancelados antes de chegarem ao fim, são concluídos com custos e prazos muito além dos estipulados e sem cumprir com todos os requisitos especificados para o produto (Cordeiro, 1999) (Glass, 1998), o que traz grandes prejuízos para o setor. Os profissionais de desenvolvimento de software passam mais tempo corrigindo defeitos do que com a inovação de sistemas.

A dinamicidade do processo de software dificulta o gerenciamento efetivo de projetos de software, devido às alterações constantes nos planos de projetos, redistribuição de atividades, inclusão/exclusão de atividades, adaptação de cronogramas, re-alocação de recursos, novos acordos com os clientes e entregas intermediárias não previstas. Um projeto de software também deve adaptar-se ao ambiente, dependendo da disponibilidade de recursos, ferramentas e habilidades da equipe (Roullier, 2001).

Portanto, o problema da melhoria da eficiência do processo de desenvolvimento de software ainda apresenta-se como um dos grandes desafios a serem superados nos próximos anos, não só devido ao aumento contínuo da demanda pela sociedade, mas principalmente pela sua magnitude econômica crescente. Observa-se que ainda permanecem associados ao desenvolvimento de software problemas crônicos como a

baixa qualidade dos produtos gerados, o alto grau de insatisfação dos clientes e o constante descumprimento das metas orçamentárias e prazos estabelecidos (Augusto, 1997).

Normalmente, pode-se verificar que muitas das tomadas de decisões vitais ligadas aos problemas de qualidade são tomadas sem fundamentação em fatos e dados concretos. Frequentemente prevalecem as experiências do gerente, acrescidas de opiniões emitidas pelos membros da equipe, muitas vezes sem maior embasamento teórico ou prático (Augusto, 1997).

Felizmente, na última década, houve uma grande preocupação com a modelagem e melhorias do processo de software. Há uma forte tendência de que nos próximos anos as organizações sejam pressionadas por seus concorrentes a reduzir substancialmente os prazos para entrega de produtos. As organizações que forem capazes de integrar, harmonizar, e acelerar seus processos de desenvolvimento e manutenção de software terão a primazia do mercado (Rocha, 2001).

4.3 – A Utilização do CMM

Desde 1993, a evolução da qualidade nas empresas de software no Brasil vem sendo acompanhada a partir de pesquisas diretas, conduzidas a cada dois anos pela Secretaria de Política de Informática e Automação do Ministério da Ciência e Tecnologia, no âmbito do Subcomitê Setorial da Qualidade e Produtividade em Software, do Programa Brasileiro da Qualidade e Produtividade (Weber, 2001). No último levantamento, publicado em 2002, foi constatado que o CMM é o modelo mais utilizado no Brasil, quando comparado com a Norma ISO/IEC 12207 e projeto SPICE (PBQP, 2002).

O CMM aparece como sendo conhecido por 75% e utilizado por 21% das organizações pesquisadas. Percentuais menores foram apurados para conhecimento (61%) e uso (4%) do SPICE (PBQP, 2002), como mostra o Figura 4.1.

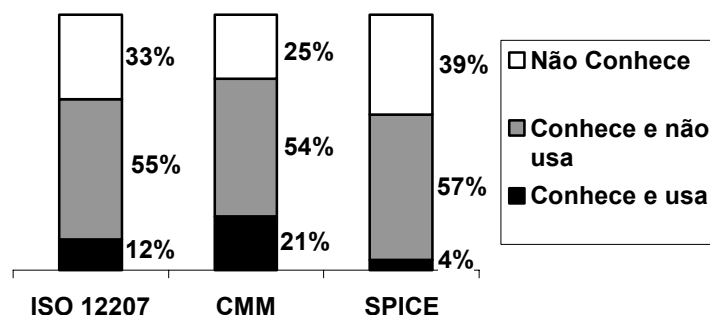


FIGURA 4.1 - Conhecimento do Modelo CMM, Norma ISO 12207 e SPICE.

FONTE: PBQP (2002).

As tabelas 4.1 e 4.2 mostram, respectivamente, os dados das duas últimas pesquisas realizadas (1999 e 2001), onde pode se ver um acentuado crescimento do conhecimento e utilização do SPICE. No entanto, o CMM ainda aparece como mais conhecido e utilizado.

TABELA 4.1 - Conhecimento dos Modelos CMM e SPICE – 1999.

Categorias	CMM		SPICE	
	Nº	%	Nº	%
Conhece e usa sistematicamente	8	1,8	1	0,2
Conhece e começa a usar	36	8,1	14	3,2
Conhece, mas não usa	165	37,2	121	27,3
Não conhece	234	52,8	308	69,4
Base	443	100	444	100

FONTE: PBQP (2001).

TABELA 4.2 - Conhecimento dos Modelos CMM e SPICE – 2001.

Categorias	CMM		SPICE	
	Nº	%	Nº	%
Conhece e usa sistematicamente	16	3,9	4	1,0
Conhece e começa a usar	71	17,1	13	3,2
Conhece, mas não usa	223	53,7	232	56,7
Não conhece	105	25,3	160	39,1
Base	415	100	409	100

FONTE: PBQP (2002).

O nível de conhecimento do CMM mais que triplicou, em termos percentuais, passando de 14%, em 1995, para 47% em 1999 (Weber, 2001), conforme o Figura 4.2. O SEI tem registro de 16 organizações brasileiras que se submeteram a uma avaliação CMM, e alcançaram até o Nível 3 de maturidade (Report, 2003).

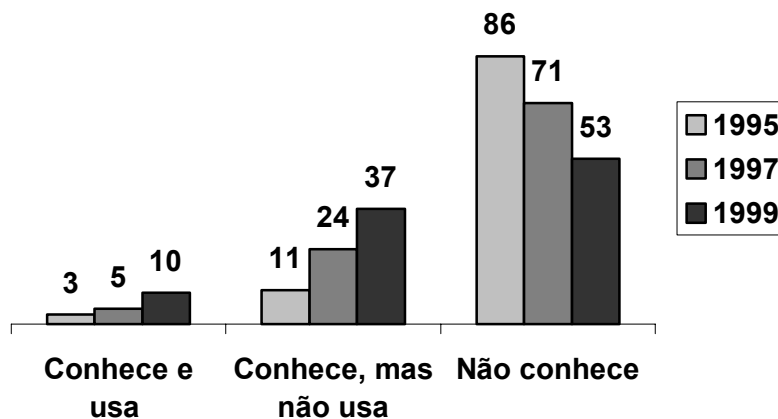


FIGURA 4.2 - Conhecimento no Modelo CMM (%).

FONTE: (Weber, 2002).

O MCT trabalha numa proposta de incentivo financeiro para estimular as empresas de desenvolvimento de software no Brasil a aprimorar a qualidade. A idéia inicial, estudada pelo grupo que coordena o Programa Brasileiro da Qualidade e Produtividade é a de garantir recursos que financiem a certificação de empresas com base em normas internacionais. A medida, considerada fundamental para o amadurecimento do setor, deverá resultar em três importantes benefícios: 1) softwares mais qualificados; 2) equilíbrio na balança comercial do setor; e 3) maior participação do Brasil no mercado internacional de software. Todos os benefícios conjugados resultariam na aceleração dos bons resultados que o setor tem conseguido nos últimos 10 anos. O CMM é o modelo que deverá ser adotado pelo Brasil para a certificação das empresas (SOFTEX, 2003).

Em termos mundiais, o sucesso e a crescente aceitação do CMM é comprovado pelo número de avaliações formais realizadas com base neste modelo. Segundo dados do SEI, entre os anos de 1987 e 2002, foram realizadas 2616 avaliações formais em 1978 organizações em 51 países. O número de reavaliações (mais de 500 organizações) também comprova a eficiência do modelo (Report, 2003). Em dez anos de existência, o CMM serviu para trilhar os caminhos de melhoria (formal e informalmente) em cerca de 5000 organizações (Report, 2003).

Devido a recente criação do CMMI, poucos registros ainda se têm sobre a implementação deste modelo, visto que a eficiência de um modelo só se avalia a médio e longo prazo da implementação de suas práticas. No entanto, quando analisado sob o ponto de vista de seu antecessor - o CMM - pode-se prever que o CMMI também será um sucesso. O CMMI incorpora toda a teoria de melhoria de processo do CMM, com conceitos que foram sendo gerados em anos de experiências práticas de implementação deste modelo em inúmeras organizações. Assim, pode-se afirmar que o CMMI é um modelo tão ou mais eficiente que o CMM.

Segundo dados do Relatório Anual de 2002 do SEI (Report, 2003), há registro de organizações com experiência em SW-CMM que estão migrando rapidamente para o

CMMI e adequando seus níveis de maturidade aos deste modelo. Além disso, verifica-se também um grande e crescente número de pessoas que recorrem aos cursos de CMMI oferecidos pelo SEI. As organizações que estão migrando para este modelo consideram esta experiência bastante positiva (Carter, 2002).

4.4 – O CMM e o Porte das Organizações

No Brasil, quando a implantação do CMM é analisada sob o ponto de vista do porte, verifica-se o reduzido número de micro e pequenas organizações que utilizam sistematicamente o modelo, conforme mostra a tabela 4.3. Porém, é importante destacar que o CMM foi criado para ser usado por todos os tipos de organizações, contudo, para organizações menores, ele precisa ser interpretado (CMMI, 2002b). A interpretação do CMM deve ser feita de modo que práticas já existentes na organização sejam aproveitadas e, se necessário, melhoradas para que contribuam para atingir as metas do CMM (Fiorini, 1999).

TABELA 4.3 – Conhecimento do CMM Segundo o Porte da Organização (%).

Categorias	Total	Micro	Pequena	Média	Grande
Conhece e usa sistematicamente	3,9	0,7	2,9	2,5	11,4
Conhece e começa a usar	17,1	3,4	20,4	30,0	29,5
Conhece mas não usa	53,7	62,2	48,9	47,5	48,9
Não conhece	25,3	33,8	27,7	20,0	10,2

FONTE: PBQP (2002).

Estudos mostram que há uma certa relação entre o nível de maturidade e o tamanho da organização, onde o nível de maturidade é diretamente proporcional ao tamanho da organização. As pequenas organizações tendem a apresentar níveis de maturidade mais baixos, enquanto as organizações maiores encontram-se em níveis mais elevados (McLain, 2001) (Report, 2003). Segundo o SEI, cerca de 28% das organizações

avaliadas com base no CMM no mundo são organizações com menos de 50 empregados (Report, 2003).

A definição do porte das organizações é um tanto quanto complexa, visto que há algumas divergências sobre o assunto. Uma organização classificada como de pequeno porte no Brasil pode não apresentar esta mesma classificação em outros países. No entanto, para se entrar em um consenso, tendo em vista a realidade do mercado nacional, é possível classificar as organizações por porte considerando sua força de trabalho, quer seja efetiva (sócios, dirigentes e empregados efetivos) quanto total (efetivos mais terceiros prestadores de serviço, bolsistas e estagiários), adotando-se como critério: microempresas (de 1 a 9 pessoas), pequena (de 10 a 49 pessoas), médias (de 50 a 99 pessoas) e grandes (100 ou mais pessoas) (PBQP, 2002). Sob este critério, confirma-se a predominância de micro e pequenas empresas, com percentual de 69% (força efetiva) e 61% (trabalho total), segundo a Figura a seguir.

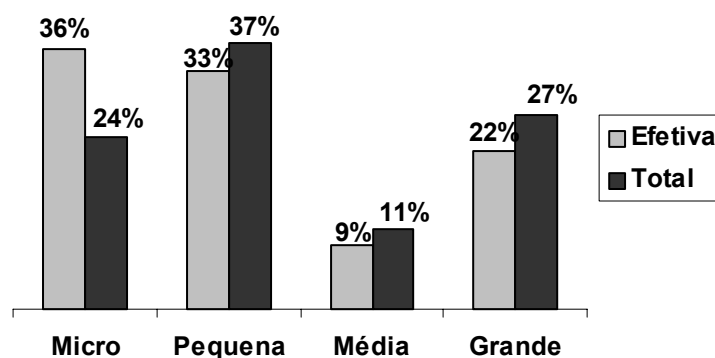


FIGURA 4.3 – Porte das empresas quanto a força de trabalho efetiva e total (2000).

FONTE: PBQP (2002).

Rosa (1997), na elaboração de uma proposta de avaliação baseada no SPICE, verificou que, no Brasil, a maioria das organizações de software é de pequeno porte, que muitas vezes possuem bons produtos, mas não têm condições financeiras de expandir suas atividades (Rosa, 1997).

Segundo dados do Plano Anual de Serviços do Instituto Brasileiro de Geografia e Estatística de 2000, empresas com até cinco pessoas representam 88% das empresas de Tecnologia da Informação e Comunicação (TICs), que lidam com o desenvolvimento de software, de soluções e aplicativos, processamento de dados, atividades de banco de dados, serviços de transmissão de dados, voz e imagem e hospedagem de sites (Araújo, 2003).

Logo, as organizações classificadas como de micro e pequeno porte juntas têm uma grande importância para o mercado econômico brasileiro. Porém, nestas mesmas organizações registram-se os menores esforços direcionados à qualidade na produção de software. A maioria das organizações que já implantaram ou estão em fase de implantação de programas de qualidade total, sistemas da qualidade ou similares, são organizações classificadas como de grande porte (PBQP, 2002).

Tendo em vista os dados mostrados na tabela 4.4, das organizações que tem implantado algum programa de qualidade, cerca de 21% caracterizam-se por micro e pequenas empresas, enquanto as médias aparecem com 37% e as grandes com 41%. Fica claro que por algum motivo micro e pequenas organizações tem mostrado pouco interesse na implantação destes programas.

TABELA 4.4 - Implantação de Programas de Qualidade (%).

Categorias	Total	Micro	Pequena	Média	Grande
Implantado	25,1	5,6	22,1	47,6	53,1
Em estudo ou implantação	26,2	23,6	32,4	23,8	22,9
Não tem	48,7	70,8	45,5	28,6	24,0

FONTE: PBQP (2002).

4.5 – Custos e Benefícios do CMM

Segundo Pádua (2003), a organização imatura comete erros clássicos. Muitos desses erros são relativos ao produto, decorrentes dos requisitos. Outros decorrem de enganos relativos aos fatores da produção: processos, pessoas e tecnologia.

Dos três fatores de produção, a tecnologia é o mais atraente. Existe um otimismo natural quanto aos resultados da aplicação de novas tecnologias. Porém, a tecnologia tem seu próprio ritmo de evolução. A tecnologia só oferece retorno do investimento quando colocada nas mãos de pessoas capacitadas, trabalhando dentro de processos adequados. Investir na capacitação de pessoas é absolutamente necessário. Entretanto não é fácil introduzir e manter um programa consistente de capacitação de pessoas. Formar pessoas é difícil, caro e demorado. Além disso, muitas pessoas produzem menos que a sua capacidade permitiria, que pode acontecer por falta de liderança, deficiência de apoio e inadequação de processos. Dos investimentos nos fatores de produção, as melhorias no processo trazem retornos em prazos relativamente curtos (Pádua, 2003).

Os anos de experiência com o modelo constataram que quanto maior o nível de maturidade alcançado, maior os índices de produtividade (medida pela habilidade de prever orçamento e cronogramas, pessoal eficiente, detecção precoce de defeitos, número de linhas de código produzidas por unidade de tempo), menor o tempo gasto no desenvolvimento de um produto, maior a qualidade do produto desenvolvido (medida pela satisfação dos clientes, número de defeitos, gastos relacionados com a manutenção corretiva) e maiores os benefícios alcançados como retorno dos investimentos (McLain, 2001).

Alguns dos benefícios que podem ser esperados pela implementação efetiva do CMM são: retorno de investimentos em menor prazo; diminuição de re-trabalho; redução de custos de re-testes; comunicação melhorada; diminuição de problemas de integração; redução de custos gerais; diminuição da média de taxa de defeito com aumento do tamanho dos produtos; aumento da produtividade; orçamentos coerentes e planejamento

de tempo adequado; aumento da moral do empregado; diminuição da pressão sobre o engenheiro de software, e respeito de outras organizações e clientes em função da clareza, prazos e boa documentação liberada.

Para algumas empresas já certificadas, os principais motivadores da melhoria de processo variam em torno da organização interna e da possibilidade de adotar processos que tornem o crescimento mais consistente. A organização interna e seu conseqüente ganho de eficiência parecem ser o principal benefício percebido pelas empresas recém certificadas. Mudanças contínuas e sistemáticas permitem a definição clara dos papéis de cada um dentro da empresa, os processos tornam-se previsíveis, sendo possível estimar prazos mais realistas subsidiados por argumentos técnicos.

Relatórios apontam que na relação custo benefício, observou-se um retorno de 5 para 1, ou seja, para cada dólar investido, economizaram 5 dólares posteriormente, em termos de redução de custos de re-trabalho e aumento de produtividade e qualidade (Spinola, 1997).

A única forma de medir o resultado da implementação de processos de qualidade é a medição contínua e agregada ao longo do tempo. Aumentos de produtividade superiores a 100%, e reduções de defeitos na entrega superiores a 70% são freqüentemente relatados. A qualidade dá resultado e se paga facilmente, mas é necessário esperar sua maturação. Falta de qualidade sai muito mais caro.

Seria de se indagar em quanto tempo um determinado nível CMM pode ser atingido. A passagem do Nível 1 (imaturo) para o Nível 2 pode levar mais tempo, a movimentação entre os outros níveis geralmente levará cerca de dois anos (Gonçalves, 2001). O SEI também reporta esses números com base nas avaliações realizadas no mundo, mostrando que o tempo médio de maturação de um nível para o outro é de 2 anos (do Nível 1 ao 2: 23 meses; do Nível 2 ao 3: 22 meses; do Nível 3 ao 4: 28 meses; do Nível 4 ao 5: 17 meses). Embora haja um intervalo de oscilação para esses números, uma organização precisaria ter uma razão muito boa para acreditar que pode fazer isso mais

rapidamente que a média mundial (Furlan, 2001). Os Níveis 2 e 3 são os que trazem maior retorno imediato (Fiorini, 1999).

A dificuldade em se alcançar os níveis depende do contexto da organização que busca a melhoria. Se uma organização começa do início (Nível 1), então é provável que achará o Nível 2 muito difícil. Satisfazer o Nível 2 pode ser uma tarefa complexa, pois as pessoas que trabalham no Nível 1 usualmente encontram-se pressionadas com os limites de prazos e custos, trabalham em ambientes pouco organizados, vendo as melhorias meramente como esforço adicional, e não como uma forma de reduzir esforços através de uma melhor organização do trabalho (Fiorini, 1999). Se conseguir atingir o Nível 2 é provável que considere os níveis sucessivos mais fáceis. Se já possui um programa de melhoria de processo implementado, baseado em outro modelo (por exemplo, ISO 9000), então poderia evoluir para o Nível 3 relativamente depressa. O Nível 4 seria o próximo grande desafio, pois inclui o uso de métodos estatísticos para controle do desenvolvimento de software, o que requer mais conhecimento de estatística aplicada do que o gerente de software comum possui. O Nível 5, finalmente, seria relativamente direto para os que já atingiram o Nível 4 (Furlan, 2001).

Ao mesmo tempo em que demanda altos investimentos, o processo de conquista da CMM não é garantia de novos negócios, ou mesmo de transição interna tranqüila. Alguns desafios se colocam para as organizações e os benefícios podem ser percebidos ao longo prazo.

O custo de um programa de melhoria varia de acordo com o contexto em que ocorre. As avaliações com o intuito de obtenção de certificação formal no CMM são caras e consomem muito tempo. Assim, muitas organizações encontram dificuldades em executa-las (Wieggers, 2000). Várias organizações têm buscado aperfeiçoar o seu processo de desenvolvimento a partir de referências do modelo CMM, mas não buscam uma certificação formal, apenas usam o modelo para estabelecer um caminho de melhoria no processo de desenvolvimento e manutenção do software. Estas organizações deverão adaptar o CMM à sua realidade e implementar melhorias

focalizadas (uma ou outra área chave). Neste caso, a organização pode optar por realizar a avaliação por sua conta, baseada nos documentos do SEI, ou contar com o auxílio de consultores especializados. Esta abordagem tem custos mais baixos.

4.6 – CMM em Pequenas Organizações

No início deste trabalho foi feita uma pesquisa empírica em organizações brasileiras que obtiveram certificação formal no CMM e com aquelas que se apóiam neste modelo informalmente. Neste estudo, foram abordadas as maiores dificuldades que estas organizações encontram na implementação do CMM.

A primeira dificuldade encontrada é a escolha do modelo a utilizar (CMM, SPICE ou outras normas ISO). As organizações não têm conhecimento sobre qual modelo atenderá melhor às suas necessidades, evidenciando o desconhecimento sobre o conteúdo dos modelos disponíveis e a necessidade de uma maior publicidade sobre o tema.

A complexidade do CMM é um fator que dificulta sua implementação. Até mesmo as organizações maiores, encontram barreiras para interpretar as práticas do CMM. Outra dificuldade é a interpretação das respostas do Questionário de Maturidade do CMM (Zubrow, 1994) para o levantamento da qualidade do processo. O fato da maioria da documentação do CMM estar em inglês também é considerado um obstáculo à implementação do modelo.

Outra barreira comum é a necessidade do envolvimento e a disponibilidade das pessoas, para reuniões, planejamentos e workshops. O apoio diretivo também é decisivo. O sucesso do método depende da interação das pessoas com o CMM, a falta de treinamento dificulta o envolvimento das pessoas.

A institucionalização das práticas também é dificultosa, visto que, muitas vezes uma prática é executada, mas diante de qualquer problema, os procedimentos são esquecidos

e parte-se para a codificação. Muitas organizações que desenvolvem softwares acabam desconsiderando os procedimentos quando aumentam as pressões de final de prazo.

O grau de dificuldade encontrado na implementação do CMM varia muito, dependendo de diversos fatores como cultura, número de profissionais envolvidos, conhecimentos do modelo, apoio de consultoria externa, objetivos de negócios, entre outros.

A eficiência do CMM é indiscutível, o que muitas vezes se questiona é a viabilidade de uma certificação formal em organizações menores, devido aos recursos que são indispensáveis à avaliação e melhoria do processo.

Segundo Rosa (1997), os modelos de avaliação e melhoria de processo de software mais conhecidos e utilizados por grandes organizações são difíceis de serem aplicados a organizações menores. Esta dificuldade ocorre porque estes modelos não se adaptam facilmente a realidade da pequena organização, onde o número de funcionários e o faturamento são reduzidos. Além disso, freqüentemente há orientações e responsabilidades citadas nestes modelos que são totalmente desconhecidos ou inadequados a um ambiente menor (Roullier, 2001).

São normalmente duas as razões que distanciam as organizações de pequeno porte da adoção de um modelo de melhoria de processo, como o CMM.

- Dificuldade e/ou impossibilidade de satisfazer as práticas CMM.
- Custo para implementar um programa de melhoria de processo (Broadman, 1994).

O CMM mostra-se eficiente em empresas de médio e grande porte, onde níveis sucessivos do CMM são crescentemente atingidos. Contudo, as pequenas empresas freqüentemente enfrentam problemas de falta de recursos, o que torna difícil manter um foco e interesse em atingir o próximo nível CMM, o qual representa uma grande melhoria ao nível corrente.

É verdade que o modelo CMM foi desenvolvido de início tendo-se em mente as grandes empresas que forneciam software para a Força Aérea Americana. Entretanto, foi feito um esforço para que o texto do CMM fosse tão genérico quanto possível, de modo a permitir a aplicação do modelo a organizações de qualquer tamanho.

O núcleo do problema é que o texto do CMM é passível de interpretação, isto é, é necessário não entender muito literalmente algumas coisas. A forma mais literal de interpretação do CMM aplica-se principalmente a organizações grandes. A necessidade destes vários grupos, por exemplo, pode assustar muita gente em organizações menores. A impressão que dá é que terá mais gente nestes grupos do que desenvolvendo o software propriamente dito. O problema está em se interpretar o conceito destes grupos. (Belloquim, 1999). Na verdade, tudo é uma questão de se trabalhar com "papéis", isto é, uma mesma pessoa pode assumir diferentes papéis ao longo do projeto, participando de vários daqueles grupos. O importante é que as atividades sejam cumpridas e que se saiba claramente quem são as pessoas responsáveis por cada uma. Portanto, é possível que estes grupos existam em organizações de qualquer tamanho.

É importante destacar que a interpretação do CMM para atender as necessidades da pequena empresa não é um trabalho simples (Broadman, 1994). No processo de implantação do CMM em uma organização pequena é necessário saber interpretar as regras gerais em um contexto menor. Deve-se, também, ter muito cuidado para não torna-lo uma burocracia em pequenas organizações (Fiorini, 1999).

Segundo Paulk (1998), um dos "pais" do CMM, os objetivos das áreas de processo, definidas pelo CMM, são suficientemente gerais para serem aplicadas em organizações ou projetos de qualquer tamanho, embora sua implementação possa ser radicalmente diferente. Paulk afirma que o objetivo primário da pequena empresa é a sobrevivência. O principal problema no início do processo de melhoria é decidir se a situação atual é mesmo insatisfatória, em que a melhoria ajudará, encontrar os recursos e assumir responsabilidades para a melhoria do processo e para definir e dispor de processos

existentes. Um dos requisitos para o sucesso do CMM é investir tempo e dinheiro na melhoria do processo (Paulk, 1999).

As dificuldades de implementação do CMM, relacionadas com a adequação das práticas em ambientes menores, já foram abordadas e soluções foram propostas como: o Processo Pessoal de Software, ou PSP - *Personal Software Process* (Humphrey, 2000a); o Processo de Software para Equipes, ou TSP - *Team Software Process* (Humphrey, 2000b), e o próprio CMMI.

Com o intuito de iniciar a melhoria de processo em nível pessoal, Humphrey propôs uma série de processos pessoais. O PSP é um modelo pessoal, que objetiva treinar o programador para que este seja capaz de definir seus próprios processos de desenvolvimento e medir o seu trabalho. Em resumo, o PSP descreve uma seqüência de pequenos projetos, que devem ser realizados seguindo rigorosamente os processos, que incluem um conjunto de formulários, scripts e relatórios predefinidos (Humphrey, 2000a). Como uma seqüência natural, Humphrey introduziu o TSP, que é uma disciplina de desenvolvimento em equipe baseada no PSP (Humphrey, 2000b).

É notório que qualquer tipo de atividade que provoque mudanças causa algum tipo de desconforto. A implementação do CMM não é trivial, obviamente que apresenta algumas dificuldades para todos os tipos de organização que queira implementá-lo, pode-se admitir também que em pequenas organizações estas dificuldades são ainda maiores (como custo e interpretação das práticas). Porém, organizações menores têm algumas vantagens em situações de mudança de cultura, pois geralmente apresentam menos inércia e burocracia, além da maior facilidade de comunicação e acompanhamento de projetos.

CAPITULO 5

MELHORIA DO PROCESSO DE SOFTWARE

5.1 – Diretrizes Iniciais

São muitos os obstáculos enfrentados pelas organizações na busca pela melhoria da qualidade de seu processo de desenvolvimento e manutenção do software. No entanto, os benefícios colhidos pela implementação de um programa de melhoria bem direcionado são comprovadamente gratificantes.

Começar é talvez a mais importante e mais difícil parte de qualquer trabalho. No caso da melhoria do processo, o levantamento de como o software está sendo desenvolvido na organização é a primeira atividade a ser executada. O sucesso de um programa de melhoria do processo depende grandemente da confiabilidade dos dados adquiridos na fase de avaliação, visto que uma avaliação competente permite a elaboração de um plano de melhoria mais eficiente e compatível com a realidade da organização. O objetivo de um programa de melhoria de processo não é simplesmente impor regras pré-definidas de como fazer as coisas, mas é necessário levar em consideração as características e necessidades da organização em questão.

Este trabalho busca minimizar as dificuldades que as organizações encontram para sair da inércia da imaturidade do processo, isto é, aqui são propostas diretrizes para a criação de um método que permita a elaboração de uma avaliação inicial menos rigorosa, mas efetiva. Porém, é difícil, se não impossível, uma proposta que possa ser útil a todos os tipos de organizações. Melhorar a qualidade do processo envolve algumas variáveis, tais como: as características e necessidades da organização, as pessoas envolvidas, as práticas já existentes, os recursos disponíveis, entre outras.

A necessidade de se adaptar os modelos existentes à realidade da organização que deseja implementá-los, comprova a subjetividade quanto ao direcionamento de um

programa de melhoria. Segundo Fiorini (1999), não existem soluções gerais, simplesmente porque cada caso é um caso. Qualquer processo de desenvolvimento utilizado precisa ser adaptado à organização e aos projetos por ela desenvolvidos.

Para que este trabalho pudesse ser realmente útil na introdução aos conceitos de qualidade de processo, considerando a impossibilidade de atender a todas as organizações, foi necessária a definição clara do escopo da organização a ser enfocada. Depois de escolhido o escopo da organização, foi necessário a escolha do modelo mais adequado às necessidades da mesma.

5.2 – Escolha do Escopo da Organização

As características de uma avaliação variam de acordo com o tipo da organização e seus interesses. Assim, é preciso a escolha e a determinação clara do ambiente para o qual a avaliação será desenvolvida.

O termo organização, aqui utilizado, deve ser compreendido como uma empresa privada, instituição governamental ou um departamento dentro de uma organização, que desenvolva software como produto de venda ou para uso próprio.

Como constatado anteriormente, as organizações que desenvolvem software, classificadas como de micro e pequeno porte (PBQP, 2002), são de grande importância para a economia nacional (Araújo, 2003). Uma infeliz coincidência é que estas mesmas organizações são as que mais se distanciam dos termos de qualidade de software. Trabalhos direcionados a estas organizações são de grande valia para a engenharia de software.

No entanto, a classificação por porte ainda é muito vasta para o contexto deste trabalho, pois as organizações classificadas como de micro e pequeno porte são aquelas com até 49 pessoas (PBQP, 2002). A introdução de um programa de melhoria em uma

organização que possui 10 pessoas desenvolvendo software, pode ser bastante distinta de um programa direcionado a uma organização com quase 50 pessoas.

Com o intuito de limitar mais o conceito de “pequenas organizações” aqui utilizado, este trabalho está direcionado a organizações que possuem de 5 a 15 profissionais envolvidos no desenvolvimento de software. Esta é uma média razoável, visto que, em organizações muito pequenas (neste caso, com menos de 5 profissionais), o estabelecimento de padrões ou procedimento específicos para o processo pode ser inviável, justamente pelo acentuado grau de informalidade presente na cultura das mesmas. Por outro lado, quanto maior o número de pessoas envolvidas, maior a complexidade para a introdução de padrões e mudanças de procedimentos. Para garantir a eficiência de um programa de melhoria em organizações com mais profissionais (no caso, mais de 15 pessoas), pode se fazer necessário um apoio especializado e maior disponibilidade de recursos.

O escopo da organização a que se destina este trabalho, não se define apenas pelo número de profissionais. Há ainda outros fatores a serem observados. Este trabalho propõe as diretrizes para que organizações com recursos limitados possam caminhar em direção ao mínimo de qualidade no processo de desenvolvimento. Assim, a primariedade das práticas aqui propostas, não serão úteis para organizações que já tiveram contato com um programa de melhoria.

Em resumo, os três fatores definidores do escopo organizacional são: o tamanho da organização, a capacidade do processo atual e a disponibilidade de recursos. Portanto, este trabalho se destina a pequenas organizações (de 5 a 15 profissionais de software), que apresentam um processo de software imaturo e poucos recursos disponíveis a um programa de melhoria.

É válido salientar que uma organização caracterizada como imatura pode desenvolver produtos de software com qualidade, mas seu desempenho depende da competência das pessoas. Mudando as pessoas a qualidade pode cair. Assim, os maiores problemas são

de ordem gerencial e não de ordem técnica (Rocha, 2001). A maturidade de uma organização em engenharia de software mede o grau de competência técnica e gerencial que esta organização possui para produzir software de boa qualidade, dentro de prazos e custos razoáveis e previsíveis (Pádua, 2003).

5.3 - Escolha do Modelo de Melhoria

Na corrida pela qualidade do processo de desenvolvimento de software, uma dos primeiros desafios é a escolha do modelo de melhoria a ser utilizado. A escolha do modelo deve ser baseada nas necessidades da organização.

Como já mencionado (na seção 4.3), o CMM é um modelo já consagrado e começa a ser fortemente utilizado no Brasil. Este trabalho é baseado na mais nova evolução deste modelo - o CMMI. Este modelo, além de reunir as mais novas práticas de excelência em engenharia de software, apresenta facilidade de adaptação aos mais variados tipos de organização e é compatível com outras normas como SPICE.

Outro atrativo do CMM (e seus derivados) é a disponibilidade de informações, tanto do modelo em si, quanto a sua base conceitual e a interpretação por avaliadores (Paula, 2003), na Internet pelo endereço <http://www.sei.cmu.edu>

A estrutura flexível do CMMI é bastante interessante, permitindo que este modelo possa se adaptar às mais diferentes realidades. O CMMI oferece muitas opções visando a adaptabilidade do modelo ao escopo da organização. Assim, são necessárias outras escolhas referentes a este modelo, como: a abordagem, a perspectiva, a disciplina e as áreas de processo.

5.4 - Escolha da Abordagem

O CMMI é apresentado em duas abordagens: abordagem em estágios e abordagem contínua (que foram respectivamente apresentadas nas seções 3.2.3 e 3.2.4). A escolha

da abordagem a ser utilizada também deve ser cuidadosa. Enquanto a abordagem em estágios usa os níveis de maturidade para medir a melhoria do processo, a abordagem contínua utiliza níveis de capacidade.

Os níveis de maturidade, que pertencem à abordagem em estágios, avaliam a maturidade da organização como um todo. Há 5 níveis de maturidade e cada nível compreende um conjunto pré-definido de áreas de processo.

Os níveis de capacidade, utilizado pela abordagem contínua, avaliam a melhoria de cada área de processo. Pode-se dizer que a abordagem contínua é mais flexível, mas mais complexa de administrar. É indicada para organizações que não tem necessidade de implementar todas as áreas de processo, podendo priorizar a melhoria de áreas mais importantes ou urgentes. Além de dar flexibilidade de escolha na ordem de prioridade na melhoria destas áreas (CMMI, 2002b).

Uma outra situação em que a organização se beneficia da abordagem contínua é quando existe uma carência de suporte pela alta administração. Se um gerente de nível médio quer implementar o CMM, mas não consegue sensibilizar a alta administração, ele pode, em escala reduzida, implantar um conjunto menor de áreas de processo, com maior rapidez e custo mais baixo, de modo a ser capaz de mostrar resultados que facilitem “a venda” do CMM para a alta administração (Belloquim, 2002).

Porém, a representação contínua pode dar a falsa impressão de que as áreas de processo do CMMI são completamente independentes, que é possível ser Nível 5 em algumas áreas e Nível 0 em outras. Embora esta representação seja mais flexível na hora de priorizar a implementação de práticas, existem dependências entre as áreas que não podem ser ignoradas (CMMI, 2002b).

A abordagem contínua, em particular, pode ser de grande valia para organizações que estejam começando sua jornada nos caminhos da melhoria da qualidade dos processos

de software (Belloquim, 2002). Portanto, neste trabalho foi escolhida a abordagem contínua, por ser a mais adequada para o escopo da organização em questão.

5.5 - Escolha da Perspectiva

O CMMI pode ser implementado sob duas perspectivas: na forma de uma auto-avaliação para melhoria do processo interno ou na forma de uma avaliação externa para fins de comparações em processos de terceirização (como mostradas na seção 3.2.5).

A avaliação aqui proposta não tem o objetivo de enquadrar uma organização em um nível de maturidade/capacidade do CMMI. Este modelo está sendo utilizado na determinação das práticas necessárias a melhoria do processo interno de desenvolvimento e manutenção do software. Este trabalho propõe uma estratégia para que a organização auto-avalie seu processo de software, de modo a identificar e priorizar os pontos que necessitem de maiores esforços de melhoria.

Além disso, este trabalho propõe uma avaliação informal. Uma avaliação é formal quando realizada por um auditor certificado pelo SEI. A avaliação aqui proposta é bem menos rigorosa, isto é, não cumpre com todos os requisitos de uma avaliação CMMI definidos no ARC (2001), assim seus resultados não podem ser comparados com avaliações formais deste modelo.

5.6 - Escolha da Disciplina

Os modelos CMMI são direcionados em áreas de interesse ou disciplinas (Engenharia de Software, Engenharia de Sistemas, Desenvolvimento de Processo e Produto Integrado e Contratos de Fornecedores, apresentadas na seção 3.2.1). A escolha da disciplina depende das características da organização e do que se quer melhorar, lembrando que uma das características do CMMI é o oferecimento de modelos que integram estas disciplinas.

O presente trabalho foi baseado no modelo CMMI direcionado à disciplina de Engenharia de Software, sendo assim, foi utilizado o modelo CMMI-SW (CMMI, 2002b). Porém, vale citar que, como neste trabalho foram utilizadas somente as práticas básicas, tais práticas são encontradas em todos os modelos CMMI que incluem a disciplina Engenharia de Software.

As escolhas feitas até o momento, permitiram a definição de uma base para o levantamento da qualidade do processo de software. Em resumo, este trabalho propõe as diretrizes para o desenvolvimento de um método de auto-avaliação menos rigorosa, baseado no modelo CMMI para Engenharia de Software em sua abordagem contínua - o CMMI-SW Contínuo (CMMI, 2002b), direcionado a pequenas organizações que disponibilizam de poucos recursos para iniciar um programa de melhoria de processo de software.

5.7 - Escolha das Áreas de Processo

A escolha das áreas a serem avaliadas na abordagem contínua não é uma tarefa simples, pois depende a realidade da organização em questão (como seus objetivos de negócio, recursos disponíveis, entre outros). O CMMI descreve 22 áreas de processo. As organizações que estão dando seus primeiros passos em direção a melhoria de processo podem encontrar dificuldades na escolha das áreas a serem enfocadas em um primeiro momento. Obviamente que, para a obtenção da excelência em qualidade de software, o CMMI aconselha que todas suas áreas sejam enfocadas gradualmente.

A observação do histórico de sucesso do SW-CMM (Paulk, 1993) pode auxiliar na escolha destas áreas. O SW-CMM baseou-se em algumas premissas para organizar parte do conhecimento em engenharia de software em um modelo que concretamente pudesse auxiliar na melhoria do processo. Uma das premissas é a de que os maiores problemas das organizações de software eram gerenciais e não técnicos. Essas premissas apontavam para soluções que, em um primeiro momento, focassem na utilização de princípios básicos de gerencia de projetos para “arrumar a casa”, gerar resultados

imediatos e preparar a organização para as próximas etapas de melhoria. Sem uma gerência de projetos bem estabelecida, o risco de qualquer outra iniciativa não produzir os resultados esperados é muito grande.

As primeiras áreas de processo a serem satisfeitas pelo SW-CMM (e também CMMI) são referentes a aspectos gerenciais, que visam orientar a organização para que esta seja capaz de planejar suas atividades e cumpri-las. Este aspecto dos modelos CMM é muito importante, pois muitas vezes a organização possui conhecimento técnico, para o desenvolvimento de software, mas faltam os conhecimentos administrativos, o que é essencial para que a organização cresça e seja competitiva.

O SW-CMM define 5 níveis incrementais de maturidade a ser buscada, dos quais o primeiro nível é apenas um ponto de referência (como no CMMI em estágios). O Nível 2 é a primeira referência a ser buscada, é focado exatamente no estabelecimento de uma gerência de projetos. Os processos básicos de gestão de projeto são estabelecidos para acompanhar custo, cronograma e funcionalidade. O Nível 2 é caracterizado pela existência de um processo de gerenciamento de software onde os controle sobre os procedimentos, compromissos e atividades são bem fundamentados.

A eficiência comprovada do SW-CMM, com suas características de melhoria contínua e em níveis, confirma a eficiência de um processo gradual de melhorias. Como um nível de maturidade é construído sobre outro e a tentativa de saltar um nível é improdutiva, pode-se afirmar que as áreas de processo do Nível 2, as primeiras a serem enfocadas, são a base para as áreas dos níveis seguintes.

Na rota do SW-CMM, o caminho para a melhoria dos processos começa pela capacitação em cumprir compromissos. A organização deve aprender a cumprir o que promete a seus clientes em termos de requisitos, de custos, de prazos e de qualidade. Isso não significa eliminar as incertezas inerentes a qualquer planejamento, mas sim que os erros de previsão deverão se tornar pequenos e aleatórios. O comprometimento é o

tema central do Nível 2 do SW-CMM, suas áreas de processo têm como foco o cumprimento dos compromissos (Pádua, 2003).

As principais metas de melhoria de processo no Nível 2 de maturidade do SW-CMM podem ser resumidas em:

- Definir e gerenciar requisitos do projeto, incluindo as mudanças destes requisitos.
- Comprometer-se apenas com o trabalho estimado e planejado.
- Acompanhar a execução do trabalho em relação ao planejado.
- Garantir que o trabalho satisfaça os padrões de qualidade estabelecidos.
- Manter um controle rigoroso dos produtos, incluindo as mudanças destes produtos.
- Garantir que todos os fornecedores sigam estes padrões.
- Monitorar o andamento de todos os itens anteriores.

Diante da importância do Nível 2 no processo de melhoria baseado no CMM e o escopo da organização aqui enfocada, este trabalho propõe que a escolha das áreas a serem avaliadas seja feita entre as que compõem o Nível 2 do SW-CMM.

O CMMI em estágios possui a mesma estrutura do SW-CMM. O nome dos níveis de maturidade foi modificado para dar maior precisão. Muitas áreas de processo tiveram pequenas modificações de nome e conteúdo. A área de processo Medições e Análises passou a ser uma área do Nível 2 (antes compreendia um grupo de atividades responsável pela institucionalização das outras áreas de processo).

Apesar de o Nível 2 de maturidade ter nomes distintos nos modelos (Nível Repetitivo no SW-CMM e Nível Gerenciado no CMMI em estágios), possuem o mesmo enfoque. A tabela 5.1 mostra a correspondência entre as áreas de processo do Nível 2 do SW-CMM e CMMI em estágio.

TABELA 5.1 Áreas de Processo do Nível 2.

SW-CMM – Nível Repetitivo	CMMI em estágios – Nível Gerenciado
Gerenciamento de Requisitos	Gerenciamento de Requisitos
Planejamento do Projeto	Planejamento de Projeto
Visão Geral e Acompanhamento do Projeto	Controle e Monitoração de Projeto
Gerenciamento de Subcontratos	Gerenciamento de Contrato de Fornecedores
Garantia da Qualidade de Software	Garantia de Qualidade de Produto e Processo
Gerenciamento de Configuração	Gerenciamento da Configuração
	Medições e Análises

FONTE: (Paulk, 1993) e (CMMI, 2002b).

Assim, mantendo a base deste trabalho no modelo CMMI, recomenda-se que as primeiras áreas a serem enfocadas sejam:

- Gerenciamento de Requisitos.
- Planejamento de Projeto.
- Controle e Monitoração de Projeto.
- Gerenciamento de Contrato de Fornecedores.
- Garantia de Qualidade de Produto e Processo.
- Gerenciamento da Configuração.
- Medições e Análises.

A grande maioria das organizações que desenvolvem software encontram-se no Nível 1 de maturidade. O caminho para o Nível 2 é talvez o mais difícil de ser percorrido, pois trata-se de tirar a organização da inércia da imaturidade. Considerando os níveis de maturidade do SW-CMM (de 1 a 5), seria razoável esperar que uma organização-padrão de software estivesse nivelada em 3. Mas este não é o caso. Mesmo depois de vários anos de experiência da indústria de software com o modelo CMM, os níveis 4 e 5 foram conquistados por poucas organizações. Apesar da popularidade do SW-CMM, somente uma pequena parcela do universo de organizações de software se submeteu a uma avaliação formal. Nos relatórios publicados pelo SEI, com o resumo das avaliações mostrando evolução na maturidade, a média ainda é nível 2. Assumindo-se que

organizações que não se preocupam com melhoria de processo de software apresentam baixa maturidade, ter atingido o nível 2 representa uma realização extraordinária.

As áreas de processo do Nível 2 são as áreas que as organizações que buscam melhorar seu processo de desenvolvimento devem focar em um primeiro momento. Porém, mesmo dentro destas áreas pode-se destacar algumas de maior prioridade. Priorizar algumas áreas não significa que elas sejam mais importantes que outras, pois na maioria das vezes, o sucesso de uma área depende da eficiência de outra. Indicar algumas áreas mais relevantes pode ser útil para organizações que não querem ou não podem focar todas as áreas em uma primeira avaliação.

Se os recursos (financeiros e pessoal) que a organização têm disponíveis a um programa de melhoria são limitados, é interessante que as melhorias sejam introduzidas de forma mais lenta e objetiva. Melhorias bem direcionadas e incorporadas à organização gradualmente, evita desperdícios de recursos e pode ser uma garantia de que estas melhorias sejam duradouras e eficientes. Além disso, mudanças graduais são menos traumáticas. Mudanças radicais podem provocar aversão e desestimular os profissionais envolvidos.

Rosa (1997), recomenda às pequenas organizações atenção maior para as áreas: Planejamento de Projetos, Gerenciamento de Configuração, Gerenciamento de Requisitos e Garantia da Qualidade de Software.

Segundo Pádua (2003), requisitos, custo e prazo formam os vértices do triângulo crítico da engenharia de software, que deve ser sustentado pelas disciplinas: gerenciamento de requisitos, planejamento e controle do projeto, garantia da qualidade, gerenciamento da configuração e gerenciamento de contratos (em caso de terceirização).

Os requisitos são as características que definem os critérios de aceitação de um produto. Um dos grandes problemas da engenharia de software é o levantamento e a documentação dos requisitos do produto. Quando este levantamento e a documentação

são bem-feitos, os requisitos têm maior chances de serem atendidos. Porém, os requisitos são bastante instáveis. Esta instabilidade tem um custo muito alto, geralmente significa perder trabalho já feito, desfazer algumas coisas e remendar outras. O Gerenciamento dos Requisitos é a disciplina da engenharia de software que procura manter sob controle o conjunto dos requisitos de um produto, mesmo diante de algumas alterações inevitáveis (Pádua, 2003).

Porém, para um produto ser viável, não basta que atenda aos requisitos desejados, tem que ser produzido dentro de parâmetros de custo e prazo. Se isto não for possível, o produto pode não ser viável do ponto de vista de mercado, ou pode ser preferível adquirir outro produto, ainda que sacrificando alguns dos requisitos. Para cumprir compromissos de prazos e custos, esses compromissos devem ser assumidos com base em requisitos bem-levantados, analisados e documentados. Os planos dos projetos devem ser feitos com boas técnicas de estimativas e análise de tamanho, esforços, prazos e riscos. Essas técnicas pertencem à disciplina de Planejamento de Projetos (Pádua, 2003).

Todo plano comporta incertezas. A produtividade dos profissionais de software pode ser feita com base em projetos anteriores da organização, mas é afetada por muitas variações, que dependem de pessoas, processos e tecnologia. Riscos previstos e imprevistos podem ser materializados. Ao longo do projeto, os gerentes normalmente enfrentam problemas e devem controlar variáveis que afetam o cumprimento de seus compromissos. Algumas vezes, os problemas podem ser resolvidos por meio de contratação e remanejamento de pessoal ou de melhoria no uso de ferramentas. Outras vezes, é necessário renegociar requisitos, prazos e custos. Para renegociar é necessário re-planejar, atualizando as estimativas para levar em conta os outros dados. A disciplina complementar ao planejamento que auxilia neste momento é o Controle de Projetos (Pádua, 2003).

Um erro conceitual comum é imaginar que é possível trocar prazo, e talvez custo, por qualidade. É possível, em alguns casos, reduzir custos e prazos através da redução dos

requisitos do produto. A qualidade é consequência do processo, das pessoas e da tecnologia. A relação entre a qualidade e esses fatores, é complexa. Por isso, é mais difícil controlar o grau de qualidade de um produto do que controlar seus requisitos. Em todas as fases do desenvolvimento as pessoas acabam introduzindo alguns defeitos. Em todo bom processo existem atividades de Garantia da Qualidade, tais como: revisões, testes e auditorias. Estas atividades removem parte dos defeitos introduzidos. O tempo gasto no desenvolvimento é geralmente reduzido com o aumento da qualidade do processo (Pádua, 2003).

Um produto de software é composto por muitos artefatos (código executável, código fonte, modelos, relatórios, e outros). A maioria dos artefatos evolui ao longo do projeto, e até ao longo de toda a vida de um produto. Mesmo terminado um projeto, é importante que seus resultados sejam documentados e guardados, para eventuais manutenções. Organizar e controlar a proliferação dos artefatos é o objetivo da disciplina Gerenciamento da Configuração. Sem esta disciplina é impossível atingir sequer níveis razoáveis de qualidade (Pádua, 2003).

Algumas organizações terceirizam o desenvolvimento de software. O esforço de uma organização em melhorar a qualidade de seus sistemas informatizados pode ser perdido por causa de falhas nos contratados. Para que isso não ocorra, a organização contratante deve estar capacitada no Gerenciamento de Contratos (Pádua, 2003).

Diante destas considerações, o presente trabalho prioriza algumas áreas e recomenda que a escolha das áreas de processo a serem avaliadas em uma avaliação inicial, considere as seguintes áreas de processo do Nível 2 do CMMI em estágio (CMMI, 2002a):

- Gerenciamento de Requisitos.
- Planejamento de Projeto.
- Controle e Monitoração de Projeto.
- Garantia de Qualidade de Produto e Processo.
- Gerenciamento da Configuração.

As demais áreas de processo do Nível 2 do CMMI em estágios: Medições e Análises e Gerenciamento de Contrato de Fornecedores devem ser enfocadas em um segundo momento, sendo a última em caso de terceirização.

O número de áreas a serem avaliadas, depende da necessidade e recursos da organização. O importante é que todas elas sejam enfocadas em um processo de melhoria contínua. No entanto, a dependência existente entre estas cinco áreas, faz com que a real eficiência de uma, só é atingida com a implementação das outras.

É importante deixar claro que o CMMI em estágios só está sendo usado na escolha das áreas a serem enfocadas em um primeiro momento, devido a sua estrutura em estágio ser semelhante ao SW-CMM e ao sucesso deste modelo. O conteúdo e a estratégia de implementação das áreas escolhidas devem ser observados com base no CMMI contínuo (CMMI, 2002b).

5.8 – Importância do Gerenciamento de Projetos

Uma das grandes contribuições do CMM foi consolidar a importância da gerência de projetos para a engenharia de software. O foco das áreas Gerenciamento de Requisitos, Planejamento de Projeto, Controle e Monitoração de Projeto, Garantia de Qualidade de Produto e Processo, Gerenciamento da Configuração é o estabelecimento de procedimentos básicos de Gerenciamento de Projetos (estas áreas estão descritas em maiores detalhes no Anexo 2). A falta de planejamento é a primeira coisa a ser atacada, sem a qual qualquer esforço de melhoria é baldado. Uma equipe que desenvolve seu trabalho aleatoriamente não pode se beneficiar de técnicas de inspeção, metodologias de desenvolvimentos ou métricas.

Segundo Koontz e O'Donnell (Koontz, 1986), gerenciar consiste em executar atividades e tarefas que têm como propósito planejar e controlar atividades de outras pessoas para atingir objetivos que não podem ser alcançados caso as pessoas atuem por conta própria.

A importância deste conceito é facilmente percebida em um ambiente de projeto de software onde, com frequência, existem pessoas sem tarefas definidas e onde muitas atividades importantes deixam de ser realizadas (Rocha, 2001).

Alguns estudos e pesquisas que foram realizados nos anos 90 (Roullier, 2001) demonstraram que o gerenciamento de projeto é a causa mais evidente das falhas na execução e entrega de produtos e serviços de software.

Já em 1989, Humphrey (1989) constatou que a ausência de práticas administrativas no desenvolvimento de software é a principal causa de sérios problemas enfrentados pelas organizações, como: atrasos em cronogramas, custo maior do que o esperado e presença de defeitos, ocasionando uma série de inconveniências para os clientes e enorme perda de tempo e de recursos.

Na atual cultura das organizações de software, o planejamento, quando ocorre, é feito de forma superficial (Weber, 2001). Muitos projetos de software são realizados sem um planejamento de como a idéia modelada pelo levantamento de requisitos e necessidades dos clientes pode ser transformada em produto. Muitas organizações têm dificuldade na elaboração de cronograma a ser cumprido e também na estimativa de custos (Rosa, 1997).

Os gerentes de projeto estão desacostumados a estimar. Quando estimam, costumam basear-se em estimativas passadas, mesmo sabendo que elas podem estar incorretas (não sabem também precisar o quanto elas estão incorretas). Há gerentes que se recusam a estimar somente por julgarem perda de tempo e correrem o risco de obter resultados incorretos. Boas estimativas de custo, esforço e prazo de software requerem um processo ordenado que defina a utilização de métricas de software, método e ferramenta de estimativa. As organizações de software, de forma geral, não detêm conhecimentos e recursos para isso (Roullier, 2001).

Segundo Humphrey (1989), com poucas exceções, estimativas iniciais de recursos e cronogramas são inaceitáveis. Isto não ocorre por falta de competência dos programadores, mas porque os clientes geralmente desconhecem que construir software leva tempo.

Assim, as negociações de planejamento são testes críticos ao time gerencial. Este time deve tratar o plano inicial como ponto de partida e, quando prazos e custos tiverem que ser reduzidos, o escopo do trabalho deve ser igualmente cortado.

Observa-se que na prática, em ambientes de desenvolvimentos caóticos, qualquer esforço de melhoria, por mais simples que sejam, causam melhorias na produtividade. O simples esforço por melhorar a documentação do projeto e as verificações de qualidade desde o início, reduzem significativamente o tempo gasto em testes e re-trabalho.

As pequenas organizações também possuem dificuldades de parte gerencial. Na maioria das vezes, o conhecimento técnico das pessoas que trabalham em pequenas empresas não envolve assuntos relacionados à formação de empreendedores (Rosa, 1997). Outros fatores ainda agravam os problemas de gerenciamento de projetos de software em organizações de pequeno porte, tais como: inexistência de um processo definido, recursos pessoais e financeiros limitados, falta e/ou pouca cultura em processos, pouco treinamento em engenharia de software, imaturidade metodológica, crescimento ocorrido por demanda, falta de experiência administrativa por parte dos gerentes e diretores e falta de definição das metas organizacionais (Roullier, 2001).

As dificuldades de se implantar um processo para gerenciamento de projetos de software em organizações de pequeno porte está principalmente na falta de cultura em qualidade que as mesmas possuem. O alto rodízio de pessoal e o constante deslocamento de recursos humanos para suprir a demanda de um outro projeto, também dificulta o bom desempenho de um processo de gerenciamento de projetos de software (Roullier, 2001).

Rosa (1997), em uma pesquisa empírica feita em pequenas organizações de software no interior de São Paulo, verificou que a falta de formalismo é característica clara de organizações de pequeno porte, onde as tarefas de um modo geral são feitas sem procedimentos explícitos. As pequenas organizações não possuem um processo de software que possa ser considerado típico, pois quando realizam atividades de engenharia de software, as fazem informalmente. Atividades como documentação e controle de mudanças, quando realizadas, não são utilizados métodos e procedimentos específicos. Muitos profissionais consideram estas atividades muito burocráticas e não lhes dão a devida importância (Rosa, 1997).

A carência de práticas de gerenciamento distancia estas organizações dos conceitos básicos de qualidade em software. A formalização do gerenciamento de projeto é a base sobre a qual se constrói um processo de software eficiente. Um projeto gerenciado reflete no custo, planejamento e qualidade do produto desenvolvido. Este trabalho busca diminuir a barreira da informalidade que separa as pequenas organizações do caminho da excelência em qualidade de software.

Tendo em vista a carência de gerenciamento nas pequenas organizações, o presente trabalho prioriza algumas práticas indispensáveis à introdução de formalismo na gerencia de projetos. Tais práticas e suas referidas áreas de processo estão apresentadas no Anexo 2.

CAPÍTULO 6

LEVANTAMENTO DA QUALIDADE DO PROCESSO DE SOFTWARE

6.1 – Avaliações CMMI

A implementação do CMMI consiste em um ciclo composto pelo levantamento do estado atual do processo da organização (avaliação propriamente dita), comparação com o próximo nível de maturidade/capacidade, elaboração de um plano para reduzir a distância entre o estado atual e o almejado, e a execução das ações planejadas.

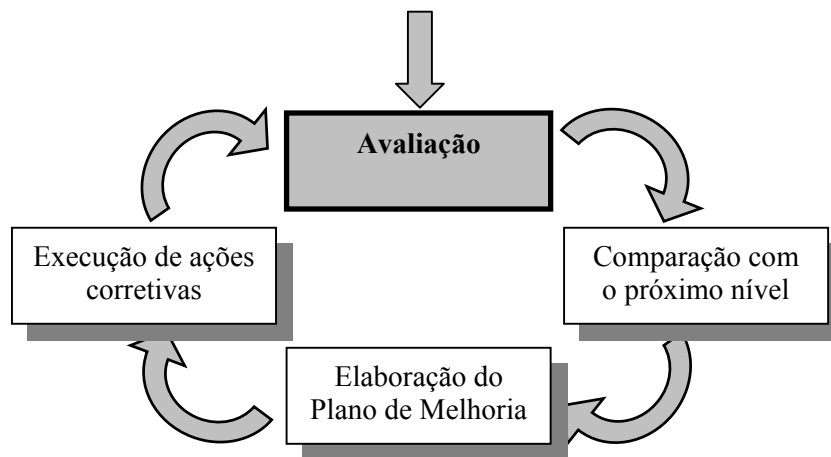


FIGURA 6.1 – Avaliação CMMI.

De modo geral os princípios básicos de avaliações CMMI são os mesmos princípios históricos de avaliações SW-CMM, que são (ARC, 2001):

- Iniciar com um modelo de referência (neste caso o CMMI-SW).
- Usar um processo de avaliação formalizado.
- Envolver a alta administração como os responsáveis pela avaliação (*sponsor*).
- Endereçar a avaliação aos objetivos de negócio do *sponsor*.
- Observar rigorosa confidência e não atribuição dos dados.
- Focalizar em ações de melhoria de processo.

Todos os requisitos para execução de avaliações CMMI estão descritos no ARC (2001). Contudo, as avaliações do CMMI não precisam cumprir rigorosamente todos os critérios definidos pelo ARC. A avaliação deve ter como foco principal os objetivos do *sponsor* (por exemplo, a alta administração ou um gerente de departamento). O ARC define 3 classes de avaliação: classe A, classe B e classe C. As classes se diferenciam basicamente no grau de rigorosidade da investigação e classificação dos processos, pessoal e recursos envolvidos, custo e duração. A avaliação Classe A é a mais rigorosa e deve satisfazer todos os requisitos do ARC.

O Método de Avaliação CMMI Padrão para Melhoria de Processo, ou SCAMPII - *Standard CMMI Assessment Method for Process Improvement*, é um método de avaliação Classe A (SCAMPI, 2001). Este método é utilizado em avaliações formais (internas e externas) do CMMI, sendo também compatível com a ISO 15504.

Uma avaliação CMMI se resume nas seguintes atividades (SCAMPI, 2001):

- Plano e preparo da avaliação.
 - o Analisar os requisitos da avaliação.
 - o Desenvolver um plano de avaliação.
 - o Selecionar e preparar o grupo avaliador.
 - o Definir os objetivos a serem alcançados e analisar as informações necessárias para atingir os objetivos.
 - o Preparar a coleta destas informações.
- Execução da avaliação.
 - o Coletar as informações.
 - o Examinar as informações.
 - o Verificar a validade das informações.
 - o Documentar as informações.
 - o Gerar resultados da avaliação
- Relatório de resultados.
 - o Entregar os resultados da avaliação aos interessados.
 - o Arquivar os documentos da avaliação.

Quando uma organização não busca a certificação formal no CMMI, ela pode criar seu método de avaliação próprio, baseado neste modelo, observando suas verdadeiras necessidades. Este trabalho apresenta aqui as diretrizes para a elaboração de um método de avaliação menos rigoroso, tendo em vista as características da organização e as escolhas referentes ao modelo CMMI, apresentadas no capítulo anterior.

6.2 – Avaliações Alternativas

A avaliação do SEI é bastante rigorosa e, certamente, traumática. Por isso, dificilmente deve ser utilizada como pontapé inicial na implantação do CMM. O primeiro ciclo de análise/planejamento/ação tem características especiais. Um método menos abrangente e ao mesmo tempo flexível pode reduzir os traumas e liberar a criatividade (Pires, 1998). A avaliação inicial deve ser o menos traumática possível, para que as pessoas não criem aversão ao modelo já neste momento. Embora, para fins de certificação, exista apenas uma forma de avaliação válida, definida pelo SEI (denominada, neste trabalho, de avaliação formal), existem outras técnicas que podem minimizar as dificuldades e encurtar o caminho até o próximo nível.

Uma avaliação formal do CMM é bastante abrangente, cara e consome tempo. Assim, muitas organizações encontram dificuldades em executá-las com frequência. Para amenizar as dificuldades e acompanhar cada passo do programa de melhoria, muitas organizações vêm criando técnicas de avaliação em pequena escala que são aplicadas entre avaliações completas. Estas mini-avaliações (*mini-assessment*) normalmente focalizam poucas áreas de processo, servindo para estimular o processo de melhoria e verificar se uma organização está preparada para uma avaliação em escala completa (Wieggers, 2000). As grandes organizações que objetivam alcançar níveis de maturidade mais elevados, freqüentemente fazem uso destas mini-avaliações (Carter, 2002) (Dunaway, 1999). Normalmente, a maioria das avaliações formais é precedida de mini-avaliações. Organizações como Motorola, Xerox e Kodak, têm aplicado esta estratégia com bastante sucesso, alcançando altos níveis de maturidade (Wieggers, 2000).

Avaliações em pequena escala mostra-se como uma alternativa também para organizações que buscam melhorar seu processo de desenvolvimento sem a necessidade de uma certificação formal. Técnicas de mini-avaliações podem ser eficientes e de custo acessível (Carter, 2002).

As avaliações classe C, por terem um conjunto menor de requisitos, mostram-se como uma alternativa bastante interessante para uma avaliação inicial e em pequena escala. Estas avaliações são uma opção para organizações menores que não possuem recursos para uma avaliação completa. Portanto, avaliações classe C são as que mais se enquadram no contexto deste trabalho. Ressaltando que a classe da avaliação tem relação com a rigorosidade da mesma e não com o tamanho da organização.

O SEI oferece todo o apoio teórico para avaliações classe A, através do ARC e do SCAMPI, pois seu objetivo primordial é a certificação formal das organizações de software. Um método para avaliações menos rigorosas pode ser definido pela adaptação do SCAMPI a um contexto menor, baseando nos requisitos classe C do ARC. Um método de avaliação mais rigoroso, como o SCAMPI (2001) pode ser utilizado como guia para elaboração de uma avaliação em pequena escala. Obviamente a grupo avaliador deve extrair deste método apenas o que julgar importante para a melhor obtenção de resultados.

Na implementação informal do CMMI, isto é, sem a necessidade de certificação, a avaliação pode ser executada ou por profissionais da própria organização, ou através do apoio de consultorias especializadas, ou ainda por ambos.

Quando a iniciativa de avaliação vem dos próprios profissionais da organização, o SEI recomenda que eles tenham experiência em avaliações de processo e conhecimento no CMM. Porém, é importante considerar a hipótese da ausência destas experiências. Esta não é a situação mais indicada pelo SEI para a execução de avaliações, mas isso não exclui totalmente a possibilidade de que a avaliação ocorra. Neste caso, é fundamental que estes profissionais tenham no mínimo larga experiência em desenvolvimento de

software. Assim, antes de qualquer iniciativa é imprescindível que os profissionais delegados à execução da avaliação, se instrua sobre o modelo. No Brasil, há consultorias que disponibilizam cursos de CMM. Além disso, uma importante fonte de informação é o próprio SEI, que disponibiliza aos interessados a documentação do CMM (e seus derivados) via Internet, sem qualquer custo. O sucesso neste caso vai depender da dedicação e desenvoltura destes profissionais.

Tendo em vista a disponibilidade destas fontes de informação, o menor problema destes profissionais ainda será conhecer o modelo e suas especificações. O maior problema será adaptar as informações adquiridas em um contexto reduzido e em uma avaliação inicial, considerando ainda as características da organização a ser avaliada.

6.3 – Preparação da Avaliação

Na preparação da mini-avaliação aqui proposta é necessário considerar: o escopo da avaliação (avaliação inicial em pequena escala) e o escopo da organização (pequenas organizações que buscam sair da imaturidade do processo), adicionando ainda a escassez de recursos para a avaliação e a falta de experiência em CMM de seus profissionais.

Pode até parecer contraditório, mas apesar da informalidade da avaliação tratada neste trabalho, busca-se com ela justamente a formalização dos processos da organização e a melhoria dos mesmos. Assim, é importante que a avaliação tenha uma fundamentação teórica, obedecendo as regras de seu modelo de referência, no caso o CMMI. Como as regras de avaliação deste modelo são descritas no ARC, é necessário conhecer e cumprir com o mínimo de requisitos do mesmo. Desta forma, é possível ter: mais controle sobre a avaliação, o conhecimento do caminho correto para conduzi-la, maior garantia de sucesso na obtenção dos resultados, além de preparar a organização para uma futura e eventual avaliação mais rigorosa.

O primeiro passo para executar uma avaliação do CMMI é estabelecer o escopo da avaliação, através do levantamento as características da organização, escolha das áreas de processo a serem investigadas e o nível a ser avaliado. Estas informações foram tratadas no capítulo anterior.

O passo seguinte é a seleção do método de avaliação, dos membros do grupo avaliador e de outros participantes da avaliação (pessoas que serão submetidas a questionários e/ou entrevistas, responsáveis pelas entradas da avaliação), o estabelecimento das saídas (evidenciar o que se espera classificar) e restrições da avaliação, com base nas informações obtidas no passo anterior.

6.4 – Método para Mini-avaliação

Na execução de mini-avaliações, quando a organização tem o apoio de consultores especializados, estes normalmente têm seus próprios métodos de avaliação. Alguns auditores credenciados pelo SEI desenvolveram seu próprio método da avaliação em pequena escala. Obviamente que, para ter acesso a estes métodos é necessário contratar os serviços das consultorias que os desenvolveram.

Grandes organizações também criaram seu próprio método de avaliação, que são executados entre avaliações formais. Um exemplo destes métodos é o Método de Mini-avaliação Modular, ou MMA – *Modular Mini-assessment Method* (Wiegers, 2000), desenvolvido pela Kodak e também utilizado por outras organizações. O MMA é composto das fases: planejamento – reunião de abertura – orientação CMM – administração do questionário – análise das respostas do questionário – discussão com os participantes – geração de resultados – apresentação dos resultados. O grupo avaliador do MMA é representado por apenas 2 avaliadores. O MMA é baseado no CMM, mas não cumpre com todos os requisitos da CAF.

O planejamento é feito pelos avaliadores, estes se reúnem com os líderes dos projetos para colher informações sobre seus projetos e identificação dos participantes da

avaliação (as pessoas-chave dos projetos). Depois os avaliadores documentam o plano, identificando os objetivos da avaliação, as áreas de processo a serem investigadas, os projetos a serem avaliados, os participantes e o cronograma das atividades da avaliação.

A reunião de abertura é a primeira oportunidade para que as equipes dos projetos se informem sobre a mini-avaliação e estratégias para implementação de melhorias. Além disso, esta reunião pode servir também para aplicação dos questionários, ou então pode ser feita outra reunião para esse fim.

A fase de orientação sobre CMM varia de acordo com a realidade da organização. Esta fase pode compreender desde uma pauta na reunião de abertura até um curso de 4 horas sobre as características do modelo.

O principal instrumento de coleta de dados é um questionário de maturidade. Os dados colhidos devem ser mantidos em caráter confidencial. A escolha das pessoas que responderão o questionário deve ser feita pelos avaliadores com o auxílio dos líderes dos projetos (normalmente isso é feito na fase de planejamento). Na administração dos questionários, devem ser dados esclarecimentos sobre as áreas de processo avaliadas. Além das respostas, é interessante que os participantes sejam encorajados a fazer comentários.

Se a fase para discussão com os participantes está planejada, os comentários feitos pelos participantes serão discutidos nesta fase. Se esta fase não é necessária, os avaliadores utilizam estes comentários em suas conclusões. É recomendável que esta fase exista, pois servirá como fonte para consolidação dos dados dos questionários.

A geração dos resultados pode ser feita pelos avaliadores ou em reunião com a equipe de cada projeto avaliado. O avaliador tira suas conclusões depois da discussão (ou do questionário, quando não houver a discussão), utilizando todos os dados coletados durante a avaliação.

O último passo é a apresentação dos resultados. Se os avaliadores tiraram suas conclusões sem a participação das equipes de projeto, os resultados devem ser apresentados a elas e aos gerentes. Se as equipes participaram na geração de conclusões, os resultados devem ser apresentados aos gerentes. Além dos resultados da avaliação, nesta apresentação devem ser apontados os próximos passos para a implementação de melhorias.

É importante a criação de um banco de dados com as informações da avaliação como: data, duração, número de participantes, o questionário usado, as áreas de processo avaliadas, os resultados dos questionários, as áreas que são selecionadas para desenvolvimento do plano de melhoria e o tempo gasto em cada fase da mini-avaliação. Estes dados serão úteis para uma próxima avaliação.

O custo de uma avaliação deste tipo é de 5% a 10% do que seria gasto em uma avaliação formal. Os participantes gastam uma média de 4 horas e os avaliadores 48 horas nas atividades da avaliação. O tempo de esforço dos avaliadores diminui à medida que aumentam suas experiências (Wieggers, 2000).

A simplicidade e objetividade deste método vão ao encontro à proposta deste trabalho. Assim, MMA, pode ser utilizado como um roteiro para a que organização prepare seu próprio método de avaliação. Porém, o conhecimento sobre os requisitos de avaliações CMMI é também de grande importância na elaboração do método de avaliação.

6.5 – Requisitos de Avaliações Classe C

Os requisitos para métodos de avaliação classe C são relacionados com as responsabilidades dos profissionais envolvidos com a avaliação, a documentação do método, o planejamento e a preparação da avaliação, coleta, consolidação e validação dos dados e o relatório dos resultados.

Os requisitos de avaliação do ARC concentram todas as regras para se ter uma avaliação bem sucedida. A eficiência dos requisitos de avaliações CMM, visíveis desde a CAF (Masters, 1995), é comprovada pela indiscutível satisfação dos que já se submeteram a uma avaliação deste modelo. O ARC possui as mais efetivas práticas percebidas ao longo dos anos de utilização da CAF, adicionadas a uma flexibilidade que permite a satisfação dos mais diversos tipos de organizações.

Os requisitos classe C do ARC são os que mais se conformam à proposta deste trabalho. Os requisitos mínimos de avaliações classe C são descritos no ARC da seguinte forma (ARC, 2001):

- Responsabilidades:

- 1- O método de avaliação deve definir as responsabilidades do *sponsor*, que no mínimo deve incluir:
 - Verificar se os membros do grupo avaliador têm experiência, conhecimento e habilidade para assumir responsabilidades e liderar a avaliação.
 - Garantir que as unidades organizacionais apropriadas (por exemplo, projetos, unidade funcional) participem da avaliação.
 - Garantir que o método apóie a não atribuição dos dados aos participantes.
 - Garantir que os recursos necessários sejam disponíveis para a avaliação.
 - Revisar e aprovar os objetivos da avaliação para que o grupo avaliador possa iniciar a coleta de dados.
- 2- O método deve definir as responsabilidades do líder do grupo avaliador que no mínimo deve incluir:
 - Garantir que a avaliação seja conduzida de acordo com processo documentado do método.
 - Confirmar o comprometimento do *sponsor* para prosseguir com a avaliação.
 - Garantir que todos os participantes da avaliação estejam informados sobre o propósito, escopo e enfoque da avaliação.
 - Garantir que todos os membros do grupo avaliador tenham: experiência, conhecimento e habilidade no modelo de referência da avaliação e no

método da avaliação; a competência necessária para usar os instrumentos ou ferramentas escolhidas para apoiar a avaliação, e garantir que tenham acesso a documentos que descrevam como executar as tarefas definidas para a avaliação.

- Verificar e documentar que os requisitos do método de avaliação estão sendo mantidos.

- Documentação do método:

- 1) O método deve ser documentado e no mínimo incluir (ou pelo menos o item *a e d*):
 - Identificação dos modelos CMMI (versão, disciplina, abordagem) para os quais o método pode ser usado.
 - Identificação da versão ARC em que o método é baseado.
 - Identificação dos requisitos de avaliação CMMI satisfeitos pelo método, determinando em que classe de avaliação CMMI (A, B, ou C) o método está relacionado.
 - Descrição das atividades e dos recursos para implementação de cada requisitos da avaliação.
- 2) A documentação do método deve fornecer informações que permitam:
 - Identificação dos propósitos, objetivos e restrições da avaliação.
 - Determinação da adaptabilidade do método de avaliação em relação aos propósitos, objetivos e restrições (identificados no item anterior).
- 3) A documentação do método deve identificar o escopo do modelo(s) CMMI(s) a ser usado pela avaliação:
 - Áreas de processo ser investigadas (na abordagem em estágios ou contínua).
 - O nível da capacidade a ser investigado para cada área de processo (na representação contínua).
- 4) A documentação do método deve permitir a identificação da organização (ou unidade organizacional) a ser avaliada:
 - O sponsor (patrocinador) da avaliação e seu relacionamento com a organização a ser avaliada (cargo).

- Os projetos da organização que serão avaliados.
 - Os elementos funcionais da organização que deverão participar (gerentes, engenheiros, programadores).
 - Os nomes e função (na organização) dos participantes da avaliação.
- 5) A documentação do método deve permitir a seleção dos membros do grupo avaliador e os critérios para qualificação, incluindo:
- Experiência técnica na disciplina específica.
 - Experiência em gerenciamento.
 - Experiência, conhecimento e habilidades no modelo de referência da avaliação e no método de avaliação.
- 6) A documentação do método deve fornecer critério de qualificação do líder do grupo avaliador:
- Treinamento e experiência usando o modelo de referência da avaliação.
 - Treinamento e experiência usando o método de avaliação.
 - Experiência em ministrar treinamentos, em gerenciamento de grupos, liderança, versatilidade em discussões e apresentações.
- 7) A documentação do método deve determinar o tamanho apropriado do grupo avaliador.
- 8) A documentação deve oferecer informações:
- Sobre as regras e responsabilidades do grupo avaliador.
 - Para endereçamento das responsabilidades do *sponsor*.
 - Para endereçamento das responsabilidades do líder do grupo.
 - Para estimar os recursos requeridos na condução da avaliação (incluindo o tempo para a avaliação).
 - Sobre a logística da avaliação.
 - Para coleta dos dados relevantes da organização e para associação dos dados às práticas genéricas e específicas do modelo de referência da avaliação.
 - Para tomadas de decisões, incluindo os pontos fracos e fortes do processo da organização em relação ao modelo de referência da avaliação.
 - Para manter os dados em caráter confidencial e garantir a não atribuição dos dados aos participantes da avaliação.

9) A documentação deve fornecer orientação para (1) registrar o relacionamento entre os dados coletados durante a avaliação e as conclusões e/ou classificações, (2) reservar e proteger os registros da avaliação, e (3) reunir e manter um registro que apóie as conclusões e/ou classificações do grupo avaliador e que contenha no mínimo:

- Datas da avaliação.
- Entradas da avaliação.
- Identificação do método usado na avaliação (versão) e com algumas eventuais opções de execução deste método.
- Conclusões.

- Planejamento e preparação da avaliação:

1) O método deve fornecer aos participantes da avaliação no mínimo:

- O propósito da avaliação.
- O escopo da avaliação.
- O enfoque da avaliação.
- As regras e responsabilidades dos participantes na avaliação.
- O cronograma das atividades da avaliação.

2) A documentação do método deve fornecer as prioridades de entradas da avaliação para início da coleta dos dados pelo grupo avaliador.

3) No mínimo, as entradas da avaliação devem especificar:

- A identidade do *sponsor* da avaliação e seu relacionamento com a organização.
- Os propósitos da avaliação, incluindo o alinhamento com os objetivos de negócios.
- O escopo do modelo de referência, incluindo: (1) as áreas de processo que precisam ser investigadas e (2) o nível de maturidade ou capacidade a ser investigado em cada área no escopo da avaliação.
- A organização (ou unidade organizacional) que será submetida a avaliação.
- O contexto do processo, que deve incluir no mínimo: (1) o tamanho da organização; (2) dados demográficos (quantidade de funcionários); (3)

domínio da aplicação dos produtos ou serviços desenvolvidos na organização; (4) tamanho, a complexidade e características críticas dos produtos ou serviços; e (5) características de qualidade dos produtos ou serviços (como densidade de defeitos, confiabilidade).

- Restrições da avaliação, que deve incluir no mínimo: (1) disponibilidade de recursos; (2) o máximo de tempo a ser gasto na avaliação; (3) áreas de processo específicas ou entidades organizacionais a serem excluídas da avaliação; (4) o mínimo, o máximo ou o tamanho específico da amostra que se pretende avaliar; (5) restrições quanto ao uso das saídas da avaliação, (6) informações sobre um acordo de confidencialidade; (7) não atribuição dos dados da avaliação às fontes.
 - Identificar o(s) modelo(s) CMMI utilizado, incluindo versão, disciplina e abordagem (estágios ou contínua).
 - Os critérios de conhecimento, experiência e habilidades dos membros do grupo avaliador.
 - Identificação dos membros do grupo avaliador, incluindo suas responsabilidades específicas na avaliação.
 - Identificação dos participantes (nome e função) dos participantes e suas responsabilidades na avaliação.
 - Algumas informações adicionais a serem coletadas durante a avaliação que servirão de apoio no alcance aos objetivos da avaliação.
 - Descrição das saídas planejadas incluindo classificações a serem geradas (área de processo, nível de maturidade).
 - Prever a execução de seguintes (como relatório de resultados, planos de ação, reavaliação).
 - Planejar a implementação do método e atividades associadas, incluindo o tamanho da amostra e cobertura da organização.
- 4) O método deve determinar que as entradas da avaliação e possíveis mudanças nestas entradas devem ser aprovadas pelo *sponsor* (ou autoridade delegada) e documentado no registro da avaliação.

5) O método deve requerer o desenvolvimento de um plano que contenha no mínimo:

- As entradas da avaliação.
- As atividades a serem executadas na condução da avaliação.
- Recursos e programas destinados à avaliação.
- Avaliação logística.
- Passos para atenuar os riscos associados à execução da avaliação.

- Coleta dos dados da avaliação:

O grupo avaliador tira suas conclusões baseadas nos dados coletados por uma ou mais fontes. Em avaliações classe C a coleta dos dados deve ser feita por pelo menos uma das fontes abaixo.

- 1) O método deve coletar dados pela administração de instrumentos (como questionários, inspeções).
- 2) O método deve coletar dados pela condução de entrevistas (com os líderes do projeto, gerentes e envolvidos).
- 3) O método deve coletar dados pela revisão de documentos (como políticas organizacionais, procedimentos de projetos).

- Consolidação e Validação:

- 1) O método deve oferecer um mecanismo para consolidar precisamente os dados coletados durante uma avaliação, de acordo com os critérios:
 - As observações foram derivadas de evidências vistas ou ouvidas durante a sessão de coletas de dados.
 - As observações estão claramente escritas, expressas sem atribuições e na terminologia usada na organização.
 - As observações são relevantes para o modelo de referência da avaliação e pode ser associado com um componente específico do modelo.

- Relatório dos resultados:

- 1) O método deve requerer documentação e relatórios das conclusões e/ou classificações para o *sponsor* e para a organização avaliada.
- 2) O método deve requerer também que o relatório da avaliação seja fornecido ao *sponsor* para ser arquivado.

6.6 – Interpretação dos Requisitos

A os requisitos classe C e o método MMA (*Modular Mini-assessment Method*) (Wieggers, 2000) podem ser utilizados como roteiros para a elaboração de um método de avaliação menos rigoroso e que atenda as necessidades da organização. Porém, algumas observações quanto o ARC podem auxiliar na interpretação dos requisitos.

O primeiro passo é na elaboração de uma avaliação é a escolha do líder da avaliação (*sponsor*), o grupo avaliador e o líder deste grupo (ARC, 2001). Neste momento o interesse de buscar a melhoria do processo começa a sair do pessoal e atingir o coletivo.

É muito importante o apoio da alta gerência para o sucesso da avaliação. É interessante que o *sponsor* seja algum profissional deste nível hierárquico. A obtenção do patrocínio do gerente sênior é fundamental para melhoria da capacidade organizacional (Paulk, 1999). É possível que em alguns momentos o andamento da avaliação não seja satisfatório, podendo ocorrer, entre outras coisas, um desvio quanto aos verdadeiros objetivos da avaliação, ou pode faltar colaboração de alguns profissionais envolvidos. Neste momento é necessária a intervenção do *sponsor* para estimular estes profissionais, impor regras, cobrar a colaboração. Ações deste tipo não surtem mais efeitos quando desempenhada por alguém que tenha maiores poderes na organização. O envolvimento da alta gerência é importante também no patrocínio aos recursos necessários à avaliação (tempo, pessoal, treinamentos, etc).

O *sponsor* também é responsável pela seleção dos membros do grupo avaliador e seu líder (SCAMPI, 2001). Dependendo do tamanho e complexidade da organização e da

urgência em se obter resultados, o grupo avaliador pode ser composto por 3 a 10 pessoas.

Na literatura este grupo é chamado de SEPG – *Software Engineering Process Group* (Pires, 1998). Este grupo pode ser composto por consultores especializados em CMM ou certificados pelo SEI (Pires, 1998). Porém, se a organização não possui recursos para isso é recomendável que seja aproveitada a experiência de profissionais da própria organização que já tenha tido algum contato com o CMM, com avaliações de processo ou tenha vivenciado um programa de melhoria, ou pelo menos tenha bastante experiência em desenvolvimento de software. Neste último caso, é necessário que estes profissionais recebam treinamentos, verbas para adquirir fontes de informação e tempo para estudo.

Uma alternativa para reduzir os investimentos em treinamento é a possibilidade de uma organização se unir a outras para dividir os custos de treinamento. Um profissional faz um curso e distribui o conhecimento para os demais profissionais envolvidos. O importante é que os componentes do grupo estejam familiarizados com os termos do CMM antes de qualquer outra iniciativa. O grupo avaliador é o coração do projeto, aquilo que vai garantir que todo o processo de implantação do modelo flua da maneira adequada. A experiência do grupo é sempre maior que a maior das experiências individuais de cada um de seus componentes.

É importante considerar que a organização não se faz apenas de processos e tecnologias, as pessoas são um aspecto essencial para o sucesso de ambos. Como as pessoas são responsáveis pelo processo e o processo não fala por si só, o bom esclarecimento destas pessoas é crucial para a obtenção de resultados consistentes com a realidade. Também nesta fase é necessário fazer circular pela organização informações a respeito do CMM. A eficiência do programa dependerá grandemente dos esforços despendidos nesta fase. É necessário que seja divulgando principalmente que não se trata de uma reengenharia (Pires, 1998) e que não há previsão de demissões. Afinal, a informação evita boa dose

de resistência. A divulgação das informações pode ser feita através de jornais internos, palestras, circulares e e-mails com pequenos resumos.

Depois de devidamente treinado, o trabalho do grupo avaliador começa pela seleção de projetos típicos do trabalho da organização, em diferentes fases de desenvolvimento. Não é necessário avaliar todos os projetos, mas é importante que os escolhidos sejam representativos das práticas habituais e cultura da organização. Os demais participantes da avaliação serão selecionados mediante sua relação com estes projetos escolhidos. Devem ser selecionadas as pessoas chave de cada projeto, ou seja, aquelas que detêm o conhecimento de como é o desenvolvimento e a manutenção do software na organização. Esta escolha deve considerar a área de processo do CMMI que está sendo investigada. Estas pessoas são as responsáveis pelos dados de entrada da avaliação. É necessário que estas pessoas sejam conscientizadas da sua importância para a avaliação e também sobre os benefícios da melhoria de processo. À medida que o processo de avaliação é executado, outras pessoas podem ser selecionadas posteriormente.

Uma organização pode ter várias maneiras de executar uma mesma tarefa, cada uma dessas maneiras representa um processo diferente. Mais de uma metodologia pode estar sendo utilizada ou até mesmo nenhuma. É importante que o grupo avaliador tome conhecimento de todos esses processos. O conhecimento dos processos existentes é útil no momento de propor melhorias, aproveitando ao máximo as experiências bem sucedidas dentro da própria organização.

O grupo avaliador deverá elaborar e documentar o plano da avaliação, aprovar um cronograma de trabalho e atualizar este cronograma. Este plano é um instrumento de apoio à avaliação, é importante que seja levado a sério, mas também pode ser alterado para o melhor andamento da avaliação. O cronograma de atividades, por exemplo, deve ser mudado sempre que isso se fizer necessário ao processo de avaliação, sendo importante que tudo isto seja documentado. Normalmente cronogramas com macro atividades (Pires, 1998) funcionam melhor do que um excessivamente detalhado. A aplicação de questionário e/ou entrevistas com as pessoas-chave devem ser marcadas de

acordo com a necessidade e disponibilidade, mas as reuniões do grupo avaliador devem acontecer de maneira periódica e de acordo com o previsto no cronograma.

O grupo avaliador deve elaborar questionários e determinar as estratégias de entrevistas tendo em mente sempre a área de processo do CMMI a ser explorada. Na elaboração do questionário, grupo avaliador deve mapear as práticas CMMI, da área de processo a ser avaliada, para a realidade da organização. As pessoas-chave de cada projeto devem ser esclarecidas sobre a metodologia de avaliação. O processo de avaliação começa com a aplicação dos questionários, análise das respostas, iniciando um ciclo de entrevistas voltadas para esclarecer falhas e contradições encontradas no preenchimento dos questionários. As entrevistas servem também para coletar preocupações e sugestões para o programa de melhoramento.

O ideal é que sejam feitas reuniões com as pessoas-chave para a entrega dos questionários, pois se surgir alguma dúvida membros do grupo avaliador poderá esclarece-la. No contexto deste trabalho é interessante que seja feita uma reunião para cada área de processo. Nesta reunião é necessária que seja dada uma explanação da área de processo em questão, se possível com a leitura das questões. Não é produtivo utilizar muitos questionários em uma mesma reunião. Quando um mesmo indivíduo desempenha papéis diferentes em um mesmo projeto, ou participa de vários projetos, é necessário que o grupo avaliador se preocupe em não tornar cansativa a aplicação de questionários, podendo dividir as reuniões por projetos também. Neste caso é necessário que o participante seja instruído sobre seus papeis no projeto que está sendo avaliado.

Nesta reunião, os membros do grupo avaliador devem anotar as maiores dúvidas encontradas pelos participantes, que poderá auxiliar na consolidação das respostas. É necessário, para se ter sucesso no levantamento, falar a mesma língua da organização, estar coerente com sua realidade. Os participantes deverão ter o prazo de alguns dias para entregar suas respostas e anotações por escrito.

Depois de entregues as respostas, o grupo avaliador deverá comparar suas anotações com estas respostas para preparar as respostas consolidadas. Cada participante do grupo avaliador deve receber uma cópia das respostas consolidadas e uma reunião de revisão deve ser marcada.

Durante o processo as informações específicas de cada projeto devem ser mantidas confidenciais para estimular a fidelidade das respostas, já que não existe o objetivo de avaliar os projetos individualmente.

Nesta reunião de revisão, devem ser levantados possíveis pontos de ambigüidade e verificada a necessidade de maiores esclarecimentos (necessidade de entrevistas e/ou revisão de documentação). Se isto for necessário, estas atividades devem ser executadas e novas reuniões devem ser marcadas, até que se tenham os dados necessários sobre os projetos da organização.

Depois disso, os resultados da avaliação devem ser mapeados e comparados com as práticas CMMI. O objetivo não é enquadrar a organização em um nível de capacidade, mas é importante estar sempre mantendo o vínculo com o modelo. O registro destes dados pode ser útil em uma próxima avaliação. Deve então ser feito um relatório final da avaliação, identificando os pontos positivos e negativos, e apresentados os resultados aos interessados. Assim é finalizado o processo de avaliação.

Os resultados da avaliação são registrados no relatório final, que deve conter os seguintes assuntos:

- Transcurso da avaliação: pequeno histórico, os eventuais problemas encontrados e recomendações para as próximas avaliações.
- Área de processo avaliada e práticas CMMI atendidas: define o estado atual da organização, de acordo com os critérios do SEI.
- Pontos fortes: capacidades existentes na organização ou em projetos individuais, que podem ser facilmente difundidas.

- Deficiências: práticas ruins e capacidades ausentes ou que devem ser melhoradas para se atingir o próximo nível de maturidade, conseqüências na organização e exemplos.
- Recomendações: pontos que devem ser melhorados no processo de desenvolvimento de software.

Após a apresentação do relatório final é necessário desenvolver um plano, identificando cada capacidade que deve ser criada ou melhorada, responsabilidades e prazos para o seu desenvolvimento e implantação. Para garantir a realização do plano, é muito importante que ele seja realista e coerente com a estratégia de longo prazo da organização. É fundamental que se use as práticas CMMI como guia implementação de melhorias.

O TSP - *Team Software Process*, pode também ser considerado (Paulk, 1999) na elaboração e execução de melhorias direcionadas. O TSP (Humphrey, 2000b) caracteriza-se por um controle rigoroso de tamanhos, esforços, prazos e defeitos, também presentes no PSP. Porém, o TSP enfatiza as áreas de processo: gerenciamento de requisitos, planejamento e controle de projetos, garantia da qualidade e gerenciamento da configuração.

Somente após a realização de todas estas etapas é que os benefícios serão colhidos.

A partir da avaliação, em que se encontram os pontos fracos do processo, é necessário fazer com que todos dentro da organização comprem a idéia de melhoria, de cima para baixo, ou seja, desde a alta administração. Isto implica na necessidade de se construir uma disciplina de comprometimento. Todos os envolvidos devem acostumar a assumir compromissos com seriedade. Isto significa, entre outras coisas, não assumir responsabilidades que não possam cumprir, isto é, não prometer o que não pode cumprir e cumprir o que prometeu. Isto é especialmente importante em relação a tratamento de prazos

Sem ação não há melhoria. Obviamente a avaliação do processo por si só não trás nenhum benefício, apenas benefícios para o conhecimento. Uma avaliação é uma parte do investimento para uma organização produzir melhorias no processo de software. É importante para verificar o que precisa ser mudado no processo, evitando desperdícios de investimentos em pontos desnecessários, e mais, para criar nos participantes céticos o verdadeiro comprometimento com a organização para melhoria da capacidade do processo (Wiegers, 2000).

6.7 – Questionário de Maturidade

Nas avaliações CMM são normalmente utilizados questionários, entrevistas e revisões de documentação na investigação dos processos da organização. No MMA recomenda-se a utilização de questionários como instrumento de coleta de dados.

O Questionário de Maturidade, ou MQ - *Maturity Questionnaire*, (Zubrow, 1994) tem sido utilizado como uma ferramenta de coleta de dados no auxílio à implementação dos métodos de avaliação baseados no SW-CMM, como o SCE (Byrnes, 1996) e CBA IPI (Dunaway, 1996). O MQ é utilizado como fonte primária de coleta de dados, a partir de sua aplicação é que outros questionários são criados e as entrevistas e revisões são direcionadas, visando maiores esclarecimentos do grupo avaliador.

O objetivo do MQ é auxiliar o grupo avaliador a chegar a uma conclusão sobre a situação da organização como um todo. Os dados obtidos pelo questionário devem ser confirmados através da revisão dos documentos e de entrevistas, ou seja, somente estes dados não podem ser levados em consideração para o julgamento da capacidade do processo de software.

As áreas de processo, no SW-CMM, são chamadas de áreas chave de processo, ou KPA - *Key Process Areas*. O MQ focaliza essencialmente o processo de desenvolvimento de software e é organizado em KPA's (que são no total 18).

No MQ as questões são apresentadas em forma de teste. As respostas possíveis são: sim, não, não se aplica (quando a pessoa considera que a questão não deve ser aplicada ao projeto) e não sei (quando a pessoa não tem certeza sobre a resposta).

O MQ apresenta 124 questões que são apresentadas formando grupos de acordo com a KPA a que elas se referem, ou seja, para cada KPA é disponível uma média de sete questões. As questões se referem ao cumprimento e institucionalização das KPA's. Além das questões, é disponível um espaço onde é possível comentar e esclarecer as respostas ou apontar referências como documentação de suporte às mesmas.

Não há regras sobre como as respostas do questionário de maturidade devem ser analisadas. Para se chegar a uma conclusão sobre a avaliação é necessário levar em consideração as evidências de: implementação ou não das práticas chave do CMM, práticas alternativas que concordam com objetivos das KPA's, pontos fracos e fortes do processo de software.

Apesar de indicar o questionário para coleta de dados, o CMMI não é acompanhado por questionários criados pelo SEI. A característica flexível deste modelo implica também na flexibilidade na criação de questionários, visto que o que se quer avaliar depende da realidade da organização e das características da avaliação.

A origem e a estrutura do MQ não permitem que este seja tão eficiente em avaliações baseadas na abordagem contínua do CMMI. Apesar de ser fonte primária de coleta de dados em avaliações SW-CMM, o MQ apresenta algumas desvantagens, até mesmo no modelo para o qual foi criado. Este questionário é bastante extenso, pois cobre todas as áreas de processo definidas pelo SW-CMM, sem qualquer distinção das características e maturidade da organização avaliada. Do ponto de vista de quem responde, o MQ pode tornar-se cansativo, o que é improdutivo. Além de disso, deve-se considerar que na avaliação de organizações imaturas, questionamentos sobre práticas avançadas, podem trazer a sensação de desânimo devido à sobrecarga de informação. Diante da comum

aversão inerente a mudanças, esta sobrecarga pode dar a sensação de que as mudanças são mais dramáticas do que realmente são.

6.8 – Uma Proposta de Questionário

A disponibilidade e facilidade de obtenção dos documentos a respeito do CMM permitem que organizações se aventurem a implementação do modelo por conta própria. Aqui é proposta uma estratégia para a criação e utilização de questionários, que podem servir como fonte de coleta de dados da avaliação.

A estratégia utilizada para a criação dos questionários é baseada na observação de dois itens:

- As práticas das áreas de processo do CMMI (práticas básicas).
- As ocupações profissionais das pessoas envolvidas com o desenvolvimento de software (papéis).

Neste momento, é importante considerar o escopo da organização e o que quer avaliar. Partindo do princípio que a organização está saindo do estado de inércia e dando seus primeiros passos em direção a qualidade, considera-se que esta se encontra em um estado imaturo. Sendo assim, serão questionadas as práticas que são indispensáveis a saída da situação de imaturidade. Por este motivo, somente as práticas consideradas básicas pelo CMMI serão avaliadas. Esta estratégia evita questionamentos desnecessários sobre práticas que não fazem parte do contexto da avaliação.

Outro ponto importante é a escolha das pessoas que serão submetidas aos questionários. As questões devem ser endereçadas às pessoas corretas, evitando que alguém seja questionado sobre uma atividade que não seja de sua competência. Cada pessoa responderá sobre as atividades que desenvolve na organização. Assim, os questionários serão mais produtivos, visto que, se uma pessoa desconhece a execução de alguma prática na organização, é porque simplesmente esta prática não é executada.

6.8.1 – Papéis

A integração harmoniosa de métodos, ferramentas e pessoas é que determina o funcionamento eficaz de um processo de software (Rocha, 2001). Como parte integrante deste trio, as pessoas têm um papel relevante no funcionamento de um processo. Dentro de uma organização há uma ou mais pessoas dedicadas a cada atividade, e o bom andamento da organização como um todo depende do bom desempenho destes profissionais individualmente.

Devido a esta diversidade de atividades é improdutiva a aplicação de um questionário único para pessoas que desempenhem atividades diferentes, sob pena de se obter resultados errôneos. Sendo assim, a avaliação proposta neste trabalho trata-se de um questionário para cada categoria profissional.

O Questionário de Maturidade (MQ) (Zubrow, 1994) não faz distinção entre profissionais, ou seja, um mesmo questionário é submetido aos indivíduos selecionados, independente de suas responsabilidades no desenvolvimento do software. O MQ é extenso e nem todas as questões são relevantes para uma mini-avaliação.

Na construção dos questionários devem ser consideradas 3 categorias. A escolha destas categorias deve-se ao fato de serem as categorias básicas presentes em qualquer organização de desenvolvimento de software. As categorias consideradas foram: gerente de projeto, analista de sistemas e programador.

Apesar desta distinção, sabe-se que na realidade das organizações de software, as obrigações da cada categoria acabam se confundindo. É muito comum, principalmente em organizações menores, a existência de analista-programador, ou seja, um mesmo indivíduo que executa a função de programador e de analista. Outra situação, o analista de sistema, normalmente assume também a função de gerente do projeto. Objetivando atender a organizações com estas características, neste trabalho é utilizado o conceito de “papéis”. Assim, questionários diferentes devem ser criados para cada um dos 3 papéis

(gerente de projeto, analista de sistema e programador). Na prática isto significa que, em situações em que um mesmo indivíduo assume a ocupação de analista e programador, e ele seja escolhido para responder os questionário, ele assumirá o papel de analista para responder o questionário desta categoria e posteriormente assumirá o papel de programador, para responder o questionário desta segunda categoria. Isso vale para situações de gerentes-analistas também. Profissionais que assumem múltiplas funções em organizações de software é uma característica bastante comum em organizações nacionais.

Antes de elaborar as questões de cada papel, é necessário identificar claramente as obrigações profissionais de cada categoria. Na prática, pode até parecer uma tarefa simples identificar o que cada um destes profissionais faz dentro de uma organização, mas seria bastante oportuno buscar uma base teórica para esta identificação.

Esta não é uma tarefa muito simples, visto que falta na literatura uma padronização para as ocupações de cada categoria. Mesmo os poucos trabalhos teóricos que existem neste sentido, muitas vezes não são aplicáveis à realidade, justamente por esta característica multidisciplinar dos profissionais de software.

Para tentar identificar as ocupações dos profissionais de cada categoria, foi utilizada a última edição da Classificação Brasileira de Ocupações – CBO, disponibilizada pelo Ministério do Trabalho e Emprego, em 2002. A CBO, desenvolvida com o apoio de pesquisadores da Unicamp, UFMG e Fipe/USP e profissionais do Serviço Nacional de Aprendizagem Industrial – Senai, é o documento que reconhece, nomeia e codifica os títulos e descreve as características das ocupações do mercado de trabalho brasileiro. Sua última edição acompanha as profundas mudanças ocorridas no cenário cultural, econômico e social do país nos últimos anos, que implicam em alterações estruturais no mercado de trabalho (CBO, 2002).

A CBO descreve estas categorias profissionais de forma bastante concisa da seguinte forma:

- GERENTES DE TECNOLOGIA DA INFORMAÇÃO: Gerenciam projetos e operações de serviços de tecnologia da informação; identificam oportunidades de aplicação dessa tecnologia; administram pessoas e equipes e interagem com outras áreas (CBO, 2002).

- ANALISTAS DE SISTEMAS COMPUTACIONAIS: Desenvolvem e implantam sistemas informatizados dimensionando requisitos e funcionalidade do sistema, especificando sua arquitetura, escolhendo ferramentas de desenvolvimento, especificando programas, codificando aplicativos. Administram ambientes informatizados, prestam suporte técnico ao cliente e o treinam, elaboram documentação técnica. Estabelecem padrões, coordenam projetos, oferecem soluções para ambientes informatizados e pesquisam tecnologias em informática (CBO, 2002).

- TÉCNICOS DE DESENVOLVIMENTO DE SISTEMAS E APLICAÇÕES: (Programador de Computador): Desenvolvem sistemas e aplicações, determinando interface gráfica, critérios ergonômicos de navegação, montagem da estrutura de banco de dados e codificação de programas; projetam, implantam e realizam manutenção de sistemas e aplicações; selecionam recursos de trabalho, tais como metodologias de desenvolvimento de sistemas, linguagem de programação e ferramentas de desenvolvimento. Planejam etapas e ações de trabalho (CBO, 2002) (maiores informações a respeito destas ocupações estão apresentadas no Anexo 1).

Na elaboração de uma avaliação deve-se considerar as necessidades e características da organização. Assim, na primeira reunião para preparação da avaliação, o grupo avaliador pode identificar outros papéis presentes na organização (como por exemplo, gerente de qualidade e gerente de configuração, apesar de não serem profissionais comuns em pequenas organizações) e novos questionários podem ser criados para ser submetido a estes papéis.

6.8.2 – Práticas Básicas

Este trabalho propõe a criação de um questionário para cada área de processo do CMMI, permitindo assim, que a organização possa escolher as áreas a serem avaliadas. Observando a prioridade entre as áreas já mencionadas.

Como já dito, cada área de processo definida pelo CMMI é composta por objetivos específicos e genéricos, organizados em práticas específicas e genéricas respectivamente.

Na abordagem contínua as práticas específicas são divididas em: práticas básicas e práticas avançadas. Todas as práticas específicas com nível de capacidade 1 são chamadas de práticas básicas. As práticas específicas de nível de capacidade 2 ou superior são chamadas de práticas avançadas (CMMI, 2002b).

Considerando o escopo da organização para a qual este trabalho se direciona, é desnecessário o questionamento sobre a execução de práticas avançadas. Desta forma, esta avaliação propõe a análise somente de práticas básicas, visando a maior eficiência da avaliação, mediante a criação de questões que fazem parte da realidade destas organizações. Evitando o desperdício de tempo com questionamentos desnecessários, permitindo que a avaliação seja de certa forma rápida, objetiva e menos cansativa (as práticas básicas das áreas de processo Planejamento de Projeto, Garantia de Qualidade de Produto e Processo, Gerenciamento de Requisitos e Gerenciamento da Configuração são apresentadas no Anexo 2).

Este trabalho indica as áreas de processo a serem enfocadas em uma avaliação inicial, e a profundidade em que cada área será investigada, o que é determinado pela observação das práticas básicas de cada área. Dentro da grande variedade de opções que a engenharia de software oferece para a melhoria do processo, esta estratégia visa limitar horizontal (áreas de processo) e verticalmente (práticas básicas) o enfoque da avaliação, como ilustra a Figura 6.2.

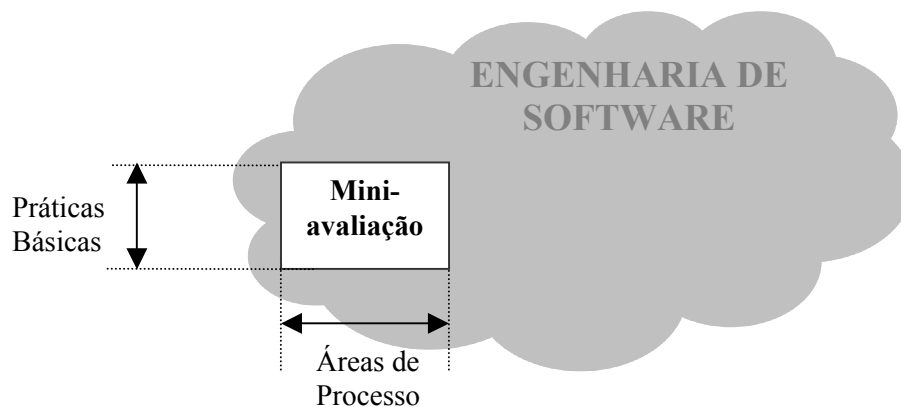


FIGURA 6.2 – Limites da avaliação.

6.8.3 – Metodologia para Criação dos Questionários

Lembrando que, cada área de processo é composta por várias práticas básicas. Quando todas as práticas básicas de uma área de processo são executadas, a organização atinge o Nível 1 de capacidade naquela área. O cumprimento de cada área envolve o trabalho de vários profissionais da organização. Aqui foram considerados 3 categorias de profissionais, que desempenham papéis diferentes na organização.

- Os questionários:

Com o objetivo de dar liberdade para a organização escolher a área de processo a ser avaliada, propõe-se a criação de um questionário para cada área de processo, de forma independente, apesar de algumas áreas apresentarem certa dependência.

Outro fator que deve ser considerado é as obrigações de cada profissional (gerente, analista e programador) aqui chamado de “papel”.

Sendo assim, os questionários devem ser feitos pelo cruzamento de duas informações: áreas de processo e ocupações profissionais das pessoas envolvidas, como ilustra a Figura 6.3. Cada área de processo então terá 3 questionários, sendo cada questionário feito para um papel. Por exemplo, para a área de processo ‘Planejamento de Projeto’ haverá 3 questionários, um para cada profissional.

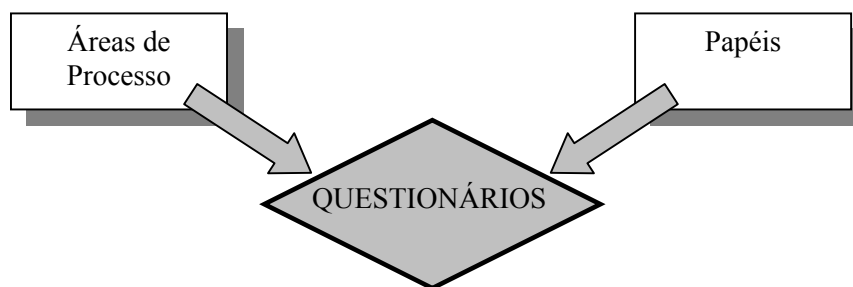


FIGURA 6.3 – Esquema para elaboração dos questionários.

- As questões:

As questões de cada questionário, obviamente, devem ser criadas também pelo cruzamento das informações: práticas básicas da área de processo em questão e obrigações do profissional da categoria para a qual está sendo feito o questionário. Considerando como exemplo a área de processo ‘Planejamento de Projeto’ e a categoria ‘Analista’, as questões para este questionário serão feitas observando as obrigações do analista no planejamento de projetos, ou seja, as questões avaliarão o trabalho deste profissional no planejamento de projeto.

O objetivo de estruturar o questionário desta forma é minimizar erros causados pelo questionamento de atividades realizadas por terceiros. Assim, cada profissional responderá sobre suas próprias atividades. O objetivo não é constranger o profissional responsabilizando-o pelos possíveis erros ou fracassos do projeto, mas sim conscientizá-lo de que ele é componente ativo da “máquina” do desenvolvimento e se cada um disciplinar seu trabalho, o resultado será o bom desempenho da máquina como um todo. Valorizar o profissional é uma das formas de incentivá-lo a melhorar continuamente.

Considerando a característica imatura das organizações a que se destina este trabalho, deve-se considerar a imaturidade também de seus profissionais quanto aos termos técnicos do CMMI. No desenvolvimento das questões é importante utilizar uma linguagem mais simplificada, em ocasiões em que isso for impossível, procurar esclarecer ao máximo significado do termo. Muitas vezes o profissional pode não compreender a questão e dar respostas que fogem da realidade. Obviamente o grau de detalhamento também vai depender da cultura da organização e seus profissionais.

Outro cuidado a ser tomado na formulação das questões é com relação ao tipo de resposta que o avaliador julgar mais convenientes. Como já mencionado, os questionários devem explorar as atividades dos próprios profissionais a que são submetidos, ou seja, devem induzir respostas pessoais. Assim, as questões podem ser de três tipos:

- Questões qualitativas: são questões que permite o profissional descrever a forma como ele executa uma determinada prática. Por exemplo: “Descreva a forma como são obtidas estimativas de custo e prazos”. As questões qualitativas (ou descritivas) são mais fáceis de serem elaboradas pela grupo avaliador e dão maior liberdade para o profissional se expressar. Porém, são mais trabalhosas na concatenação dos dados para obtenção do resultado final.
- Questões teste ou objetivas: estas questões podem ser do tipo *booleanas*, ou seja, que permitem as respostas Sim ou Não. Por exemplo: “São utilizadas ferramentas para automatizar o gerenciamento de versões?”. As questões booleanas são as mais difíceis de serem elaboradas, sendo necessário uma certa perícia do grupo avaliador na sua elaboração. Estas questões limitam as respostas, e conseqüentemente faz-se necessário maior número de questões para detalhar uma determinada prática (quando comparada com questões qualitativas). Um questionário inteiro com questões deste tipo pode se tornar muito extenso. Por outro lado, questionários desta forma permitem maior facilidade de obtenção de resultados e comparações entre profissionais de uma mesma categoria.

- Questões quantitativas: são questões relacionadas com números, por exemplo: “Qual o tempo médio gasto na análise de requisitos de software?”. As questões quantitativas nem sempre são convenientes no contexto de todas as práticas.

O grupo avaliador deve encontrar a melhor maneira para coletar os dados. Na maioria das vezes, torna-se necessário o emprego de mais um tipo de questões. O importante é que se usem termos condizentes com a cultura dos profissionais, que os questionários não sejam muito extensos e que sejam úteis no levantamento de dados reais. Qualquer ambigüidade nas respostas faz necessário o uso de entrevistas.

Muitos profissionais não estão acostumados com termos técnicos, é importante que, na elaboração dos questionários, seja utilizada a linguagem da organização. Pois a não compreensão ou compreensão errônea de termos pode causar distorção nos resultados da avaliação.

No Anexo 1 e Anexo 2 deste trabalho estão descritos respectivamente as obrigações dos profissionais (gerente de projeto, analista, programador) e as áreas de processo consideradas mais relevantes no contexto deste trabalho (Planejamento de Projeto, Garantia de Qualidade de Produto e Processo, Gerenciamento de Requisitos e Gerenciamento da Configuração). Estas informações podem ser úteis na elaboração das questões e para maiores esclarecimentos sobre as áreas de processo do CMMI.

CAPÍTULO 7

CONCLUSÃO

“Nós odiamos mudanças e amamos ao mesmo tempo, o que realmente queremos é que tudo continue como está... mas que melhore” (autor desconhecido) (Pinho, 2001). Esta frase resume o anseio e o receio, sentimentos tão contraditórios e característicos na expectativa de mudanças. Mudança cultural é difícil. Inserir novos conceitos nas práticas de trabalho requer conhecimento, mas também persistência. A princípio, quando novos processos são implementados, é comum que se crie uma situação desconfortável, uma sensação de trabalho a mais, que desaparece com o tempo, quando estes processos passam a fazer parte da rotina. É nesse momento que se constata um acréscimo de maturidade e novos desafios podem ser lançados.

A introdução de procedimentos visando o aumento de qualidade pode causar, em um primeiro momento, uma pequena redução da produtividade, devido à crítica fase de aprendizado. Isto acontece com a introdução de qualquer tecnologia em um ambiente de produção. É necessário que se tome cuidado para que os programas de melhoria não sejam abandonados neste momento, fazendo com que os recursos investidos se percam, baixando a moral dos envolvidos.

Muitas pessoas resistem a melhorias por sentirem-se ameaçadas, uma vez que melhores práticas e um acompanhamento mais eficaz pode trazer à tona deficiências de proficiência pessoal (Fiorini, 1999). Muitos também resistem, pois já possuem práticas estabelecidas, algumas vezes eficazes, e não gostariam que estas práticas fossem alteradas, pois as dominam. Portanto, a forma como as pessoas são envolvidas com a implementação de um programa de melhoria, é crucial.

O objetivo dos modelos CMM é uma evolução consciente, estruturada e contínua. A vantagem em propor níveis diferentes de maturidade e capacidade, é a de tornar mais

fácil a implementação das tarefas e possibilitar que a melhoria do processo da organização ocorra de forma gradativa, dando oportunidade de mudança de cultura.

Avaliar as práticas básicas de uma área de processo não significa que o trabalho acabou, pois uma das principais metas do CMM e seus modelos derivados é o contínuo aprimoramento do processo.

O que há de positivo para a indústria brasileira de software, é que existe um amplo mercado a ser explorado e, principalmente, a importância dos softwares para que o mundo funcione em todos os campos, do trabalho ao lazer, passando por conflitos armados até missões de paz e programas sociais. O que há de negativo é que a indústria brasileira ainda tem um longo caminho a percorrer para assumir uma posição importante no contexto mundial. A qualificação do processo de desenvolvimento é vista como um poderoso recurso para vencer este desafio.

Nos dias de hoje, a qualidade é um dos requisitos para garantir a sobrevivência de um software no mercado. Portanto, podemos concluir que as organizações mais competitivas são as que trabalham sob a ótica da melhoria contínua dos processos para aumentar a qualidade do processo de desenvolvimento e, conseqüentemente, aumentar a qualidade do produto final. A qualidade não é um estado permanente, mas uma busca constante.

A qualidade também não é privilégio restrito a um tipo de organização. Todas as organizações, inclusive as pequenas que dispõem de poucos recursos, podem e devem introduzir um mínimo de qualidade no seu processo de desenvolvimento e manutenção de software. Esta não é uma tarefa fácil, mas também não é impossível. Neste trabalho, foi apresentado um caminho para que estas organizações possam se aproximar dos conceitos mais básicos da busca da excelência em engenharia de software.

7.1 – Considerações sobre o Desenvolvimento

Durante o desenvolvimento deste trabalho, algumas modificações foram ocorrendo com relação à proposta inicial. Ao longo do tempo, muitas das expectativas iniciais foram se mostrando inviáveis e novas idéias foram surgindo.

A meta era desenvolver um trabalho que auxiliasse no levantamento da qualidade do processo de software e que fosse útil a um vasto número de organizações. Assim, foi proposta inicialmente a criação de uma forma de avaliação baseada no CMM, que pudesse ser executada por um questionário. O questionário teria como base o MQ (Maturity Questionnaire), mas procurando extrair deste as suas desvantagens. O questionário não deveria ser extenso e cansativo; deveria considerar a maturidade da organização, evitar questionamentos sobre práticas de níveis mais elevados e utilizar um vocabulário coerente com a cultura dos profissionais; cada profissional responderia um questionário referente à sua ocupação na organização; as questões seriam *booleanas* (Sim/Não); à medida que o questionário fosse sendo respondido, novas questões seriam apresentadas dependendo das respostas anteriores; o questionário seria automatizado, podendo ser respondido via Internet; no final da avaliação seria impresso um relatório sobre o levantamento da qualidade do processo da organização avaliada.

O levantamento da qualidade do processo de software desta forma, quase que a um passe de mágica, seria sem dúvida um verdadeiro achado para a engenharia de software. À medida que esta idéia foi amadurecendo, foram surgindo os obstáculos. Foi verificado que um questionário desta forma deveria ser suficientemente longo para que não deixasse nenhuma dúvida a respeito do processo da organização, principalmente considerando-o como única fonte de coleta de dados. Para um levantamento detalhado seria necessária a criação de muitas questões. Assim, a tentativa de desenvolver um questionário menos extenso e cansativo seria descartada.

Além disso, foi verificada a impossibilidade de se criar um questionário, com as características descritas anteriormente, que pudesse avaliar precisamente uma

organização e ser utilizado por outras com a mesma eficiência. Era necessário diminuir o universo de cobertura da avaliação, definindo detalhadamente o escopo da organização para a qual a proposta seria útil.

A imaturidade desta proposta inicial foi sendo constatada a medida que se acumulavam os estudos sobre sua viabilidade. Um questionário deste tipo, para ser eficiente, deveria ser direcionado a um universo muito restrito de organizações, diminuindo significativamente o número de organizações que pudesse se beneficiar com o trabalho. Assim, alguns conceitos iniciais foram sendo abandonados, modificados e outros criados, para que fosse possível uma proposta concreta para apoiar o levantamento da qualidade do processo de software de um maior número de organizações. O resultado disso foi o presente trabalho, cujo escopo da organização pode atender a um universo bem maior de organizações.

A proposta inicial não foi completamente abandonada. Ela foi inserida dentro de um contexto maior, ou seja, ela está contida em uma estratégia de avaliação mais completa, aqui apresentada. Assim, o presente trabalho tem uma característica mais genérica. Nestes termos a proposta do questionário continua válida, mas deve ser implementada tendo em vista as características específicas da organização a ser avaliada.

Outra consideração que merece nota é a migração para o CMMI. A princípio, a proposta era baseada no CMM pelo seu contexto histórico de sucesso. Tendo em vista a adaptabilidade do CMM ao escopo da organização definida neste trabalho, foi verificado que a flexibilidade do CMMI era mais indicada. Neste momento, o trabalho passou a basear-se no CMMI. Porém, por ser um modelo recente, falta literatura que fundamente sua eficiência. Dessa forma, toda a fundamentação teórica sobre os benefícios de se apoiar no CMMI é feita com base em anos de utilização do SW-CMM.

No decorrer deste trabalho outros fatores foram observados e válidos de nota:

- Foi verificado o aumento do número de analistas em relação ao número de programadores em organizações de software. Assim, está havendo uma

conscientização de que construir software é muito mais do que simplesmente criar inúmeras linhas de códigos.

- Os profissionais que já conhecem e/ou se beneficiaram com a melhoria da qualidade de processo, estão muito abertos à divulgação de suas experiências. Todos os que foram procurados durante o desenvolvimento deste trabalho (profissionais que vivenciaram a implementação de modelos de melhoria em suas organizações, especialistas em CMM [inclusive auditores certificados pelo SEI] e pesquisadores) se mostraram bastantes solícitos à colaboração.
- São muitos os profissionais de software que desconhecem completamente a existência de modelos para melhoria da qualidade de processo. Muitos destes se interessavam pelo assunto e solicitavam fontes de pesquisa para maior interação com os modelos. Talvez seja necessária uma maior divulgação.
- Muitos são os profissionais, até mesmo de grandes organizações, que simplesmente desconhecem termos básicos de qualidade. Os que conhecem, muitas vezes não aplicam em seu trabalho por “falta de tempo”, ou seja, seus projetos estão sempre com cronogramas atrasados, então abandonam vários procedimentos devido à pressão de gerentes e clientes.

7.2 – Dificuldades Encontradas

No decorrer deste trabalho vários obstáculos foram sendo superados. É importante registrar algumas das dificuldades pessoais encontradas, que podem ser úteis para outras pessoas que queiram aventurar-se neste caminho. As principais dificuldades estão relacionadas a:

- Falta de experiência pessoal no desenvolvimento de software. Por não ter sentido na prática o problema da falta de qualidade, ou usufruído os benefícios desta, foram muitas as dificuldades em propor uma maneira de busca-la.
- A grande diversidade de organizações de software (como o tamanho, o tipo de software, as necessidades e carências, a metodologia de desenvolvimento ou falta de uma, entre outros). Tudo isso dificulta a proposta de uma solução

genérica para o problema de desenvolvimento, que possa atender as necessidades de várias organizações.

- A inexistência, na literatura, de uma “receita de bolo” onde se possa encontrar o “como” que o CMM não define. O CMM não é prescritivo, ou seja, ele não diz à organização “como” melhorar, mas sim “o que” melhorar. O “como” é uma particularidade de cada organização, pois depende de uma série de fatores (Pinho, 2001).

7.3 - Considerações Finais

Não se pretendia com este trabalho discutir a aplicabilidade do CMM em pequenas organizações, nem apresentar uma proposta revolucionária para o problema da falta de qualidade, pois é difícil algo nesta área que ainda não tenha sido proposto. O problema na engenharia de software não é a falta de propostas teóricas, o problema é a falta de por em prática estas propostas. Este trabalho apresenta uma estratégia para diminuir a distância entre a teoria e a prática que ainda assombra a engenharia de software.

O presente trabalho foi resultado da concatenação conclusões extraídas de: experiências de mini-avaliações (*mini-assessment*) que foram bem sucedidas em outras organizações; sugestões já consagradas de adaptação do CMM em pequenas organizações; e, técnicas que se mostraram eficientes em avaliações iniciais.

Espera-se que a necessidade eminente da qualificação dos processos, a importância das pequenas organizações no contexto nacional e as dificuldades iniciais da busca pela qualidade, expostas neste trabalho, sirvam de motivação para que outros trabalhos sejam feitos neste sentido. Que as diretrizes aqui apresentadas contribuam positivamente para que as organizações descrevam seus próprios métodos de avaliação. E que estes métodos sejam eficientes no levantamento e, posteriormente, na melhoria da qualidade do processo.

Neste trabalho não foi proposta uma “receita de bolo” para elaboração e execução de uma avaliação de processo, mas apenas as diretrizes para a criação de um método efetivo. Que este trabalho sirva de incentivo às pequenas organizações de software na introdução de práticas mais efetivas de gerenciamento de projetos e que elas possam colher os benefícios destas práticas. Para que um modelo de melhoria seja positivamente implementado é necessário entender como funciona a organização, o que precisa ser modificado e o que precisa ser adicionado, e principalmente documentar os procedimentos e formas de implementação de cada área de processo enfocada.

Avaliar as práticas básicas de uma área de processo não significa que o trabalho acabou, pois uma das principais metas do CMM e seus modelos derivados é o contínuo aprimoramento do processo. Neste trabalho foram apresentados primeiros passos de um longo, mas bastante gratificante caminho em busca da qualidade do software.

REFERÊNCIAS BIBLIOGRÁFICAS

Araújo, E. E. R. Oportunidades e desafios para o desenvolvimento de uma indústria de software nacional. **Revista Ciência e Cultura**, v.55, n.2, p.42-45, jun, 2003.

Argyris, C.; Schon, D. A. **Organizational learning II: theory, method, and practice**. 2^a ed. New York: Editora Adison-Wesley, USA, 1995. 305 p.

Augusto Neto, Á. **Uma estratégia para gerência de qualidade e produtividade no desenvolvimento de software**. São José dos Campos, 1997. 172 p.

Dissertação (Mestrado em Engenharia Eletrônica e Computação), Instituto Tecnológico da Aeronáutica, 1997.

Barros, C. D. C **Qualidade & participação: o caminho para o êxito**. São Paulo: Editora Nobel, 1999. 129 p.

Bastos Jr, P. A. - INOVAÇÃO: Garantia de competitividade sustentável às organizações. **Revista Eletrônica Acadêmica**. Rio de Janeiro, set/out. 2000. Disponível em: <<http://www.academica.cjb.net/>> Acesso em : 02/10/2002.

Belloquim, A. CMM em pequenas organizações: é possível? **Infochoose** n. 10. Ano I. maio, 1999. Disponível em:<<http://www.choose.com.br/infochoose/infochoose10.htm>> Acesso em: 01/06/2003.

Belloquim, A. CMMI: O futuro do CMM. InfoChoose n. 42. Ano III. Janeiro/Fevereiro 2002. Disponível em: < <http://www.choose.com.br/infochoose/infochoose42.htm#1>> Acesso em: 02/03/2003.

Broadman, J. G; Johnson, D. L. **What Small Business and Small Organizations Say About the CMM**. Sorento: ACM Digital Library 1994. Experience Report, p.331-340

Byrnes, P.; Phillips, M. **Software capability avaluation**. Version 3.0. Method Description. Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Technical Report (CMU/SEI-96-TR-002/ESC-TR-96-002), Apr 1996. 192 p.

Capability Maturity Model Integration (CMMI). **CMMI for software engineering version 1.1(CMMI-SW, V1.1)**. Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Staged Representation, Improving processes for better products (CMU/SEI-2002-TR-029/ESC-TR-2002-029), Aug., 2002a. 639 p.

Capability Maturity Model Integration (CMMI), **CMMI for software engineering version 1.1 (CMMI-SW, V1.1)**. Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Continuous Representation, Improving processes for better products (CMU/SEI-2002-TR-028/ESC-TR-2002-028), Aug 2002b. 645 p.

Cardoso, O. R. **Foco da qualidade total de serviços no conceito de produto ampliado**, Florianópolis, SC. Tese de Doutorado em Engenharia de Produção, Universidade Federal de Santa Catarina, 1995. Disponível em:
<http://www.eps.ufsc.br/teses/olga/volume1/> Acesso em: 01/02/2003.

Carter, L.; Graettinger C.; Patrick, M.; Wemyss, G.; Zasadni S. **The road to CMMI: results of the first technology transition**. Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Technical Report (CMU/SEI-2002-TR-007/ESC-TR-2002-007), Feb 2002. 127 p.

Cordeiro, M. A. Modelos de qualidade de desenvolvimento de software. **Jornal Bate Byte - CELEPAR** Companhia de Informática do Paraná, jul 2000. Disponível em:
<http://www.pr.gov.br/celepar/celepar/batebyte/edicoes/2000/bb99/modelos> Acesso em: 04/12/2002.

Crosby, P. B. **Qualidade é Investimento**. Rio de Janeiro: Editora José Olympio, 1991. 327 p.

Deming, W. E. **Qualidade: a revolução da administração**. Rio de Janeiro: Marques-Saraiva, 1990. 368 p.

Dunaway, D. K. e Masters, S. **CMM based appraisal for internal process improvement (CBA IPI): method description**. Pittsburgh : Software Engineering Institute, Carnegie Mellon University. Technical Report (CMU/SEI-96-TR-007/ ESC-TR-96-007), Apr 1996. 57 p.

Dunaway, D.; Berggren, R; Rochettes, G; Iredale, P; Lavi, I.; Taylor, G. **Why do organizations have assessments? do they pay off?** Pitsburg: Software Engineering Institute, Carnegie Mellon University. Technical Report (CMU/SEI-99-TR-012/ESC-TR-99-012), July 1999. 35 p.

Ferreira, A. B. H - **Novo Aurélio Século XXI**. O dicionário da língua portuguesa. Rio de Janeiro: Editora Nova Fronteira, 1999. 2128 p.

Fiorini, S. T. ; Staa, A.; Baptista, R. M. **Engenharia de software com CMM**. Rio de Janeiro: Editora Brasport, , 1999. 354 p.

Furlan, J. D. **CMM versus CMMI**. Qual é o melhor modelo para minha organização? Comunidade Digital, 2001. Disponível em:
<http://www.jdfurlan.com.br/Download/CMMxCMML.PDF> Acesso em: 04/08/2003.
2001.

Garvin, D. A. **Gerenciando a Qualidade**. Rio de Janeiro: Editora Qualitymark, 2002. 357 p.

Glass, R.L. **Software runaways, lessons learned from massive software project failures**. Upper Saddle River, NJ: Prentice Hall, 1998. 245 p.

Gonçalves, J. M; Villas Boas, A. **Modelo de maturidade da capacidade de software (CMM)**. Campinas: Fundação CPqD. Versão 1.2. Tradução não oficial do CMU/SEI-93-TR-24-(CMM V1.1), 2001. 60 p.

Henry, J; Kafura, D; Henry S.; Matherson, L. Improving Software Maintenance at Martin Marietta. **IEEE Software**, v. 11, n. 4 , p. 8 - 11. jul 1994.

Humphrey, W. **Managing the software process**. Massachussets: Addison-Wesley, 1989. 512 p.

Humphrey, W. **The personal software process (PSP)**. Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Technical Report (CMU/SEI-2000-TR-022/ESC-TR-2000-022), Nov 2000a. 40 p.

Humphrey, W. **The team software process (TSP)**. Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Technical Report (CMU/SEI-2000-TR-023 – ESC-TR-2000-023), Nov 2000b. 36 p.

International Standard Organization (ISO) **ISO/IEC 12207**. Standard for information technology - software life cycle processes. Geneva: ISO, 1995. 75 p.

International Standard Organization (ISO). **ISO/IEC 9126-1-1** Information technology. Part 1: quality model. software product quality. Geneva: ISO, 1999. 40 p.

International Standard Organization (ISO). **ISO/IEC TR 15504 - Information technology. Part 1: concepts and introductory guide, process software assessment**. Geneva: ISO, 1997. 28 p.

Jones, C. Software project management in the 21st century. **American Programmer**, v.11, n. 2, p. 12 – 18, Feb, 1999.

Juran, J. M. **A qualidade desde o projeto - novos passos para o planejamento da qualidade de produtos e serviços**. São Paulo: Pioneira, 1992. 538 p.

Juran, J. M. **Juran planejando para a qualidade**. São Paulo: Pioneira, 1990. 376 p.

Kan, S. H. **Metrics and models in software quality engineering**, Massachusetts. Addison-Wesley, 1995. 560 p.

Koontz, H.; O'Donnell, C. **Management**. 7^a. ed. New York: Editora McGraw-Hill, 1986. 608 p.

Lamprech, J.L. **ISO 9000 e o setor de serviços**. Rio de Janeiro: Editora Qualitymark, p. 237-253, 1997. p. 268.

Masters, S.; Bothwell, C. **CMM appraisal framework CAF, version 1.0** . Pittsburgh: Software Engineering Institute, Carnegie Mellon University . Technical Report (CMU/SEI-95-TR-001/ESC-TR-95-001), Feb. 1995. 76 p.

McLain, J. **Impact of CMM based software process improvement**. Dissertação de Mestrado em Sistema de Informação, Divisão de Estudos Profissionais. Honolulu: Hawaii Pacific University, 2001. 230 p.

Ministério da Ciência e Tecnologia / Secretaria de Política de Informática Programa Brasileiro de Qualidade e Produtividade em Software (PBQP). **Qualidade e produtividade no setor de software brasileiro**: resultados da pesquisa bienal de 2001. Brasília: Editora Brazilian Software, n. 4, 2002. 258 p.

Ministério do Trabalho e Emprego (MTE). **CBO - Classificação brasileira de ocupações CBO2002** – Livro 1: Códigos Títulos e Descrições, , Brasília-DF, ago. 2002. 650 p. Disponível em: <http://www.mtecbo.gov.br>. Acesso em: 12/02/2003.

Paula Filho, W. P. **Engenharia de software** – fundamentos, métodos e padrões. 2ª edição. Rio de Janeiro: LTC, , 2003. 600p.

Paulk, M. C. Using the software CMM in small organizations. In: Pacific Northwest Software Quality Conference, 16., with International Conference on Software Quality, 8.,13-14 Oct, 1998, Portland, Oregon. **Proceedings**. Portland: PNSQC/ASQ, 1998, p. 350-361.

Paulk, M. C. **Using the software CMM with good judgment**. ASQ Software Quality Professional. Pttsburgh: SEI, v. 1, n. 3, June 1999, p. 19-29.

Paulk, M. C.; Curtis, B; Chississ, M; Weber, C. V. **Capability maturity model version 1.1**. Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Technical Report (CMU/SEI-93-TR-024/ESC-TR-93-177), Feb. 1993. 82 p.

Pereira, M. J. L. B.; Fonseca, J. G. M. **Faces da Decisão, as mudanças de paradigma e o poder da decisão**. São Paulo: Makron Books, 1997. 278 p.

Pfleeger, S. L. ; Rombach H. D. Measurement Based Process Improvement. **IEEE Softwrae**, v. 11, n. 4 , p. 8 - 11. Julho 1994.

Pires, A, ; Mendes, J. R. B. **CMM**: o difícil é dar o primeiro passo. **Developers Magazine**, n. 30 - Ano 3, 1998.

Porter, M. E. **A Vantagem competitiva das nações**. Rio de Janeiro: Campus. 1993. 932 p.

Pressman, R. S. **Engenharia de Software**. São Paulo: Makron Books, 1995. 1056 p.

Reed, K. Software engineering – a new millennium? **IEEE Software**, p. 17 – 201 July/ago, 2000.

Rocha, A. R. C. ; Maldonado, J.C. ; Webwr, K.C. **Qualidade de software – teoria e prática**. São Paulo: Prentice Hall, , 2001. 303 p.

Rosa, P. G. **Modelo de avaliação do processo de software para a pequena empresa**. São Paulo. Dissertação (mestrado em Ciências Matemática e de Computação) – Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, 1997. 90 p.

Roullier, A. C. **Gerenciamento de projetos de software para empresas de pequeno porte**. Recife: 2001, 175p. Tese (Doutorado em Ciência da Computação), UFPE. Engenharia de Software e Qualidade de Software. 2001.

Software Engineering Institute (SEI). **ARC assessment requirements for CMMI**. Versão 1.1. Pittsburg: Software Engineering Institute, Carnegie Mellon Institute. Technical Report (CMU/SEI-2001-TR-034/ ESC-TR-2001-034), Dez. 2001. 49 p.

SOFTEX – **Ministério discute incentivo a melhoria da qualidade**. Disponível em: <http://www.softex.br/> Acesso em: 01/08/2003.

Software Engineering Institute, Carnegie Mellon University. **2002 SEI annual report published** . Pittsburgh: CMU, Feb, 2003. 76 p.

Software Engineering Institute (SEI). **SCAMPI standard CMMI appraisal method for process improvement, version 1.1**: method definition document. Pittsburgh : Software Engineering Institute, Carnegie Mellon University – Handbook (CMU/SEI-2001-HB-001), , Dec., 2001. 245 p.

Spinola, M.; Pessoa, A.; e Volpe, R.L.D. Uma experiência na implantação do modelo CMM., In: Simpósio Brasileiro de Engenharia de Software, WQS'97 – Workshop Qualidade de Software, 1997, Fortaleza. **Anais**. Fortaleza: UFC, 1997.

Staa, A. **Engenharia de programas**. Rio de Janeiro: Ed LTC - Livros Técnicos, 1987. 293 p.

Teboul, J. **Gerenciando a dinâmica da qualidade**. Rio de Janeiro: Editora Qualitymark, 1991. 292 p.

Tingey, M.O. **Comparing ISO 9000, Malcolm Baldrige, and the SEI CMM for software: a reference and selection guide**. Upper Sadle River, NJ: Prentice Hall, 1997. 312 p.

Vieira, C.G.G. **Uma metodologia para melhoria de processos**, Florianópolis, SC. Tese (Mestrado em Engenharia de Produção), Universidade Federal de Santa Catarina, 1995. 156 p.

Weber, K.C. ; Rocha, A.R.C. ; Nascimento, C, J. **Qualidade e produtividade em software**. 4ª ed. São Paulo: Makron Books, 2001. 188 p.

Wiegers, K. E. A modular software process mini-assessment method. **IEEE Software**, , v. 17, n. 1, p. 62 – 69, Jan/Feb. 2000.

Zubrow, D.; Hayes, W.; Siegel, J. e Goldenson, D. **Maturity questionnaire** . Pittsburgh: Software Engineering Institute, Carnegie Mellon University. Special Report (CMU/SEI-94-SR-7), July 1994. 57 p.

APÊNDICE A

1 - O que é a CBO2002

A Classificação Brasileira de Ocupações - CBO é o documento que normaliza o reconhecimento, a nomeação e a codificação dos títulos e conteúdos das ocupações do mercado de trabalho brasileiro.

A CBO codifica empregos e outras situações de trabalho para fins estatísticos de registros administrativos (por exemplo, Relação Anual de Informações Sociais, Seguro Desemprego, Declaração do Imposto de Renda de Pessoa Física, dentre outros), censos populacionais e outras pesquisas domiciliares. Inclui códigos e títulos ocupacionais e a descrição sumária.

Além disso, a CBO também inventaria detalhadamente as atividades realizadas no trabalho, os requisitos de formação e experiência profissionais e as condições de trabalho. A CBO é utilizada nos serviços de recolocação de trabalhadores, na elaboração de currículos e na avaliação de formação profissional, nas atividades educativas das empresas e dos sindicatos, nas escolas, nos serviços de imigração, enfim, em atividades em que informações do conteúdo do trabalho sejam requeridas.

A CBO 2002 é estruturada em Famílias Ocupacionais. Para efeitos práticos, define-se a ocupação como o conjunto de postos de trabalho substancialmente iguais quanto a sua natureza e as qualificações exigidas. Constitui-se de tarefas, obrigações e responsabilidades atribuídas a cada trabalhador. Pode-se ainda conceituar a ocupação como o conjunto articulado de funções, tarefas e operações destinadas à obtenção de produtos ou serviços (representado pelo código total de 4 dígitos, cada dígito referente a um grupo e subgrupos ocupacionais) (CBO, 2002).

2 – Famílias Ocupacionais

A CBO foi utilizada como fonte de caracterização dos profissionais de software mencionados neste trabalho.

2.1 – Gerentes da Tecnologia da Informação

O papel de Gerente de Projetos, utilizado neste trabalho, é definido na Família 1425 da CBO, que é composta pelas seguintes ocupações: Gerente de Rede, Gerente de Desenvolvimento de Sistemas e Gerente de Produção de Tecnologia da Informação.

Condições gerais de exercício: Os profissionais dessa família ocupacional podem exercer suas funções em instituições financeiras, em empresas de tele-processamento, de segurança, de suporte e manutenção à informática, de manutenção e expansão de redes, de processamento e comunicação de dados, em setores empresariais de desenvolvimento e produção de tecnologia da informação empresarial, entre outros.

Formação e experiência: Essas ocupações são exercidas por pessoas com escolaridade de ensino superior, acrescida de curso básico com mais de quatrocentas horas, além de constantes cursos de especialização e aperfeiçoamento. O exercício pleno das funções ocorre após o período de cinco anos de experiência profissional.

Áreas de Atividade:

a) Gerenciar a operações de serviços de Tecnologia da Informação: garantir aos usuários a disponibilidade dos serviços de tecnologia da informação; garantir cumprimento das políticas da empresa; avaliar performance de recursos de tecnologia da informação; avaliar indicadores de ocorrências; validar soluções propostas para ocorrências; otimizar o uso de recursos; propor melhorias nos processos operacionais; acompanhar implementação de correções e mudanças no processo; monitorar operacionalização de projetos (operação assistida); revisar pontos que sofreram auditorias; garantir atualização do plano de contingência; acionar plano de contingência; acompanhar atividades de gestão de mudanças; monitorar resultados; garantir integração dos produtos; acompanhar execução orçamentária; garantir manutenção do conhecimento de tecnologia da informação na organização, e garantir qualidade de serviços e projetos.

b) Gerir projetos de Tecnologia da Informação: assegurar cumprimento de normas e padrões; alocar recursos humanos, materiais e tecnológicos; controlar escopo do projeto; controlar custos, prazos de execução e andamento do projeto; gerir prestação de serviços terceirizados; analisar desvios na execução de projetos e serviços; corrigir

desvios; negociar mudanças de escopo com clientes e com fornecedores; aprovar produtos do projeto; homologar equipamentos e softwares, e coordenar implantação do projeto.

c) Identificar oportunidades de aplicação de Tecnologia da Informação: identificar escopo das necessidades de clientes; prospectar soluções tecnológicas; estimar prazo, custo e benefícios de soluções propostas; presumir riscos e analisar qualidade de soluções propostas, e apresentar soluções e alternativas.

d) Planejar atividades de Tecnologia da Informação: prever impactos de projetos e serviços e soluções para situações de risco; definir escopo do projeto e serviço; analisar escopo do negócio; definir estratégia de desenvolvimento e necessidades de recursos humanos, materiais (infra-estrutura) e recursos tecnológicos (software e hardware); estabelecer prazos, custos de execução e prioridades; elaborar propostas de projetos, serviços e orçamento; definir matriz de responsabilidades e padrões de performance e qualidade; estabelecer acordo de nível de serviços, metas, política de segurança e pontos de controle; traçar plano de comunicação e plano de contingência; homologar plano de contingência, e planejar simulação e execução de plano de contingência.

e) Administrar pessoas e equipes: disponibilizar recursos de trabalhos; distribuir serviços à equipe; promover integração; administrar conflitos; supervisionar produtividade individual; avaliar desempenho; distinguir potencialidades individuais; aprovar promoção de funcionários; prover meios para treinamento e desenvolvimento; especificar perfis dos profissionais da área; selecionar profissionais; aprovar programação de férias, e dispensar de funcionários.

f) Interagir com outras áreas: participar da criação de normas e procedimentos, da avaliação de impactos de novas tecnologias, da definição de padrões de desenvolvimento de projetos e da definição de padrões de segurança; negociar execução de serviços de apoio (elétrica, civil, logística, redes, sistemas); participar da simulação de plano de contingência; subsidiar elaboração de orçamento de outras áreas; acompanhar processos de auditoria interna e externa, e relatar andamento de projetos; divulgar planos, metas e resultados da área.

2.2 – Analista de Sistemas Computacionais

O papel de Analista, utilizado neste trabalho, é definido na Família 2124 da CBO, que é composta pelas seguintes ocupações: Analista de Desenvolvimento de Sistemas, Analista de Redes e de Comunicação de Dados, Analista de Suporte Computacional e Analista de Sistemas de Automação.

Condições gerais de exercício: Os profissionais podem trabalhar em atividades industriais, comerciais e de serviços de informática e atividades conexas, em atividades econômicas como as da área financeira, das comunicações, das comerciais em geral. Seu trabalho se realiza em equipe, com supervisão ocasional.

Formação e experiência: O exercício dessas ocupações requer curso superior completo em Ciência da Computação. Para os profissionais com outra formação de nível superior, o mercado de trabalho tem valorizado especialização e pós-graduação na área de informática. O exercício pleno das atividades ocorre, em média, após dois anos de experiência. Em função da inovação tecnológica, a permanência no mercado de trabalho requer atualização contínua dos profissionais.

Áreas de atividade:

a) Desenvolver sistemas informatizados: estudar as regras de negócio inerentes aos objetivos e abrangência de sistema; dimensionar requisitos e funcionalidade de sistema; fazer levantamento de dados; prever taxa de crescimento do sistema; definir alternativas físicas de implantação; especificar a arquitetura do sistema; escolher ferramentas de desenvolvimento; modelar dados; especificar programas; codificar aplicativos; montar protótipo do sistema; testar sistema; definir infra-estrutura de hardware, software e rede; aprovar infra-estrutura de hardware, software e rede, e implantar sistemas.

b) Administrar ambiente informatizado: monitorar performance do sistema; administrar recursos de rede, banco de dados e ambiente operacional, executar procedimentos para melhoria de performance de sistema; identificar e corrigir falhas no sistema; controlar acesso aos dados e recursos; administrar perfil de acesso às informações, e realizar auditoria de sistema.

c) Prestar suporte técnico ao Cliente: orientar áreas de apoio; consultar documentação técnica e fontes alternativas de informações; simular problema em ambiente controlado; acionar suporte de terceiros, e instalar e configurar software e hardware.

d) Treinar Cliente: consultar referências bibliográficas: preparar conteúdo programático, material didático e instrumentos para avaliação de treinamento; determinar os pré-requisitos do treinando, recursos áudio visuais, hardware e software; configurar e validar ambiente de treinamento, e ministrar treinamento.

e) Elaborar documentação para ambiente informatizado: descrever processos; desenhar diagrama de fluxos de informações; elaborar dicionário de dados, manuais do sistema e relatórios técnicos; emitir pareceres técnicos; inventariar software e hardware; documentar estrutura da rede, níveis de serviços, capacidade, performance e soluções disponíveis; divulgar documentação, e elaborar propostas técnicas e comerciais, estudos de viabilidade técnica e econômica e especificação técnica.

f) Estabelecer padrões para ambiente informatizado: estabelecer padrão de hardware e software; criar normas de segurança; definir requisitos técnicos para contratação de produtos e serviços e metodologias a serem utilizadas; padronizar nomenclatura; instituir padrão de interface com usuário; divulgar utilização de novos padrões, e especificar procedimentos para recuperação de ambiente operacional.

g) Coordenar projetos em ambiente informatizado: selecionar equipe de trabalho; preparar cronograma de atividades e financeiro; administrar recursos internos e externos; delegar funções à equipe; acompanhar execução do projeto; realizar revisões técnicas; avaliar qualidade de produtos gerados, e validar produtos junto a clientes em cada etapa.

h) Oferecer soluções para ambientes informatizados: propor mudanças de processos e funções; prestar consultoria técnica; identificar necessidade do cliente; avaliar proposta de fornecedores; negociar alternativas de solução com cliente e fornecedor; adequar soluções a necessidade do cliente; demonstrar alternativas de solução; divulgar solução; propor adoção de novos métodos e técnicas, e organizar fóruns de discussão.

i) Pesquisar tecnologias em informática: pesquisar padrões, técnicas e ferramentas disponíveis no mercado; identificar fornecedores; solicitar demonstrações de produto; avaliar novas tecnologias por meio de visitas técnicas; construir plataforma de testes; analisar funcionalidade do produto; comparar alternativas tecnológicas, e participar de eventos para qualificação profissional.

j) Comunicar-se: desenvolver expressão oral e escrita e capacidade de negociação; assimilar idéias; desenvolver expressão em forma gráfica; interpretar gráficos, diagramas e símbolos; adaptar linguagem, e trabalhar em equipe.

2.3 – Técnicos de Desenvolvimento de Sistemas e Aplicações

O papel de Programador, utilizado neste trabalho, é definido na Família 3171 da CBO, que é composta pelas seguintes ocupações: Programador de Sistemas de Informação, Programador de Multimídia, Programador de Máquinas com Comando Numérico e Programador de Internet.

Condições gerais de exercício: Trabalham em atividades de informática e conexas, presentes em todas as atividades econômicas. Trabalham individualmente e com supervisão ocasional, exceto o programador de internet, o programador de multimídia e o programador de sistemas de informação, que podem, eventualmente, trabalhar em equipe. Em algumas ocupações, é possível o trabalho a distância.

Formação e experiência: Para o exercício dessas ocupações requer-se ensino técnico de nível médio de informática ou superior incompleto em áreas como ciências exatas, informática, engenharia. A atualização profissional permanente é condição para o seu exercício. O desempenho pleno das atividades do programador requer de dois a quatro anos de experiência, dependendo da ocupação.

Áreas de Atividades:

a) Desenvolver sistemas e aplicações: desenvolver interface gráfica; aplicar critérios ergonômicos de navegação em sistemas e aplicações; montar estrutura de banco de dados; codificar, compilar e testar programas; prover sistemas de rotinas de segurança;

gerar aplicativos para instalação e gerenciamento de sistemas, e documentar sistemas e aplicações.

b) Realizar manutenção de sistemas e aplicações: alterar sistemas e aplicações e estrutura de armazenamento de dados; atualizar informações gráficas e textuais; converter sistemas e aplicações para outras linguagens ou plataformas; atualizar documentações de sistemas e aplicações; fornecer suporte técnico, e monitorar desempenho e performance de sistemas e aplicações.

c) Implantar sistemas e aplicações: instalar programas; adaptar conteúdo para mídias interativas; homologar sistemas e aplicações junto a clientes; treinar usuários; verificar resultados obtidos, e avaliar objetivos e metas de projetos de sistemas e aplicações.

d) Projetar sistemas e aplicações: identificar demanda de mercado; coletar dados; desenvolver leiaute de telas e relatórios; elaborar anteprojeto, projetos conceitual, lógico, estrutural, físico e gráfico; definir critérios ergonômicos e de navegação em sistemas e aplicações; definir interface de comunicação e interatividade; elaborar croquis e desenhos para geração de programas em CNC; projetar dispositivos, ferramentas e posicionamento de peças em máquinas; dimensionar vida útil de sistema e aplicações, e modelar estrutura de banco de dados.

e) Selecionar recursos de trabalho: selecionar metodologias de desenvolvimento de sistemas, linguagem de programação e ferramentas de desenvolvimento; especificar configurações de máquinas e equipamentos (hardware); especificar máquinas, ferramentas, acessórios e suprimentos; compor equipe técnica; especificar recursos e estratégias de comunicação e comercialização, e solicitar consultoria técnica.

f) Planejar etapas e ações de trabalho: definir cronograma de trabalho; reunir-se com equipe de trabalho ou cliente; definir padronizações de sistemas e aplicações; especificar atividades e tarefas e distribuir tarefas.

OBS: CBO completa pode ser obtida pela internet no endereço:
<http://www.mtecbo.gov.br>.

APÊNDICE B

1 – Áreas de Processo

Neste trabalho foram levantadas algumas áreas de processo consideradas mais críticas no gerenciamento de projetos, que são: Planejamento de Projeto, Controle e Monitoração de Projeto, Gerenciamento de Requisitos, Gerenciamento da Configuração e Garantia de Qualidade de Processo e Produto. Estas áreas devem ser enfocadas mais urgentemente em um programa de melhoria.

A descrição destas áreas é apresentada a seguir. O intuito desta descrição é introduzir os objetivos do CMMI para cada uma destas áreas.

Quando uma área de processo não é executada ou parcialmente executada, ela encontra-se no Nível 0 – Incompleto. Para alcançar um nível de capacidade, o objetivo genérico e objetivos específicos do referente nível devem ser alcançados. O objetivo genérico é comum para todas as áreas. Cada nível de capacidade possui um único objetivo genérico, que é composto por práticas genéricas. As práticas determinam o que deve ser feito para atingir os objetivos.

Objetivo Genérico do Nível 1: Alcançar os Objetivos Específicos.

Para uma área de processo alcançar o nível Executado (Nível 1), a organização deve executar os objetivos específicos da referida área. Este objetivo só possui uma prática genérica.

Prática Genérica 1: Executar as práticas básicas.

Para se atingir o objetivo genérico 1 é necessário executar todas as práticas específicas consideradas básicas da área de processo em questão.

Cada área de processo possui um ou mais objetivos específicos, composto por práticas específicas. Nem todos os objetivos e práticas específicas são mencionados aqui, apenas os que se relacionam com as práticas básicas.

O CMMI utiliza os conceitos de “produtos de trabalho” e “componentes do produto”.

Os produtos de trabalho (*work products*) são alguns artefatos produzidos pelo processo, como os arquivos, documentos, partes do produto, serviços, especificações, processos, e outros. Os componentes do produto (*product component*) são integrados na construção um produto (no caso o software). A diferença entre produtos de trabalho e componente do produto é que o primeiro não precisa necessariamente fazer parte do produto final a ser entregue ao cliente. Os componentes do produto fazem parte do produto final e podem ser necessários para o uso do produto.

A seguir, são apresentados as áreas de processo, consideradas prioritárias em um programa de melhoria, e os objetivos específicos relacionados com as práticas básicas.

1.1 – Área de Processo Planejamento de Projeto

O propósito do Planejamento de Projeto é estabelecer e manter planos que definam as atividades do projeto.

O planejamento inicia-se com os requisitos que definem o produto e o projeto. Planejar inclui estimar os atributos dos produtos e as tarefas, determinar os recursos necessários, negociar de comprometer, produzir um cronograma, identificar e analisar os riscos do projeto. A interação entre estas atividades é necessária para o estabelecimento do plano razoável. O plano fornece a base para executar e controlar as atividades do projeto que correspondem os compromissos com o cliente.

O plano deve ser revisado quando houver: mudanças nos requisitos e nos compromissos, estimativas imprecisas, ações corretivas e mudanças no processo. Esta área de processo possui práticas específicas para planejamento e re-planejamento do projeto.

Objetivo Específico 1 – Estabelecer Estimativas.

Estimativas de parâmetros de planejamento devem ser estabelecidas e mantidas. Os fatores que normalmente são considerados na estimativa destes parâmetros são:

- Os requisitos do projeto, incluindo requisitos do produto, requisitos impostos pela organização, os requisitos impostos pelo cliente, e outros requisitos que causam impacto no projeto.
- O escopo do projeto.
- As tarefas identificadas e produtos de trabalho.
- Abordagem técnica.
- O modelo de ciclo-de-vida escolhido para o projeto (por exemplo, tamanho e complexidade).
- Os atributos dos produtos e das tarefas.
- Os modelos ou dados históricos de tempo e custo em projetos passados.
- A metodologia (modelos, dados, algoritmos) usada para determinar a necessidade de material, treinamento, horas de trabalho e custo.

As Práticas Básicas relacionadas com este objetivo são:

- 1- Estimar o escopo do projeto.
- 2 - Estabelecer estimativas de produtos e atribuição de tarefas.
- 3 - Definir as fases do ciclo de vida do projeto.
- 4 - Determinar estimativas de esforços e custos do projeto.

Objetivo Específico 2 – Desenvolver um plano para o projeto.

Deve ser estabelecido e mantido um plano como base para o gerenciamento do projeto. Um plano é um documento formal, aprovado e usado para gerenciar e controlar a execução do projeto. É baseado nos requisitos do projeto e em estimativas estabelecidas. O plano deve considerar todo o ciclo de vida do projeto e garantir que todos os planos que afetem o projeto sejam consistentes com o plano geral do projeto.

As Práticas Básicas relacionadas com este objetivo são:

- 1 – Estabelecer e manter o orçamento e o cronograma do projeto.
- 2 – Identificar e analisar os riscos do projeto.
- 3 – Planejar o gerenciamento dos dados do projeto.
- 4 – Planejar os recursos do projeto.
- 5 – Planejar as necessidades de treinamentos e qualificação para a execução do projeto.
- 6 – Planejar o envolvimento dos interessados no projeto.
- 7 – Estabelecer e manter plano geral do projeto.

Objetivo Específico 3 – Obter o comprometimento com o plano.

Compromissos com o plano devem ser estabelecidos e mantidos.

As Práticas Básicas relacionadas com este objetivo são:

- 1 – Revisar todos os planos que afetam os projetos.
- 2 – Conciliar o plano do projeto com os recursos estimados e disponíveis.
- 3 – Obter o comprometimento das pessoas relevantes responsáveis pela execução e apoio ao plano.

1.2 – Área de Processo Controle e Monitoração de Projeto

O objetivo do Controle e Monitoração de Projeto é permitir um entendimento do progresso do projeto e tomada de ações corretivas apropriadas, quando a execução do projeto desvia significativamente do plano estabelecido. Estas ações podem requerer re-planejamento, revisão do plano original, estabelecimento de novos compromissos e adição de novas atividades no plano corrente.

Um plano documentado do projeto é a base para monitorar atividades, comunicar *status* e executar ações corretivas.

Objetivo Específico 1 – Monitorar o contraste entre o plano e o projeto.

O desempenho atual e o progresso do projeto são confrontados com o plano do projeto.

As Práticas Básicas relacionadas com este objetivo são:

- 1 – Monitorar os parâmetros do planejamento do projeto.
- 2 – Monitorar os compromissos.
- 3 – Monitorar os riscos do projeto.
- 4 – Monitorar o gerenciamento dos dados do projeto.
- 5 – Monitorar o envolvimento das pessoas interessadas com o projeto.
- 6 – Conduzir revisões progressivas.
- 7 – Conduzir revisões formais em marcos pré-determinados do desenvolvimento.

Objetivo Específico 2 – Gerenciar ações corretivas.

O gerenciamento de ações corretivas é necessário quando a execução do projeto ou os resultados desviam significativamente do plano.

As Práticas Básicas relacionadas com este objetivo são:

- 1 – Coletar e analisar os pontos onde há contraste entre o plano e o andamento do projeto.
- 2 – Tomar ações corretivas.
- 3 – Gerenciar ações corretivas.

1.3 - Área de Processo Gerenciamento de Requisitos

O propósito do Gerenciamento de Requisitos é gerenciar os requisitos do produto e de seus componentes e identificar inconsistências entre estes requisitos, o plano e o produto desenvolvido.

O Gerenciamento de Requisitos gerencia todos os requisitos recebidos ou gerados pelo projeto, incluindo requisitos técnicos e não técnicos, assim como os requisitos determinados pela organização. Se a área Gerenciamento de Requisitos é implementada, normalmente seus processos gerarão também requisitos de produto e componentes do produto, que também deverão ser gerenciados.

Os requisitos devem ser revisados entre as entidades que os definem (como o cliente, a organização) e quem deve aceitá-lo (por exemplo, gerente, analista). Uma vez chegado a um acordo, os demais participantes do projeto devem se comprometer com estes requisitos.

Os requisitos tendem a ser bastante voláteis, isto é, podem mudar bastante durante o desenvolvimento do projeto. Gerenciar os requisitos é também documentar as mudanças de requisitos, e manter um controle bi-direcional entre os requisitos definidos e os requisitos do produto e componentes do produto.

Objetivo Específico 1 – Gerenciamento de requisitos.

Os requisitos devem ser gerenciados e inconsistências com os planos do projeto e produtos de trabalho deve ser identificado.

As Práticas Básicas relacionadas com este objetivo são:

- 1 – Obter o entendimento dos requisitos.
- 2 – Obter comprometimento dos participantes do projeto com os requisitos.
- 3 – Gerenciar as mudanças nos requisitos durante a execução do projeto.
- 4 – Manter um controle entre os requisitos, o plano e os produtos de trabalho.
- 5 – Identificar inconsistências entre o plano do projeto, os produtos de trabalho e os requisitos.

1.4 - Área de Processo Gerenciamento de Configuração

O propósito do Gerenciamento de Configuração é estabelecer e manter a integridade dos produtos de trabalho usando identificação de configuração, controle de configuração, relatório do *status* de configuração e auditoria da configuração.

Os produtos de trabalhos mantidos sob gerenciamento da configuração incluem os produtos que são entregues ao cliente, componentes internos do produto, produtos adquiridos, ferramentas, e outros itens usados na criação e descrição destes produtos de trabalho. Exemplos destes produtos são: os planos, descrição de processo, requisitos, dados de projeto, diagramas, especificação do produto, código-fonte, arquivos de dados do produto e publicações técnicas do produto.

O gerenciamento da configuração pode ser executado em vários níveis, ou seja, os itens de configuração podem ser decompostos em componentes de configuração e unidades de configuração. Em um sistema composto por hardware e software, o software pode ser considerado como um único item de configuração, ou o software pode ser decomposto em múltiplos itens de configuração. Nesta prática o termo item de configuração é usado para referir os elementos sob controle de configuração.

As linhas básicas (*baselines*) fornecem uma base estável para a evolução contínua dos itens de configuração. As linhas básicas são adicionadas ao sistema de gerenciamento de configuração quando são desenvolvidas. Mudanças na construção de linhas básicas e *releases* dos produtos de trabalho de um sistema de gerenciamento de configuração são

sistematicamente controladas e gerenciadas através de controle de configuração, gerenciamento de mudanças e auditoria da configuração.

Esta área de processo se aplica não somente ao gerenciamento da configuração, mas também para gerenciar a configuração dos produtos de trabalho como padrões, procedimentos e reuso de bibliotecas.

Objetivo Específico 1 – Estabelecer linhas básicas.

As práticas específicas deste objetivo são relacionadas com o estabelecimento de linhas básicas e seleção dos itens de configuração. A seleção destes itens deve obedecer a algum critério documentado. Cada item deve ter uma identificação única, suas características devem ser especificadas, além da identificação dos responsáveis por cada item de configuração.

Um sistema de gerenciamento de configuração deve incluir os meios, os procedimentos e as ferramentas para acesso ao sistema de configuração. É necessário um mecanismo de níveis de controle para gerenciar a configuração. Armazenar e recuperar os itens de configuração e a versão destes itens. Criar relatórios dos itens de configuração. Revisar a estrutura de configuração quando necessário.

As Práticas Básicas relacionadas com este objetivo são:

- 1 - Identificar os itens de configuração, componentes e produtos de trabalho relacionados que serão mantidos sob gerenciamento da configuração.
- 2 – Estabelecer e manter um sistema de gerenciamento de mudanças e gerenciamento da configuração, para controlar os produtos de trabalho.
- 3 – Criar linhas básicas para uso interno e para ser entregue ao cliente.

Objetivo Específico 2 – Acompanhar e controlar mudanças.

Mudanças nos produtos de trabalho mantidas sobre gerenciamento de configuração devem ser acompanhadas e controladas. Mudanças não controladas nos itens pode gerar defeitos e falhas nos produtos. É necessário análise do impacto das mudanças nos itens.

As Práticas Básicas relacionadas com este objetivo são:

- 1 – Acompanhar os pedidos de mudança para os itens de configuração.

2 – Controlar as mudanças nos itens de configuração.

Objetivo Específico 3 – Estabelecer a integridade.

A integridade das linhas básicas deve ser estabelecida e mantida.

As Práticas Básicas relacionadas com este objetivo são:

1 – Estabelecer e manter registros com descrição dos itens de configuração.

2 – Executar auditoria da configuração para manter a integridade das linhas básicas.

1.5 – Área de Processo Garantia de Qualidade de Processo e Produto

Esta área de processo apóia a entrega de produtos e serviços de alta qualidade por fornecer ao gerente e pessoal envolvido no projeto a visão geral da eficácia do processo e produtos associados em todo o ciclo de vida deste projeto.

No CMMI, as práticas desta área objetiva que os processos planejados sejam implementados.

A objetividade na avaliação da qualidade é determinante para o sucesso do projeto. A objetividade é alcançada pela independência e pelo uso de critérios. Normalmente, quando o grupo de qualidade é independente do projeto, fica mais fácil manter esta objetividade. Porém, há organizações que a função de garantir a qualidade está embutida no processo. Neste caso, é necessário ter o cuidado de focalizar na objetividade. Todos os envolvidos com atividades de garantia da qualidade devem ser treinados. As atividades de garantia da qualidade devem ser separadas das atividades diretamente envolvidas com o desenvolvimento e manutenção dos produtos de trabalho.

Os casos de não conformidade verificados devem primeiramente ser tratados entre as pessoas envolvidas com o projeto e, se possível, serem resolvidas ali. Quando não forem resolvidas nesta instância devem ser conduzidas para uma instância de gerenciamento apropriada. Assim, é necessário que exista um canal independente de comunicação com a gerência.

Atividades de garantia da qualidade devem começar nas primeiras fases de um projeto para estabelecer planos, processos, padrões e procedimentos com o objetivo de adicionar valor ao projeto e satisfazer os requisitos do projeto considerando a política organizacional. Estas atividades devem acompanhar todo o ciclo de vida do projeto. As revisões e auditorias são aplicadas em vários pontos durante o desenvolvimento e servem para descobrir defeitos enquanto estes ainda são relativamente baratos para serem encontrados e tratados.

Objetivo Específico 1 – Avaliar objetivamente os processos e produtos de trabalho.

Avaliar objetivamente a conformidade entre a execução dos processos, produtos de trabalho e serviços e a descrição do processo, padrões e procedimentos e planos aprovados.

As Práticas Básicas relacionadas com este objetivo são:

- 1 - Avaliar objetivamente buscando garantir de que os planos, descrições, padrões e procedimentos definidos para o processo estão sendo cumpridos.
- 2 - Avaliar objetivamente buscando garantir de que os planos, descrições, padrões e procedimentos definidos para os produtos de trabalho e serviços sendo cumpridos.

Objetivo Específico 2 – Providenciar o cumprimento dos objetivos.

Casos de não conformidade devem ser objetivamente investigados e comunicados, e soluções devem ser encontradas.

As Práticas Básicas relacionadas com este objetivo são:

- 1 – Comunicar questões de qualidade e encontrar soluções para os casos de não conformidade com o gerente e demais pessoas envolvidas.
- 2 – Estabelecer e manter registros de atividades de garantia de qualidade.

2 - Relacionamento entre as Áreas de Processo

Há um forte relacionamento entre estas áreas de processo, dentre os quais podem ser citados:

- O Planejamento de Projetos e o Gerenciamento de Requisitos relacionam-se mutuamente na obtenção de maiores informações sobre os requisitos necessários

no planejamento e re-planejamento e também sobre a necessidade de revisão no plano quando há mudanças nos requisitos.

- O Controle e Monitoração do Projeto e Planejamento do Projeto se completam.
- O Gerenciamento de Requisitos, por sua vez, relaciona-se com o Gerenciamento da Configuração na obtenção de maiores informações sobre as linhas básicas e controle de mudanças para documentação da configuração dos requisitos.
- O Gerenciamento de Configuração relaciona-se também com o Planejamento de Projeto na determinação dos itens de configuração.

Foram apresentadas aqui as principais considerações a respeito das práticas básicas destas áreas de processo. No modelo CMMI-SW são disponibilizadas informações detalhadas sobre cada uma destas áreas (CMMI, 2002b).