

**FACULDADE DE MINAS – FAMINAS-BH
CURSO DE SISTEMAS DE INFORMAÇÃO**

**MARLEY FERNANDES
SAMUEL DA SILVA CORREA**

OS MODELOS DE MATURIDADE DE PROCESSO DE SOFTWARE E O MPS.BR

**BELO HORIZONTE
2008**

MARLEY FERNANDES
SAMUEL DA SILVA CORREA

OS MODELOS DE MATURIDADE DE PROCESSO DE SOFTWARE E O MPS.BR

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da FAMINAS-BH – Faculdade de Minas como exigência para obtenção do título de graduação em Sistemas de Informação, tendo como orientador o Prof. Ricardo Terra Nunes Bueno Villela.

BELO HORIZONTE

2008

Marley Fernandes
Samuel da Silva Corrêa

Os modelos de maturidade de processo de software e o MPS.BR.

Objetivo: demonstrar a funcionalidade proporcionada pelo modelo MPS.BR na resolução dos problemas verificados no desenvolvimento de software, apresentando-se como um modelo eficiente na organização e padronização dos procedimentos fundamentais desta atividade.

Área de concentração: desenvolvimento e melhoria dos processos de software.

Data de aprovação: 12/12/2008

FAMINAS-BH – Faculdade de Minas
Curso de Sistemas de Informação

Dedicamos este trabalho aos nossos cônjuges, que nos apoiaram e acreditaram no nosso sonho de tornar realidade o nosso curso superior.

AGRADECIMENTOS

Agradecemos a Deus por iluminar nosso caminho e nos dar forças para seguirmos sempre em frente.

Agradecemos ao nosso orientador, professor Ricardo Terra, por sua dedicação, paciência, sensibilidade, apoio ao longo desta jornada e amizade construída durante a realização deste trabalho.

Agradecemos a todos os professores que nos acompanharam no curso e nos levaram ao amadurecimento, conhecimento e nos fizeram enxergar um mundo em várias dimensões.

Agradecemos aos colegas: há uns que nos falam e não ouvimos, há outros que nos tocam e não os sentimos, há alguns que nos ferem e nem cicatrizes nos deixam, mas há aqueles que simplesmente vivem e nos marcam por toda a vida.

Eu, Marley, agradeço à minha amada e querida filha, Danielle Fernandes Santos, que soube entender a ausência necessária para que este trabalho fosse realizado e pelo apoio dado em todos os momentos.

Eu, Samuel, agradeço à minha querida esposa Elizângela Soares Corrêa, pelo apoio, companheirismo e compreensão. Agradeço também a minha querida mãe Maria Rita da Silva Corrêa por ter estado sempre ao meu lado me apoiando.

Enfim, a todos aqueles que de alguma forma contribuíram para esta monografia tornasse realidade, seja pela ajuda constante ou por uma palavra de amizade ou por compartilhar alegrias, sorrisos, tristezas, angústias e até desespero! O nosso MUITO OBRIGADO.

RESUMO

A evolução tecnológica trouxe às pessoas e às empresas, inúmeras inovações que facilitaram tanto a comunicação quanto o desempenho de processos, atividades e funções essenciais aos negócios realizados. No entanto, devido ao seu caráter mutável e dinâmico, os processos tecnológicos precisam ser continuamente aperfeiçoados e melhorados, visando reduzir riscos, minimizar problemas e aumentar o seu nível de aplicabilidade e satisfação dos usuários. Inseridos nesse contexto, os sistemas de software se apresentam como produtos que necessitam de um alto grau de organização, sistematização e gerenciamento ao longo de seu processo de desenvolvimento, envolvendo o interesse e a dedicação das empresas desenvolvedoras no sentido de identificar os principais riscos e problemas existentes ao longo do processo e buscando suas soluções. Diante disso, o modelo de MPS.BR – Melhoria de Processo do Software Brasileiro tem oferecido uma metodologia eficaz e adequada às empresas brasileiras produtoras de sistemas de software que necessitam de melhorias significativas em seus processos. Assim, este trabalho demonstra como o MPS.BR lida com os problemas verificados no desenvolvimento de software, apresentando-se como um modelo eficiente na organização e padronização dos procedimentos fundamentais. Por meio de um estudo bibliográfico, este trabalho apresenta os principais conceitos e características relacionados ao modelo MPS.BR.

Palavras-chave: Desenvolvimento de Software; Modelos de Software; CMM; CMMI; MPS.BR.

ABSTRACT

Technological evolution has brought to people and to business several innovations that facilitated both the communication and the performance of essential processes, activities, and functions to the business conducted. However, their changeability and their dynamic behavior force the technological processes to be incrementally refined and improved aiming at reducing risks and minimizing problems to improve their level of applicability and satisfaction of users. In this context, systems of the software are presented as products that require a high degree of organization, systematization and management over their own development process, involving the interest and dedication of software companies to identify the main risks and problems throughout the process and to look for their solutions. Due to this, MPS.BR (translated by Improvement of Procedure of the Brazilian Software) model has offered an effective and adapted methodology for Brazilian software companies to gain significant improvement in their processes. Thus, this work demonstrates how the MPS.BR model tackles the existent problems in software development, presenting itself as an efficient process in the organization and in the standardization of basic activities. Through a bibliographical study, this work presents the main concepts and features related to MPS.BR model.

Keywords: Software Development; Software Models; CMM; CMMI; MPS.BR.

LISTA DE FIGURAS

Figura 1: Níveis de maturidade do modelo CMM.....	19
Figura 2: Objetivos principais das ACPs em cada nível do modelo CMM.....	20
Figura 3: Níveis de maturidade do CMMI.....	24
Figura 4: Categorias de custos – CMM e CMMI.....	28
Figura 5: Níveis de maturidade do MPS.BR.....	32
Figura 6: Processos e subprocessos do MPS.BR.....	34
Figura 7: Componentes do MPS.BR.....	35
Figura 8: Estrutura compacta do modelo do MPS.BR.....	36
Figura 9: Mapeamento entre os Níveis de Maturidade do MPS.BR e CMMI.....	39
Figura 10: Ascensão do CMMI.....	43
Figura 11: Evolução do MPS.BR.....	45

LISTA DE QUADROS

Quadro 1: Competências dos níveis de maturidade do CMMI	25
---	----

LISTA DE SIGLAS

- ACP – Áreas-Chave do Processo.
- BID – Banco Interamericano de Desenvolvimento.
- CMM – *Capability Maturity Model*.
- CMM-ACQ – *Capability Maturity Model for Acquisition*.
- CMMI – *Capability Maturity Model Integration*.
- CMMI DEV – *Capability Maturity Model Integration for Development*.
- CMMI-SW – *Capability Maturity Model Integration for Software*.
- CMMI-SE – *Capability Maturity Model Integration - Systems Engineering*.
- ETM – Equipe Técnica do Modelo.
- FCC – Fórum de Credenciamento e Controle.
- FINEP – Financiadora de Estudos e Projetos.
- IA – Instituições Avaliadoras.
- IEC – *International Electrotechnical Commission*.
- ISO – *International Organization for Standardization*.
- KPAs – *Key Process Areas*
- MA-MPS – Método de Avaliação para Melhoria de Processo de Software.
- MCT – Ministério da Ciência e Tecnologia.
- MN-MPS – Modelo de Negócio para Melhoria de Processo de Software.
- MPS.BR – Melhoria de Processo do Software Brasileiro.
- MR-MPS – Modelo de Referência para Melhoria de Processo de Software.
- NBR – Normas Brasileiras (ABNT).
- SEI – *Software Engineering Institute*.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro.
- SPICE – *Software Process Improvement and Capability Determination*.
- SW -CMM – *Capability Maturity Model for Software*.
- TO - Treinamento Organizacional.

SUMÁRIO

1 INTRODUÇÃO.....	11
2 PROCESSOS DE MELHORIA DA QUALIDADE DE SOFTWARE.....	14
2.1 Riscos do processo de desenvolvimento.....	14
3 CAPABILITY MATURITY MODEL (CMM) E CAPABILITY MATURITY MODEL INTEGRATION (CMMI).....	18
3.1 <i>Capability Maturity Model (CMM)</i>	18
3.1.1 <i>Níveis de maturidade do CMM</i>	18
3.1.2 <i>Áreas-chave de processo do modelo CMM</i>	19
3.1.3 <i>Descontinuação do CMM</i>	21
3.2 <i>Capability Maturity Model Integration (CMMI)</i>	21
3.2.1 <i>Objetivos do CMMI</i>	22
3.2.2 <i>Níveis de maturidade do CMMI</i>	23
3.2.3 <i>Processo de certificação no CMMI</i>	26
3.3 <i>Custos de CMM e CMMI</i>	27
3.3.1 <i>Custos diretos e indiretos</i>	27
3.3.2 <i>Custos de treinamentos</i>	29
4 MELHORIA DE PROCESSOS DO SOFTWARE BRASILEIRO (MPS.BR).....	31
4.1 <i>Níveis de maturidade e processos do MPS.BR</i>	32
4.2 <i>MPS.BR e seus procedimentos</i>	35
4.3 <i>Aspectos comparativos entre os modelos MPS.BR e CMMI</i>	37
4.4 <i>Normas Internacionais ISO/IEC 12207 e ISO/IEC 15504</i>	40
5 CONSIDERAÇÕES FINAIS.....	42
REFERÊNCIAS.....	46

1 INTRODUÇÃO

Os constantes avanços observados nas tecnologias utilizadas como ferramentas de trabalho nas organizações têm demonstrado grande importância em relação à funcionalidade, agilizando processos, reduzindo tempo de trabalho, diminuindo riscos e problemas e proporcionando maior interatividade entre setores, empresas e agentes participantes da cadeia produtiva e estratégica das organizações. Nesse sentido, a informática e seus diversos aplicativos, software e hardware, são responsáveis por grande parte da evolução que propicia a otimização das atividades e tarefas desempenhadas dentro das empresas.

Entretanto, ainda que situados em um contexto marcado por constantes transformações e avanços, esses sistemas também têm apresentado problemas e riscos, principalmente no que se refere à sua concepção, incluindo as etapas de planejamento, projeto e desenvolvimento. Esses riscos estão diretamente associados à qualidade do produto, especificamente do software, e à satisfação das expectativas dos clientes, o que, em muitas situações, compromete todo o seu processo de desenvolvimento.

Contudo, convém salientar que os profissionais das organizações desenvolvedoras de software (gerentes, desenvolvedores, analistas) e das empresas consumidoras (gestores, diretores e operacionais) têm consciência, ou já tiveram a experiência, de que uma das principais causas dos problemas do desenvolvimento de software é a desorganização do processo e a inexistência de padrões documentados focando o desenvolvimento e a manutenção do software.

Segundo SOFTEX (2006), visando a fornecer subsídios que auxiliem de forma efetiva as organizações a disporem de processos devidamente organizados, documentados, definidos e sincronizados, dentro do ciclo de desenvolvimento e aplicação de software, surgiram dois mecanismos: o CMM – *Capability Maturity Model* e o CMMI – *Capability Maturity Model Integration*. Esses dois modelos se preocupam com a qualidade do produto a ser desenvolvido nas organizações produtoras de software.

De acordo com Fiorini, Staa e Baptista (2000), o CMM descreve os estágios através dos quais as organizações de software evoluem quando elas definem, implementam, mensuram, controlam e melhoram seus processos de software. Essa

evolução passa a diferenciar as organizações maduras das demais. Já o CMMI é considerado uma versão evoluída do CMM e procura estabelecer um modelo único para o processo de melhoria corporativo, integrando diferentes modelos e disciplinas. Atualmente, o CMMI apresenta dois modelos: (1) *CMMI for Development* (CMMI-DEV) que se dirige ao processo de desenvolvimento de produtos e serviços; (2) *CMMI for Acquisition* (CMM-ACQ) que é voltado para os processos de aquisição e terceirização de bens e serviços.

Na esteira da concepção e disponibilização desses modelos, surgiu um modelo destinado às necessidades e expectativas do ambiente e dos problemas verificados na situação do desenvolvimento e gerenciamento de software brasileiro. O MPS.BR – Melhoria de Processo do Software Brasileiro é, conceitualmente, um programa que busca a excelência do software desenvolvido em território nacional. Ele está focado, ainda que não de maneira exclusiva, a atender micros, pequenas e médias empresas brasileiras produtoras de sistemas de software e que necessitam de melhorias significativas em seus processos, conforme SOFTEX (2006).

Para isso, é preciso que o MPS.BR esteja adequado ao perfil de empresas de diferentes tamanhos e com características variadas, sejam elas do setor público ou privado. Também é necessário que o sistema seja compatível com os padrões de qualidade aceitos internacionalmente e que tenha como pressuposto o aproveitamento de toda competência existente nos padrões, conceitos e processos dos modelos já disponíveis.

Assim, o objetivo geral deste estudo é demonstrar a funcionalidade proporcionada pelo modelo MPS.BR na resolução dos problemas verificados no desenvolvimento de software, apresentando-se como um modelo eficiente na organização e padronização dos procedimentos fundamentais dessa atividade. Assim, foram estabelecidos os seguintes objetivos específicos:

- Conceituar e contextualizar o modelo MPS.BR dentro do ambiente tecnológico, ressaltando suas características essenciais em comparação ao modelo CMMI;
- Demonstrar a importância e a aplicabilidade que o MPS.BR e seus procedimentos possuem para a melhoria das atividades implementadas pelas empresas desenvolvedoras de software;

- Identificar as principais vantagens e desvantagens do modelo MPS.BR em relação ao ambiente tecnológico e à melhoria da qualidade do software.

Inicialmente, vale ressaltar que a escolha do tema abordado neste trabalho se procedeu em razão da relevância que o assunto possui, nos dias atuais, tanto para o ambiente tecnológico quanto para o setor empresarial e acadêmico. Por tratar-se de um tema ainda novo e de pouco conhecimento público, acredita-se que o desenvolvimento e a apresentação de um estudo que vise caracterizar de forma consistente, embasado em uma bibliografia selecionada de forma criteriosa, pode proporcionar maior conhecimento sobre o assunto, destacando ainda algumas de suas características mais relevantes.

Assim, por se tratar de um trabalho que envolve uma pesquisa teórica, de baixo custo e que, dentre outras vantagens, proporciona ao pesquisador o desenvolvimento pleno de seus objetivos e propósito em tempo hábil e de forma menos complexa que outras existentes, acredita-se que a viabilidade global do presente trabalho consiste em uma das formas pelas quais o mesmo se justifica.

Em uma outra perspectiva, a elaboração, desenvolvimento e apresentação deste trabalho, com foco aprofundado no tema proposto e a partir de critérios adequadamente pré-estabelecidos, possibilitam a difusão dos conceitos e teorias relativas ao assunto enfocado, facilitando a sua compreensão, servindo como fonte de consulta para profissionais da área estudada e, ainda, estabelecendo-se como referência ou base para o desenvolvimento de novos estudos, projetos e pesquisas que possuam enfoque e objetivos similares.

O Capítulo 2 aborda os processos de melhoria da qualidade de software, descrevendo os diversos riscos do processo de desenvolvimento. No Capítulo 3, são apresentados os modelos de melhoria no desenvolvimento de software *Capability Maturity Model* (CMM) e *Capability Maturity Model Integration* (CMMI), suas principais características, objetivos e níveis de maturidade. O Capítulo 4 aborda especificamente melhoria de processos do software brasileiro (MPS.BR), suas características, níveis de maturidade e a importância do modelo e de seus procedimentos.

E, por fim, o Capítulo 5 apresenta as considerações finais.

2 PROCESSOS DE MELHORIA DA QUALIDADE DE SOFTWARE

Os procedimentos relativos ao desenvolvimento e à gestão de software estão, continuamente, sendo aprimorados e necessitando de novas ferramentas e modelos de trabalho que procurem otimizar seus resultados e adequar o produto às novas necessidades do mercado e aplicações que surgem a cada dia. Nesse sentido, existem alguns modelos e processos alternativos, bastante atuais e eficientes, que podem identificar problemas existentes ao longo do processo de desenvolvimento do produto e, de maneira satisfatória, prover soluções para que, ao seu final, ele possa estar em total adequação com o planejamento realizado inicialmente por seus analistas em conjunto com os clientes.

No entanto, antes que se procure dimensionar, analisar e/ou descrever o conteúdo desses processos de melhoria e de otimização da qualidade do desenvolvimento de software (definição, objetivos, características, níveis de maturidade e aplicações), torna-se imprescindível proceder à apresentação e descrição dos principais problemas que podem ocorrer no processo de desenvolvimento de software. Isso se torna necessário para que, de maneira conceitual, sejam explicitados as principais deficiências e riscos para os quais os processos de melhoria e qualidade direcionam suas ações.

2.1 Riscos do processo de desenvolvimento

Os riscos ou problemas que se relacionam às etapas correspondentes ao processo de desenvolvimento de software podem surgir em diversas formas e com diferentes níveis de ameaça ao projeto. Por isso, comumente os autores e especialistas da área apontam, como principais riscos, aspectos e fatores que estão ligados a diferentes etapas ou camadas correspondentes ao mesmo processo.

Pressman (2002) aponta que os riscos inerentes ao desenvolvimento de software incluem normalmente duas características básicas: incerteza – o risco pode ou não acontecer, ou seja, não existem riscos que sejam totalmente prováveis; e perda – se o risco torna-se real, conseqüências ou perdas indesejadas certamente

irão ocorrer. Assim, o autor considera ainda que “quando os riscos são analisados, é importante quantificar o nível de incerteza e o grau de perda, associados com cada risco”, PRESSMAN (2002, p.140).

Como forma de obter um cenário favorável a essa quantificação, Pressman (2002) sugere que os riscos podem ser classificados em diferentes categorias, o que facilitaria a sua identificação, análise e sistematização corretiva ou eliminação. São as suas classificações:

- **Riscos de projeto:** são aqueles problemas que ameaçam o plano do projeto. Quer dizer que, se os riscos do projeto tornam-se reais, é provável que o cronograma do projeto se atrase e que os custos relativos à sua execução aumentem. Riscos de projeto identificam problemas potenciais orçamentários, de cronograma, de pessoal (quantidade e organização), de recursos, do cliente e de requisitos e seu impacto no projeto de software.
- **Riscos técnicos:** são aqueles problemas que ameaçam a qualidade e a pontualidade do processo produtivo do software. Se um risco técnico se mostra iminente, a implementação pode mostrar-se impraticável ou, no mínimo, custosa. Os riscos técnicos estão associados a potenciais problemas de projeto, de implementação, de interface, de verificação, de controle e de manutenção. “Riscos técnicos ocorrem porque o problema é mais difícil de resolver do que pensamos”, PRESSMAN (2002, p.141).
- **Riscos do negócio:** são aqueles problemas que ameaçam a viabilidade do software ser construído e, frequentemente, comprometem o projeto ou o produto. Os prováveis riscos do negócio que Pressman (2002) relaciona são: (1) riscos de mercado: construir um produto ou um sistema excelente que ninguém realmente quer; (2) risco estratégico: construir um produto que não atende a atual estratégia geral de negócios da empresa; (3) risco comercial: construir um produto que a equipe de vendas não sabe como ou para quem vender; (4) risco gerencial: perda de apoio da gerência superior devido a mudança de enfoque ou de pessoal; (5) risco de orçamento: perda de comprometimento orçamentário ou de pessoal.

- **Riscos conhecidos:** são aqueles problemas que, segundo Charette (1989), citado por Pressman (2002, p.141), se refere aos riscos que podem ser descobertos após a avaliação cuidadosa do plano de projeto, do ambiente técnico e comercial, no qual o projeto está sendo desenvolvido, e de outras fontes de informação confiáveis. Dentro dessa categoria surgem dois tipos de riscos: (1) riscos previsíveis: são extrapolados de experiência de projetos anteriores (rotatividade de pessoal, comunicação deficiente com o cliente, diluição do esforço de pessoa na medida em que os pedidos de manutenção que surgem são atendidos); (2) riscos imprevisíveis: podem e efetivamente ocorrem, mas a sua identificação prévia possui elevado grau de dificuldade.

Em outra abordagem, Paula Filho (2003) analisa os diversos tipos de riscos que podem ocorrer com maior frequência durante o processo de desenvolvimento de software a partir da descrição e avaliação dos mesmos dentro de quatro categorias que, de acordo com o autor, representam a totalidade simplificada dos elementos imprescindíveis ao processo como um todo. Seus riscos são em relação:

- **Ao produto:** decorrem de possíveis defeitos referentes à sua definição, e, em muitos casos, a definição se mostra deficiente desde a origem; em outros casos, torna-se deficiente ao longo do processo. “Podem também ser introduzidas pelos desenvolvedores durante o desenho ou a implementação por falta de disciplina, agravada por desatenção do gerente de projeto”. (PAULA FILHO, 2003, p.57).
- **Aos processos:** Os riscos relativos a processos são bastantes comuns em organizações que utilizam processos informais, assumindo um caráter de fragilidade perante a esse tipo de erros.
- **As pessoas:** Muitos riscos que ocorrem com o processo de desenvolvimento ou planejamento de software podem ser atribuídos às pessoas que estão envolvidas. “Uma consequência frequente é que informação vital deixa de ser repassada pelos usuários aos desenvolvedores”. (PAULA FILHO, 2003, p.58)
- **À tecnologia:** Alguns erros e riscos podem ser causados ao processo de desenvolvimento de software em razão do fascínio que a tecnologia de ponta

exerce sobre as pessoas que, equivocadamente, acreditam que a sua utilização plena é incapaz de gerar qualquer tipo de problema. Os principais riscos relativos a esta categoria, de acordo com Paula Filho (2003), são:

- A crença indiscriminada nas soluções tecnológicas faz com que muitos desenvolvedores não percebam as limitações da própria tecnologia;
- Mudanças de ferramentas no decorrer do projeto;
- A falta de automação de algumas atividades.

Apesar da enorme variedade de problemas que caracterizam a crise do software, engenheiros de software e gerentes de projeto para desenvolvimento de sistemas computacionais tendem a concentrar suas preocupações no seguinte aspecto: *“a enorme imprecisão das estimativas de cronogramas e de custos de desenvolvimento”*. De fato, a experiência tem mostrado que custos podem ser ultrapassados em ordens de grandeza e cronogramas podem ser desrespeitados por prazos e meses ou anos. Na realidade, esse aspecto constitui a manifestação mais visível de outras dificuldades inerentes ao processo de desenvolvimento de software. (REZENDE, 2005, p.9).

Como podemos observar na citação acima, genericamente em relação aos riscos que podem ocorrer durante o planejamento, projeto ou desenvolvimento de software, Rezende (2005) trata tais problemas a partir de um conjunto denominado “crise do software”.

3 CAPABILITY MATURITY MODEL (CMM) E CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

3.1 Capability Maturity Model (CMM)

Existem vários sistemas que promovem a melhoria dos processos e ferramentas de desenvolvimento de software, sendo que dentre eles o *Capability Maturity Model (CMM)*, ou Modelo de Maturidade da Capacidade, recebe grande destaque por ter sido um dos primeiros a ser desenvolvido com essa finalidade e por, posteriormente, ter servido de modelo para o desenvolvimento de outros sistemas mais atuais e adaptados a outras condições e ambientes.

O CMM não é um modelo, pois não estabelece ações específicas a serem seguidas à risca. É um modelo que precisa ser estudado, compreendido e adaptado às características de cada organização. O modelo não diz *como* implementar determinadas práticas, apenas determina *o que* deve ser feito. (FIORINI, STAA, BAPTISTA, 2000, p.54).

Baseado nos estudos de Fiorini, Staa e Baptista (2000), o CMM é um processo sistematizado, concebido pelo SEI, *Software Engineering Institute*, na *Carnegie Mellon University* situada nos Estados Unidos da América, visando avaliar e melhorar a capacidade de organizações que produzem software, propondo um caminho gradual que leva as organizações a se aprimorarem continuamente em busca da própria solução de problemas inerentes ao desenvolvimento sistemático e modular de software.

3.1.1 Níveis de maturidade do CMM

O modelo CMM propõe cinco níveis distintos de maturidade conforme pode ser observado na Figura 1. Cada um com características próprias, com o objetivo de proporcionar uma determinação precisa e confiável do nível de capacidade da organização em relação à execução do processo de desenvolvimento de software.

Rezende (2005) esclarece que, com exceção do nível 1, cada nível determinado pelo CMM inclui várias Áreas-Chave de Processo (ACP), do inglês KPAs *Key Process Areas* que, por sua vez, são compostas por metas de melhoria do processo de cada nível a serem atingidas.

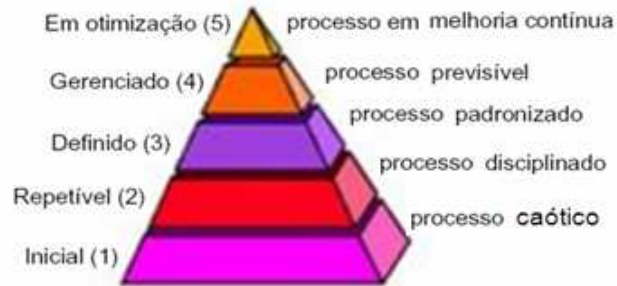


Figura 1: Níveis de maturidade do modelo CMM.¹
 Fonte: Adaptado de Fiorini, Staa e Baptista (2000).

Segundo explicam Rezende (2005) e Fiorini, Staa e Baptista (2000), os cinco níveis de maturidade do modelo CMM implicam em gestão ou ingestão em diferentes níveis e demonstram, individualmente, o grau de capacidade que a organização possui no que se refere aos aspectos como organização, controle e padronização dos procedimentos relativos ao processo de desenvolvimento e qualidade do produto.

3.1.2 Áreas-chave de processo do modelo CMM

As ACPs estão presentes em todos os níveis existentes no modelo CMM, com exceção do primeiro nível, e são detalhadas em práticas-chave de processo. “Essas áreas são exatamente o que a organização deve focar para melhorar o seu processo de desenvolvimento de software” (BARRETO, apud REZENDE, 2005, p.139). De acordo com os autores, para que uma organização possa demonstrar

¹ De acordo com Fiorini, Staa e Baptista (2000), no nível 1, chamado de Inicial, o desenvolvimento é caótico, pois não existem procedimentos padronizados, estimativas de custos e planos de projeto. O sucesso depende de esforços individuais. Não existe documentação e não há mecanismos de controle para que o gerente possa saber o que está acontecendo, identificando assim os problemas e riscos e agir de acordo. Como consequência, os desvios não são corrigidos e ocorrem os problemas tais como: prazos não cumpridos, orçamentos estourados, software sem qualidade e usuários insatisfeitos.

que se encontra situada em um determinado nível de maturidade CMM, é preciso que esteja continuamente realizando os processos relacionados às áreas-chave daquele nível e, dessa forma, cumprindo seus objetivos centrais, conforme ilustrado na Figura 2.

As ACPs estão organizadas em cinco seções chamadas de *Características Comuns* e que determinam as características de institucionalização ou de implementação das *práticas-chave* que, por sua vez, determinam as atividades e a infra-estrutura necessária para a melhoria de uma ACP. (REZENDE, 2005, p.137).

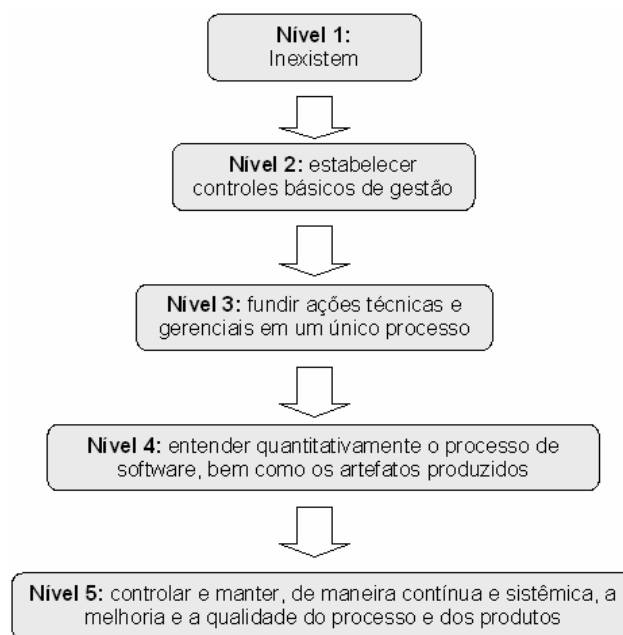


Figura 2: Objetivos principais das ACPs em cada nível do modelo CMM.
Fonte: Adaptado de Rezende (2005, p.139).

As áreas-chave de processo em cada nível do modelo CMM, segundo expõe Rezende (2005), são as seguintes:

- **Nível 1:** com relação ao processo inicial do projeto: pessoas competentes e heróis.
- **Nível 2:** com relação aos processos de gerenciamento de projetos: visão geral, supervisão e acompanhamento; garantia de qualidade do software, gestão de configuração do software; gestão de requisitos; gestão de contratos de software e sub-contratos; planejamento do projeto de software.

- **Nível 3:** com relação aos processos de engenharia e apoio: coordenação entre os grupos; gestão integrada de software; definição do processo organizacional; foco do processo organizacional; programa de treinamento; engenharia de produto de software; revisão conjunta por parceiros.
- **Nível 4:** com relação à qualidade do produto e do processo: gestão quantitativa dos processos; gestão da qualidade de software.
- **Nível 5:** Com relação ao melhoramento contínuo do processo: gestão da evolução e mudanças dos processos; gestão de mudanças tecnológicas; prevenção de defeitos.

3.1.3 Descontinuação do CMM

Embora o CMM fosse útil para as organizações, a aplicação de múltiplos modelos não integrados torna-se complicado em termos de treinamentos, avaliações e atividades de melhoria. Para que as organizações alcançassem êxito nas melhorias, seria necessário um modelo integrado, com suporte coerente para treinamento e avaliação, portanto, com a descontinuação do modelo CMM, as organizações foram migrando progressivamente para o CMMI, pois, de acordo com o SEI (2008), as avaliações de CMM não são mais válidas desde 31/12/2007.

3.2 Capability Maturity Model Integration (CMMI)

Assim como o CMM, o *Capability Maturity Model Integration* (CMMI) foi concebido pelo SEI e disponibilizado como um modelo de metodologia que procura promover melhorias no processo de desenvolvimento de software e no próprio produto, aumentando sua qualidade, eliminando riscos e provendo soluções aos problemas que, ocasionalmente, podem surgir.

De acordo com Sotille (2006), o CMMI, apesar de ser um modelo baseado no CMM, mas com maior abrangência e que reúne propostas dinâmicas de soluções para áreas específicas do processo, possui características próprias e diferentes daquelas demonstradas pelo primeiro modelo. Assim, “a proposta do CMMI é promover a melhoria dos processos das organizações e a habilidade de gerenciar, desenvolver e manter os seus produtos (software)” (SOTILLE, 2006, p.1).

3.2.1 Objetivos do CMMI

Os objetivos fundamentais do CMMI estão voltados, em primeiro plano, para o oferecimento de modelos, de forma sistematizada e padronizada, que possam agir como ferramentas eficazes na construção e sustentação de um processo de desenvolvimento de software que não apresente riscos para os desenvolvedores e que resulte em um produto de alta qualidade para os clientes.

Na visão de Souza (2005), os objetivos essenciais do CMMI encontram-se fundamentados em:

Ser um guia destinado a melhorar os processos organizacionais e a habilidade desses em gerenciar o desenvolvimento, a aquisição e a manutenção de produtos e serviços. O CMMI organiza as práticas, que já são consideradas efetivas, em uma estrutura que visa auxiliar a organização a estabelecer prioridades para melhoria e também fornece um guia para a implementação dessas melhorias. (SOUZA, 2005).

Para Sotille (2006), as principais metas e objetivos do modelo CMMI estão relacionadas à sua capacidade sistêmica em organizar práticas e processos vitais para o processo de desenvolvimento do produto, sua manutenção, análise e implementação contínua e sistematizada de melhorias. Assim, o autor entende que os objetivos desse modelo se associam à sua proposta de enquadrar uma empresa em um determinado nível de maturidade conforme sua situação gerencial e de empreendimento de melhorias no processo de desenvolvimento e no próprio software como produto final.

Para que esses objetivos possam ser alcançados de forma satisfatória e adequados às expectativas de cada empresa, Souza (2005) recomenda a adoção de

algumas práticas genéricas, às quais ele agrupa em quatro categorias, conforme demonstrado a seguir:

- **Comprometimento com a execução:** agrupa práticas relacionadas à definição de políticas e responsabilidades, descrevendo ações para assegurar que o processo se estabeleça e seja duradouro;
- **Habilitação para execução:** agrupa práticas contendo pré-condições para o projeto, de forma a permitir a implementação adequada do processo;
- **Direcionamento a implementação:** agrupa práticas relacionadas ao gerenciamento do desempenho do processo;
- **Verificação da implementação:** agrupa práticas para revisão junto à alta gerência e avaliação objetiva da conformidade com processos, procedimentos e padrões.

É importante ressaltar ainda que, para que o trabalho de melhoria proposto pelo modelo CMMI possa ser bem sucedido, as empresas precisam focar seus esforços na definição de metas específicas que, por sua vez, devem ser definidas de maneira criteriosa, respeitando limitações estruturais e estratégicas da organização. Souza (2005), no artigo mencionado acima, afirma que:

As metas específicas, na maioria das vezes, estão focadas no negócio da empresa e buscam alinhar a metodologia CMMI às necessidades próprias; por sua vez as metas comuns focam em aspectos inerentes a qualquer empresa e devem ser considerados para a correta implementação da metodologia de forma a garantir a maximização dos resultados (SOUZA, 2005).

3.2.2 Níveis de maturidade do CMMI

Assim como o modelo CMM, o modelo CMMI apresenta cinco níveis de maturidade que se relacionam com a forma como a empresa gerencia suas atividades e processos ligados ao planejamento, análise, controle, manutenção e

melhoria do desenvolvimento dos produtos. De acordo com Souza (2005), esses níveis são:

- **Nível 1 – Realizado:** nível inicial;
- **Nível 2 – Gerenciado:** gerenciamento de requisitos, planejamento de projeto, monitoramento e controle de projeto, gerenciamento de fornecedores, medição e análise, garantia de qualidade do processo e do produto, gerenciamento de configuração;
- **Nível 3 – Definido:** desenvolvimento de requisitos, solução técnica, integração do produto, verificação e validação, foco no processo organizacional, definição do processo organizacional, treinamento organizacional, gerenciamento de riscos, gerenciamento integrado do projeto, análise da decisão e resolução;
- **Nível 4 – Quantitativamente gerenciado:** gerenciamento quantitativo do projeto, performance do processo organizacional;
- **Nível 5 – Otimização:** análise causa e resolução, inovação organizacional e implantação.

A figura 3 ilustra os níveis de maturidade do CMMI.

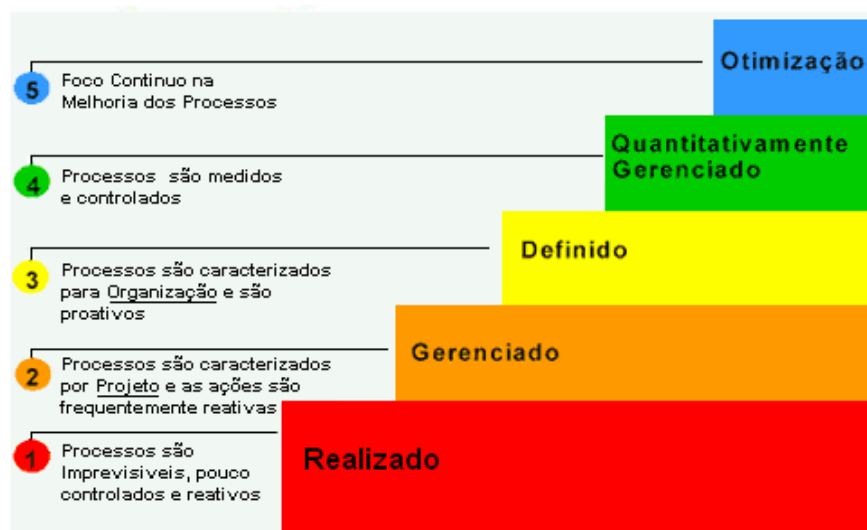


Figura 3: Níveis de maturidade do CMMI
Fonte: Adaptado de Souza (2005).

As competências relativas a cada estágio podem ser observadas de forma detalhada no Quadro 1.

Nível	Competências
1 - Realizado	Estágio inicial – completa falta de planejamento e controle dos processos. Os funcionários estão focados basicamente em atividades corretivas que surgem a todo o momento.
2- Gerenciado	São estabelecidos processos básicos de gerenciamento de projeto para planejar e acompanhar custos, prazos e funcionalidades. Compromissos são firmados e gerenciados. A disciplina de processo permite repetir sucessos de projetos anteriores em aplicações similares. Tipicamente, possui gerenciamento de projetos estabelecido; alguns procedimentos técnicos escritos; acompanhamento de qualidade; gerência de configuração inicial; atividades básicas de medição e análise. O sucesso depende basicamente do gerenciamento do projeto.
3 - Definido	Atividades de gerenciamento básico e as de Engenharia de Software são documentadas, padronizadas e integradas em processos-padrão. Todos os projetos de desenvolvimento ou manutenção de software utilizam uma versão de um desses processos adaptada às características específicas de cada projeto. Possui processos gerenciais e técnicos bem definidos, possibilidade de avaliação do processo; ferramentas e metodologias padronizadas; medições iniciais de desempenho; inspeções e auditorias rotineiras; testes padronizados; gerência de configuração; evolução controlada dos processos técnicos e gerenciais.
4- Quantitativamente gerenciado	Métricas detalhadas do processo de software e da qualidade do produto são coletadas. Tanto o processo quanto o produto de software são quantitativamente compreendidos, avaliados e controlados. Relatórios estatísticos são gerados. Tipicamente, encontra-se estabelecido e em uso rotineiro um programa de medições, a qualidade é planejada por um grupo dedicado, sendo rotineiramente avaliada e aprimorada
5 - Otimização	A melhoria contínua do processo é estabelecida por meio de sua avaliação quantitativa e da implantação planejada e controlada de tecnologias e idéias inovadoras. Projetos-piloto são realizados para a absorção e internalização de novas tecnologias. Tipicamente, um alto nível de qualidade e de satisfação dos clientes é alcançado rotineiramente, com grande foco na melhoria contínua.

Quadro 1: Competências dos níveis de maturidade do CMMI.

Fonte: Souza (2005).

Em entrevista com Barbieri (2008), tanto o CMM quanto o CMMI surgiram da mesma fonte, ambos são programas do governo americano concebidos pelo SEI, ambos se preocupam com a qualidade do produto definindo modelos de qualidade e maturidade no desenvolvimento de software.

Como existiam diversos processos não integrados dentro do CMM, surgiu então o CMMI que procura estabelecer o modelo único para os processos de melhoria, integrando diferentes modelos e disciplinas. Tendo suas representações como:

- **Representação Contínua:** Possibilita a organização utilizar a ordem de melhoria que melhor atender os objetivos de negócio da empresa. É caracterizado por Níveis de Capacidade (*Capability Levels*, em inglês).
- **Representação Por Estágios:** Disponibiliza uma seqüência pré-determinada para melhoria baseada em estágios que não deve ser desconsiderada, pois cada estágio serve de base para o próximo. É caracterizado por Níveis de Maturidade (*Maturity Levels*, em inglês).

Existem algumas diferenças básicas entre CMM e CMMI, tais como: O CMM possui diversos modelos com distintos processos de avaliação; O CMM possui elevado custo para uma empresa adotar mais de um modelo CMM; O CMM não é compatível com a norma ISO/IEC 15540; O CMM contempla apenas a representação por Estágio. O CMMI apresenta de maneira hábil e eficaz uma avaliação e melhoria dos diversos processos distintos em uma empresa, com uma visão comum e integrada de melhoria para todos seus elementos, custos reduzidos de formação e avaliação; Em cada modelo do CMMI, encontram-se disponíveis as duas formas de representações: por Estágio (níveis de maturidade com áreas de processo) e Contínua (áreas de processos com níveis de capacidade).

3.2.3 Processo de certificação no CMMI

O movimento de certificação e a necessidade de melhoria de qualidade dos produtos levam as empresas a adequar-se às novas regras de competitividade. Um dos indicativos é a certificação CMMI, aplicada também no Brasil em empresas que utilizam o modelo de fábrica de software como instrumento de desenvolvimento de sistemas.

Segundo Pressman (2002), a qualidade de software visa garantir a confiabilidade das propriedades do produto final e, para isto, utiliza uma combinação de modelos, ferramentas e de pessoal. No geral, um dos modelos utilizados para

garantir a qualidade é a padronização, ou seja, certificação de processos com base nas normas e modelos de qualidade.

3.3 Custos de CMM e CMMI

Segundo Della Volpe, Jamori e Zabeu (2007), os modelos que visam à introdução e sustentação de melhorias no processo de desenvolvimento de software – CMM e CMMI – apresentam alguns custos para que possam ser implementados ou otimizados em determinado período de tempo. As previstas adaptações e incorporações de novas atividades como forma de contribuir com maior nível de qualidade no desenvolvimento do produto confere às empresas maior responsabilidade em relação aos seus custos amplos, sendo que, no presente estudo, foram destacados os principais: custos diretos, custos indiretos e custos com treinamento profissional.

3.3.1 Custos diretos e indiretos

Ainda de acordo com Della Volpe, Jamori e Zabeu (2007), tanto no caso do CMM quanto do CMMI, os custos diretos e indiretos se encontram diretamente associados à qualidade dos processos, uma vez que, quando não existe conformidade em relação aos procedimentos de desenvolvimento de software, a empresa desenvolvedora precisa arcar com custos relativos à adequação dos mesmos e, por ocasião dessa adequação, alguns custos invariavelmente devem ser absorvidos.

Diante disso, observa-se que os custos da qualidade englobam os custos de não conformidade e os custos de conformidade. Os custos de não conformidade podem ser descritos como custos extras em face do fato de um produto ou serviço não ter sido desenvolvido (planejamento, processamento e acabamento) corretamente na primeira vez em que foi elaborado. Os custos da não conformidade podem incluir tanto as falhas internas quanto as falhas externas durante o

desenvolvimento do produto. Os custos de conformidade estão associados às necessidades de prevenção e avaliação de falhas e riscos ao projeto que está sendo desenvolvido, ainda que o mesmo esteja em conformidade com os requisitos imprescindíveis ao seu nível elevado e adequado de qualidade, segundo Della Volpe; Jamori e Zabeu (2007) e conforme ilustrado na figura 4.

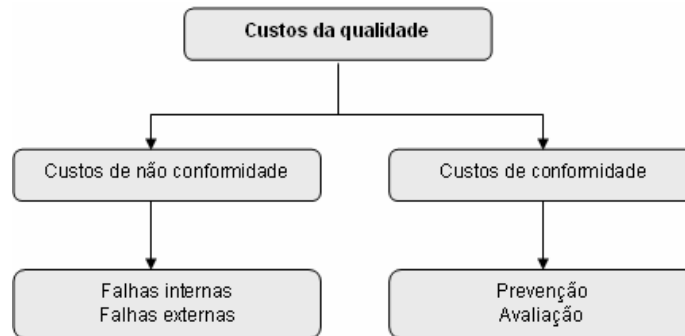


Figura 4: Categorias de custos – CMM e CMMI.

Fonte: Adaptado de Della Volpe, Jamori e Zabeu (2007).

Diretamente, pode-se compreender que os custos diretos e indiretos, associados à qualidade do produto desenvolvido, contemplam, na prática, as falhas internas, falhas externas, prevenção e avaliação de riscos, falhas e problemas identificados ao longo da realização dos processos que constituem o desenvolvimento de software:

- Falhas internas: custos associados aos defeitos encontrados antes da implementação;
- Falhas externas: custos associados aos defeitos encontrados durante e/ou após o produto estar implementado;
- Prevenção: custos associados à prevenção de defeitos, riscos e falhas;
- Avaliação: custos associados à observação de defeitos e propostas de melhorias ou eliminação dos mesmos.

3.3.2 Custos de treinamentos

Por conferirem maior qualidade aos produtos desenvolvidos, organização interna e eliminação das margens de erros, falhas e riscos, os modelos CMM e CMMI exigem grande capacidade operacional e treinamento especializado dos profissionais das empresas desenvolvedoras de software. Os custos com esses treinamentos são considerados relativamente elevados em razão de alguns aspectos fundamentais, destacados por Della Volpe, Jamori e Zabeu (2007):

- Existem poucas instituições e empresas especializadas em treinamento de pessoal para a área de desenvolvimento de software, especificamente para as metodologias CMM e CMMI;
- Por se tratar de metodologias relativamente novas e pouco difundidas no país, o treinamento de pessoal voltado para as metodologias CMM e CMMI constitui um investimento alto e, ao mesmo tempo, imprescindível para as empresas desenvolvedoras de software que pretendem otimizar seus processos em níveis elevados de qualidade;
- O treinamento organizacional, no que concebe à melhoria da qualidade de processos por meio da utilização do CMM e do CMMI, utiliza metodologia específica, atualizada e por meio de atividades consideradas inovadoras em relação às responsabilidades de alto grau de comprometimento dos profissionais treinados;
- O treinamento das habilidades voltadas para a utilização dos modelos CMM e CMMI engloba todos os profissionais que atuam na empresa e encontram-se envolvidos no processo amplo de desenvolvimento do produto.

De acordo com Della Volpe, Jamori e Zabeu (2007), a área de processo Treinamento Organizacional (TO) é responsável pelo desenvolvimento das habilidades e conhecimentos das pessoas para que elas possam desempenhar seu papel de forma efetiva e eficiente. Também é responsável por desenvolver, nos colaboradores, habilidades e conhecimentos que possibilitem realizar seus trabalhos

de forma efetiva e eficiente. Alguns objetivos específicos dessa área de processo estabelecem treinamentos das capacidades da organização em que os treinamentos são estabelecidos e mantidos. Também são previstos treinamentos individuais necessários para que o funcionário desenvolva suas tarefas mais eficientemente.

Os requisitos relacionados com o fornecimento de pessoal, infra-estrutura, instalações e o ambiente de trabalho necessário são tratados pela área de processo Planejamento do Projeto, pois é no plano de desenvolvimento do projeto que esses aspectos são abordados.

4 MELHORIA DE PROCESSOS DO SOFTWARE BRASILEIRO (MPS.BR)

O MPS.BR é um programa para Melhoria de Processo do Software Brasileiro, que começou a ser desenvolvido em dezembro de 2003, coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), e que contou com apoio do Ministério da Ciência e Tecnologia (MCT), da Financiadora de Estudos e Projetos (FINEP) e do Banco Interamericano de Desenvolvimento (BID).

A coordenação do Programa MPS.BR conta com duas estruturas de apoio para o desenvolvimento de suas atividades, o Fórum de Credenciamento e Controle (FCC) e a Equipe Técnica do Modelo (ETM). Através dessas estruturas, o MPS.BR obtém a participação de representantes de universidades, instituições governamentais, centros de pesquisa e de organizações privadas, os quais contribuem com suas visões complementares que agregam qualidade ao empreendimento.

O FCC tem como principal objetivo assegurar que as instituições implementadoras e as instituições avaliadoras sejam submetidas a um processo adequado de credenciamento e que suas atuações não se afastem dos limites éticos e de qualidade esperados, além de avaliar e atuar sobre o controle dos resultados obtidos pelo MPS.BR.

Paralelamente, a ETM atua sobre os aspectos técnicos relacionados ao Modelo de Referência (MR-MPS) e ao Método de Avaliação (MA-MPS), tais como a concepção e evolução do modelo, elaboração e atualização dos Guias do MPS.BR, preparação de material e definição da forma de treinamento e de aplicação de provas, publicação de Relatórios Técnicos e interação com a comunidade visando a identificação e aplicação das melhores práticas, conforme SOFTEX (2006).

Segundo explicam Koscianski e Soares (2007), a principal meta do MPS.BR está voltada para a definição e o aprimoramento de um modelo de melhoria e avaliação de processo de software, visando preferencialmente as micros, pequenas e médias empresas, de forma a atender as suas necessidades de negócio e ser reconhecido nacional e internacionalmente como um modelo aplicável às indústrias de software. O MPS.BR também estabelece um método de avaliação, o qual dá sustentação e garante que o MPS.BR está sendo empregado de forma coerente com as suas definições.

4.1 Níveis de maturidade e processos do MPS.BR

Os níveis de maturidade do MPS.BR, assim como dos outros sistemas de melhoria do processo de desenvolvimento de software apresentados, propõem o estabelecimento de patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos na organização. O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos. O MR-MPS define sete níveis de maturidade conforme pode ser observado na Figura 5. Que são: A (em otimização), B (gerenciado quantitativamente), C (definido), D (largamente definido), E (parcialmente definido), F (gerenciado) e G (parcialmente gerenciado). A escala de maturidade se inicia no nível G e progride até o nível A.

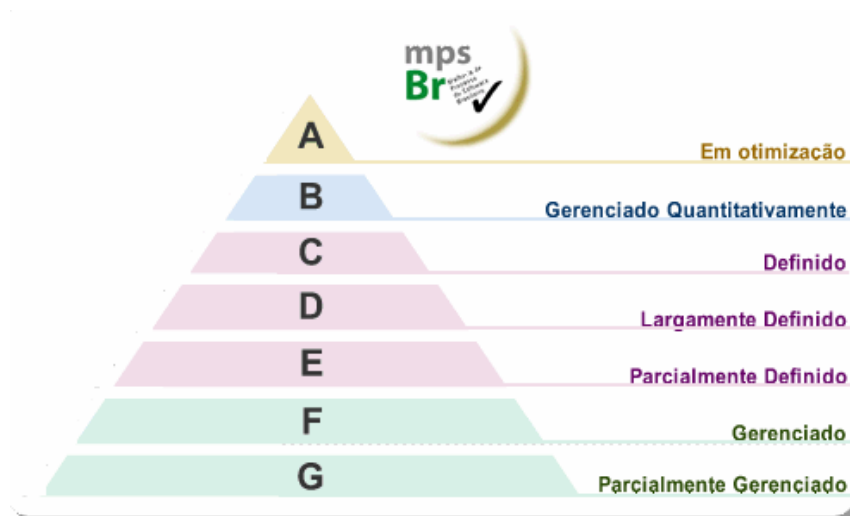


Figura 5: Níveis de maturidade do MPS.BR.
 Fonte: Adaptado de Koscianski e Soares (2007).

Na concepção de Ribeiro e Noemi (2007), existem fortes razões para que os modelos – como no caso do MPS.BR – sejam utilizados como referência. Algumas delas são:

- Modelos definem os requisitos a que os processos devem atender, apresentando flexibilidade em relação a como atendê-los;
- Modelos, especialmente os estruturados “por estágio”, definem um caminho evolucionário para a melhoria do processo;

- Modelos são repositórios de melhores práticas que vêm sendo utilizadas ao longo de vários anos com sucesso;
- Modelos permitem avaliações dos processos de forma objetiva e a identificação de pontos fracos e fortes.

A respeito dos propósitos conjuntos objetivados pelos níveis de maturidade do MPS.BR, a SOFTEX aponta que:

Para cada um destes sete níveis de maturidade, é atribuído um perfil de processos que indicam onde a organização deve colocar o esforço de melhoria. O progresso e o alcance de um determinado nível de maturidade MPS se obtém quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e dos atributos de processo estabelecidos para aquele nível (SOFTEX, 2006, p.15).

De acordo com a SOFTEX (2006), os processos no MR-MPS são descritos em termos de propósito, resultados e informações adicionais.

- Propósito: descreve o objetivo geral a ser atingido durante a execução do processo;
- Resultados: os resultados esperados do processo estabelecem os resultados a serem obtidos com a efetiva implementação do processo. Esses resultados podem ser evidenciados por um artefato produzido ou uma mudança significativa de estado ao se executar o processo;
- Informações adicionais: são referências que podem ajudar na definição do processo pela organização. Normalmente, elas citam o processo ou subprocesso da NBR ISO/IEC 12207 e a área de processo do CMMI-SE/SW que estão relacionados ao processo do MR-MPS. Essas referências fornecem descrições de atividades, tarefas e melhores práticas que podem apoiar a definição e implementação do processo nas organizações.

Conforme pode ser observado na Figura 6, os processos são agrupados, por uma questão de organização, de acordo com a sua natureza, ou seja, o seu objetivo

principal no ciclo de vida de software. Esse agrupamento resultou em três classes de processos, que são:

- **Processos fundamentais:** atendem o início e a execução do desenvolvimento, operação ou manutenção dos produtos de software e serviços correlatos durante o ciclo de vida de software;
- **Processos de apoio:** auxiliam um outro processo e contribuem para o sucesso e qualidade do projeto de software;
- **Processos organizacionais:** uma organização pode empregar esses processos em nível corporativo para estabelecer, implementar e melhorar um processo do ciclo de vida.

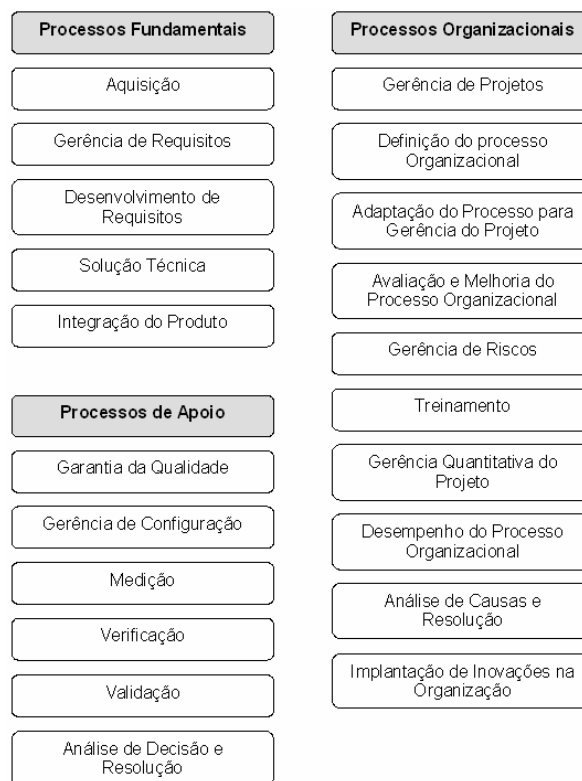


Figura 6: Processos e subprocessos do MPS.BR.
Fonte: SOFTEX (2006).

4.2 MPS.BR e seus procedimentos

O programa mobilizador MPS.BR está dividido em três componentes: Modelo de Negócio (MN-MPS), Modelo de Referência (MR-MPS) e Método de Avaliação (MA-MPS), conforme ilustra a figura 7.

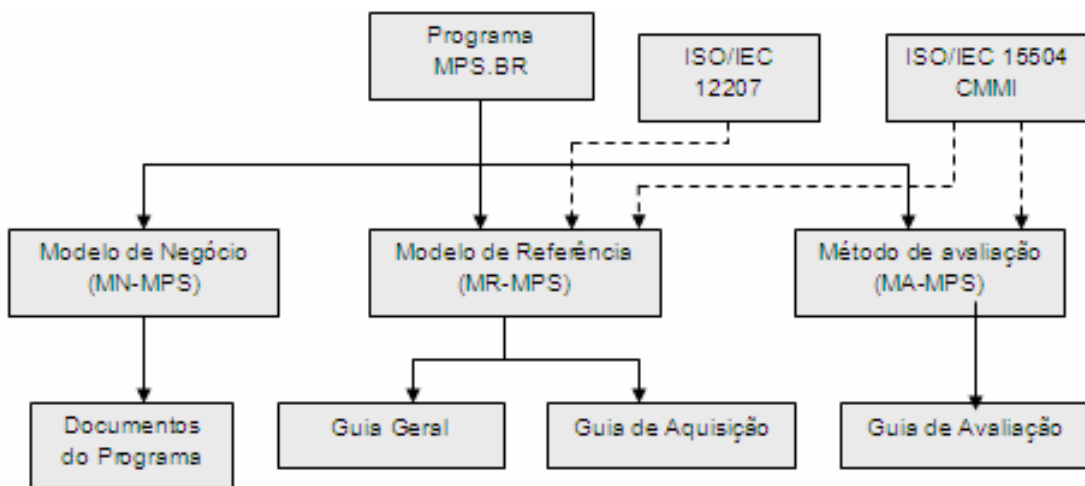


Figura 7: Componentes do MPS.BR.

Fonte: SOFTEX (2006), adaptado de acordo com Barbieri (2008).

O Modelo de Negócio descreve regras de negócio para implementação do MR-MPS pelas instituições implementadoras, avaliação seguindo o MA-MPS pelas Instituições avaliadoras, organização de grupos de empresas para implementação do MR-MPS e avaliação MA-MPS pelas instituições organizadoras de grupos de empresas, certificação de consultores de aquisição e programas anuais de treinamento por meio de cursos, provas e *workshops* MPS.BR.

O Modelo de Referência contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com os padrões de qualidades estabelecidos. Ele contém as definições dos níveis de maturidade, processos e atributos do processo.

O Guia de Aquisição, por sua vez, é um documento complementar destinado a organizações que pretendam adquirir software e serviços correlatos. O Guia de Aquisição não contém requisitos do MR-MPS, mas boas práticas para a aquisição de software e serviços correlatos.

O Guia de Avaliação contém o processo e o método de avaliação MA-MPS, os requisitos para os avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA). O processo e o método de avaliação MA-MPS está em conformidade com a norma ISO/IEC 15504-2 de 2003.

O Método de Avaliação contém o processo e os métodos recomendáveis de avaliação do modelo, os requisitos para os avaliadores líderes, avaliadores adjuntos e instituições avaliadoras.

Segundo Ribeiro e Noemi (2007), a estrutura do modelo MPS.BR, em sua forma abrangente e voltada para a qualidade de software, comporta três etapas fundamentais (qualidade do produto de software, qualidade do processo de desenvolvimento de software e modelo de maturidade) que estão embasadas em uma quarta etapa, o gerenciamento de projetos, conforme demonstra a Figura 8.

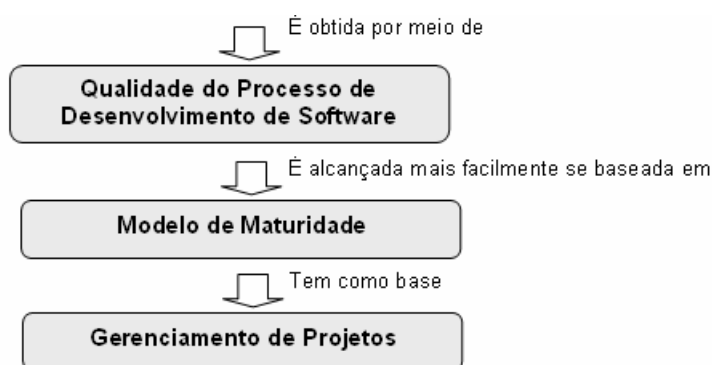


Figura 8: Estrutura compacta do modelo MPS.BR.

Fonte: Adaptado de Ribeiro e Noemi (2007).

Em um instante inicial, a importância do modelo MPS.BR pode estar associada unicamente à melhoria do processo de desenvolvimento de software pelas empresas desenvolvedoras brasileiras, ou seja, pode-se pensar que a sua aplicação se mostra relevante apenas no sentido de otimizar o nível de qualidade inerente ao produto final. No entanto, de acordo com Ribeiro e Noemi (2007), existem diversas outras importâncias que devem ser consideradas e associadas à implementação do MPS.BR, tanto em relação à qualidade do produto quanto à qualidade do processo. São elas:

- O processo é importante, pois mesmo as mais competentes e qualificadas pessoas não conseguem trabalhar de forma eficiente se o processo é problemático ou mal compreendido;

- O processo é uma junção que tende a unificar os outros aspectos;
- Investimentos em tecnologia sem um guia que defina como utilizá-la e sistematizá-la é um desperdício de recursos; não atinge os objetivos e, em pouco tempo, perde a sua verdadeira função;
- Sem processos claros e eficientes, uma empresa não consegue crescer e demonstrar evolução de maturidade em seus projetos e atividades;
- Oferece acesso à melhoria de processos a pequenas e médias empresas em larga escala;
- Fornece compatibilidade com os padrões de qualidade aceitos internacionalmente;
- Disponibiliza um caminho evolutivo mais suave e incremental que outros modelos.

Segundo Ribeiro e Noemi (2007), o modelo MPS.BR se mostra de grande importância em todas as etapas e processo relacionados ao conjunto de práticas que sustentam e possibilitam a disponibilização de um produto de software de alta qualidade para o mercado, ou seja, um produto altamente funcional, sem riscos para o usuário e para o cliente, voltado para os objetivos e expectativas a partir dos quais ele foi criado e de grande relevância no que se refere à sua atualização e aspectos inovadores em termos de sistema.

4.3 Aspectos comparativos entre os modelos MPS.BR e CMMI

Em entrevista, Barbieri (2007) define que, os modelos MPS.BR e CMMI demonstram diversos pontos comuns em suas diversas fases de implementação e também no que se refere aos aspectos organizacionais envolvidos em sua utilização sistematizada. Ambos os modelos requerem grande comprometimento das organizações desenvolvedoras de sistemas de software com os procedimentos a

serem implementados, partindo esse da alta direção das empresas e envolvendo todos os seus colaboradores e parceiros profissionais. As organizações precisam, nos dois modelos, estar dispostas a investir tempo, dinheiro e infra-estrutura na melhoria da qualidade de seus produtos e/ou serviços, ficando, dessa forma, responsáveis pelo treinamento de seus profissionais, reformulação de sua estrutura interna e adequação de sua cultura às mudanças institucionais imprescindíveis em sua nova forma de trabalhar os processos internos e externos.

Em entrevista, Barbieri (2007) ainda diz que, em sua concepção pragmática, os modelos MPS.BR e CMMI apresentam diferenças em relação aos seus níveis de maturidade. Ainda que propostos com a mesma finalidade, a de demonstrar a evolução de cada empresa de acordo com os níveis de segurança e qualidade demonstrados por elas, os níveis de maturidade dos modelos são apresentados em diferentes escalas quanto a seu processo de identificação e mensuração intangível da qualidade demonstrada pelas empresas usuárias. Conforme ilustrado na Figura 5, no caso do CMMI, cinco níveis de maturidade determinam a posição que a empresa ocupa na escala global do modelo e, conseqüentemente, essa metodologia de classificação mais reduzida, em comparação ao MPS.BR, Barbieri (2007) atenta para o fato de que no MPS.BR, os níveis de maturidade são sete e, dessa forma exigem que as empresas se despendam de maior prazo para que seja agregado um número maior de valores e requisitos essenciais para a sua escala até o último nível que assegura confiança e qualidade total nos processos desenvolvidos.

Embora a certificação tenha sido, a princípio, elaborada para micro e pequenas empresas, grandes empresas de software estão fazendo o MPS.BR. Existem empresas, como a IBM Brasil e a GVS - Global Value Soluções, cujo caminho natural é fazer o CMMI, mas que, depois de obterem a certificação, têm a intenção de adotarem o modelo brasileiro (BARBIERI, 2007).

Na concepção da SOFTEX (2006), existe uma equivalência entre o MPS.BR e o CMMI, conforme pode ser observado na Figura 9. Essa equivalência é total do ponto de vista do CMMI para o MPS.BR, isto é, todos os requisitos das áreas de processo do CMMI estão presentes no MPS.BR. Entretanto, não existe equivalência total do ponto de vista do MPS.BR para o CMMI, pelas seguintes razões:

- No nível E do MPS.BR, o processo Gerência de Recursos Humanos inclui os requisitos da área de processo Treinamento Organizacional, mas tem requisitos

relacionados à Aquisição de Pessoal e Gerência de Conhecimento que não estão presentes na área Treinamento Organizacional do CMMI;

- No nível E do MPS.BR, existe o processo Gerência de Reutilização que não existe no CMMI;
- No nível C do MPS.BR, existe o processo Desenvolvimento para Reutilização que não existe no CMMI.

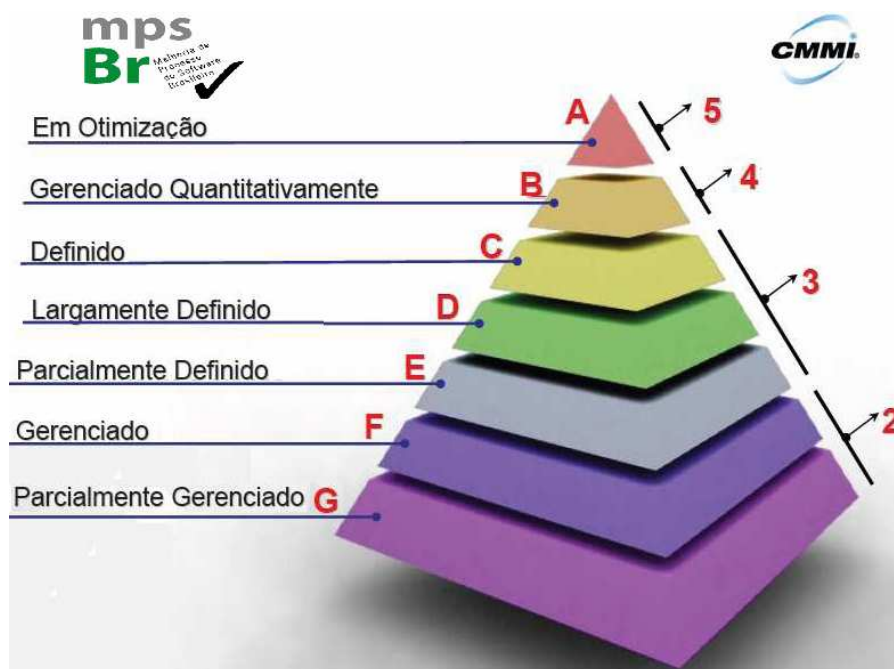


Figura 9: Mapeamento entre os Níveis de Maturidade do CMMI e MPS.BR

Fonte: Adaptado de <<http://www.softex.br>>. Acesso em: 04 out 2008.

4.4 Normas Internacionais ISO/IEC 12207 e ISO/IEC 15504

A Norma ISO/IEC 12207 foi criada pela ISO – *International Organization for Standardization* e o IEC – *International Electrotechnical Commission*, dentro de um esforço conjunto dessas organizações.

De acordo com a SOFTEX (2006), em 1988 foi proposto o desenvolvimento da norma e, em agosto de 1995, ela foi publicada como norma internacional. Em 1998, foi publicada a sua versão brasileira que tem o mesmo nome que a internacional, somente acrescida das iniciais NBR. Em outubro de 2002 e 2004, foram feitas atualizações na norma ISO/IEC 12207, chamadas de emendas 1 e 2 respectivamente, em que foram implantadas algumas melhorias. Essas melhorias criaram novos processos e expandiram escopos de alguns já existentes. Foram inseridas para esses o seu propósito e resultados e para os novos processos, suas atividades e tarefas foram definidas. Essas modificações têm o objetivo de representar a evolução da Engenharia de Software, as necessidades vivenciadas pelos usuários da norma e a harmonização com a série ISO/IEC 15504 – Avaliação de Processo.

A norma ISO/IEC 12207 e suas emendas 1 e 2 estabelecem uma arquitetura comum para o ciclo de vida de processos de software com uma terminologia bem definida. Contém processos, atividades e tarefas a serem aplicadas durante o fornecimento, aquisição, desenvolvimento, operação e manutenção de produtos de software e serviços correlatos.

Segundo a SOFTEX (2006), os processos definidos na NBR ISO/IEC 12207 devem ser utilizados como referência na implementação do MR-MPS e avaliação seguindo o MA-MPS. É possível realizar inclusões de novos processos ou exclusões e alterações de processos que não sejam pertinentes ao negócio, seguindo o processo de adaptação da NBR ISO/IEC 12207.

De acordo com Della Volpe, Jamori e Zabeu (2007), a norma ISO/IEC 15504 foi desenvolvida desde 1993 em conjunto com a comunidade internacional através do projeto SPICE (*Software Process Improvement and Capability Determination*) com base em modelos já existentes, como ISO 9000 e SW-CMM (*Capability Maturity Model for Software*). O propósito desse padrão é harmonizar modelos diferentes (incluindo SW-CMM, CMMI, ISO 9001) e métodos de avaliação.

A norma define um guia para orientação da melhoria de processo, avaliando-os segundo um modelo bidimensional, contendo a dimensão dos processos e a dimensão da capacidade. A primeira dimensão caracteriza os processos de acordo com os seus objetivos e resultados esperados e é fornecida por um modelo de processos externo à ISO/IEC 15504, gerando um perfil dos processos e identificando os pontos fracos e fortes; esses processos serão utilizados para a elaboração de um plano de melhorias.

Ainda segundo Della Volpe, Jamori e Zabeu (2007), a segunda dimensão define a forma de medição do estado desses processos. Equivalentemente, a primeira dimensão reúne práticas relacionadas a "o que fazer", organizadas em processos específicos; a segunda agrupa práticas relacionadas ao "quão bem fazer qualquer coisa que seja feita", organizadas em níveis de capacidade genéricos.

O ISO/IEC 15504, apesar de ser um esforço independente, não foi projetado para uso autônomo. Seu papel é fornecer uma estrutura que modelos existentes e que venham a ser criados possam seguir. Assim, uma das especificações do CMMI declara que ele deve ser consistente e compatível com o ISO/IEC 15504.

A norma ISO 15504 (1997) define um modelo bi-dimensional que tem por objetivo a realização de avaliações de processos de software com foco na melhoria dos processos e na determinação da capacidade dos processos viabilizando a avaliação de um fornecedor em potencial.

5 CONSIDERAÇÕES FINAIS

Diante as exposições teóricas e argumentações conceituais apresentadas ao longo do presente trabalho, tornou-se possível concluir que o processo de desenvolvimento de software mostra-se como uma atividade que requer cuidados criteriosos e contínuos para que possa se estabelecer de maneira adequada e obter os resultados esperados pelas fábricas de software, tanto no que diz respeito à sua aplicabilidade quanto no que se refere a satisfação das empresas clientes (usuárias e distribuidoras dos produtos). Esses cuidados requeridos encontram-se presentes dentro das diversas etapas que constituem o processo desenvolvedor, tais como planejamento, projeto, desenvolvimento, gerenciamento, controle e avaliação, e estão associados à identificação de riscos e problemas existentes no processo e também à implementação e controle de medidas ou modelos que possam solucioná-los, de acordo com Pressman (2002).

Dentro do ambiente de desenvolvimento de software, também pode-se compreender que é de vital importância que haja um excelente nível de harmonia e compatibilidade entre as expectativas da empresa cliente, que encomenda o desenvolvimento do produto, e aquele que se dispõe a desenvolvê-lo e entregá-lo. Nesse sentido, aspectos como o cronograma de trabalho, prazo de entrega, características e aplicações do produto e demais fatores intervenientes devem ser discutidos, planejados e analisados previamente, procurando atender as necessidades do cliente, mas sem deixar de se adequar às condições de trabalho (prazo, capacidade, capacitação, infra-estrutura) do fabricante, conforme Pressman (2002).

Em outra perspectiva, também foi possível concluir que os principais riscos relativos ao desenvolvimento de software encontram-se inseridos em quatro dimensões básicas (produto, projeto, pessoas e tecnologia) e a sua identificação e mensuração constituem atividades fundamentais para a tentativa de melhoria ou eliminação dos mesmos e ainda da consecução dos objetivos traçados pela empresa desenvolvedora em parceria com a empresa cliente. A identificação desses problemas ou riscos tende a permitir que a empresa seja capaz de avaliar, de forma criteriosa, qual o melhor modelo a ser utilizado para minimizá-los, ou até mesmo,

eliminá-los e, assim, o processo possa ser desempenhado de forma segura e compatível com as metas e propósitos traçados, de acordo com Paula Filho (2003).

Os conceitos apresentados, também permitiram que se chegasse à conclusão de que, atualmente, existem processos e metodologias eficazes para que os riscos inerentes ao desenvolvimento de software possam ser trabalhados de forma a otimizar todo o processo e, assim, dotá-lo de um grau satisfatório de organização, sistematização e controle gerencial, atendendo os requisitos de qualidade pretendidos e, contudo, satisfazendo com plenitude as exigências e expectativas dos clientes. Nesse sentido, o CMM – *Capability Maturity Model* e CMMI – *Capability Maturity Model Integration*, além das Normas de Gestão da Qualidade e Garantia da Qualidade da série ISO 9000, se apresentam como metodologias processuais, com reconhecimento em âmbito internacional, que se preocupam com a qualidade do produto a ser desenvolvido nas empresas desenvolvedoras de software. Esses modelos, fundamentalmente, procuram otimizar todo o processo de desenvolvimento de produtos das empresas desenvolvedoras de software, oferecendo-lhes, por meio de adequação aos seus níveis de maturidade, condições de evoluírem qualitativamente e de eliminarem problemas e riscos que possam comprometer aspectos como a aplicação, funcionalidade e resultados gerados por seus produtos, de acordo com Rezende (2005) e Fiorini, Staa e Baptista (2000).

A figura 10 ilustra ascensão do CMMI em relação ao CMM.

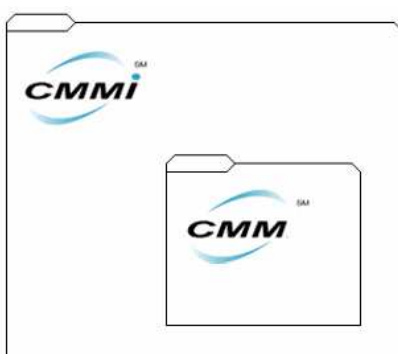


Figura 10: Ascensão do CMMI.
Fonte: Arquivo pessoal.

De acordo com a SOFTEX (2006), o procedimento de Melhoria de Processo do Software Brasileiro – MPS.BR, inspirado conceitualmente no CMM e no CMMI, se apresenta como uma ferramenta de solução eficaz para a organização,

sistematização e gerenciamento de software fabricados por empresas brasileiras por demonstrar níveis de maturidade, modelos e procedimentos adequados à realidade dessas empresas e, assim, permitir que as mesmas possam produzir com elevado nível de qualidade, segurança e confiabilidade. Nesse contexto e baseado em Ribeiro e Noemi (2007), foi possível concluir que as principais vantagens oferecidas pelo MPS.BR estão associadas a aspectos como:

- O direcionamento, por meio de estágios, do caminho evolucionário que a empresa deve tomar;
- A disponibilização de modelos que permitem avaliações precisas e objetivas de cada atividade,
- Etapa e procedimento que compõem o processo de desenvolvimento do produto;
- Economia de recursos e investimentos através da adequação da capacidade operacional às exigências propostas por cada nível de maturidade do modelo;
- Fornecimento de processos e metodologias transparentes e eficazes para que a empresa cresça e demonstre qualitativamente um nível satisfatório de maturidade em todos os seus projetos, atividades e procedimentos;
- O fornecimento de compatibilidade com os padrões de qualidade aceitos internacionalmente.

De forma geral, pode-se concluir que a Melhoria de Processo do Software Brasileiro – MPS.BR, enquanto metodologia que visa a implementação de melhorias escalonadas e gradativas em todas as etapas presentes no processo de desenvolvimento de software, oferece recursos que identificam e minimizam riscos, proporcionando maior flexibilidade no atendimento dos requisitos aos quais o modelo deve se submeter e, de forma sistemática, organiza todas as etapas, bem como suas atividades, que estão presentes ao longo do processo de desenvolvimento, facilitando, assim, o seu planejamento, implementação, controle, gerenciamento e avaliação contínua. Essas características fazem com que o

MPS.BR seja visto como um modelo adequado às empresas brasileiras que buscam a evolução da qualidade de seus produtos, projetos e capacitação dos recursos empregados no desenvolvimento do software produzido.

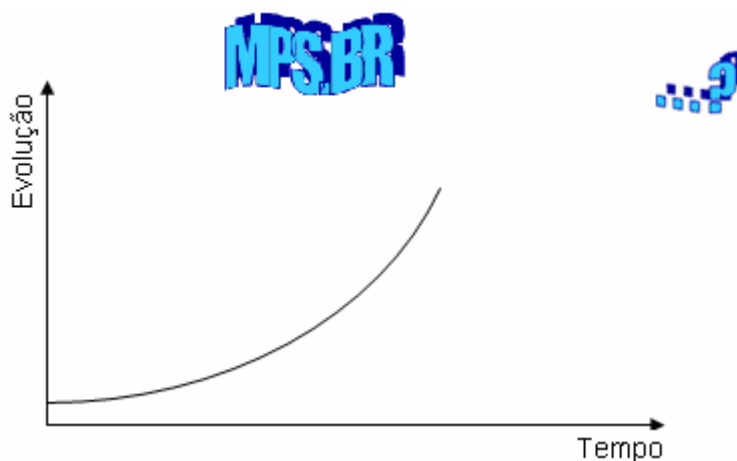


Figura 11: Evolução do MPS.BR.

Fonte: Arquivo pessoal.

A figura 11 ilustra a evolução do MPS.BR desde sua criação em dezembro de 2003. Como será a evolução do MPS.BR? Será que conseguirá atender empresas a nível internacional?

Em entrevista, Barbieri (2007) menciona as metas e desafios do modelo MPS.BR. Sua principal meta consiste na disseminação regional desse modelo na Argentina e no Chile e posteriormente divulgação em outros países latino-americanos, como Peru e Uruguai. Seus principais desafios são: comprometer mais pessoas qualificadas com o modelo, instituições experientes e organizações interessadas; aprimorar anualmente os Guias do MPS.BR; reforçar a capacitação de pessoas por meio de cursos, provas e *workshops*; certificar consultores; credenciar novas instituições implementadoras e avaliadoras.

REFERÊNCIAS

BARBIERI, Carlos. **MPS.BR**. Entrevista concedida a Vinícius Manhães Teles, em 12 de julho de 2007. Disponível em: < <http://blog.improveit.com.br/articles>>. Acesso em: 17 jun 2008.

BARBIERI, Carlos. **MPS.BR**, 1:39 de gravação, em 10 out 2008. Entrevista concedida a Marley Fernandes e Samuel da Silva Correa.

CARVALHO, A. M. B. R.; CHIOSSI, T. C. S. **Introdução à engenharia de software**. Campinas: Unicamp, 2001.

CHARRETE, R. N. **Software Engineering Risk Analysis and Management**. SL: McGraw-Hill/Intertext, 1989.p. 49 apud PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: McGraw-Hill, 1995.

CHIAVENATO, Idalberto. **Gestão de pessoas: o novo papel dos recursos humanos**. Rio de Janeiro: Elsevier, 1999.

DELLA VOLPE, Renato Luiz; JAMORI, Sérgio Massao; ZABEU, Ana Cecília P. **Benefícios, aplicação prática de CMMI e MPS.BR e resultados brasileiros**. São Paulo: SIMPROS – Simpósio Internacional de Melhoria de Processos de Software, 2007.

FIORINI, Soeli T.; STAA, Arndt Von; BAPTISTA, Renan Martins. **Engenharia de software com CMM**. Rio de Janeiro: Brasport, 2000.

KOSCIANSKI, André; SOARES, Michel dos Santos. **Qualidade de software**. São Paulo: Novatec, 2007.

MARTINS, José Carlos Cordeiro. **Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML**. Rio de Janeiro: Brasport, 2006.

PAULA FILHO, Wilson de Pádua. **Engenharia de software: fundamentos, métodos e padrões**. Rio de Janeiro: LTC, 2003.

PRESSMAN, Roger S. **Engenharia de software**. 1ª edição. São Paulo: Makron Books, 1995.

PRESSMAN, Roger S. **Engenharia de software**. 5ª edição. Rio de Janeiro: McGraw-Hill, 2002.

REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**. Rio de Janeiro: Brasport, 2005.

RIBEIRO, Adriele; NOEMI, Marta. **Gerenciamento de projetos, MPS.BR e qualidade de software**. Belo Horizonte: Project Management Institute (PMI), 2007.

SEI (*Software Engineering Institute*). **Reminder of the CMMI Product Suite Version 1.1 and SW-CMM Sunset Dates**. *Carnegie Mellon University* (2008). Mensagem Eletrônica. Disponível em <<http://www.sei.cmu.edu/about/press/releases>>. Acesso em: 13 out 2008.

SOFTEX – Associação para Promoção da Excelência do Software Brasileiro.
MPS.BR – Melhoria de Processo do Software Brasileiro. Brasília: SOFTEX, 2006. Disponível em: <http://www.softex.br/mpsBr/_home/default.asp>. Acesso em: 15 de março de 2008.

SOTILLE, Mauro Afonso. **O gerenciamento de projetos de software e a implementação do CMMI (Capability Maturity Model Integration)**. Artigo. PMTEC, setembro de 2006. Disponível em: <http://www.pmttech.com.br/artigos/GP&CMMI-Mauro_Sotille.pdf>. Acesso em: 12 de maio de 2008.

SOUZA, Adilson Moreira de. **Implementando o CMMI (Capability Maturity Model Integration) como ferramenta para gerenciamento de projetos de software**. Artigo. Cosmo On Line, 01 de dezembro de 2005. Disponível em: <<http://kplus.cosmo.com.br/materia.asp?co=30&rv=Vivencia>>. Acesso em: 12 de maio de 2008.