

**Sandro Marcelo Melhoretto**

Agradecimentos



Para que este trabalho pudesse ser feito devo agradecer  
aos meus pais, que sempre estiveram presentes por que passou o  
período que mais me custou a escrever e a transferir o texto

## **Um Método para a Melhoria das 3 Dimensões do Processo de Software**

Dissertação apresentada ao curso de Mestrado em  
Informática Aplicada da Pontifícia Universidade Católica  
do Paraná, como requisito parcial para obtenção do título  
de “Mestre em Ciências”.

Orientador: Prof. Dr. Roberto Cesar Betini

Curitiba

1999

## Agradecimentos

Para que este trabalho pudesse ter sido feito devo agradecer especialmente aos meus pais, que nos momentos mais difíceis por que passei durante o período que compreendeu o mestrado não deixaram de transferir o apoio e o conforto necessários para dar continuidade a árdua tarefa de aliar o estudo e o trabalho. Não poderia deixar de citar aqui as inestimáveis dicas, considerações e apoio do engenheiro Paulo Ricardo Stark, do sempre amigo e colega de mestrado Marco Paludo e do incentivo oportuno dos colegas Luís Marcelo, Bertholdi e Paulo César, cujas companhias foram muito gratificantes.

Agradeço o empenho e os esforços despendidos pelo meu orientador Dr. Roberto Cesar Betini na condução para que este trabalho obtivesse êxito, e também pela sua colaboração na avaliação e análise de cada linha por mim escrita durante todo o período que compreendeu o desenvolvimento desta dissertação.

Agradeço a Siemens Ltda. pela oportunidade a mim dada para fazer este mestrado e poder melhorar os meus conhecimentos e minhas habilidades, fornecendo-me todo o apoio possível para que tivesse êxito na conclusão deste trabalho.

Agradeço também ao coordenador do mestrado, Dr. Robert Burnet pela sua disposição e esforço em prover-nos com todos os meios necessários para a realização das atividades que compreenderam a realização desta dissertação.

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>O Modelo IDEAL</b>	<b>7</b>
2.1	Fase 1 - Inicial	8
2.2	Fase 2 - Diagnose	9
2.2.1	<i>CMM Appraisal Framework (CAF)</i>	10
2.2.2	<i>CMM Based Appraisal Internal Process Improvement (CBA IPI)</i>	15
2.2.3	<i>SPICE Assessment</i>	19
2.2.4	<i>Questionário de maturidade integrado</i>	20
2.3	Fase 3 - Estabelecimento	22
2.4	Fase 4 - Ação	23
2.5	Fase 5 - Aprendizagem	26
2.6	Aplicando o Modelo	27
2.7	O IDEAL no Método Proposto	28
<b>3</b>	<b>Modelos de Melhoria de Processos</b>	<b>29</b>
3.1	CMM – Capability Maturity Model	29
3.2	SPICE – ISO 15504	34
3.3	PSP – Personal Software Process	41
3.4	TSP – Team Software Process	47
<b>4</b>	<b>O Método Integrado</b>	<b>54</b>
4.1	Objetivo	54
4.2	Relação entre os modelos	55
4.3	Impacto no desempenho	61
4.4	Ciclos de Melhoria	63
4.4.1	<i>Primeiro Ciclo de Melhoria</i>	<i>64</i>
4.4.1.1	Práticas a serem Implementadas	65
4.4.1.2	Impacto no Desempenho da Organização	70
4.4.1.3	Questionário Integrado para o Primeiro Ciclo	71
4.4.2	<i>Segundo Ciclo de Melhoria</i>	<i>75</i>
4.4.2.1	Práticas a serem Implementadas	75
4.4.2.2	Impacto no Desempenho da Organização	82
4.4.2.3	Questionário Integrado para o Segundo Ciclo	83

4.4.3	<i>Terceiro Ciclo de Melhoria</i> .....	87
4.4.3.1	Práticas a serem Implementadas.....	87
4.4.3.2	Impacto no Desempenho da Organização.....	91
4.4.3.3	Questionário Integrado para o Terceiro Ciclo.....	92
4.4.4	<i>Quarto Ciclo de Melhoria</i> .....	93
4.4.4.1	Práticas a serem Implementadas.....	94
4.4.4.2	Impacto no Desempenho da Organização.....	95
4.4.4.3	Questionário Integrado para o Quarto Ciclo.....	95
4.5	Resumo do Método Integrado.....	96
<b>5</b>	<b>Estudo de Caso</b> .....	<b>98</b>
5.1	O Processo de Engenharia.....	99
5.2	Motivação.....	102
5.3	A Implementação do Programa de Melhoria pela Siemens.....	103
5.4	Processos Implementados.....	108
5.5	Resultados Obtidos.....	117
<b>6</b>	<b>Conclusão</b> .....	<b>120</b>

## Índice de Figuras

Figura 1 – Definição de processo de software.....	2
Figura 2 – Atributos para o sucesso de um programa de melhoria de processos.....	3
Figura 3 – As três dimensões da melhoria de processos.....	5
Figura 4 – O modelo IDEAL.....	8
Figura 5 – Diagrama conceitual de uma avaliação baseada no CMM.....	12
Figura 6 – Atividades primárias de um método de avaliação obediente ao CAF.....	13
Figura 7 – Fluxo de atividades de um método de avaliação.....	14
Figura 8 – Cronologia das Atividades <i>in loco</i> .....	17
Figura 9 – Avaliação do processo de software pelo SPICE.....	20
Figura 10 – Visão bidimensional da atividade de melhoria de processo.....	27
Figura 11 – Os cinco Níveis do CMM.....	30
Figura 12 – Estrutura do CMM para Software.....	31
Figura 13 – Áreas Chave do Processo para cada nível de maturidade.....	32
Figura 14 – Resultado de 852 avaliações efetuadas e sua distribuição nos níveis de maturidade.....	33
Figura 15 – Arquitetura do modelo SPICE.....	35
Figura 16 – Relacionamento entre as categorias de processo.....	39
Figura 17 – Evolução do PSP.....	43
Figura 18 – Densidade de defeitos em cada fase do ciclo de desenvolvimento.....	47
Figura 19 – Estrutura do TSP.....	49
Figura 20 – Exemplo de um componente de perfil de qualidade.....	53
Figura 21 – Grau de cobertura do CMM em relação ao SPICE.....	56
Figura 22 – Evolução da estimativa com a melhoria do processo.....	62
Figura 23 – Evolução do tempo médio em cada fase do desenvolvimento através do PSP.....	77
Figura 24 – Melhora na estimativa de tempo de implementação.....	78
Figura 25 – Resultados obtidos quanto a melhora na qualidade da implementação.....	78
Figura 26 – Perfil de qualidade do projeto analisado.....	89
Figura 27 – Esforços em diferentes organizações.....	101
Figura 28 – Seqüência adotada na Siemens para a condução da avaliação do processo.....	104
Figura 29 – As 4 fases de um projeto de melhoria na Siemens.....	106
Figura 30 – Responsabilidades do PITs e do MSG.....	107
Figura 31 - Estrutura das Categorias e Áreas de Processo do SyDeP.....	111
Figura 32 – Descrição reduzida do modelo de processos.....	115

## Índice de Tabelas

Tabela 1 – Período de tempo típico de uma avaliação através do CBA IPI. ....	17
Tabela 2 – Recursos requeridos segundo o método CBA IPI.....	18
Tabela 3 – Resumo dos resultados de melhoria de processos de software utilizando o CMM. ....	34
Tabela 4 – Passos no PSP0. ....	44
Tabela 5 – Estrutura dos procedimentos do PSP. ....	46
Tabela 6 – Dimensões do perfil de qualidade dos projetos segundo o TSP. ....	52
Tabela 7 – Relacionamento entre os modelos CMM SW v1.1 e SPICE v1.0. ....	55
Tabela 8 – Comparação entre as atividades do ciclo de vida e os modelos. ....	57
Tabela 9 – Como o PSP e o TSP abrangem as áreas chave do processo do CMM. ....	58
Tabela 10 – Processos e atividades a serem implementadas em cada ciclo de melhoria do método integrado. ....	60
Tabela 11 – Quantidade de práticas/atividades implementadas no primeiro ciclo. ....	69
Tabela 12 – Quantidade de práticas/atividades implementadas neste segundo ciclo. ....	82
Tabela 13 – Quantidade de práticas/atividades implementadas neste terceiro ciclo. ....	91
Tabela 14 – Tabela comparativa entre o método de avaliação da Siemens e da SEI. ....	105
Tabela 15 – Exemplo de relacionamento entre comitês, áreas chave de processo e funções operacionais. ....	116
Tabela 16 – Dados obtidos após o primeiro ciclo de melhoria. ....	118

## Abreviaturas

BOS	– Business Opportunity Scanning
CAF	– CMM Appraisal Framework
CBA IPI	– CMM Based Appraisal Internal Process Improvement
CMM	– Capability Maturity Model
CMU	– Carnegie Mellon University
CUS	– CUsomer-Supplier
ENG	– ENGEineering
IDEAL	– Initial, Diagnosing, Establishing, Acting, Learning
ISO	– International Standard Organization
KPA	– Key Process Area
ORG	- ORGanizational
PIP	– Process Improvement Proposals
PNQ	– Prêmio Nacional da Qualidade
PRO	– PROject
PSP	– Personal Software Process
SEI	– Software Engineering Institute
SPI	– Software Process Improvement
SPICE	– Software Process Improvement and Capability dEtermination
SUP	– SUPport
SyDeP	– System Development Process
TSP	– Team Software Process

## Resumo

Depois de duas décadas de promessas não cumpridas sobre ganhos de produtividade e qualidade na aplicação de novas metodologias e tecnologias para o desenvolvimento de software, a indústria e as organizações governamentais - principalmente americanas - estão percebendo que seu problema fundamental é a incapacidade de gerenciar o processo de software. Em várias organizações, os projetos atrasam excessivamente e o orçamento planejado chega a ser dobrado ao seu final. Em tais circunstâncias, a organização não está fornecendo a infra-estrutura e o suporte necessário para manter os projetos de acordo com o planejado e evitando que os problemas ocorram.

Mesmo em organizações que não possuem maturidade no seu processo de software, entretanto, alguns projetos de software específicos produzem excelentes resultados. Quando tais projetos alcançam êxito, é geralmente devido ao esforço heróico de uma equipe dedicada, e não da repetição de um processo de software maduro. O sucesso que repousa somente na disponibilidade de indivíduos específicos não fornece bases para melhoria da produtividade e qualidade de longa duração através de toda a organização. A melhoria contínua pode ocorrer somente através de esforços enfocados e sustentados em direção a construção de uma infra-estrutura de processo com práticas efetivas de engenharia de software e de gerenciamento.

Para obter esta melhoria do processo de software o SEI – Software Engineering Institute da Carnegie Mellon University desenvolveu alguns modelos que enfocam diferentes aspectos do processo de software. O CMM – Capability Maturity Model fornece uma orientação para a melhoria do processo de software de toda a organização, o PSP – Personal Software Process provê, ao engenheiro de software, uma seqüência de exercícios para melhorar o processo de software individual e o TSP – Team Software Process fornece uma orientação para as equipes de projeto de software para a melhoria do seu processo de software. A ISO, através do projeto SPICE – Software Process Improvement and Capability dEtermination, criou a norma 15504 que fornece um modelo de avaliação de processo de software. O modelo SPICE tem como foco a

## Resumo

Depois de duas décadas de promessas não cumpridas sobre ganhos de produtividade e qualidade na aplicação de novas metodologias e tecnologias para o desenvolvimento de software, a indústria e as organizações governamentais - principalmente americanas - estão percebendo que seu problema fundamental é a incapacidade de gerenciar o processo de software. Em várias organizações, os projetos atrasam excessivamente e o orçamento planejado chega a ser dobrado ao seu final. Em tais circunstâncias, a organização não está fornecendo a infra-estrutura e o suporte necessário para manter os projetos de acordo com o planejado e evitando que os problemas ocorram.

Mesmo em organizações que não possuem maturidade no seu processo de software, entretanto, alguns projetos de software específicos produzem excelentes resultados. Quando tais projetos alcançam êxito, é geralmente devido ao esforço heróico de uma equipe dedicada, e não da repetição de um processo de software maduro. O sucesso que repousa somente na disponibilidade de indivíduos específicos não fornece bases para melhoria da produtividade e qualidade de longa duração através de toda a organização. A melhoria contínua pode ocorrer somente através de esforços enfocados e sustentados em direção a construção de uma infra-estrutura de processo com práticas efetivas de engenharia de software e de gerenciamento.

Para obter esta melhoria do processo de software o SEI – Software Engineering Institute da Carnegie Mellon University desenvolveu alguns modelos que enfocam diferentes aspectos do processo de software. O CMM – Capability Maturity Model fornece uma orientação para a melhoria do processo de software de toda a organização, o PSP – Personal Software Process provê, ao engenheiro de software, uma seqüência de exercícios para melhorar o processo de software individual e o TSP – Team Software Process fornece uma orientação para as equipes de projeto de software para a melhoria do seu processo de software. A ISO, através do projeto SPICE – Software Process Improvement and Capability dEtermination, criou a norma 15504 que fornece um modelo de avaliação de processo de software. O modelo SPICE tem como foco a

melhoria do processo a nível organizacional, possuindo muitas práticas de gerenciamento de projeto semelhantes ao CMM. A nível organizacional serão adotados o uso conjunto dos modelos CMM e SPICE.

O objetivo desta dissertação é propor um método que integre estas três dimensões do processo de software - organizacional, individual e de equipe. Este método deve fornecer à organização os passos a serem efetuados quando da implantação de um programa de melhoria de processos e quando cada abordagem deve ser tratada pelo programa. O método deve prever uma seqüência de ciclos de melhoria onde cada ciclo deve seguir o modelo IDEAL – Inicial, Diagnose, Estabelecimento, Ação e Aprendizagem. Este modelo, também desenvolvido pela SEI, permite a adoção das três dimensões de melhoria, ou seja, organizacional, individual e equipe, fornecendo uma seqüência de atividades e tarefas a serem executadas para que a sua implementação e implantação obtenha sucesso.

Para comprovar a viabilidade deste método de melhoria de processo de software será apresentado um estudo de caso, na Siemens Ltda.

melhoria do processo a nível organizacional, possuindo muitas práticas de gerenciamento de projeto semelhantes ao CMM. A nível organizacional serão adotados o uso conjunto dos modelos CMM e SPICE.

O objetivo desta dissertação é propor um método que integre estas três dimensões do processo de software - organizacional, individual e de equipe. Este método deve fornecer à organização os passos a serem efetuados quando da implantação de um programa de melhoria de processos e quando cada abordagem deve ser tratada pelo programa. O método deve prever uma seqüência de ciclos de melhoria onde cada ciclo deve seguir o modelo IDEAL – Inicial, Diagnose, Estabelecimento, Ação e Aprendizagem. Este modelo, também desenvolvido pela SEI, permite a adoção das três dimensões de melhoria, ou seja, organizacional, individual e equipe, fornecendo uma seqüência de atividades e tarefas a serem executadas para que a sua implementação e implantação obtenha sucesso.

Para comprovar a viabilidade deste método de melhoria de processo de software será apresentado um estudo de caso, na Siemens Ltda.

## Abstract

After two decades of unfulfilled promises about productivity and quality gains from applying new software methodologies and technologies, industry and government organizations are realizing that their fundamental problem is the inability to manage the software process. In many organizations, projects are often excessively late and double the planned budget. In such instances, the organization frequently is not providing the infrastructure and support necessary to help projects avoid these problems.

Even in undisciplined organizations, however, some individual software projects produce excellent results. When such projects succeed, it is generally through the heroic efforts of a dedicated team, rather than through repeating the mature software process. Success that rests solely on the availability of specific individuals provides no basis for long-term productivity and quality improvement throughout an organization. Continuous improvement can occur only through focused and sustained effort towards building a process infrastructure of effective software engineering and management practices.

To obtain this software process improvement the SEI – Software Engineering Institute at Carnegie Mellon University developed some models which focus on different aspects of the software process. CMM – Capability Maturity Model provide a guideline to the wide organization software process improvement, PSP – Personal Software Process provide to the software engineer, a sequence of exercises to improve his software process, and TSP – Team Software Process provide a guideline to the software project teams on how to improve their software process. The ISO, for way of the project SPICE – Software Process Improvement and Capability dEtermination, build the standard 15504 which provide a software process assessment model. The SPICE model has as focus the process improvement on the organizational level, having many practices similar to the CMM. On organizational level will be adopted the both CMM and SPICE models.

## Abstract

After two decades of unfulfilled promises about productivity and quality gains from applying new software methodologies and technologies, industry and government organizations are realizing that their fundamental problem is the inability to manage the software process. In many organizations, projects are often excessively late and double the planned budget. In such instances, the organization frequently is not providing the infrastructure and support necessary to help projects avoid these problems.

Even in undisciplined organizations, however, some individual software projects produce excellent results. When such projects succeed, it is generally through the heroic efforts of a dedicated team, rather than through repeating the mature software process. Success that rests solely on the availability of specific individuals provides no basis for long-term productivity and quality improvement throughout an organization. Continuous improvement can occur only through focused and sustained effort towards building a process infrastructure of effective software engineering and management practices.

To obtain this software process improvement the SEI – Software Engineering Institute at Carnegie Mellon University developed some models which focus on different aspects of the software process. CMM – Capability Maturity Model provide a guideline to the wide organization software process improvement, PSP – Personal Software Process provide to the software engineer, a sequence of exercises to improve his software process, and TSP – Team Software Process provide a guideline to the software project teams on how to improve their software process. The ISO, for way of the project SPICE – Software Process Improvement and Capability dEtermination, build the standard 15504 which provide a software process assessment model. The SPICE model has as focus the process improvement on the organizational level, having many practices similar to the CMM. On organizational level will be adopted the both CMM and SPICE models.

The objective of this dissertation is to propose a method that integrate these three aspects, organizational, personal and project team. This method must provide to the organization the steps to be accomplished when of the implanting of a process improvement program and when each aspect must be used into program. Method must foresee a sequence of improvement cycles where each cycle must follow the IDEAL model – Initiating, Diagnosing, Establishing, Acting and Learning. This model, also developed by SEI, allows the adoption of the three improvement aspects – organizational, personal and team - providing a sequence of activities and tasks to be executed to the successful of its implementing and roll out.

To assess the viability of this software process improvement method will be presented a case study, based on Siemens Ltda.

The objective of this dissertation is to propose a method that integrate these three aspects, organizational, personal and project team. This method must provide to the organization the steps to be accomplished when of the implanting of a process improvement program and when each aspect must be used into program. Method must foresee a sequence of improvement cycles where each cycle must follow the IDEAL model – Initiating, Diagnosing, Establishing, Acting and Learning. This model, also developed by SEI, allows the adoption of the three improvement aspects – organizational, personal and team - providing a sequence of activities and tasks to be executed to the successful of its implementing and roll out.

To assess the viability of this software process improvement method will be presented a case study, based on Siemens Ltda.

## 1 Introdução

Na maioria das organizações de software, os processos são geralmente improvisados pelos seus desenvolvedores e seus gerentes durante o curso do projeto. Mesmo que um projeto de software tenha sido especificado, ele não é rigorosamente seguido ou cumprido. Estas organizações são chamadas de imaturas [36]. Elas normalmente são reacionárias e os seus gerentes estão normalmente enfocados na resolução de crises imediatas (melhor conhecido como apagadores de incêndio). Cronogramas e orçamentos são rotineiramente excedidos porque eles não estão baseados em estimativas realistas. Quando prazos de entrega difíceis são impostos, a funcionalidade e qualidade do produto são freqüentemente comprometidas para respeitar o cronograma.

Em uma organização imatura, não há bases objetivas para avaliação da qualidade do produto ou para resolução de problemas no produto ou no processo [20]. Além disso, a qualidade do produto é difícil de prognosticar. As atividades pretendidas para aumentar a qualidade, tais como revisões e testes, são freqüentemente encurtadas ou eliminadas quando o cronograma dos projetos aperta.

A organização não fornece um ambiente estável para desenvolvimento e manutenção de software. Quando uma organização necessita efetuar práticas de gerenciamento, os benefícios de boas práticas de engenharia de software são minadas por planejamentos ineficazes e por sistemas de comprometimento que induzem reações ao contrário.

Durante uma crise, os projetos abandonam os procedimentos planejados e reverterem para codificação e teste. O sucesso depende inteiramente em ter um gerente excepcional e uma equipe de software afinada e eficaz. Ocasionalmente, gerentes de software capazes e vigorosos podem resistir as pressões para tomar atalhos no processo de software; mas quando eles deixam o projeto, sua influência de estabilização sai com eles. Mesmo um forte processo de engenharia não pode superar a instabilidade criada pela ausência das práticas de gerenciamento.

A capacidade do processo de software destas organizações é imprevisível porque o processo de software é constantemente alterado ou modificado quando o trabalho avança. Cronogramas, orçamentos, funcionalidade e qualidade de produto são geralmente imprevisíveis. O desempenho depende das capacidades dos indivíduos e varia com suas habilidades inatas, seu conhecimento e suas motivações. Há poucos processos de software em evidência e o desempenho pode ser previsto apenas por indivíduos ao invés da capacidade organizacional.

A implantação de melhores práticas de engenharia de software e de gerenciamento para o controle dos processos de desenvolvimento e manutenção de software são imprescindíveis para melhorar a capacidade e a maturidade da organização [39].

O termo processo possui significados diferentes para coisas diferentes e para pessoas diferentes [34]. Logo, é importante definir claramente o que se entende como processo no contexto do desenvolvimento e suporte de software.

Segundo o dicionário Webster's, um processo é "um sistema de operações na produção de alguma coisa ... uma série de ações, mudanças ou funções que atingem um fim ou resultado." O IEEE define um processo como "uma seqüência de passos desempenhados para um dado propósito" [1]. Um *processo de software* pode ser definido como "uma série de atividades, métodos, práticas e transformações que as pessoas utilizam para desenvolver e manter software e os produtos associados". A Figura 1 ilustra esta definição [36].

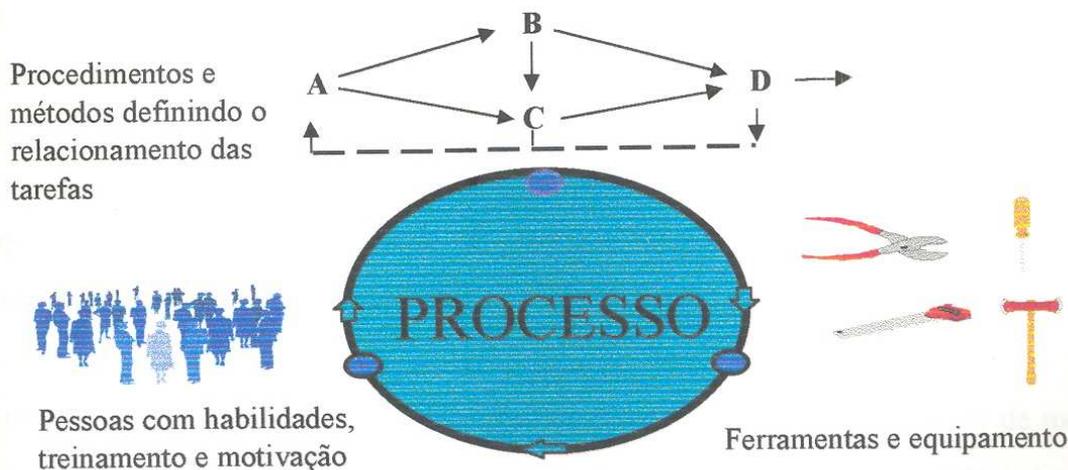


Figura 1 – Definição de processo de software.

A melhoria do processo de software pode ser compreendida como um quebra-cabeça onde cada peça é um atributo que garantirá a evolução do nível de maturidade da organização. Para que o programa de melhoria seja bem sucedido, um acordo com metas realistas e que possam ser medidas deve ser estabelecido. Tais metas devem ser planejadas previamente, deve haver a supervisão contínua e um controle conveniente para se atingí-las, as barreiras – humanas, financeiras e técnicas – devem ser identificadas e removidas, as informações sobre o progresso e os sucessos obtidos pelo programa devem ser continuamente fornecidas, os recursos requisitados devem ser providos e o suporte ao programa deve ser duradouro e visível pela gerência da organização. A Figura 2 ilustra os atributos necessários para que tal programa seja bem sucedido.

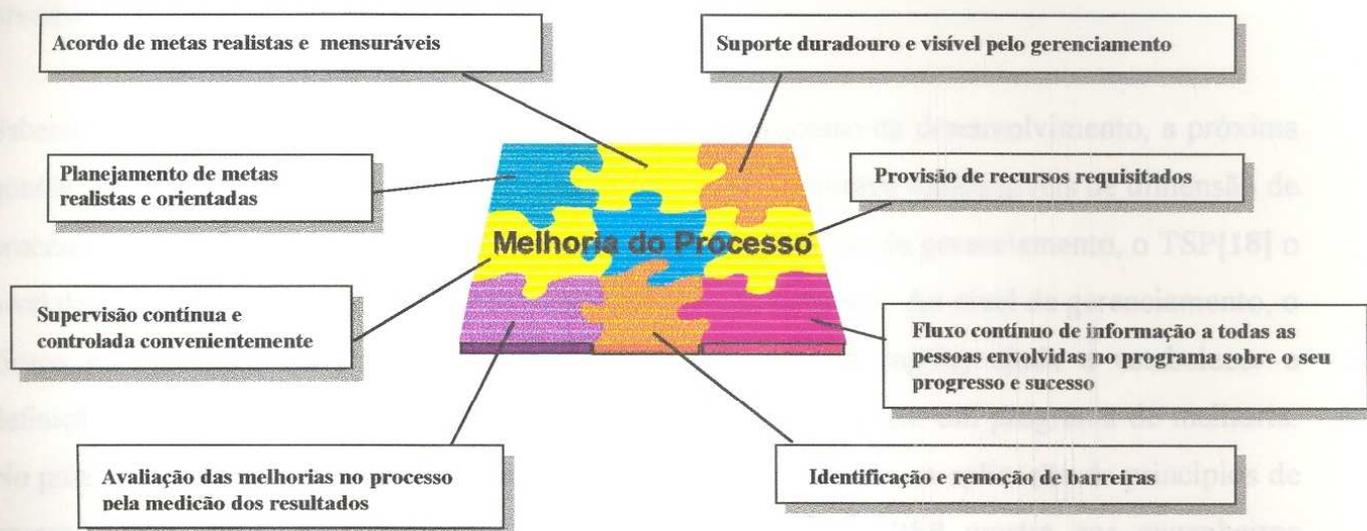


Figura 2 – Atributos para o sucesso de um programa de melhoria de processos.

Um programa de melhoria de processo deve utilizar, para garantir que todos os atributos acima descritos sejam considerados, uma abordagem que sirva de guia para a sua implementação.

A primeira pergunta a ser respondida em qualquer programa de melhoria é “Por que melhorar?” Sem a evidência da importância de um programa de melhoria e de seus benefícios, as organizações não iniciam um programa de melhoria de processos bem sucedido até que elas tenham um comprometimento com as razões de negócio da organização [14].

Um dos motivadores mais eficazes é a pressão do cliente. No caso dos modelos de melhoria de processos, a sua grande utilização está servindo de motivação para que outras organizações também estabeleçam programas de melhoria de processos. Muitas organizações de software

desejam estabelecer e manter uma posição competitiva forte em relação a qualidade dos fornecedores de sistemas.

Estando a questão “Por quê?” respondida, a próxima questão imediata é “O que deve ser feito para adquirir uma capacidade de software maior?” [17]. Esta é a principal questão abrangida pelos modelos de melhoria de processos. A estrutura de níveis de maturidade e um sistema de avaliação relacionando-os ajuda as organizações a entenderem suas capacidades. Através da comparação de suas práticas atuais no seu processo de desenvolvimento com as práticas apresentadas pelos modelos CMM[36] e SPICE[43] pode-se verificar quais atividades elas precisam adicionar ou melhorar.

Sabendo a organização o que fazer para melhorar o seu processo de desenvolvimento, a próxima questão é “Como fazer estas melhorias?” [18]. Esta questão envolve vários níveis de dimensão de processo. O CMM e o SPICE abrangem o nível organizacional ou de gerenciamento, o TSP[18] o nível de equipes de projeto e o PSP[15] os engenheiros de software. Ao nível de gerenciamento, o Grupo de Processos de Engenharia de Software (SEPG, em inglês) ajuda a estabelecer a definição, o controle e a melhoria das tarefas necessárias para lançar um programa de melhoria. No próximo nível, o TSP guia as equipes de projeto e seus gerentes na aplicação de princípios de processo para atingir os objetivos do projeto. Finalmente, o PSP mostra aos engenheiros (analistas, técnicos, etc.) que desenvolvem produtos o caminho de como melhorar os seus processos pessoais de desenvolvimento de produtos de software. Para a organização atingir seus objetivos, os engenheiros e gerentes de todos estes níveis devem desempenhar suas tarefas com habilidade e competência, para isto os modelos de melhoria de processos fornecem melhores práticas para que estas características possam ser incrementadas. A Figura 3 apresenta o relacionamento entre estas camadas de gerenciamento organizacional, equipe de projeto e engenheiro de software[14].

A capacidade organizacional é construída de dentro para fora enquanto que o suporte a projeto e a motivação da engenharia são fornecidos de fora para dentro. Um funcionamento eficaz nos três níveis é requerido para que a organização de software seja plenamente produtiva. Esta dissertação de mestrado tem por objetivo propor um método de melhoria que integre as três dimensões do

processo de software. Tal método deve ser implantado em ciclos de melhoria de processo, onde em cada ciclo de melhoria serão implementadas as atividades que aumentarão a capacidade organizacional, das equipes de projeto e dos engenheiros de software da organização. Um exemplo de método integrado unindo as áreas de engenharia de software e engenharia de sistemas através da utilização dos modelos CMM - Software e CMM - Engenharia de Sistemas, pode ser verificado na literatura técnica [23].

O escopo deste método de melhoria integrado deve levar em conta as características da organização que servirá de estudo de caso, que são: alto conteúdo tecnológico envolvido no processo de desenvolvimento, produtos que envolvem o desenvolvimento de sistemas, produtos desenvolvidos para três parceiros distintos com processos de desenvolvimento estabelecidos e não subcontratação de terceiros. Estas características serão descritas e levadas em consideração durante o estabelecimento das atividades que deverão ser consideradas ao longo da descrição do método integrado.

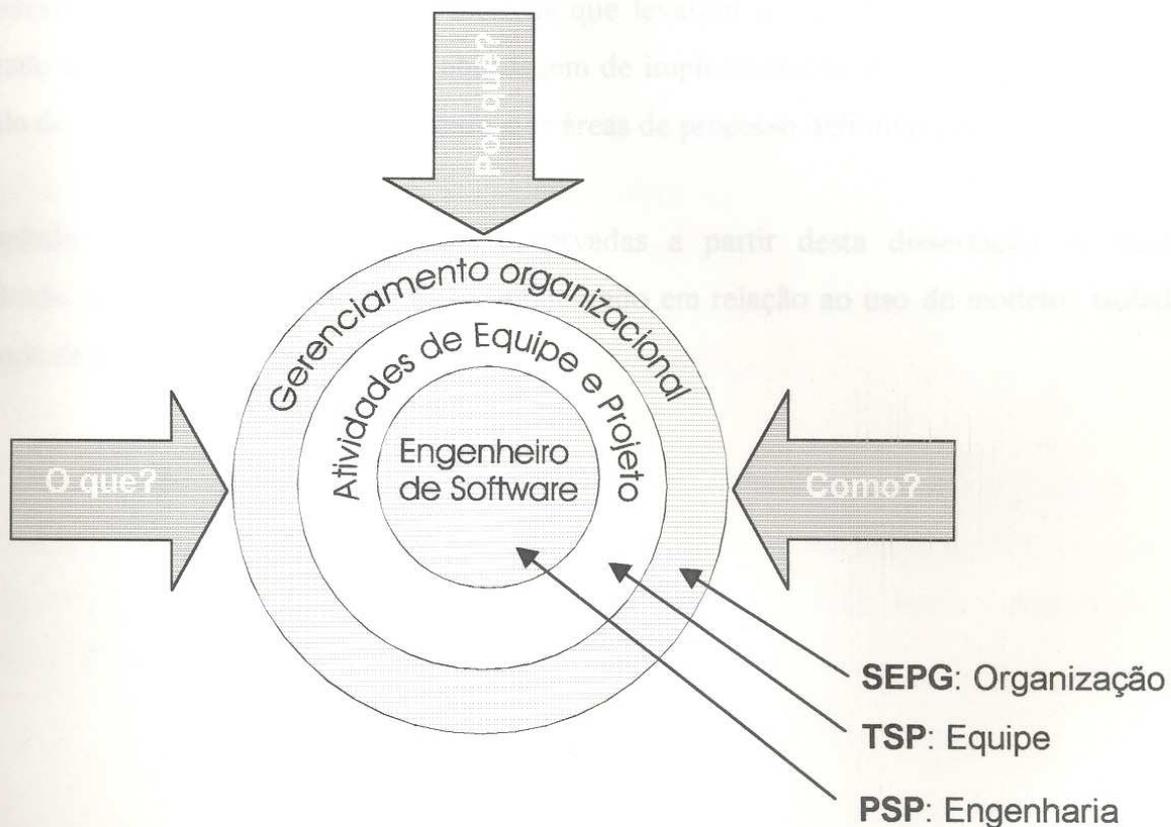


Figura 3 – As três dimensões da melhoria de processos.

O capítulo 2 desta dissertação apresenta uma revisão bibliográfica da abordagem desenvolvida pelo SEI para a adoção de um programa de melhoria de processos de software, o modelo IDEAL.

O capítulo 3 deste trabalho revisa as principais características dos modelos de melhoria de processos de software que serão utilizados pelo método integrado. Os modelos de maturidade organizacional CMM e SPICE, o modelo de melhoria do processo individual - PSP e o modelo de melhoria de equipes de projeto - TSP. Não serão abordados outros métodos integrados disponíveis na literatura técnica, pois estes se utilizam dos mesmos modelos básicos (por exemplo, CMM) que os apresentados nesta dissertação.

O capítulo 4 apresenta o método integrado de melhoria de processos, estabelecendo as atividades a serem introduzidas em cada um dos quatro ciclos de melhoria presentes no método.

O capítulo 5 apresenta o estudo de caso da implantação do método integrado em uma organização da Siemens Ltda. Serão descritos os motivos que levaram a organização a adotar um método integrado de melhoria de processo, a abordagem de implementação do programa de melhoria, o método de avaliação de processo utilizado e as áreas de processo definidas pela organização.

O capítulo 6 apresenta as conclusões observadas a partir desta dissertação de mestrado, abordando as vantagens do uso do método integrado em relação ao uso de modelos isolados de melhoria de processos de software.

## 2 O Modelo IDEAL

O modelo IDEAL [29] [10] descreve um guia de como implementar um programa de melhoria de processo de software (SPI – Software Process Improvement). Este modelo pode ser utilizado como um guia de desenvolvimento de longo prazo e um plano integrado de iniciação e gerenciamento de um programa de SPI. O propósito deste guia é fornecer aos gerentes de melhoria de processo uma descrição genérica de uma seqüência de passos recomendados para se efetuar um SPI. No contexto do método integrado de melhoria de processos aqui proposto, o modelo IDEAL serve como a abordagem a ser adotada pela organização para integrar os diferentes modelos de referência – CMM, SPICE, PSP e TSP – e os passos necessários para que as práticas e as atividades de engenharia de software sejam introduzidas e/ou institucionalizadas de maneira eficiente e eficaz.

O modelo, mostrado na Figura 4, apresenta 5 fases de uma iniciativa de SPI, que fornecem um círculo contínuo através dos passos necessários para o SPI[10]. É importante notar que o tempo para se completar um ciclo através do modelo IDEAL varia de organização para organização. Algumas organizações acharão, dependendo dos recursos que elas são capazes de comprometer com o programa de SPI, que várias atividades podem ser executadas de maneira paralela. Há também instâncias de algumas partes da organização executando atividades de uma fase do modelo enquanto outras estão executando atividades de uma fase diferente. Na prática os limites entre as fases do modelo IDEAL não são tão claramente definidas como mostrado no modelo.

É também importante notar que a infra estrutura preparada para se executar o programa de SPI desempenhará um papel significativo no sucesso ou fracasso da iniciativa de SPI. O valor que a infra estrutura traz para o programa de SPI, o entendimento de seus papéis e responsabilidades não podem ser subestimados.

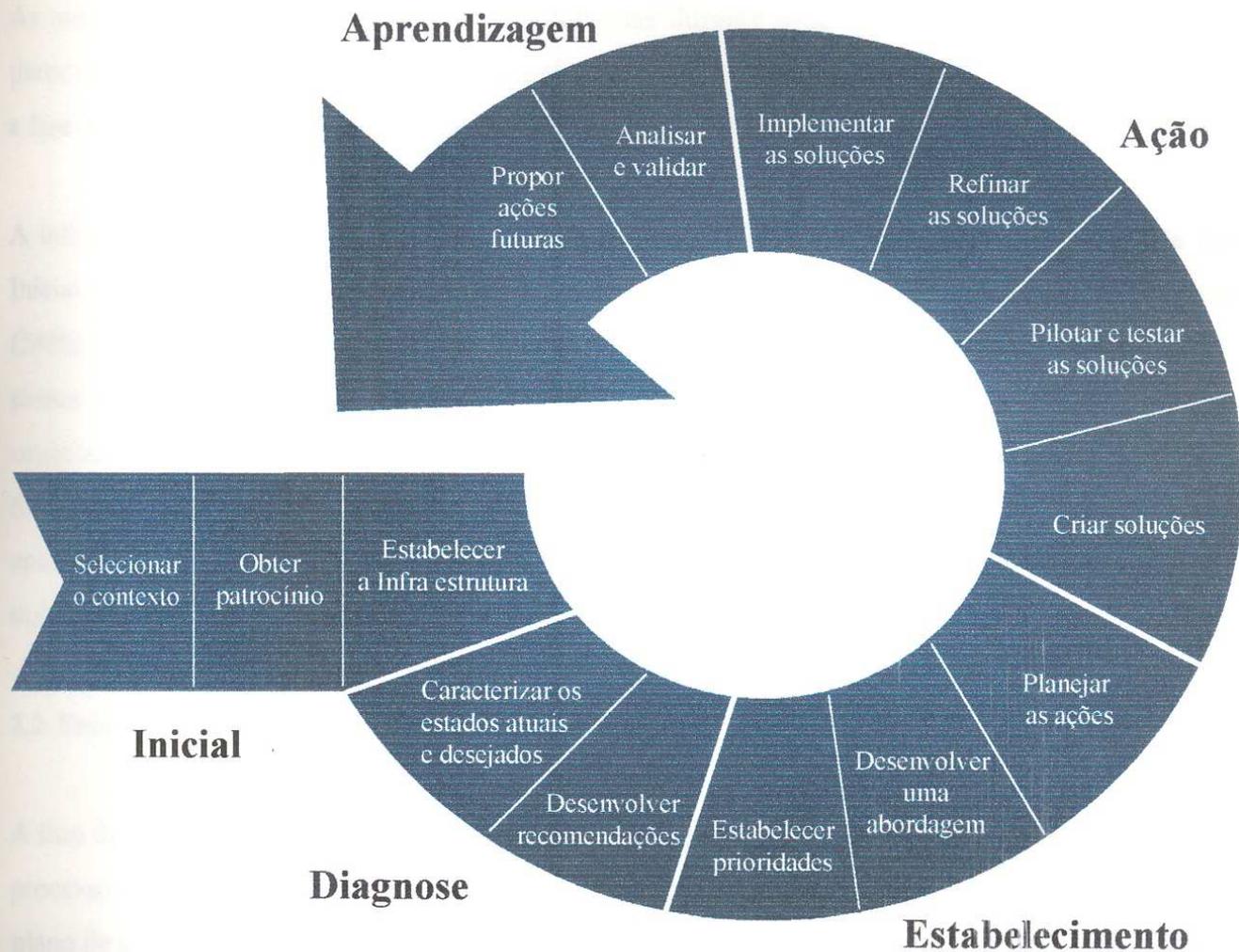


Figura 4 – O modelo IDEAL.

## 2.1 Fase 1 - Inicial

A fase 1 do modelo IDEAL é o ponto de partida. Aqui é onde a infra estrutura inicial de melhoria é estabelecida, os papéis e responsabilidades pela infra estrutura são inicialmente definidos e os recursos iniciais são designados. Nesta fase, um plano de SPI é criado para guiar a organização para conclusão das fases Inicial, Diagnose e Estabelecimento. A aprovação da iniciativa de SPI é obtida com o comprometimento de recursos futuros para o trabalho de melhoria.

As metas gerais do programa de SPI são definidas durante esta fase 1. Elas são estabelecidas a partir das necessidades de negócio da organização e serão refinadas e tornadas específicas durante a fase de Estabelecimento do modelo IDEAL.

A infra estrutura inicial para dar suporte e facilitar o SPI serão estabelecidas durante esta fase Inicial. Dois componentes chave são tipicamente estabelecidos, um *Management Steering Group* (MSG) e um *Software Engineering Process Group* (SEPG). Também durante a fase Inicial, planos são feitos para comunicar o início da iniciativa de SPI e é sugerido que a avaliação organizacional seja executada para determinar a presteza da organização para uma iniciativa de SPI. Por exemplo, nesta fase são escolhidos os membros dos grupos MSG e SEPG e que em 2 anos deve ser atingido o nível 2 de maturidade de acordo com o CMM. Estas são definições típicas a serem efetuadas na fase Inicial.

## 2.2 Fase 2 - Diagnose

A fase de Diagnose do modelo IDEAL inicia a organização no caminho da melhoria contínua do processo de desenvolvimento. Esta fase prepara as bases para as fases posteriores. Nesta fase, o plano de ação de SPI é iniciado em concordância com a visão da organização, o plano estratégico de negócios, as lições aprendidas de esforços de melhoria passados, exemplos de negócios chave enfrentados pela organização e metas de longo prazo. Atividades de avaliação são executadas para estabelecer uma linha básica do estado atual da organização. Os resultados e recomendações das avaliações e qualquer outra atividade semelhante serão reconciliadas com os esforços de melhoria existentes e/ou planejadas e incluídas dentro do plano de ação de SPI.

O item 2.2.1 descreve as características necessárias para que um método de avaliação seja compatível com o modelo CMM. Como o modelo CMM é o modelo de referência base do método integrado aqui proposto, a avaliação da organização que adote este método integrado deve também adotar um método de avaliação compatível com o proposto pelo CAF.

O item 2.2.2 apresenta o método de avaliação criado pela SEI para a avaliação do processo de software interno das organizações. Tal método serve como referência para a organização avaliar os recursos necessários para efetuar uma avaliação do seu processo de software.

O item 2.2.3 apresenta as características do método de avaliação utilizado pelo modelo SPICE, que difere do utilizado pelo CMM. No caso do modelo SPICE cada processo é classificado de acordo com o nível de capacidade adquirido, diferentemente do modelo CMM, em que o nível de capacidade é obtido devido aos processos desempenhados [9].

Para o nosso método integrado será adotado um questionário integrado de maturidade, descrito no item 2.2.4 que incorpora ao questionário de maturidade do CMM as práticas obtidas dos modelos SPICE, PSP e TSP. O método integrado de melhoria estabelece como método de avaliação o conceito utilizado pelo CMM, onde o nível de capacidade da organização é dado pelos processos implementados em cada ciclo de melhoria e verificados através do questionário integrado.

### 2.2.1 CMM Appraisal Framework (CAF)

O CAF Versão 1.0 [28] é um *framework* para desenvolvimento, definição e utilização de métodos de avaliação que utilizam o *Capability Maturity Model(CMM) for Software Version 1.1* como modelo de referência.

O CAF inclui uma arquitetura para um método de avaliação CAF genérico e uma descrição dos requisitos de um método de avaliação CAF. Quando um método de avaliação atende todos os requisitos do CAF, ele é dito ser plenamente obediente ao CAF.

Há três famílias de avaliações de processos de software:

- Avaliações cujo propósito primário é fornecer informação útil aos clientes de um fornecedor de software selecionado;
- Avaliações cujo propósito primário é fornecer informação para orientar nos esforços de melhoria de processo de software interno de um determinado fornecedor;

- Avaliações cujo propósito primário é fornecer informação para melhoria do processo conjunto cliente/fornecedor e/ou esforços de gerenciamento de risco.

Há dois exemplos de métodos de avaliação baseados no CAF e desenvolvidos pelo SEI/CMU: SCE - *Software Capability Evaluation* [4] que avalia os processos de software de fornecedores e o CBA IPI – *CMM-Based Appraisal Internal Process Improvement* – que avalia os processos de software internos da organização [7].

O CMM fornece uma estrutura para julgar a maturidade do processo de software de uma organização. O CMM e o CAF juntos descrevem “o que” deve ser executado pelos métodos de avaliação obedientes ao CAF. Os métodos de avaliação detalham “como” transformar os dados do processo de software da organização em informações de valor para atender as necessidades de negócio da organização.

A Figura 5 ilustra o relacionamento entre os componentes de uma avaliação baseada no CMM. O círculo central, o CMM, é o modelo utilizado como um *framework* para a avaliação do processo de software da organização. O CAF é o *framework* para execução de uma avaliação utilizando este modelo. O próximo disco contém os componentes comuns – os artefatos, ferramentas e técnicas que podem ser compartilhadas através dos métodos específicos. A camada externa ilustra os métodos de avaliação que são obedientes ao CAF.

Os objetivos do CAF são:

- Definir os requisitos para os métodos obedientes ao CAF;
- Fornecer orientação para comparação entre diferentes métodos;
- Definir os componentes de um método obediente ao CAF e os relacionamentos entre eles;
- O método obediente ao CAF deve tornar pública uma definição explícita de classificação do processo e seus pré-requisitos.

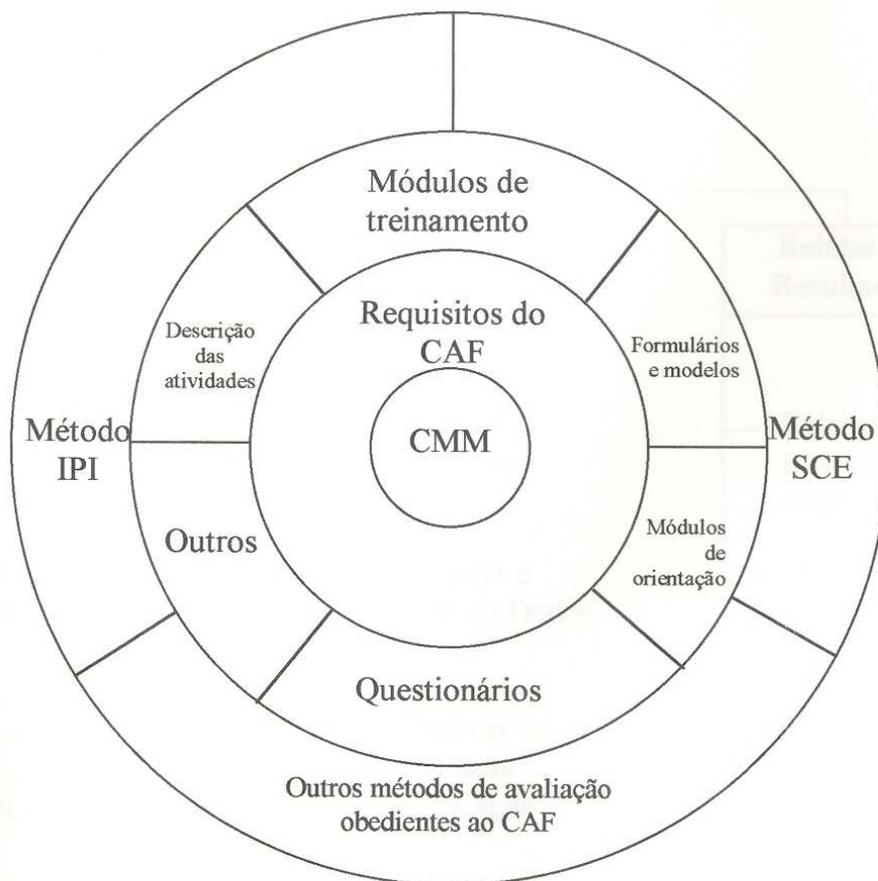


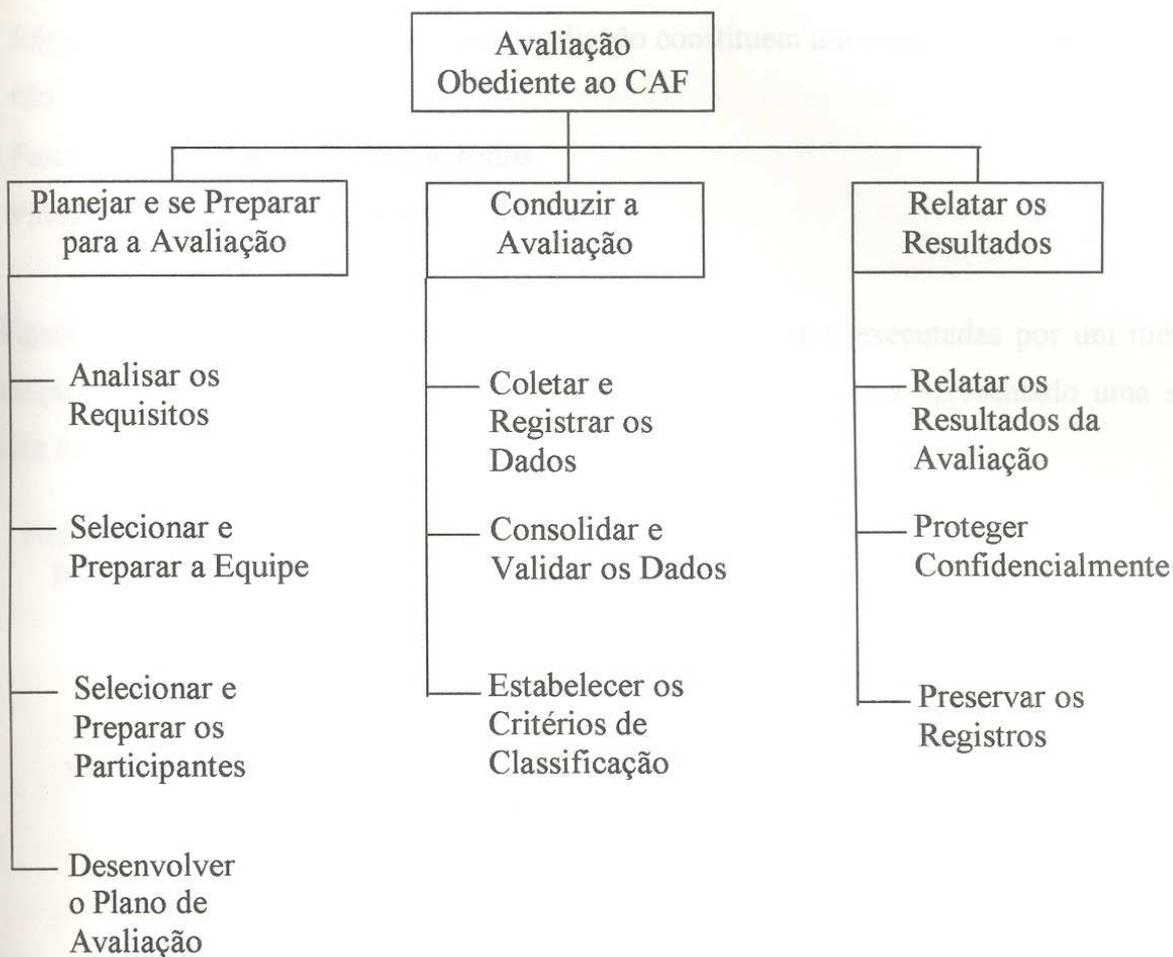
Figura 5 – Diagrama conceitual de uma avaliação baseada no CMM.

### Atividades de um Método de Avaliação

A Figura 6 apresenta as atividades primárias que tornam um método de avaliação obediente ao CAF. Estas atividades são divididas em três fases (algumas vezes sobrepostas):

- Planejar e se preparar para a avaliação;
- Conduzir a avaliação;
- Relatar os resultados.

O planejamento e a preparação para a avaliação incluem analisar os requisitos da avaliação e desenvolver um plano de avaliação para tratar aqueles requisitos e, também a seleção e a preparação dos participantes e da equipe de avaliação.



**Figura 6** – Atividades primárias de um método de avaliação obediente ao CAF.

A condução de uma avaliação inclui a coleta e o registro dos dados da avaliação, consolidando-os dentro de um conjunto gerenciável de observações, validando tais observações e fazendo uma classificação das observações baseado em critérios definidos.

O relatório dos resultados inclui a documentação e a apresentação dos produtos da avaliação.

Os dados básicos obtidos durante a avaliação devem ser transformados em produtos detalhados como relatório da avaliação. As maiores transformações incluem:

- O registro dos dados na forma de notas individuais;

- A consolidação destas notas em um conjunto gerenciável de observações;
- Julgar se as observações da equipe de avaliação constituem um conjunto válido de achados ou não;
- Fazer uma classificação nestes achados;
- Produzir relatórios da avaliação.

A Figura 7 ilustra um diagrama de fluxo de atividades a serem executadas por um método de avaliação baseado no CMM. No estudo de caso no capítulo 5 será apresentado uma situação prática de avaliação seguindo este fluxo de atividades.

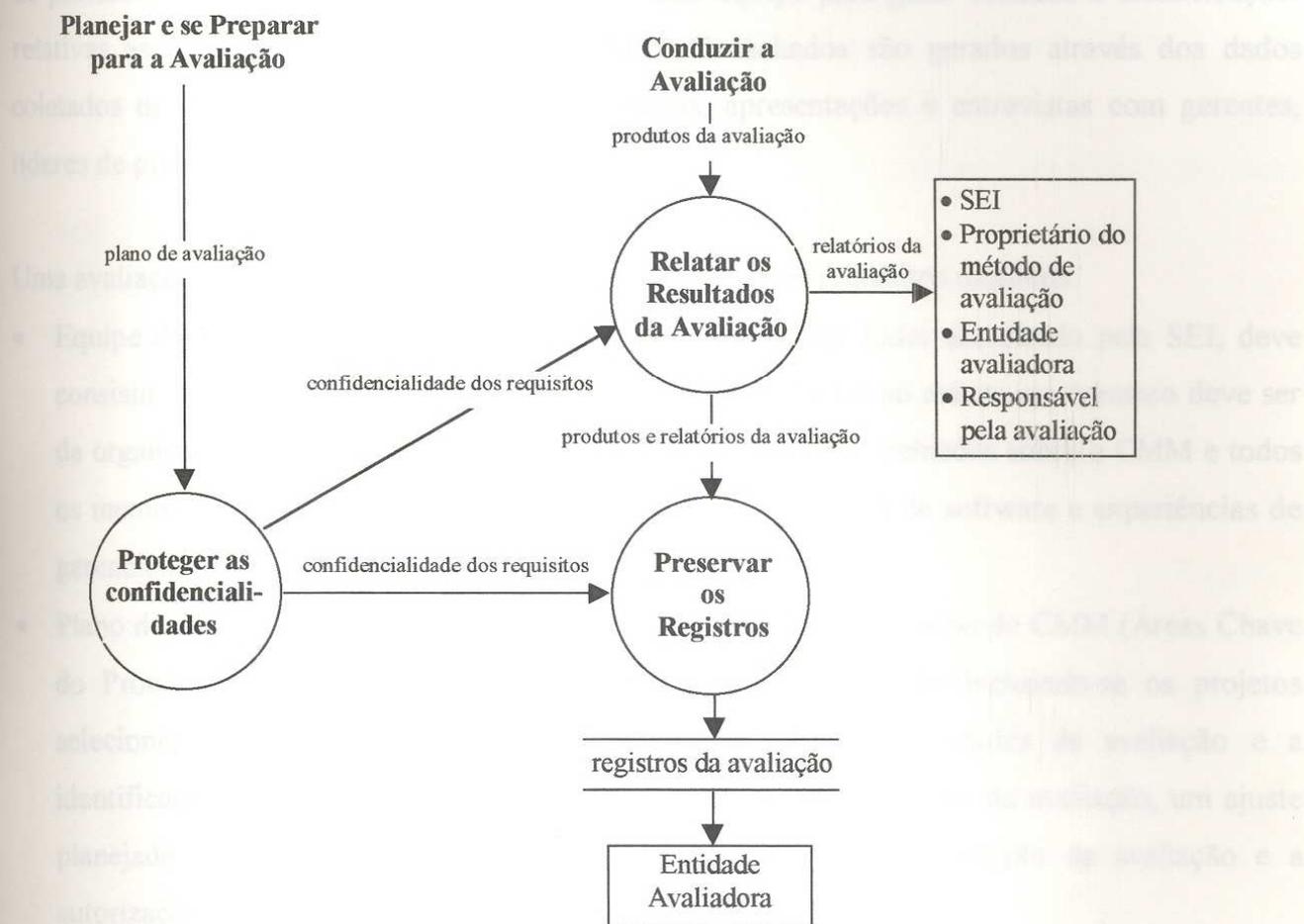


Figura 7 – Fluxo de atividades de um método de avaliação.

### 2.2.2 CMM Based Appraisal Internal Process Improvement (CBA IPI)

O método CBA IPI [7] é uma ferramenta de diagnose que habilita uma organização a ganhar maior discernimento sobre sua capacidade de desenvolvimento de software pela identificação dos pontos fortes e fracos dos seus processos atuais, relacionar estes pontos fortes e fracos ao CMM, priorizar os planos de melhoria e focar nas melhorias que são mais benéficas, dado o seu nível de maturidade atual e metas de negócio.

O método efetua a avaliação da capacidade do processo de software da organização por um grupo de profissionais treinados que trabalham como uma equipe para gerar achados e classificações relativas as áreas chave do processo do CMM. Os achados são gerados através dos dados coletados de questionários, revisão de documentos, apresentações e entrevistas com gerentes, líderes de projeto e desenvolvedores.

Uma avaliação segundo o CBA IPI deve possuir os seguintes requisitos mínimos:

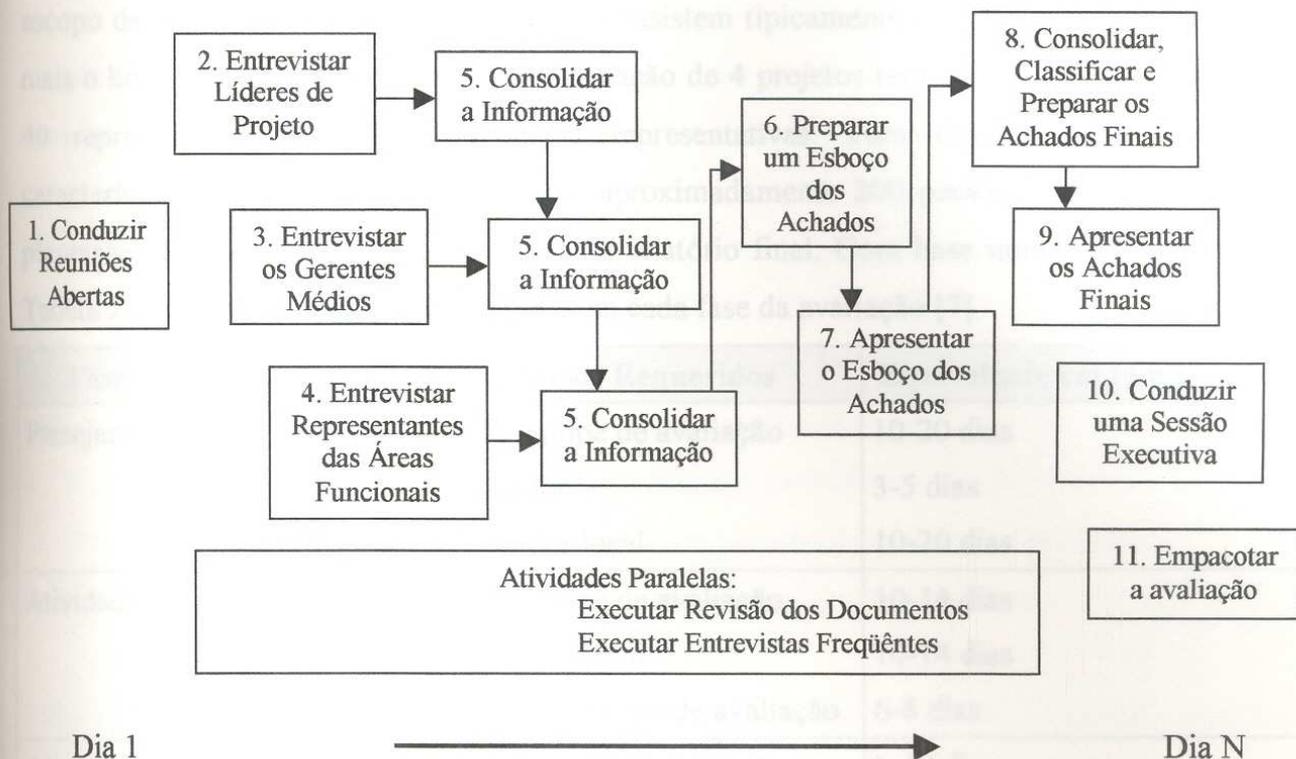
- Equipe de avaliação: Deve ser liderada por um Avaliador Líder autorizado pela SEI, deve consistir de um mínimo de 4 e um máximo de 10 membros e ao menos um membro deve ser da organização sendo avaliada, todos os membros devem ser treinados sobre o CMM e todos os membros devem selecionar as linhas básicas de engenharia de software e experiências de gerenciamento.
- Plano de Avaliação: Deve conter as metas para a avaliação, o escopo do CMM (Áreas Chave do Processo a serem examinadas) e o escopo da organização incluindo-se os projetos selecionados e os participantes, um cronograma para as atividades de avaliação e a identificação dos recursos para executar as atividades, os resultados da avaliação, um ajuste planejado do método de avaliação, riscos associados com a execução da avaliação e a autorização de um responsável para que a avaliação seja conduzida.
- Coleta de Dados: Os dados por instrumento (respostas coletadas pelo questionário de maturidade) obtidos dos líderes de projeto dos projetos selecionados, dados de entrevistas individuais com os líderes de projeto, dados de entrevistas em grupo com representantes (desenvolvedores) de áreas funcionais e com gerentes médios, dados de documentos de cada

uma das metas das Áreas Chave de Processo avaliadas e dados obtidos da revisão pelos participantes da avaliação da apresentação do esboço dos achados encontrados.

- Validação dos Dados: Devem ser corroborados por observações baseadas em ao menos duas fontes independentes, obtidas durante ao menos duas sessões de obtenção de dados diferentes e confirmadas por ao menos uma fonte de dados que reflete o trabalho atualmente sendo feito.
- Classificação: Devem ser seguidos critérios para a classificação da maturidade do processo da organização contra o CMM.
- Relatório dos Resultados da Avaliação: O relatório final deve possuir os pontos fortes e fracos de cada KPA dentro do escopo da avaliação, também o perfil da KPA que indica se as KPAs estão satisfeitas, não satisfeitas, não classificadas ou não aplicável.

O CBA IPI consiste de três fases:

- A primeira fase inclui as atividades necessárias para planejar e preparar para a avaliação, estas atividades devem ser executadas seqüencialmente e são:
  - Identificar o escopo da avaliação;
  - Desenvolver um plano de avaliação;
  - Preparar e treinar a equipe de avaliação;
  - Breve avaliação dos participantes;
  - Administrar os questionários;
  - Examinar as respostas aos questionários;
  - Conduzir a revisão do documento inicial.
- A segunda fase consiste de atividades *in loco* para a condução da avaliação, incluindo técnicas para obtenção, organização e consolidação dos dados. A Figura 8 ilustra estas atividades e a sua cronologia de execução.



**Figura 8** – Cronologia das Atividades *in loco*.

- A fase final consiste em relatar os resultados. As atividades desta fase estão ilustradas na Figura 8 nas atividades 9, 10 e 11.

A Tabela 1 apresenta um período de tempo típico para a avaliação segundo o método CBA IPI [7].

Período	Atividade
Meses 1-2	Planejamento da avaliação e treinamento da equipe
Mês 3	Avaliação <i>in loco</i>
Mês 4	Relatório Final liberado e breves recomendações
Mês 5	Desenvolvimento de um plano de ação
Meses 6-24	Implementação do plano de ação
Meses 18-30	Reavaliação

**Tabela 1** – Período de tempo típico de uma avaliação através do CBA IPI.

Os recursos requeridos para uma organização executar um CBA IPI varia de acordo com o escopo da avaliação. Porém, as avaliações consistem tipicamente de uma equipe de 8 membros mais o líder da equipe de avaliação, investigação de 4 projetos representativos e entrevistas com 40 representantes de áreas funcionais representativas. Para uma avaliação com estas características o método CBA IPI requer aproximadamente 200 pessoas-dia, iniciando com o planejamento da avaliação até a entrega do relatório final. Com base numa avaliação típica, a Tabela 2 apresenta os recursos requeridos em cada fase da avaliação [7].

<b>Fase da Avaliação</b>	<b>Recursos Requeridos</b>	<b>Equivalente em tempo integral</b>
Planejamento	Líder da equipe de avaliação	10-20 dias
	Patrocinador	3-5 dias
	Coordenador local	10-20 dias
Atividades Pré <i>in loco</i>	Líder da equipe de avaliação	10-14 dias
	Coordenador local	10-14 dias
	Membros da equipe de avaliação	6-8 dias
Atividades <i>in loco</i>	Líder da equipe de avaliação	6-10 dias
	Patrocinador	4-5 horas
	Coordenador local	6-10 dias
	Membros da equipe de avaliação	5-10 dias cada
	Outros participantes da avaliação	1.5 dias cada
Atividade Pós avaliação	Líder da equipe de avaliação	4-8 dias
	Patrocinador	1 dia
	Coordenador local	2 dias
	Membros da equipe de avaliação	1-2 dias cada
Totais	Líder da equipe de avaliação	30-52 dias
	Patrocinador	8-11 dias
	Coordenador local	28-46 dias
	Membros da equipe de avaliação	12-20 dias cada
	Outros participantes da avaliação	1.5 dias cada

**Tabela 2** – Recursos requeridos segundo o método CBA IPI.

### 2.2.3 SPICE Assessment

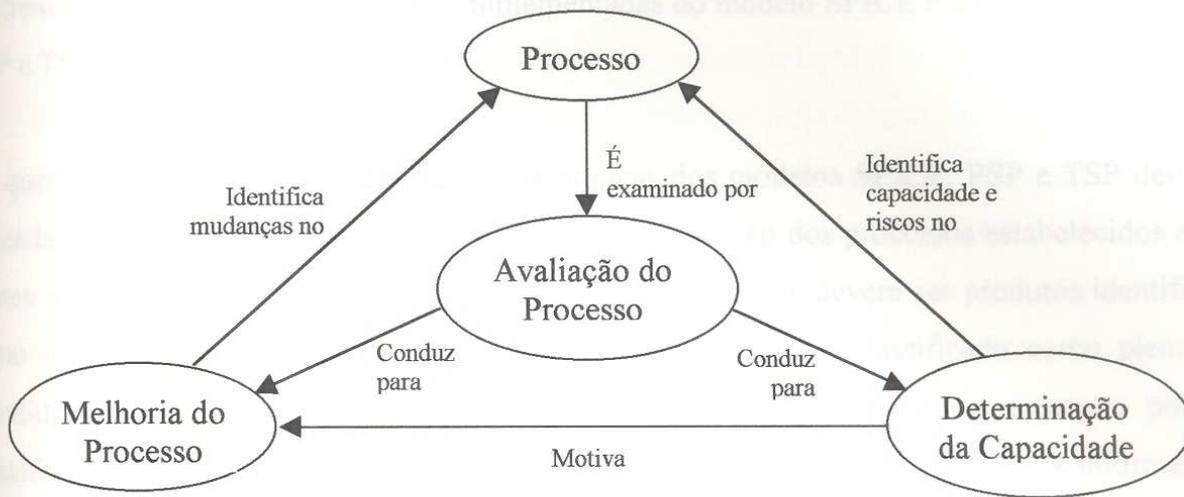
O método de avaliação do modelo SPICE – futura norma ISO15504 [43] fornece um *framework* para a avaliação dos processos de software. Este *framework* pode ser utilizado pelas organizações envolvidas no planejamento, gerenciamento, monitoração, controle e melhoria da aquisição, fornecimento, desenvolvimento, operação, evolução e suporte de software. O *framework* da avaliação do processo:

- Encoraja a auto avaliação;
- Leva em conta o contexto no qual opera os processos avaliados;
- Produz uma série de classificações de processo (um perfil de processo);
- Através das práticas genéricas, avaliam o gerenciamento dos processos avaliados;
- É apropriado a todos os domínios e tamanhos de aplicação da organização.

Dentro do contexto da melhoria de processo, a avaliação baseada no SPICE fornece o significado da caracterização das práticas atuais dentro de uma unidade organizacional em termos da capacidade dos processos selecionados. A análise dos resultados sob a luz das necessidades de negócio da organização identifica os pontos fortes, as fraquezas e os riscos inerentes nos processos. Isto fornece a habilidade para se determinar se os processos são eficazes para se atingir as metas e identificar causas significativas de baixa qualidade ou se há superação de tempo e custo. Estas informações fornecem direções para se priorizar a melhoria dos processos.

A determinação da capacidade do processo é relacionada com a análise da capacidade proposta dos processos selecionados contra um perfil de capacidade do processo alvo a fim de identificar os riscos envolvidos no empreendimento de um projeto utilizando os processos selecionados.

A Figura 9 ilustra a avaliação do processo de software segundo a norma SPICE.



**Figura 9** – Avaliação do processo de software pelo SPICE.

O contexto da avaliação do processo segundo a norma SPICE é constituída de requisitos de entrada, instrumentos de suporte, um modelo de referência e a geração de uma saída. A entrada formal nos processos de avaliação propriamente dita ocorrem com a compilação da entrada da avaliação que definem o seu propósito, escopo, restrições, responsabilidades, os processos estendidos (aqueles que são extensões dos processos definidos pelo modelo de processo do SPICE) e informações adicionais a serem coletadas. A avaliação é orientada pelo modelo de processo definida no SPICE. Este modelo bidimensional consiste de uma série de práticas básicas e genéricas. As práticas genéricas se aplicam através de todos os processos. As práticas genéricas são agrupadas em características comuns e níveis de capacidade que podem ser utilizadas para determinar quão bem o processo é gerenciado. A saída da avaliação inclui uma série de classificações do nível de capacidade do processo para cada instância de processo avaliada. Uma avaliação é suportada por um instrumento de avaliação, ou série de instrumentos, que funcionam como indicadores de processo e indicadores de gerenciamento de processo.

#### 2.2.4 Questionário de maturidade integrado

Para se avaliar a realização dos processos em cada ciclo de maturidade deste método integrado, deve-se estabelecer um método integrado de avaliação. Este método integrado de avaliação é baseado no questionário de maturidade do modelo CMM [45]. Em cada ciclo de melhoria de

processo serão avaliadas as áreas chave de processo implementadas e adicionadas questões que avaliem as práticas genéricas a serem implementadas do modelo SPICE e as práticas dos modelos PSP e TSP.

As questões a serem adicionadas quanto as práticas dos modelos SPICE, PSP e TSP devem ser baseadas na verificação de evidências geradas pela execução dos processos estabelecidos a partir destes modelos. A partir da constatação destas evidências, que devem ser produtos identificáveis como documentos, implementações, etc., o processo pode ser classificado como plenamente atendido. Caso não hajam estas evidências de forma identificável, o processo pode ser classificado como parcialmente atendido ou não atendido. Caso o processo seja informalmente realizado ele é classificado como parcialmente atendido. Se o processo não é praticado pela organização ele é classificado como não atendido.

As áreas de processo ou práticas que não foram implantadas em um determinado ciclo de melhoria e que não pertencem a este ciclo não necessitam ser avaliadas. Caso a organização deseje avaliar os processos de ciclos superiores, a sua não realização pela organização deve ser classificada como não pertinente, indicando ao grupo de melhoria de processos SEPG que o processo não pertence aquele ciclo de melhoria e portanto pode ser implementado posteriormente.

O objetivo do questionário de maturidade integrado é estabelecer uma referência cruzada entre os objetivos do ciclo de melhoria de processos e os resultados obtidos ao final de cada ciclo. Caso alguma prática não tenha sido plenamente implantada a organização deve avaliar quais foram os seus motivos e priorizar a sua implantação antes de prosseguir para o próximo ciclo.

O questionário de maturidade integrado deve ser aplicado na fase de aprendizagem de cada ciclo de melhoria. Porém, muitas das questões serão semelhantes às efetuadas na fase de diagnose no ciclo seguinte. Entretanto esta avaliação se caracteriza como uma auto avaliação informal da organização e servirá de preparação para a avaliação formal que será efetuada na fase de diagnose do ciclo seguinte e que atestará o aumento do nível de maturidade da organização segundo o método integrado.

### 2.3 Fase 3 - Estabelecimento

Após a avaliação do processo de software atual da organização na fase de Diagnose, na fase de Estabelecimento, os resultados que a organização tenha decidido encaminhar com suas atividades de melhoria são priorizadas e as estratégias para a implementação das soluções também são desenvolvidas. O esboço do plano de ação de SPI será completado de acordo com a visão da organização, do plano de negócio estratégico, das lições aprendidas de esforços de melhoria passados, de exemplos de negócios chave encarados pela organização e de metas de longo prazo.

Durante a fase de Estabelecimento, metas mensuráveis são desenvolvidas a partir das metas gerais que foram definidas na fase 1. Estas metas mensuráveis serão incluídas na versão final do plano de ação estratégico de SPI.

Métricas necessárias para monitorar o progresso também são definidas, recursos são comprometidos e treinamento é fornecido para os Technical Working Groups (TWGs). Por exemplo, o tempo e o custo dos projetos, o histórico de linha de código dos projetos e a produtividade dos desenvolvedores, além da disponibilidade dos gerentes de projeto para melhorar suas estimativas são obtidos e estabelecidos. O plano de ação desenvolvido guiará a atividade de SPI pois ela priorizou os achados e as recomendações da fase de Diagnose. Também durante esta fase, modelos do plano de ação tático são criados e tornados disponíveis para os TWGs completarem e seguirem.

Nesta fase as pessoas devem entender por que a organização deve gastar tempo e esforço com um programa de melhoria de processo de software. Para que isto seja entendido deve ser criada uma lista com as motivações em termos da diferença entre o estado atual e o estado desejado. Tais motivações devem responder as seguintes questões:

- Por que mudar?
- O que há de errado com o *status quo*?
- Por que eu devo me preocupar?
- Quando eu serei afetado(imediatamente ou algumas vezes no futuro)?

Nesta fase a organização deve reconciliar o esforço de melhoria planejado com os achados e as recomendações da avaliação efetuada na fase anterior. O SEPG deve construir uma matriz relacionando as recomendações da avaliação com as atividades de melhoria planejadas.

As metas gerais de melhoria de processo devem ser transformadas em medidas específicas mensuráveis que serão implementadas na fase seguinte. Por exemplo, uma meta geral poderia ser possuir projetos de software mais previsíveis em termos de custo e cronograma. A avaliação verificou que 80% dos projetos atuais excedem seu custo original e as estimativas de cronograma por mais de 25%. A meta transformada poderia ser melhorar aquela medida tal que 80% de todos os projetos devem se completar dentro de 10% da estimativa original dentro de 2 anos.

Para implementar estas medidas devem ser criados os TWGs – Technical Working Groups que serão compostos pelos engenheiros de software da organização escolhidos pelo SEPG. Outra atividade a ser executada é finalizar a definição das funções e das responsabilidades para o SEPG, o MSG e qualquer outro grupo de gerenciamento e coordenação do SPI. Deve-se definir, também as funções e as responsabilidades para os TWGs nos termos de suas responsabilidades, autoridade, requisitos, etc.

#### **2.4 Fase 4 - Ação**

Na fase de Ação do modelo IDEAL, as soluções endereçadas às áreas de melhoria descobertas durante a fase de Diagnose são criadas, testadas e distribuídas através da organização. Planos serão desenvolvidos para executar pilotos para testar e avaliar os processos novos e os melhorados. Depois do pleno sucesso do teste piloto dos novos processos e da determinação de sua presteza para adoção, distribuição e institucionalização em toda a organização, planos para acompanhar a sua implantação são então desenvolvidos e executados.

Nesta atividade o plano de ação tático deve ser completado nas seções não especificadas pelo MSG, revisando e restringindo o escopo do programa de melhoria de processo de software. A orientação para a determinação do escopo do projeto dos TWGs não devem exceder 6 a 9 meses.

Deve-se criar o WBS – Work Breakdown Structure – para o TWG, onde se quebra as tarefas globais em menores, tornando-se partes gerenciáveis que podem ser utilizadas como base para o planejamento, identificando-se marcos, notificações e pontos de controle [41].

O propósito desta fase é criar soluções para os problemas ou processos que a organização determinou que sejam necessários de acordo com as recomendações fornecidas pela avaliação. A solução selecionada ou desenvolvida deve ser compatível com a cultura da organização para que seja facilmente aceita e institucionalizada. Esta atividade é dividida nas duas tarefas abaixo:

- Refinar o processo (abordagem centrada no processo): A abordagem centrada no processo ocupa-se do entendimento de um processo chave específico identificado durante a fase de avaliação e aplicando-se refinamentos incrementais a este processo. Esta abordagem é útil por fornecer melhorias de longo prazo ao processo. Entretanto, devido a pressões e incertezas típicas de organizações de baixo nível de maturidade, é difícil manter este foco em tais organizações. Sustentar uma abordagem centrada no processo requer um forte comprometimento gerencial e ímpeto e entusiasmo organizacional. Para que esta abordagem tenha sucesso algumas tarefas devem ser executadas, entre elas: identificar os processos selecionados e entender suas necessidades; determinar o escopo, os limites e o contexto do processo atual; descrever o estado desejado do processo (o “será”); analisar a lacuna entre o estado “como é” e o “será”; criar o processo refinado; determinar os objetivos da modelagem do processo; modelar o novo processo; especificar as métricas do processo e implementar o processo. O resultado será a identificação dos componentes da solução: descrições, procedimentos, métricas, métodos e ferramentas do processo.
- Analisar e se fixar no problema (abordagem centrada no problema): A abordagem centrada no problema é distinta da abordagem centrada no processo pois torna mais fácil a identificação dos problemas e pode fornecer resultados mais rápidos. Quando os problemas se tornam complexos ou as soluções são difíceis de se manipular outros problemas, diferentes do problema inicialmente abordado, surgem inesperadamente. Devido ao ímpeto e ao entusiasmo no início de um programa de melhoria de processos a abordagem centrada no problema é mais útil. Para se obter o sucesso desta atividade é necessário executar as seguintes tarefas: formular o problema; definir os objetivos da solução; identificar restrições; analisar o

problema; gerar e selecionar alternativas para encaminhar o problema; definir métricas para a solução e selecionar a melhor solução das alternativas descobertas. O resultado obtido será o mesmo da abordagem centrada no processo.

As soluções propostas, tanto pela abordagem centrada no processo quanto na centrada no problema, devem ser testadas em projetos pilotos. As soluções precisarão de ajustes e refinamentos para adequá-las à todos os projetos da organização, e a pilotagem ajudará a determinar as necessidades de ajuste.

O objetivo do teste piloto é verificar a solução em um projeto real na organização e capturar as lições aprendidas. Os resultados obtidos com a pilotagem servem para refinar a solução e a sua instalação na organização.

Uma vez a solução desenvolvida e pilotada, e as necessidades de suporte de curto e longo prazo encaminhadas, a solução está pronta para ser implantada na organização. O TWG criará um modelo de plano de implantação que dêem orientação aos projetos de desenvolvimento que estarão instalando a melhoria de processo. O plano incluirá:

- Qual o treinamento necessário
- Quais as ferramentas e os métodos devem ser adquiridos
- Os passos para a instalação
- Informações de como obter suporte

As diversas soluções propostas devem ser enviadas ao SEPG para serem armazenadas em um banco de dados de processo. Como tarefa final o TWG pode fazer um relatório com as lições aprendidas que ajudarão ao MSG e ao SEPG no gerenciamento do desenvolvimento da melhoria de processo. Finalmente, o TWG tem seu trabalho finalizado e o seu esforço deve ser reconhecido e celebrado através da organização. As equipes de implementação de melhorias que compõem o TWG devem ser dissolvidas.

E finalmente, as soluções aprovadas e pilotadas devem ser implantadas em toda a organização, para isto os projetos devem ser treinados e um suporte de longo prazo deve ser fornecido. Para organizações com menor nível de maturidade nas quais há maior variação através da organização, mais adaptações são necessárias.

Nos projetos onde a melhoria foi implantada uma avaliação deve ser efetuada para se obter as lições aprendidas sobre a distribuição dos novos processos nos projetos, dando a realimentação ao SEPG para se refinar a instalação e distribuição dos novos processos. Fornecendo estas informações ao SEPG, os métodos e as técnicas utilizadas durante a implementação e implantação podem ser incorporados dentro do próximo ciclo de melhorias.

## **2.5 Fase 5 - Aprendizagem**

O objetivo da fase de Aprendizagem é fazer com que a próxima passagem através do modelo IDEAL seja mais eficaz. Neste momento, soluções são desenvolvidas, lições são aprendidas e métricas de desempenho e metas atingidas são coletadas. Estes artefatos são adicionados ao banco de dados de processo que se tornará uma fonte de informação para o pessoal envolvido na próxima passagem através do modelo.

Usando estas informações coletadas, uma avaliação da estratégia, dos métodos e da infraestrutura utilizada no programa de SPI pode ser apresentada. Fazendo isto, correções ou ajustes na estratégia, nos métodos ou na infra estrutura pode ser tornada prioritária no início.

O propósito desta fase é garantir que todas as lições aprendidas nas atividades anteriores deste ciclo estejam disponíveis para revisão. O motivo desta revisão é relembrar as atividades executadas nas fases anteriores e verificar que todas as lições aprendidas através da melhoria do processo de software estão no banco de dados de processo da organização.

Depois de obter as lições aprendidas, o SEPG deve procurar analisar as práticas e os processos de melhoria implantados, ajustar ou planejar os ajustes necessários e considerar a adição de novos processos que farão os projetos funcionarem melhor.

O objetivo é procurar por meios de fazer as coisas melhores em toda a organização no próximo ciclo do método de melhoria.

## 2.6 Aplicando o Modelo

Quando da aplicação do modelo IDEAL há dois componentes para uma atividade de melhoria do processo de software, um componente estratégico junto com um componente tático.

O componente estratégico, baseado nas necessidades e direções do negócio da organização, fornecerão a direção e a priorização das atividades táticas.

A Figura 10 mostra uma visão bidimensional da aplicação do modelo IDEAL. Este guia pretende direcionar estes dois níveis operacionais dentro de uma iniciativa de melhoria de processo:

- Um nível estratégico, no qual há processos que são responsabilidade do gerente senior.
- Um nível tático, no qual os processos são modificados, criados e executados pelos gerentes imediatos e pelos desenvolvedores.

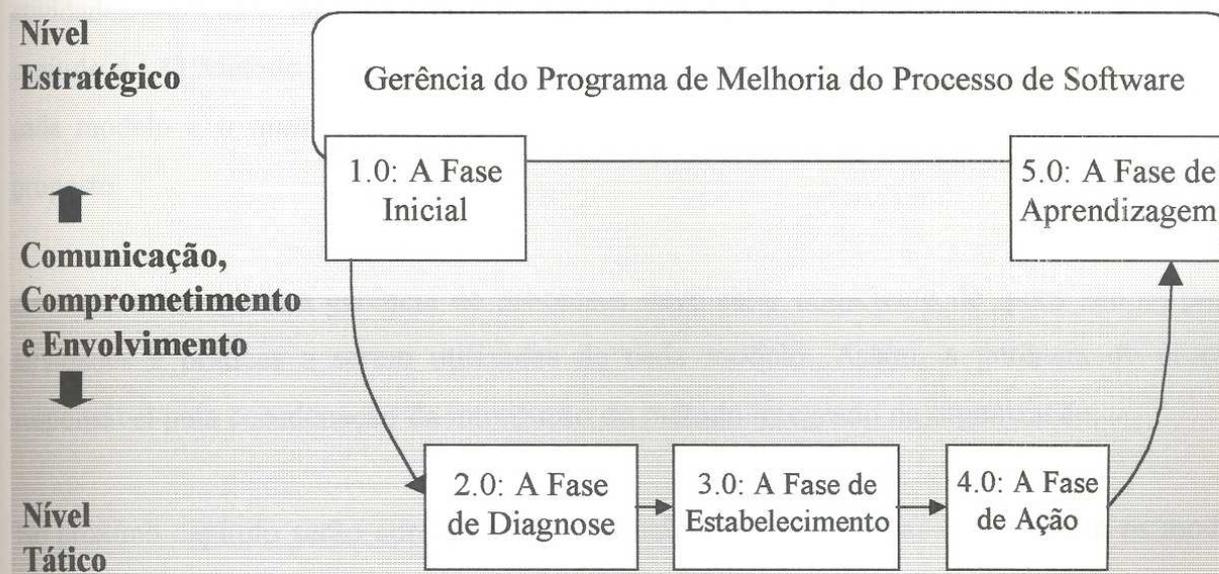


Figura 10 – Visão bidimensional da atividade de melhoria de processo.

O fluxo representado na Figura 10 pode ser visto como um fluxo contínuo. Quando a atividade de melhoria está completada, ambas as atividades dos níveis estratégico e tático retornam a fase

Aprendizagem, onde o compromisso de gerenciamento é reafirmado, novas linhas básicas são planejadas e a estratégia pode ou não ser redirecionada.

No nível estratégico há a Gerência do Programa de Melhoria do Processo de Software, o objetivo desta atividade é fornecer a supervisão dos projetos de melhoria e resolver conflitos entre diferentes programas que estejam ocorrendo em outras organizações da empresa. Em organizações muito grandes esta gerência pode ser executada por um "Executive Council" que deve estabelecer as orientações gerais para a introdução das melhorias e estabelecer quais são as prioridades de melhoria de processo. Além disso, esta unidade deve ser responsável pela troca de experiências, ou melhores práticas, verificadas nas diversas áreas de desenvolvimento da organização. Esta unidade deve guiar os SEPGs no objetivo de aumentar o nível de maturidade das suas respectivas áreas de atuação.

## **2.7 O IDEAL no Método Proposto**

A utilização integrada dos modelos de melhoria de processos ocorre no nível tático do modelo IDEAL. Esta integração ocorre quando da avaliação do processo corrente, na implementação das atividades/práticas e na implantação e disseminação destas através de toda a organização. O capítulo 4 - O Método Integrado apresenta quais os processos e práticas devem ser utilizados de cada modelo e o questionário integrado para a avaliação da introdução do método integrado proposto.

Ao nível estratégico, a gerência de processos deve estar consciente de quais são os processos, atividades e práticas a serem utilizadas de cada modelo. Assim, a gerência de processos é responsável pela coordenação da introdução do método integrado na organização.

No próximo capítulo serão apresentadas as principais características que definem os modelos de melhoria de processos utilizados no método integrado aqui proposto.

### 3 Modelos de Melhoria de Processos

#### 3.1 CMM – Capability Maturity Model

O CMM - Capability Maturity Model para Software [36] [14] [35] fornece às organizações de software um modelo de como controlar seus processos de desenvolvimento e manutenção de software e como evoluir em direção a uma cultura de excelência em engenharia e gerenciamento de software. O CMM foi criado para guiar as organizações de software na seleção de estratégias de melhoria do processo através da determinação da maturidade do processo atual e pela identificação de casos críticos para a obtenção da qualidade de software. Enfocando uma série limitada de atividades e trabalhando agressivamente para atingi-las, uma organização pode adquirir ganhos contínuos e persistentes na capacidade do processo de software [30].

A melhoria contínua do processo é baseada em vários passos pequenos e evolucionários, ao invés de inovações revolucionárias. O CMM fornece uma estrutura para a organização destes passos evolucionários dentro de cinco níveis de maturidade que configuram bases sucessivas para melhoria contínua do processo. Estes cinco níveis de maturidade definem uma escala numérica para medir a maturidade do processo de software da organização e para avaliar a sua capacidade do processo de software. Os níveis também ajudam uma organização a priorizar seus esforços de melhoria.

Um nível de maturidade é uma base evolucionária bem definida para se adquirir um processo de software maduro. Cada nível de maturidade fornece uma camada para a melhoria contínua do processo. Cada nível consiste de uma série de metas de processo que, quando satisfeitas, estabelecem um componente importante do processo de software. Alcançando cada nível da estrutura de maturidade estabelece-se um componente diferente na organização do processo, resultando no aumento da capacidade de processo da organização.

A Figura 11 apresenta os cinco níveis do CMM e as características e as ações de melhoria para o aumento da maturidade do processo de software.

Nível	Características	... para alcançar o próximo nível	Benefícios
<b>5</b> <b>Otimizando</b>	<b>Auto-adaptação</b> Processo contínuo de melhoria (realimentação)	<ul style="list-style-type: none"> <li>• Processo otimizado continuamente por um controle específico</li> <li>• Envolver todos os interessados</li> </ul>	
<b>4</b> <b>Gerenciado</b>	Quantitativo Processo medido; quantitativamente controlado	<ul style="list-style-type: none"> <li>• Aumentar a flexibilidade tecnológica</li> <li>• Analisar as áreas fracas do processo</li> <li>• Prevenir as áreas fracas do processo</li> </ul>	
<b>3</b> <b>Definido</b>	Qualitativo Processo definido, controlado e institucionalizado	<ul style="list-style-type: none"> <li>• Analisar processo de desenv. de SW</li> <li>• Medir atividades do processo</li> <li>• Redigir um plano de qualidade quantitativo</li> </ul>	
<b>2</b> <b>Repetível</b>	Intuitivo Baseado na experiência; processo depende de indivíduos	<ul style="list-style-type: none"> <li>• Empregar processo de desenv. de SW</li> <li>• Introduzir técnicas de revisão e teste</li> <li>• Treinar o staff</li> <li>• Considerar padrões e normas</li> </ul>	
<b>1</b> <b>Inicial</b>	Ad hoc Imprevisível; pessimamente controlado	<ul style="list-style-type: none"> <li>• Institucionalizar Planejamento de Projeto</li> <li>• Introduzir Gerenciamento de Projeto</li> <li>• Introduzir Gerenciamento de Configuração</li> </ul>	

Figura 11 – Os cinco Níveis do CMM.

Cada nível de maturidade é composto de várias áreas chave do processo (KPA). Cada área chave do processo é organizado em cinco seções chamadas características comuns. As características comuns especificam as práticas chave que, quando endereçadas coletivamente, acompanham as metas da área chave do processo. A estrutura do CMM é ilustrado na Figura 12.

Exceto para o Nível 1, cada um dos 5 níveis de maturidade é decomposto em várias áreas chave do processo que indicam as áreas que uma organização deve focar para melhorar o seu processo de software. As áreas chave do processo identificam os resultados que devem ser adquiridos para se atingir um nível de maturidade.

Cada área chave do processo identifica uma região de atividades relacionadas que, quando desempenhadas coletivamente, atingem uma série de metas consideradas importantes para aumentar a capacidade do processo. O caminho para atingir as metas de uma área chave do processo pode diferir entre os projetos baseadas nas diferenças entre os domínios ou ambientes das aplicações utilizadas. Entretanto, todas as metas de uma área chave do processo devem ser atingidas pela organização para satisfazer aquela área chave do processo. Quando as metas de uma área chave do processo são desempenhadas continuamente através de todos os projetos, a

organização pode ser dita como tendo institucionalizada a capacidade do processo caracterizada pela área chave do processo.

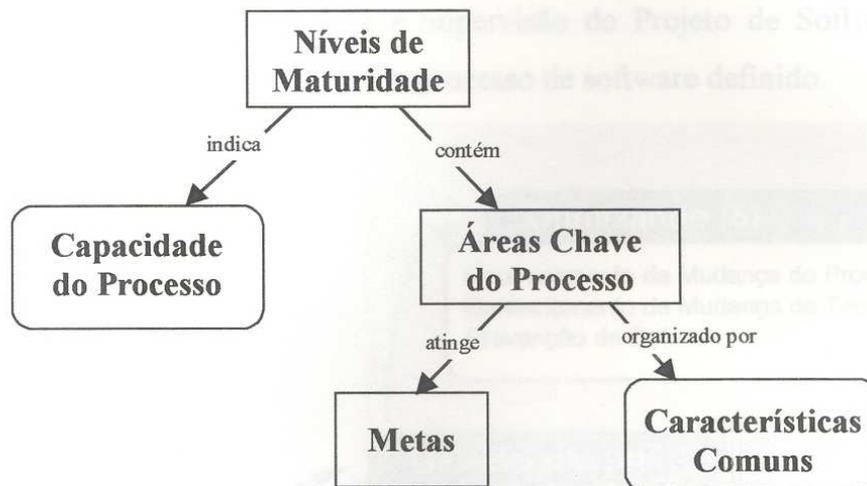


Figura 12 – Estrutura do CMM para Software.

O adjetivo “chave” implica que há áreas do processo (e processos) que não são chave para atingir um nível de maturidade. O CMM não descreve todas as áreas do processo no detalhe em que estão envolvidas no desenvolvimento e manutenção de software. Certas áreas do processo tem sido identificadas como chaves determinantes da capacidade do processo. Estas áreas estão descritas no CMM.

Embora outros resultados afetem o desempenho do processo, as áreas chave do processo foram identificadas por causa de sua efetividade na melhoria da capacidade do processo de software de uma organização. Elas podem ser consideradas como requisitos para atingir um nível de maturidade. A Figura 13 apresenta as áreas chave do processo para cada nível de maturidade. Para atingir um nível de maturidade, as áreas chave do processo daquele nível devem ser satisfeitas integralmente. As metas resumem as práticas chave de uma área chave do processo e podem ser utilizadas para determinar se uma organização ou projeto tem efetivamente implementada a área chave do processo. As metas significam o escopo, os contornos e a intenção de cada área chave do processo.

As práticas específicas a serem executadas em cada área chave do processo evoluem quando a organização atinge níveis mais altos do processo de maturidade. Por exemplo, muitas das

capacidades estimadas do projeto descritas na área chave do processo Planejamento do Projeto de Software do Nível 2 devem evoluir para ajudar os dados adicionais do projeto disponíveis nos Níveis 3, 4 e 5. O Gerenciamento Integrado de Software do Nível 3 é a evolução do Planejamento de Projeto de Software e do Rastreamento e Supervisão do Projeto de Software do Nível 2 quando o projeto é gerenciado utilizando-se um processo de software definido.

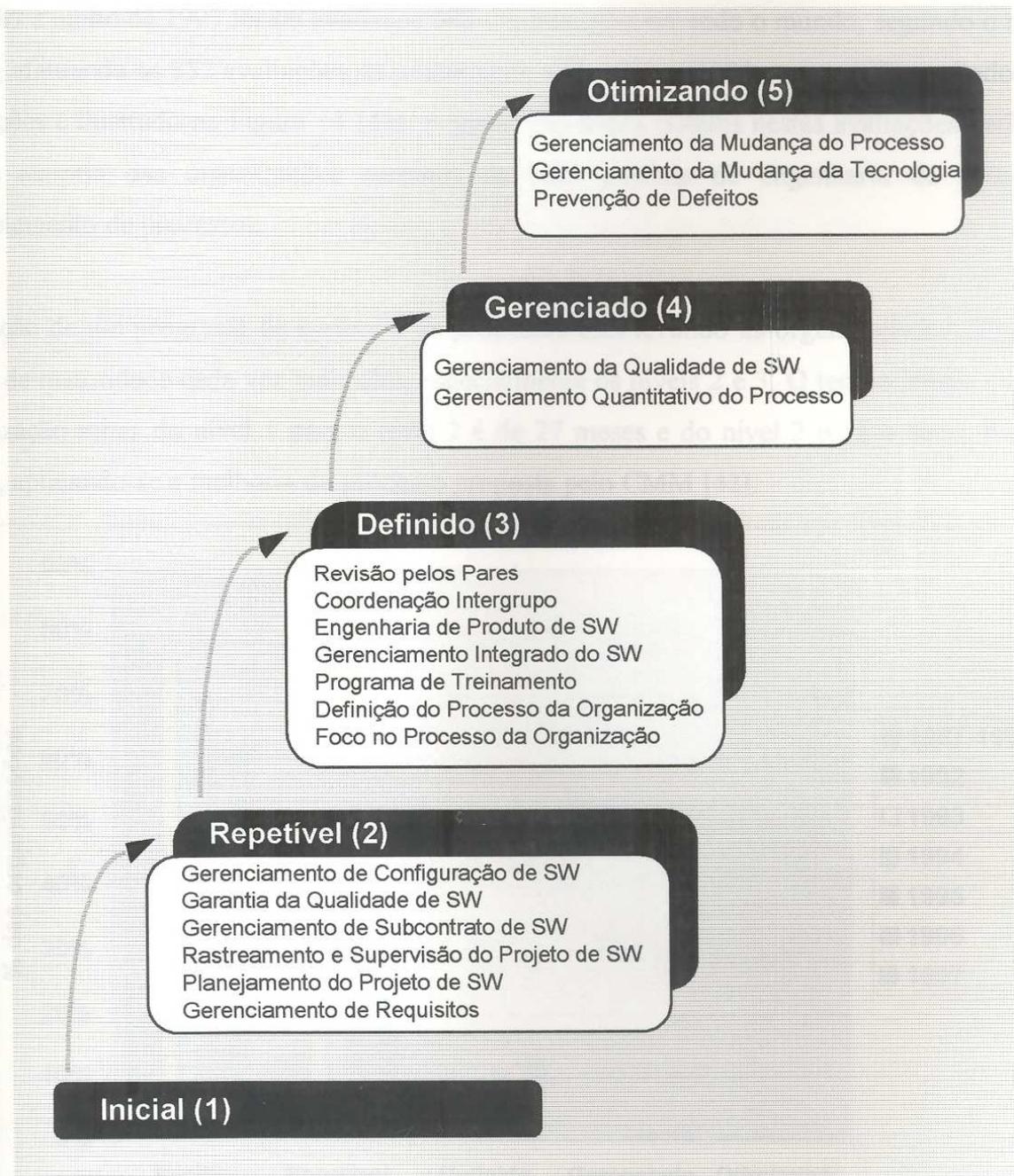


Figura 13 – Áreas Chave do Processo para cada nível de maturidade.

As áreas chave do processo do CMM representam um meio de descrever como as organizações amadurecem. Estas áreas chave do processo foram definidas baseadas em muitos anos de experiência em engenharia de software e gerenciamento e através de cinco anos de experiência com avaliações do processo de software e avaliações da capacidade de software [36][42].

De 1987 a junho de 1997 foram avaliadas 688 organizações em todo o mundo, segundo o modelo CMM, efetuando-se 852 avaliações de processo de software em 3584 projetos. O resultado destas avaliações é ilustrado na Figura 14 [42]. A partir dos dados obtidos nestas avaliações verifica-se que a maioria das organizações carecem de boas práticas de engenharia de software e gerenciamento de processos.

A adoção de um programa de melhoria de processos está levando as organizações a adquirirem níveis de maturidade cada vez mais alto, especialmente os níveis 2 e 3. O tempo médio para uma organização saltar do nível 1 para o nível 2 é de 27 meses e do nível 2 para o nível 3 é de 18 meses, aplicando-se a melhoria do processo proposta pelo CMM [42].

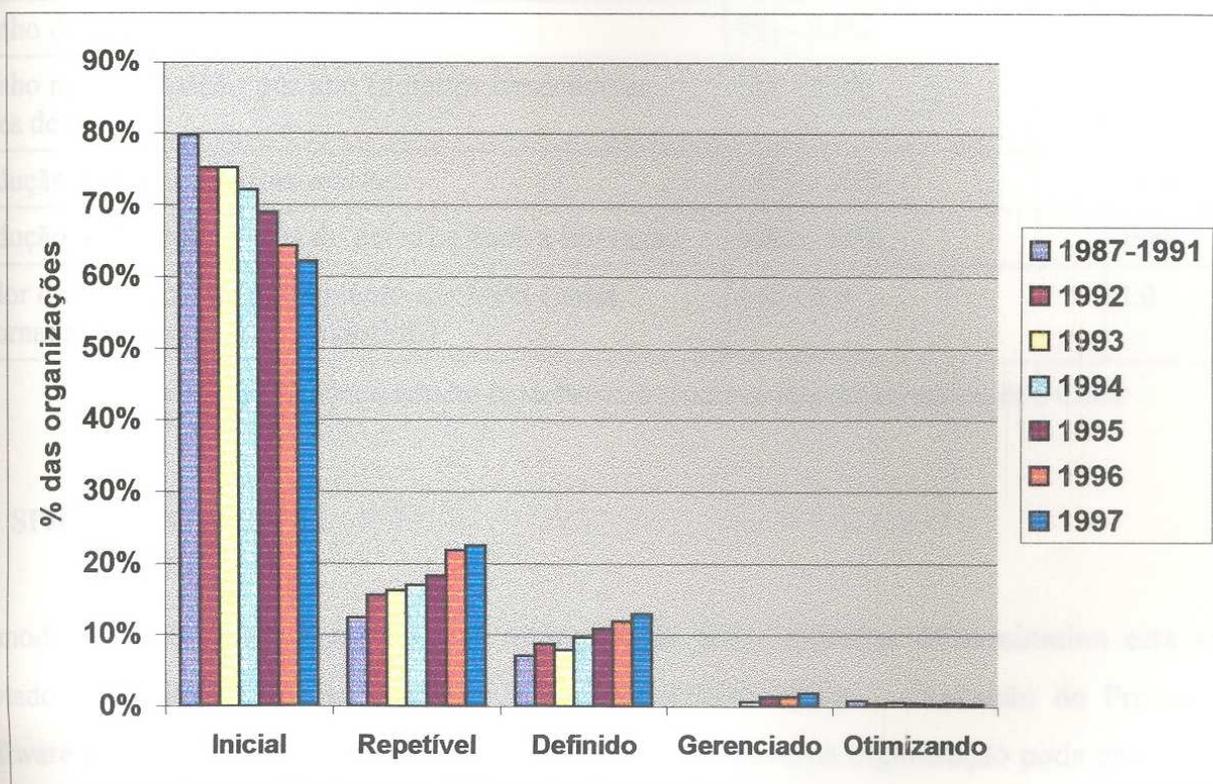


Figura 14 – Resultado de 852 avaliações efetuadas e sua distribuição nos níveis de maturidade.

Os resultados dos esforços de um programa de melhoria de processo de software utilizando o CMM podem ser representados observando-se os dados coletados de 13 organizações, resumidos na Tabela 3, que representam uma larga variedade de níveis de maturidade. Para cada um dos itens da tabela são apresentados os valores mínimo e máximo identificados nas organizações avaliadas e a mediana entre as 13 organizações. Ao observar como o desempenho mudou através do tempo dentro de cada uma das organizações, quando da implementação de um esforço de melhoria, podem ser identificados ganhos substanciais de produtividade, antecipação na detecção de defeitos, redução no *time-to-market* e aumento na qualidade dos produtos. Estes ganhos são adicionados aos retornos expressivos sobre os recursos investidos. Estes resultados mostram os tipos de ganho possíveis em um ambiente favorável, onde os atributos para um programa de melhoria de processos de software (MPS) são satisfeitos.

<b>Categoria</b>	<b>Variação</b>	<b>Mediana</b>
Total de custo anual das atividades de MPS	US\$49.000 – US\$1.202.000	\$245.000
Anos engajados em MPS	1 – 9	3.5
Custo da MPS por engenheiro de software	US\$490 – US\$2.004	US\$1.375
Ganho de produtividade por ano	9% - 67%	35%
Ganho na antecipação por ano (defeitos descobertos antes de testes)	6% - 25%	22%
Redução anual no <i>time-to-market</i>	15% - 23%	19%
Redução anual nos relatos de defeitos após a liberação	10% - 94%	39%
Valor de negócio do investimento em MPS (valor retornado para cada dólar investido)	4.0 – 8.8	5.0

**Tabela 3** – Resumo dos resultados de melhoria de processos de software utilizando o CMM.

### 3.2 SPICE – ISO 15504

O modelo SPICE – Software Process Improvement and Capability dEtermination está sendo adotado pela ISO/IEC como a norma internacional ISO 15504 – Avaliação do Processo de Software [44] [40] [27]. Este modelo descreve processos que uma organização pode desempenhar para adquirir, suprir, desenvolver, operar, evoluir e suportar software e as práticas genéricas que caracterizam a capacidade daqueles processos.

A arquitetura organiza as práticas de dois modos diferentes:

- Práticas Básicas, que são atividades essenciais de um processo específico, agrupadas dentro de processos e categorias de processo por tipos de atividades que eles endereçam;
- Práticas Genéricas, aplicável a qualquer processo, que representa as atividades necessárias para gerenciar um processo e melhorar sua capacidade de desempenho.

Os processos são classificados em níveis de capacidade, os níveis de capacidade são compostos de características comuns e as características comuns são compostas de práticas genéricas. As categorias de processo são compostas de processos, que são compostos de práticas básicas. Esta é uma decomposição por tipo de atividade. A composição destas duas hierarquias na classificação de um processo é ilustrado na Figura 15.

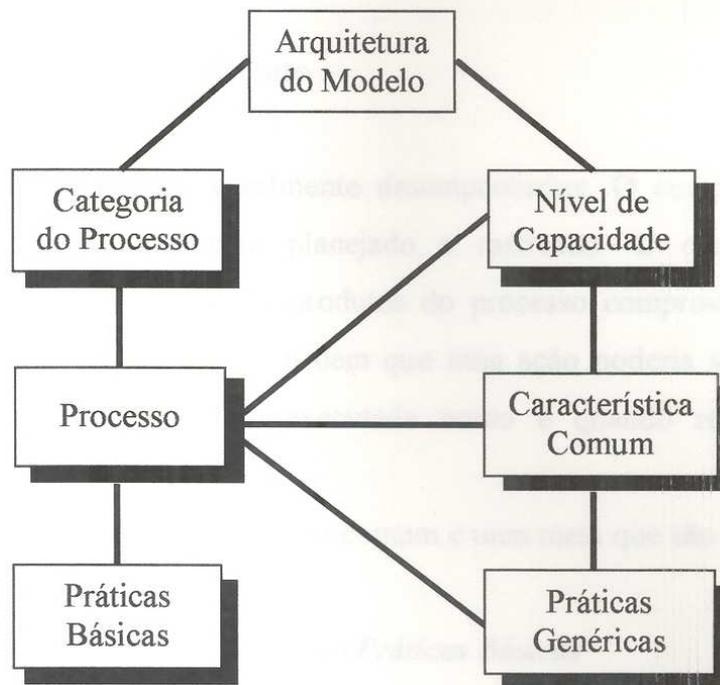


Figura 15 – Arquitetura do modelo SPICE.

A evolução da capacidade do processo é expressa em termos de níveis de capacidade, características comuns e práticas genéricas. Um nível de capacidade é uma série de características comuns (série de atividades) que funcionam juntos para fornecer um aumento maior na capacidade de desempenho de um processo. Cada nível fornece um aumento maior na capacidade

àquele fornecido pelo seu predecessor no desempenho de um processo. Eles constituem um meio racional de progredir através das práticas genéricas.

Os níveis de capacidade fornecem dois benefícios: eles reconhecem dependências entre as práticas de um processo e ajudam uma organização a identificar que melhorias ela pode desempenhar primeiro, baseado na seqüência plausível de implementação do processo. Os níveis de capacidade são apresentados a seguir:

### **Nível 0 – Não Desempenhado**

O nível **Não Desempenhado** não tem características comuns. Há uma falha geral para desempenhar as práticas básicas no processo. Não são facilmente identificáveis os produtos ou saídas do processo.

### **Nível 1 – Desempenhado Informalmente**

Práticas básicas do processo são geralmente desempenhadas. O cumprimento destas práticas básicas pode não ser rigorosamente planejado e rastreado. O cumprimento depende do conhecimento e esforço individual. Os produtos do processo comprovam o cumprimento. Os indivíduos dentro da organização reconhecem que uma ação poderia ser executada, e há uma concordância geral que esta ação é executada como e quando requisitada. Há produtos identificáveis pelo processo.

Este nível possui somente uma característica comum e uma meta que são apresentadas a seguir:

#### ***Característica Comum 1.1: Desempenhando Práticas Básicas***

##### **1.1.1 – Desempenhar o processo**

### **Nível 2 – Planejado e Rastreado**

O cumprimento de práticas básicas no processo é planejado e rastreado. O cumprimento de acordo com procedimentos especificados é verificado. As funções dos produtos estão conforme especificado em padrões e requisitos.

A distinção básica em relação ao Nível Desempenhado Informalmente é que o desempenho do processo é planejado e gerenciado e progride em direção a um processo bem definido.

As características comuns e suas respectivas metas são apresentadas a seguir:

#### ***Característica Comum 2.1: Planejando o Desempenho***

- 2.1.1 – Alocar recursos
- 2.1.2 – Assinalar responsabilidades
- 2.1.3 – Documentar o processo
- 2.1.4 – Fornecer ferramentas
- 2.1.5 – Garantir treinamento
- 2.1.6 – Planejar o processo

#### ***Característica Comum 2.2: Desempenho Disciplinado***

- 2.2.1 – Usar planos, padrões e procedimentos
- 2.2.2 – Fazer gerenciamento de configuração

#### ***Característica Comum 2.3: Verificando o Desempenho***

- 2.3.1 – Verificar a concordância com o processo
- 2.3.2 – Auditar os produtos.

#### ***Característica Comum 2.4: Rastreamento o Desempenho***

- 2.4.1 – Rastrear com medição
- 2.4.2 – Tomar ações corretivas

### **Nível 3 – Bem Definido**

Práticas básicas são desempenhadas de acordo com um processo bem definido, utilizando versões de padrões e processos documentados aprovados.

A distinção básica em relação ao Nível Planejado e Rastreado é que o processo do Nível Bem Definido é planejado e gerenciado utilizando um processo padrão em toda a organização.

As características comuns e suas respectivas metas do nível 3 são:

### ***Característica Comum 3.1: Definindo um Processo Padrão***

3.1.1 – Padronizar o processo

3.1.2 – Ajustar o processo padrão

### ***Característica Comum 3.2: Desempenhando o Processo Definido***

3.2.1 – Usar um processo bem definido

3.2.2 – Desempenhar revisões pontuais

3.2.3 – Usar dados bem definidos

## **Nível 4 – Quantitativamente Controlado**

Medidas detalhadas do desempenho são coletadas e analisadas. Isto leva a um entendimento quantitativo da capacidade do processo e uma melhoria na habilidade de prever o desempenho. O desempenho é objetivamente gerenciado. A qualidade dos produtos é quantitativamente conhecida.

A distinção básica em relação ao Nível Bem Definido é que o processo definido é quantitativamente entendido e controlado.

O nível 4 possui as seguintes características comuns com suas respectivas metas:

### ***Característica Comum 4.1: Estabelecendo Metas de Qualidade Mensuráveis***

4.1.1 – Estabelecer metas de qualidade

### ***Característica Comum 4.2: Gerenciando Objetivamente o Desempenho***

4.2.1 Determinar a capacidade do processo

4.2.2 Usar a capacidade do processo

## **Nível 5 – Melhorando Continuamente**

Metas de eficácia e eficiência quantitativas do processo para o desempenho são estabelecidas, baseadas nas metas de negócio da organização. A melhoria contínua do processo contra estas

metas é habilitado por uma realimentação quantitativa da execução de processos definidos e da adoção de idéias e tecnologias inovadoras.

A distinção básica em relação ao Nível 4, Quantitativamente Controlado, é que o processo definido e o processo padrão suporta refinamentos e melhorias contínuas, baseado no entendimento quantitativo do impacto das mudanças destes processos.

A seguir as características comuns e suas metas são apresentadas:

#### ***Característica Comum 5.1: Melhoria da Capacidade Organizacional***

5.1.1 – Estabelecer metas eficazes do processo

5.1.2 – Melhorar continuamente o processo padrão

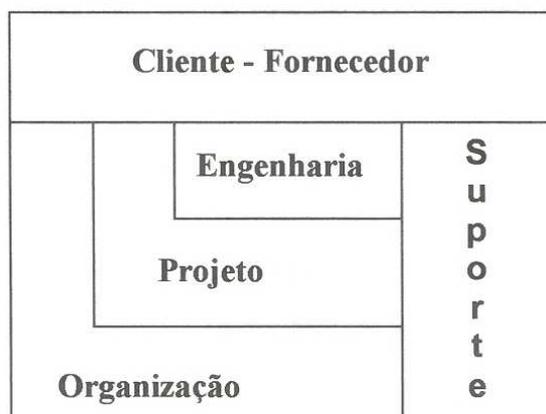
#### ***Característica Comum 5.2: Melhorando a eficácia do processo***

5.2.1 – Executar análises causais

5.2.2 – Eliminar causas de defeitos

5.2.3 – Melhorar continuamente o processo definido

As práticas básicas são agrupadas em cinco categorias de processo. A Figura 16 apresenta o relacionamento entre estas categorias de processo.



**Figura 16** – Relacionamento entre as categorias de processo.

A categoria de processo Cliente-Fornecedor consiste dos processos que impactam diretamente o cliente, o desenvolvimento do suporte, a transição do software para o cliente e o fornecimento para a sua correta operação e uso. Os processos que compõem esta categoria de processo são:

- Adquirir produto e/ou serviço de software
- Estabelecer um contrato
- Identificar as necessidades do cliente
- Executar revisões e auditorias conjuntas
- Empacotar, entregar e instalar o software
- Fornecer suporte a operação do software
- Fornecer assistência ao cliente
- Avaliar a satisfação do cliente

A categoria de processo Engenharia consiste dos processos que diretamente especificam, implementam ou mantêm um sistema e os produtos de software e sua documentação de usuário. Esta categoria de processos é compreendida pelos seguintes processos:

- Desenvolver os requisitos e o *design* do sistema
- Desenvolver os requisitos de software
- Desenvolver o *design* de software
- Implementar o *design* de software
- Integrar e testar o software
- Integrar e testar o sistema
- Manter o sistema e o software

A categoria de Projeto consiste dos processos que estabelecem o projeto, e coordenam e gerenciam seus recursos para produzir um produto ou fornecer uma assistência que satisfaça o cliente. Os processos que compõem esta categoria de processo são:

- Planejar o ciclo de vida do projeto
- Estabelecer um plano de projeto
- Construir equipes de projeto
- Gerenciar os requisitos
- Gerenciar a qualidade

- ❑ Gerenciar os riscos
- ❑ Gerenciar recursos e cronograma
- ❑ Gerenciar subcontratados

A categoria de processo Suporte consiste de processos que habilitam e suportam o desempenho de outros processos em um projeto. Os processos que compõem esta categoria de processos são:

- ❑ Desenvolver a documentação
- ❑ Executar o gerenciamento de configuração
- ❑ Executar a garantia da qualidade
- ❑ Executar a resolução de problemas
- ❑ Executar a revisão pelos pares

A categoria de processo Organizacional consiste de processos que estabelecem as metas de negócio da organização e o desenvolvimento de bens como processos, produtos e recursos que ajudarão a organização a atingir suas metas de negócio. Os processos que compõem esta categoria de processos são:

- ❑ Engenhar o negócio
- ❑ Definir o processo
- ❑ Melhorar o processo
- ❑ Executar treinamento
- ❑ Facilitar o reuso
- ❑ Fornecer um ambiente de engenharia de software
- ❑ Fornecer facilidades para o trabalho

Uma comparação entre os modelos CMM e SPICE será efetuada no capítulo 4 quando será apresentada a metodologia de integração entre os modelos analisados.

### 3.3 PSP – Personal Software Process

Embora o Capability Maturity Model (CMM) forneça uma estrutura poderosa para melhoria do processo, seu foco está necessariamente em “o que” as organizações devem fazer e não “como”

elas devem fazê-lo. Muitas organizações que utilizaram o CMM tiveram problemas na aplicação dos seus princípios. Em grupos pequenos, por exemplo, geralmente não é possível ter especialistas dedicados em processo, todos os engenheiros devem dedicar ao menos parte do tempo na melhoria do processo. Isto se deve ao custo de se manter funcionários dedicados a melhoria de processo, sendo que tais colaboradores devem possuir um alto grau de especialização e visão global do ambiente de desenvolvimento da organização, deslocando-os assim de suas funções produtivas de desenvolvimento propriamente dito para uma função não operacional que é o gerenciamento de processos.

O *Personal Software Process* (PSP) [15] [16] [17] é um processo de auto melhoria desenvolvido para ajudar no controle, gerenciamento e melhoria da maneira como um engenheiro de software trabalha. O PSP fornece um fluxo de trabalho onde o programador ou engenheiro de software faz uma seqüência de exercícios e analisa os dados, os erros e tenta entender o que está errado para melhorar o seu próprio processo de trabalho.

O objetivo do PSP é fornecer um método estruturado para, através de uma abordagem *bottom-up*, melhorar o processo de software de uma organização que está adotando a melhoria do processo pelo modelo CMM (Capability Maturity Model). O modelo CMM fornece uma abordagem *top-down*, isto é da gerência superior para os desenvolvedores.

Os passos seguintes são adotados pelo PSP para orientar o engenheiro de software na melhoria do seu processo pessoal de desenvolvimento de software:

- Identificar aqueles métodos e práticas de software de grandes sistemas que podem ser utilizadas por indivíduos.
- Definir uma subsérie destes métodos e práticas que podem ser aplicados enquanto se desenvolve pequenos programas.
- Estruturar estes métodos e práticas para que elas possam ser gradualmente introduzidas.
- Fornecer exercícios apropriados para praticar estes métodos em uma estrutura educacional.

Com o PSP o engenheiro de software segue um processo, um planejamento e medições definidas, rastreia seu trabalho, gerencia a qualidade do produto e aplica uma realimentação quantitativa para melhorar seus processos pessoais, direcionando para:

- Melhor estimativa de tempo de codificação, tamanho do software, etc.
- Melhor planejamento e rastreamento das tarefas do processo de software
- Proteção contra acúmulo de tarefas ou obrigações
- Um comprometimento pessoal com a qualidade
- O envolvimento do engenheiro em processos de melhoria contínua

A progressão do PSP é mostrada na Figura 17.

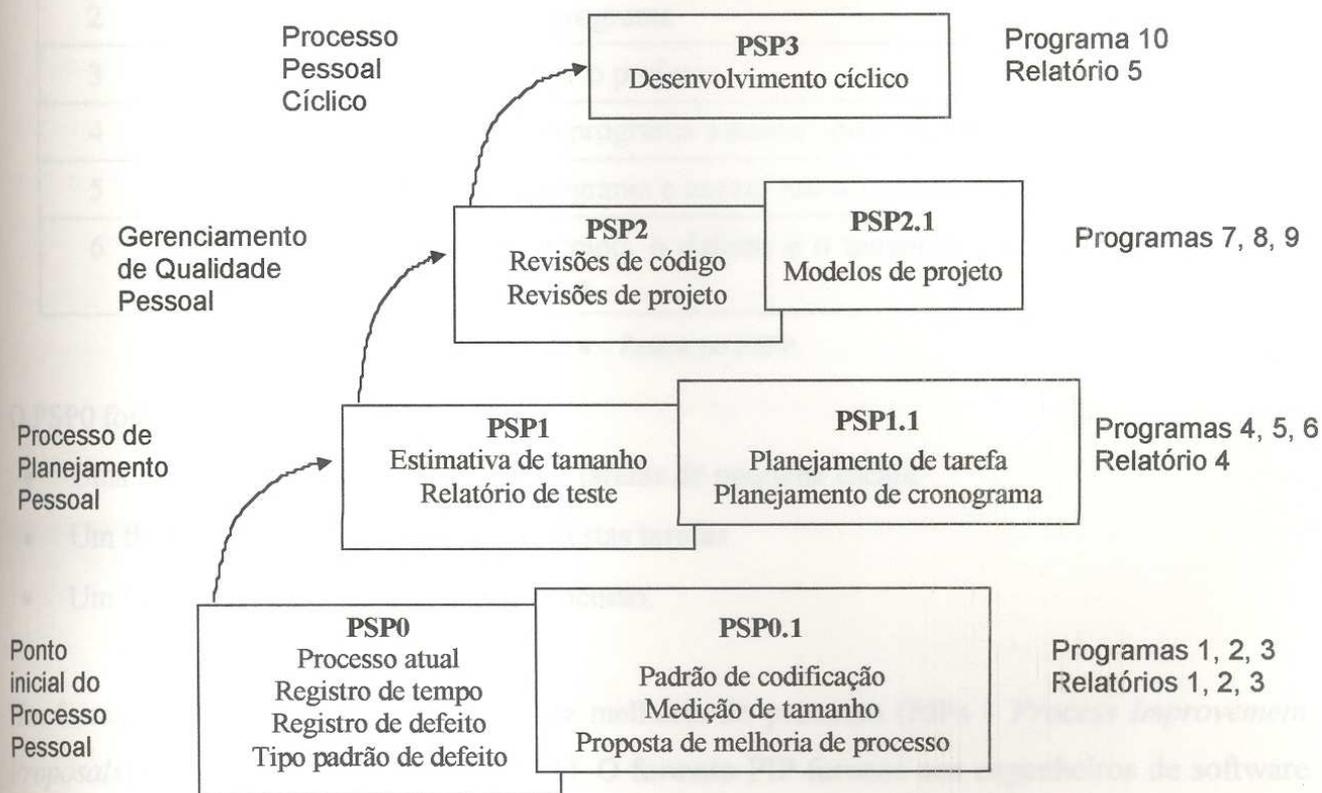


Figura 17 – Evolução do PSP.

A seguir serão apresentadas as características principais de cada uma das etapas do modelo PSP e os objetivos a serem alcançados pelo engenheiro de software nestas etapas[12].

## PSP0 e PSP0.1: O ponto inicial do Processo

O primeiro passo no PSP é estabelecer uma linha básica que inclui algumas medições básicas e um relatório. Esta linha básica fornece uma base consistente para o progresso da medição e um fundamento definido sobre o que melhorar. Os seis passos do processo PSP0 são mostrados na Tabela 4 de forma simplificada.

Passo	Fase	Descrição
1	Planejamento	Planejar o trabalho e documentá-lo
2	Projeto	Projetar o programa
3	Codificação	Implementar o projeto
4	Compilação	Compilar o programa e anotar todos os defeitos achados
5	Teste	Testar o programa e anotar todos os defeitos achados
6	Postmortem	Gravar o tempo, o defeito e o tamanho dos dados atuais no planejamento

Tabela 4 – Passos no PSP0.

O PSP0 fornece:

- Uma estrutura conveniente para fazer tarefas de pequena escala.
- Um fluxo de trabalho para medição destas tarefas.
- Um fundamento para melhoria do processo.

Um formato de gravação das propostas de melhoria do processo (PIPs – *Process Improvement Proposals*) é também introduzido (PSP0.1). O formato PIP fornece aos engenheiros de software um meio conveniente para gravar os problemas e soluções propostas para o processo.

## PSP1 e PSP1.1: Gerenciamento do Projeto Pessoal

PSP1 e PSP1.1 enfocam as técnicas de gerenciamento do projeto pessoal, introduzindo estimativas de tamanho e esforço, planejamento de cronograma e métodos de rastreamento de

cronograma. Estimativas de tamanho e esforço são feitas utilizando o método PROBE (PROBE – *PRoxy-Based Estimating*).

### **PSP2 e PSP2.1: Gerenciamento da Qualidade Pessoal**

O PSP2 e o PSP2.1 adicionam métodos de gerenciamento de qualidade ao PSP: projeto pessoal e revisões de código, uma notação de projeto, modelos de projeto, técnicas de verificação de projeto e medidas para gerenciamento do processo e qualidade do produto.

### **PSP3: Processo Pessoal Cíclico**

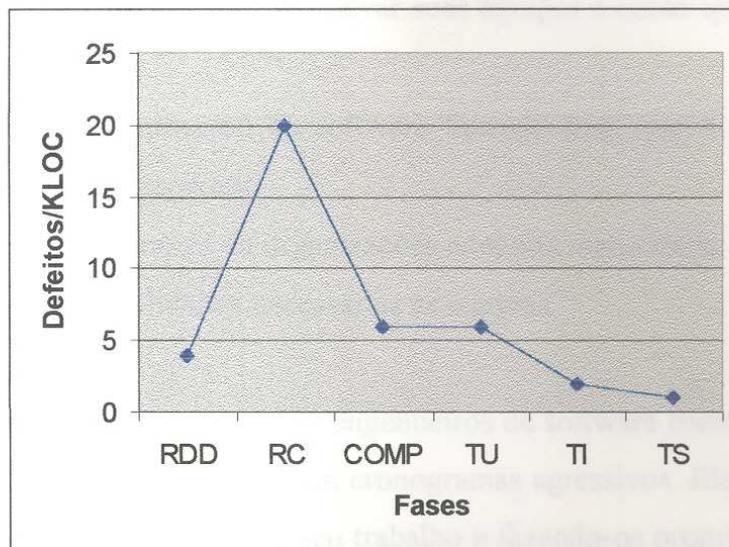
O processo pessoal cíclico, PSP3, aponta a necessidade da introdução gradual do PSP em projetos maiores sem sacrificar a qualidade ou produtividade. Para os engenheiros de software a sua produtividade é maior entre um intervalo definido de tamanho de código a ser desenvolvido. Abaixo deste intervalo, a produtividade declina devido aos custos fixos elevados. Acima deste intervalo, a produtividade declina porque o limite do processo de escalabilidade é alcançado. O PSP aponta este limite de escalabilidade pela introdução de uma estratégia de desenvolvimento cíclico onde programas maiores são decompostos em partes para o desenvolvimento e então integrados. Esta estratégia garante que os engenheiros de software estão trabalhando com produtividade máxima. Os níveis de qualidade do produto aumenta, em projetos grandes, somente com incrementos e não exponencialmente.

A Tabela 5 apresenta um resumo dos procedimentos apresentados pelo modelo PSP e seu relacionamento com o seu respectivo nível PSP.

Procedimento	Nível PSP
Estratégia do PSP e o PSP básico	PSP0
Planejamento do processo	PSP0.1
Medição do tamanho do software	PSP0.1
Estimativa do tamanho do software	PSP1
Estimativa de recursos e cronograma	PSP1.1
Medição de produtividade no PSP	PSP1.1
Revisão de projeto e código	PSP1.1
Gerenciamento da qualidade de software	PSP2
Projeto de software	PSP2 ou PSP2.1
Verificação do projeto de software	PSP2.1
Melhorando o PSP	PSP3
Definindo o processo de software e usando o PSP	PSP3

**Tabela 5** – Estrutura dos procedimentos do PSP.

Engenheiros de software treinados em PSP são capazes de rastrear o seu tempo de desenvolvimento, contar e registrar os defeitos que eles injetam no código e removem do código e medir o tamanho dos produtos que eles produzem. O PSP mostra aos engenheiros de software que tais dados não são difíceis de se obter. Tais dados são necessários para planejar seu trabalho e melhorar o seu desempenho. Estes dados podem ser obtidos em cada fase do ciclo de vida de desenvolvimento de software. Um exemplo de dados obtidos é ilustrado na Figura 18, onde a escala vertical mostra a densidade de defeitos por 1000 linhas de código (KLOC) [22]. A linha horizontal apresenta as fases de desenvolvimento: revisão do *design* detalhado (RDD), revisão de código (RC), compilação (COMP), teste de unidade (TU), teste de integração (TI) e teste de sistema (TS).



**Figura 18** – Densidade de defeitos em cada fase do ciclo de desenvolvimento.

É importante, a partir da introdução do PSP, a organização fornecer um ambiente para obtenção dos dados de desempenho dos engenheiros de software criado-se um banco de dados histórico. Este banco de dados será importante para o sucesso da implementação do TSP pela organização. Casos práticos de utilização do PSP serão apresentados no capítulo 4 quando da proposta do método integrado.

### 3.4 TSP – Team Software Process

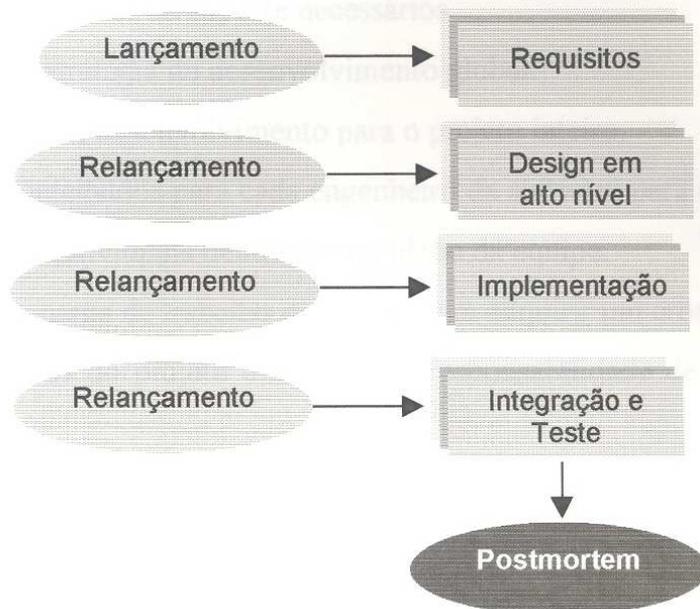
O Team Software Process (TSP) [18] [19] [21] [22] estende e refina os métodos do CMM e do PSP para guiar os engenheiros de software no seu trabalho em equipes de desenvolvimento e manutenção. Ele mostra aos engenheiros de software como construir uma equipe auto controlada e ter um desempenho como um membro eficiente da equipe. Ele também mostra aos gerentes como guiar e dar suporte a estas equipes e manter um ambiente que promova um alto desempenho da equipe. O TSP possui cinco objetivos:

- Construir equipes auto controladas que possam planejar e rastrear seu trabalho, estabelecer metas e seus próprios processos e planos. Estas equipes podem ser puramente de desenvolvimento de software ou equipes de produto integrado (IPT – Integrated Product Teams) de 3 a até aproximadamente 20 engenheiros.

- Mostrar aos gerentes como instruir e motivar suas equipes e como ajudá-las a sustentar picos de desempenho.
- Acelerar a melhoria do processo de software fazendo com que o CMM Nível 5 seja um comportamento normal e esperado.
- Fornecer melhoria na liderança de organizações com alta maturidade.
- Facilitar o ensino das habilidades necessárias às equipes.

O principal benefício do TSP é mostrar aos engenheiros de software como produzir produtos com qualidade através de custos planejados e em cronogramas agressivos. Ele faz isso mostrando aos engenheiros de software como gerenciar seu trabalho e fazendo-os proprietários de seus planos e processos.

O TSP fornece à equipe de projetos um método para executar seus objetivos. Como mostrado na Figura 19, o TSP guia as equipes através de 4 fases típicas de um projeto. Estes projetos podem começar ou terminar em qualquer fase, ou eles podem percorrer as etapas do início ao fim. Antes de cada fase, a equipe vai até um lançamento (*launch*) ou relançamento (*relaunch*) completo, onde elas planejam e organizam seu trabalho. Geralmente, uma vez que os membros da equipe são treinados em PSP, um workshop de lançamento de 3 dias fornece uma direção suficiente para a equipe completar uma fase do projeto. As equipes precisam então de um workshop de 2 dias de relançamento para começar cada uma das fases subsequentes. Estes lançamentos não são treinamento, eles são parte do projeto [18].



**Figura 19** – Estrutura do TSP.

A versão atual do TSP (preliminar) utiliza 23 *scripts*, 14 formulários e 3 normas. Os *scripts* do TSP definem 173 lançamentos e passos de desenvolvimento. Nenhum dos passos é complexo, mas cada um deles é definido com algum detalhe para que os engenheiros de software possam ver como fazer o que eles tem de fazer. Estes *scripts* guiam as equipes através dos passos de lançamento e percorre seus projetos. Casos práticos de utilização destes *scripts* não estão disponíveis na literatura técnica até o momento da publicação deste trabalho, pois o TSP está em sua versão preliminar. No método integrado proposto os conceitos inerentes ao TSP são suficientes para a análise da sua integração com os outros modelos de referência de melhoria de processos.

Para iniciar um projeto com TSP, o *script* de um processo de lançamento orienta as equipes através dos seguintes passos:

- Revisão dos objetivos do projeto com a gerência e concordância e documentação das metas da equipe.
- Estabelecer papéis dentro da equipe.
- Definir o processo de desenvolvimento da equipe.
- Fazer um plano de qualidade e um conjunto de alvos de qualidade.

- Planejar as facilidades de suporte necessários.
- Produzir uma estratégia de desenvolvimento global.
- Fazer um plano de desenvolvimento para o projeto inteiro.
- Fazer planos detalhados para cada engenheiro de software para cada nova fase do projeto.
- Unir os planos individuais dentro de um plano da equipe.
- Reequilibrar a carga de trabalho da equipe para adquirir um cronograma global mínimo.
- Avaliar os riscos do projeto e assinalar a responsabilidade de rastreamento para cada risco chave.

No passo de lançamento final, a equipe revisa seus planos e os riscos do projeto com a gerência. Uma vez o projeto iniciado, a equipe deve conduzir reuniões semanalmente e periodicamente relatar seu status para a gerência e para o cliente.

No *workshop* de lançamento de 3 dias, a equipe TSP produz:

- As metas da equipe escritas.
- Os papéis dentro da equipe definidos.
- Um plano do processo de desenvolvimento.
- Um plano de qualidade da equipe.
- Um plano de suporte de projeto.
- Um plano de desenvolvimento e um cronograma global.
- Os planos detalhados de cada engenheiro de software em cada fase.
- Uma avaliação dos riscos do projeto.
- Um relatório do status do projeto.

Talvez a mais poderosa consequência do TSP é o meio com que ele ajuda as equipes a gerenciar o seu ambiente de trabalho. O problema mais comum que as equipes de desenvolvimento de produtos encontram é a pressão de cronogramas irracionais. Embora isto seja normal, ele pode ser destrutivo. Quando as equipes são forçadas a trabalhar com cronogramas irracionais, elas são incapazes de fazer planos úteis. Todo plano que elas produzem não são aceitos. Como resultado,

elas devem trabalhar sem um plano que as guie ordenadamente e geralmente levarão muito mais tempo para completar o projeto [22].

O problema mais sério com trabalhos complexos e desafiadores é manter a disciplina e a consistência do desempenho em um nível alto. Nos esportes e nas artes, por exemplo, tem-se reconhecidamente a necessidade por habilitados treinadores, condutores e diretores. Seu trabalho é motivar e guiar os executores e também insistir para que eles atinjam altos padrões pessoais. Embora jogadores habilidosos sejam essenciais, é o treinador que de forma consistente produz equipes vencedoras. Há muitas diferenças entre grupos de software e artistas ou atletas, mas todos eles compartilham uma necessidade comum, sustentar altos desempenhos. Isto requer acompanhamento e suporte.

Os gerentes de software não tem, tradicionalmente, agidos como técnicos, mas este é seu papel no TSP. O trabalho do gerente é fornecer os recursos, efetuar a interface com a gerência mais alta e resolver problemas. Mas o mais importante, o gerente deve motivar a equipe e manter um foco implacável na qualidade e excelência.

Para se avaliar a qualidade de um projeto de software e com base nos dados obtidos através do PSP a organização pode desenvolver um gráfico do perfil de qualidade. Os limites dos processos, com base no PSP, informam que o tempo de *design* deveria ser maior que 100% do tempo de codificação, o tempo de revisão do *design* deveria ser maior que 50% do tempo de *design* e o tempo de revisão do código deveria ser maior que 50% do tempo de codificação. Além disso, os defeitos de compilação deveriam ser menores que 10/KLOC e os defeitos em teste de unidade estar abaixo de 5/KLOC[22]. Quando um fator atinge ou excede estes critérios, o perfil daquela dimensão está na borda do gráfico de perfil de qualidade. Quando o critério não é atingido, como por exemplo o tempo de revisão de *design* é de 25% ao invés de 50%, aquela dimensão poderia estar na metade do caminho entre o centro e a borda do gráfico. Estes critérios estão resumidos na Tabela 6. Toda organização deve obter seus próprios dados e selecionar seus próprios valores de perfil baseado em suas pessoas, métodos, ferramentas, áreas de aplicação e experiência.

Dimensão	Significado
Tempo de Design/Codificação	Razão entre o tempo de <i>design</i> detalhado e o tempo de codificação – quando os engenheiros não gastam tempo para produzir um <i>design</i> cuidadoso, eles geralmente cometem mais erros de <i>design</i> . Para reduzir este risco, o tempo de <i>design</i> deve ser igual a pelo menos 100% do tempo de codificação.
Tempo de Revisão de Código	O tempo gasto na revisão de código, comparado com o tempo de codificação – pela revisão pessoal do código antes de eles o compilarem, os engenheiros podem achar uma alta porcentagem de defeitos. Uma revisão de código cuidadosa pode levar 50% ou mais do tempo de codificação.
Defeitos em Compilação/KLOC	Os defeitos por KLOC achados na compilação – mesmo com bons tempos e taxas de revisão, a revisão ainda poderia ter deixado muitos defeitos. Para produtos de qualidade, defeitos na compilação devem ser menores que 10 defeitos/KLOC.
Tempo de Revisão de Design	O tempo na revisão de <i>design</i> detalhado, relacionado ao tempo de <i>design</i> detalhado – uma revisão cuidadosa do <i>design</i> detalhado deve levar 50% ou mais do tempo gasto no <i>design</i> detalhado. Qualquer coisa a menos geralmente indica uma revisão inadequada.
Defeitos em Teste de Unidade/KLOC	Os defeitos por KLOC achados no teste de unidade – o número de defeitos achados no teste de unidade é um dos melhores indicadores dos valores que depois serão achados. Quando os defeitos no teste de unidade/KLOC excede 5, problemas subsequentes são comuns.

Tabela 6 – Dimensões do perfil de qualidade dos projetos segundo o TSP.

A Figura 20 apresenta um exemplo de um componente de perfil de qualidade de um projeto. Na interpretação dos perfis de qualidade, várias linhas guia podem ajudar a identificar alguns problemas potenciais nos componentes.

1. Se os engenheiros não gastam tempo suficiente durante a fase de *design*, eles estão provavelmente efetuando o *design* enquanto estão codificando. Como avaliado pelo PSP[12] eles introduzirão duas vezes mais defeitos por hora do que fariam com uma fase de *design* cuidadosa. O perfil indica este problema por um baixo valor na relação tempo de *design*/codificação.
2. Já que os engenheiros injetam defeitos durante o *design*, eles devem gastar tempo suficiente revisando estes *designs* ou eles não acharão os defeitos. Este defeito é indicado pelo baixo valor do tempo de revisão de *design*.
3. Mesmo quando os engenheiros gastam tempo suficiente fazendo uma revisão de *design*, eles podem não fazer uma revisão eficiente. Meramente olhando o *design* por longo tempo, por

exemplo, geralmente não resultará no achado de muitos defeitos. Os engenheiros devem utilizar métodos de revisão sonoros (conversando ou explicando) para achar os problemas mais sofisticados e mesmo os mais simples. Métodos de revisão de *design* pobres são geralmente indicados por um alto número de defeitos/KLOC em teste de unidade.

4. Se os engenheiros não gastam tempo suficiente revisando seus códigos antes de eles os compilarem, eles comumente deixarão para encontrar defeitos depois na compilação e teste. O perfil indica este problema com um baixo valor no tempo de revisão de código.
5. Mesmo quando os engenheiros gastam tempo suficiente fazendo uma revisão de código, eles podem não ter feito a revisão apropriadamente. Este problema é geralmente indicado pelo alto número de defeitos/KLOC na compilação.

Nota-se que os valores dos perfis próximos do centro indicam práticas de baixa qualidade e sugerem que o componente terá defeitos achados nos últimos testes ou pelo usuário.

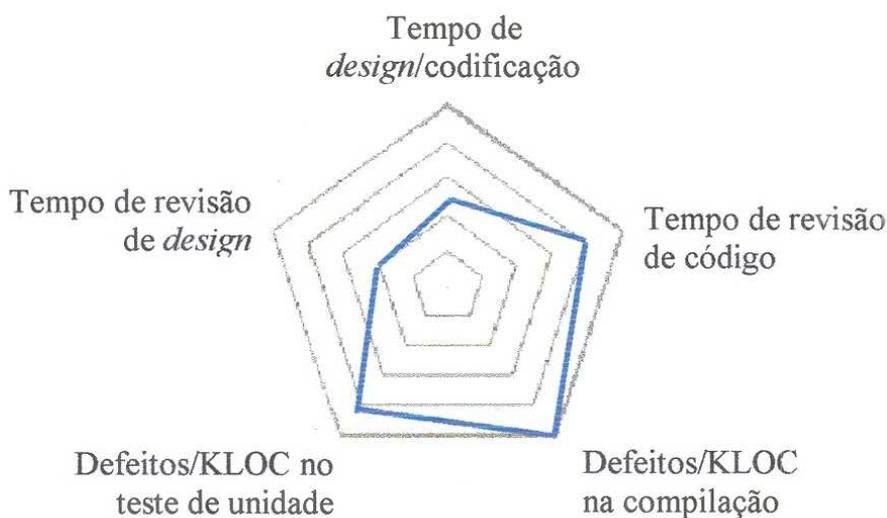


Figura 20 – Exemplo de um componente de perfil de qualidade.

## 4 O Método Integrado

### 4.1 Objetivo

O objetivo deste método de melhoria de processo de software é integrar os três aspectos, ou dimensões, da melhoria de processo de software, que são: organizacional, individual e equipe de projeto. Para melhorar as três dimensões do processo de software, o método integrado de melhoria do processo de software propõe a utilização conjunta dos modelos CMM, SPICE, PSP e TSP apresentados na seção anterior. No método integrado aqui proposto o objetivo é utilizar as melhores práticas de engenharia de software descritas nos modelos selecionados e que atendam o objetivo de melhorar as três dimensões do processo de software, a organizacional, a individual e das equipes de projeto. Não há, na literatura técnica atual, um método que faça a integração entre estes diferentes aspectos de melhoria de processos, caracterizando-se como um assunto de pesquisa inédito. Este não é um método que visa customizar qualquer um dos modelos ao interesse de uma organização em específico, como efetuado pela Bell Canada com o modelo Trillium [2], que é a adaptação do modelo CMM ao ambiente de desenvolvimento de produtos para telecomunicações. O conceito utilizado aqui é o mesmo adotado pela força aérea americana quando da integração entre o modelo CMM para software e o modelo CMM para engenharia de sistemas [23], ou seja, a utilização de parte de vários modelos que atendem determinadas características de organizações que desenvolvem um tipo de produto.

O método integrado proposto apresenta uma seqüência de ciclos de melhoria, onde a organização deve focar cada uma das dimensões de melhoria de processo de software. Para a implementação de cada um dos ciclos de melhoria deve-se adotar como abordagem o modelo IDEAL – Inicial, Diagnose, Estabelecimento, Ação e Aprendizagem – que fornece as 5 fases para a adoção de um programa de melhoria de processos de software.

Estas 5 fases criarão as condições necessárias para que cada ciclo do método obtenha o comprometimento da organização para com o foco a ser melhorado. Cada ciclo deve introduzir

algumas práticas e atividades que tem por objetivo aumentar, de forma contínua, a maturidade e a capacidade da organização quanto ao seu processo de desenvolvimento.

#### 4.2 Relação entre os modelos

Para efetuar a integração entre os modelos de melhoria de processos é necessário efetuar uma verificação do relacionamento entre estes modelos. A Tabela 7 apresenta uma relação entre as áreas chave do processo do modelo CMM SW v1.1 (seção 3.1) e as categorias do processo do modelo SPICE v1.0 (seção 3.2).

Nível CMM	Área Chave do Processo CMM	Categoria do Processo SPICE
2	RM	PRO.4
2	SPP	PRO.1, PRO.2, PRO.6
2	SPT&O	PRO.7
2	SSM	PRO.8
2	SQA	SUP.3
2	SCM	SUP.2, SUP.4
3	OPF	
3	OPD	ORG.2
3	TP	ORG.4
3	SPE	ENG.2, ENG.3, ENG.4, ENG.5, ENG.6, SUP.1, SUP.4, ORG.5, ORG.6
3	IC	CUS.3, PRO.3, ORG.1
3	ISM	
3	PR	SUP.5
4	QPM	ORG.2
4	SQM	PRO.5
5	PCM	ORG.1, ORG.3
5	TCM	

Tabela 7 – Relacionamento entre os modelos CMM SW v1.1 e SPICE v1.0.

A partir da Tabela 7 pode-se obter o grau de cobertura do CMM em relação ao SPICE, ou seja, qual o percentual das atividades descritas no modelo SPICE estão também descritas no CMM [11]. A Figura 21 ilustra este grau de cobertura. A partir do grau de cobertura verificamos quais categorias de processos presentes no modelo SPICE não são plenamente consideradas pelo

modelo CMM. Por exemplo, o modelo CMM abrange apenas 70% das atividades de engenharia descritas pelo modelo SPICE.

Este mesmo relacionamento pode ser efetuado entre os modelos CMM, SPICE e PSP, considerando-se como critério de comparação as atividades que fazem parte do ciclo de vida do desenvolvimento de software. Tais atividades podem ser implementadas como definido pelo modelo que a possui.

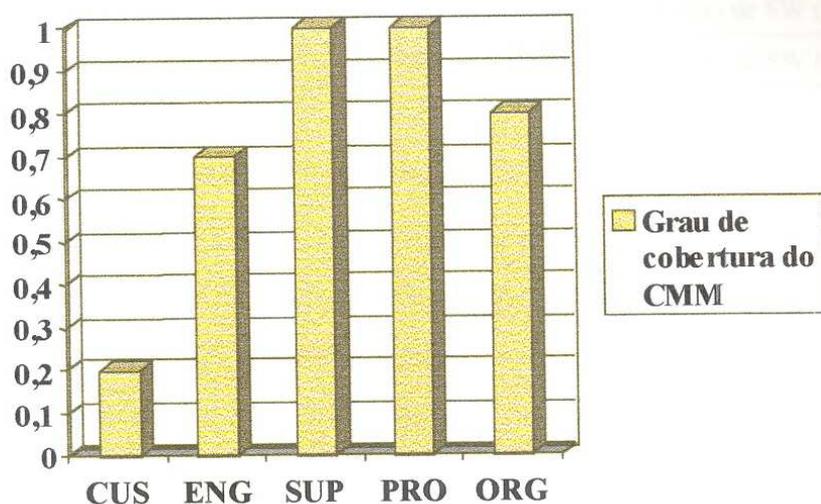


Figura 21 – Grau de cobertura do CMM em relação ao SPICE.

A Tabela 8 apresenta o relacionamento entre as atividades do ciclo de vida de software com os modelos CMM, SPICE e PSP. A partir deste relacionamento obtém-se as práticas que podem ser realizadas de acordo com o proposto pelos modelos. Por exemplo, a atividade de Requisitos e Design de Sistema é descrita pelo modelo SPICE no primeiro processo da categoria de processo de Engenharia. Quando um processo está descrito em dois ou mais modelos foi utilizado como critério de seleção o detalhamento das atividades a serem executadas pelo processo. Como exemplo pode ser citado o caso da revisão de *design* e código, em que estes processos são citados no modelo CMM como parte das atividades de *design* e implementação do código. Porém no modelo PSP estes processos são descritos detalhadamente, citando-se inclusive procedimentos para a execução do processo de revisão. Logo, para este processo, o modelo integrado deve se basear no descrito no modelo PSP. O mesmo critério é válido para todos os processos em que houve a comparação entre os modelos.

Ciclo de Vida de Desenvolvimento	Modelo / Processo
Requisitos e Design de Sistema	SPICE / ENG.1
Alocação de Requisitos de SW	CMM / Engenharia de Produto de SW (Atividade 2)
Design de SW	CMM / Engenharia de Produto de SW (Atividade 3)
Codificação	CMM / Engenharia de Produto de SW (Atividade 4)
Revisão de Design e Código	PSP / PSP2
Integrar o SW	SPICE / ENG.5
Teste de SW	CMM / Engenharia de Produto de SW (Atividade 5)
Teste de Integração	CMM / Engenharia de Produto de SW (Atividade 6)
Teste de Sistema e Aceitação	CMM / Engenharia de Produto de SW (Atividade 7)
Manutenção do Sistema e do SW	SPICE / ENG.7

**Tabela 8** – Comparação entre as atividades do ciclo de vida e os modelos.

Como mostrado na Tabela 9, o PSP e o TSP guiam os engenheiros de software e as equipes de desenvolvimento em quase todas as áreas chave do processo do CMM [18]. Estes métodos não apenas ajudam os engenheiros de software a serem mais eficazes mas também fornecem um entendimento mais profundo da necessidade de se acelerar a melhoria do processo organizacional.

Uma organização que esteja implementando um programa de melhoria do processo, pode utilizar-se do PSP para mostrar aos engenheiros de software como direcionar suas tarefas de um modo profissional. Embora relativamente novo, o PSP já tem mostrado seu potencial para melhorar as habilidades de planejamento e rastreamento do trabalho dos engenheiros de software e em produzir produtos com qualidade. Estando as equipes treinadas em PSP, elas geralmente precisam ajudar na aplicação das vantagens dos métodos de processo em seus projetos. O TSP guia estas equipes no lançamento de seus projetos e no planejamento e gerenciamento de seus trabalhos. Talvez o mais importante, o TSP mostra aos gerentes como guiar e treinar suas equipes de software a desempenhar as suas tarefas melhor e de forma consistente.

Tendo-se em vista que o PSP tem por objetivo melhorar a qualidade com que os engenheiros de software da organização desenvolvem, é necessário que exista uma infra estrutura básica de projeto para que as práticas introduzidas pelo PSP possam ser colocadas em prática. O foco

principal do primeiro ciclo de melhoria será fornecer esta base de gerenciamento e controle de projeto e a definição das fases do ciclo de vida de desenvolvimento da organização.

Nível CMM	Área Chave do Processo	PSP	TSP
5 Otimizando	Prevenção de Defeitos	X	X
	Gerenciamento de Mudança de Tecnologia	X	X
	Gerenciamento da Mudança de Processo	X	X
4 Gerenciado	Gerenciamento Quantitativo do Processo	X	X
	Gerenciamento da Qualidade de Software	X	X
3 Definido	Focos no Processo da Organização	X	X
	Definição do Processo da Organização	X	X
	Programa de Treinamento		
	Gerenciamento Integrado de Software	X	X
	Engenharia de Produto de Software	X	X
	Coordenação Intergrupo		X
	Revisão pelos Pares	X	X
2 Repetível	Gerenciamento de Requisitos		X
	Planejamento do Projeto de Software	X	X
	Rastreamento do Projeto de Software	X	X
	Garantia da Qualidade de Software		X
	Gerenciamento de Configuração de Software		X
	Gerenciamento de Subcontrato de Software		

**Tabela 9** – Como o PSP e o TSP abrangem as áreas chave do processo do CMM.

Tendo-se obtido a infra estrutura básica de projeto a organização deve focar a melhoria dos seus desenvolvedores pela adoção do PSP e a implementação de melhoria a nível organizacional. Esta melhoria organizacional, a ser efetuada em um segundo ciclo de melhoria, deve ser independente de projetos e visa consolidar as melhorias implementadas no primeiro ciclo.

Com a introdução do PSP a organização pode obter dados relevantes para a introdução do TSP, que tem por objetivo melhorar o rendimento das equipes de desenvolvimento. Além disso, a

organização deve implantar o gerenciamento da qualidade de maneira qualitativa e quantitativa. Este é o foco do terceiro ciclo de melhoria.

No quarto ciclo de melhoria a organização deve focar a melhoria contínua de seus processos pela implantação do gerenciamento de processos e pela detecção de defeitos. O objetivo deste ciclo é capacitar a organização a agir pró-ativamente, prevenindo-se da ocorrência de problemas com o processo de desenvolvimento.

A Tabela 10 apresenta os processos a serem implementados em cada um dos ciclos de melhoria de processos do método integrado, as práticas ou atividades de cada um dos processos e o modelo correspondente onde tais processos estão definidos.

Ciclo	Processos	Práticas/Atividades	Modelos
1	Planejamento	Todas	SPICE
	Projeto	Todas	SPICE
	Desenvolvimento	Todas	SPICE
	Teste	Todas	SPICE
	Implantação	Todas	SPICE
	Manutenção	Todas	SPICE
	Encerramento	Todas	SPICE
	Revisão	Todas	SPICE
	Atualização	Todas	SPICE
	Transferência	Todas	SPICE
2	Planejamento	Todas	SPICE
	Projeto	Todas	SPICE
	Desenvolvimento	Todas	SPICE
	Teste	Todas	SPICE
	Implantação	Todas	SPICE
	Manutenção	Todas	SPICE
	Encerramento	Todas	SPICE
	Revisão	Todas	SPICE
	Atualização	Todas	SPICE
	Transferência	Todas	SPICE
3	Planejamento	Todas	SPICE
	Projeto	Todas	SPICE
	Desenvolvimento	Todas	SPICE
	Teste	Todas	SPICE
	Implantação	Todas	SPICE
	Manutenção	Todas	SPICE
	Encerramento	Todas	SPICE
	Revisão	Todas	SPICE
	Atualização	Todas	SPICE
	Transferência	Todas	SPICE
4	Planejamento	Todas	SPICE
	Projeto	Todas	SPICE
	Desenvolvimento	Todas	SPICE
	Teste	Todas	SPICE
	Implantação	Todas	SPICE
	Manutenção	Todas	SPICE
	Encerramento	Todas	SPICE
	Revisão	Todas	SPICE
	Atualização	Todas	SPICE
	Transferência	Todas	SPICE

Tabela 10

Processos implementados em cada ciclo de melhoria

Ciclos	Processos	Atividades	Modelos
1	Definição do Processo da Organização	1 a 3	CMM Nível 3
	Engenharia de Produto de Software	1 a 8	CMM Nível 3
	Desenvolver Requisitos e Design do Sistema	Todas	SPICE ENG.1
	Integrar e Testar o Software	Todas	SPICE ENG.5
	Manter o Sistema e o Software	Todas	SPICE ENG.7
	Definir o Processo	Todas	SPICE ORG.2
	Revisão de Design e Código	Todas	PSP.2
	Gerenciamento de Requisitos	Todas	CMM Nível 2
	Planejamento de Projeto de Software	Todas	CMM Nível 2
	Rastreamento e Supervisão de Projeto de Software	Todas	CMM Nível 2
	Gerenciamento de Configuração de Software	Todas	CMM Nível 2
	Garantia da Qualidade de Software	Todas	CMM Nível 2
	Adquirir Produto e/ou Serviço de Software	Todas	SPICE CUS.1
	Identificar as Necessidades do Cliente	Todas	SPICE CUS.3
	Fornecer Instalações de Trabalho	Todas	SPICE ORG.7
	Gerenciamento de Mudança de Tecnologia	2 a 6	CMM Nível 5
2	Processo Pessoal de Software	Todas	PSP
	Definição do Processo da Organização	4 a 6	CMM Nível 3
	Engenharia de Produto de Software	9 e 10	CMM Nível 3
	Peer Review	Todas	CMM Nível 3
	Foco no Processo da Organização	Todas	CMM Nível 3
	Programa de Treinamento	Todas	CMM Nível 3
	Gerenciamento Integrado do Software	Todas	CMM Nível 3
	Coordenação Intergrupo	Todas	CMM Nível 3
	Gerenciamento de Mudança de Tecnologia	1, 7 e 8	CMM Nível 5
	Empacotar, Entregar e Instalar o Software	Todas	SPICE CUS.5
	Fornecer Serviços ao Cliente	Todas	SPICE CUS.7
	Avaliar a Satisfação do Cliente	Todas	SPICE CUS.8
	Facilitar o Reuso	Todas	SPICE ORG.5
3	Processo da Equipe de Software	Todas	TSP
	Gerenciamento Quantitativo do Processo	Todas	CMM Nível 4
	Gerenciamento da Qualidade de Software	Todas	CMM Nível 4
	Construir Equipes de Projeto	Todas	SPICE PRO.3
	Dar Suporte a Operação do Software	Todas	SPICE CUS.6
4	Prevenção de Defeitos	Todas	CMM Nível 5
	Gerenciamento da Mudança do Processo	Todas	CMM Nível 5

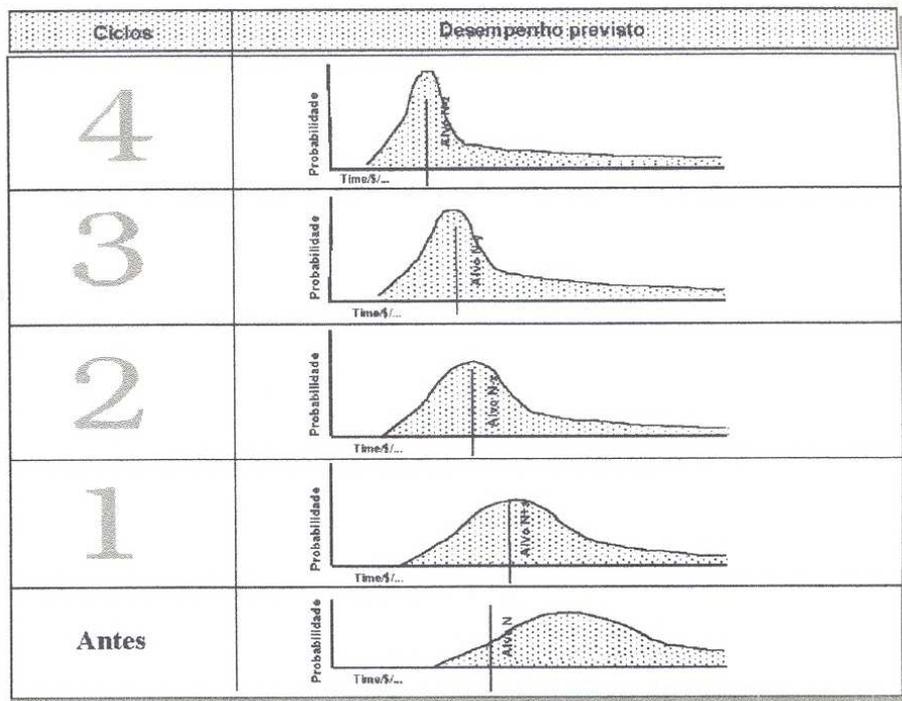
Tabela 10 – Processos e atividades a serem implementadas em cada ciclo de melhoria do método integrado.

### 4.3 Impacto no desempenho

A dificuldade de medição do desempenho dos processos e a comparação entre seus resultados é extremamente difícil sem um processo de software definido. Para guiar o uso de medições dentro de uma organização, possibilitando a análise do impacto da adoção de uma melhoria de processo, recomenda-se um modelo de engenharia de software que enfoque sobre as entradas, saídas e resultados. Este modelo deve capturar a criação dos produtos de trabalho e os impactos que eles tem sobre o desenvolvimento de software ou o desempenho do negócio da companhia. A análise do impacto da melhoria do processo de software deve ser dividida em três grupos [13].

A primeira envolve a identificação da medição que indicará o sucesso de uma estratégia de negócio, a definição cuidadosa dos procedimentos de medição e implementação para a coleta dos dados atuais. Esta medição tem por objetivo obter o desempenho da organização quanto aos aspectos financeiros, satisfação do cliente, processos internos e atividades de inovação e melhoria.

Na segunda, a partir de qualquer série de medidas, espera-se verificar que as mudanças estão em uma direção positiva através do tempo. A Figura 22 ilustra como uma organização pode medir o desempenho de seus processos de software com o decorrer do tempo, considerando-se custo(\$), cronograma(T), qualidade(Q) e funcionalidade(F) do processo de desenvolvimento. Assume-se que a organização está interessada em ambos na sua habilidade de predizer o seu desempenho e no seu desempenho atual em medidas como custo e cronograma. Nesta figura, uma organização está comparando sua estimativa aos resultados atuais de seus processos. O resultado atual de um projeto é primeiro normalizado a outros resultados de projetos similares na organização. Por exemplo, o custo pode ser normalizado expressando-o como uma porcentagem do custo orçado antes do projeto.



**Figura 22** – Evolução da estimativa com a melhoria do processo.

Há ao menos três tipos de melhorias que podem ser rastreadas com estes gráficos. Quando as melhorias são efetuadas, os valores devem se mover mais e mais em direção a estimativa, tornando o processo de software mais previsível. Na Figura 22, a distribuição dos valores se encontram mais próximas do centro da estimativa. Além disso, a distribuição dos valores deve se tornar mais agrupada de forma estreita em torno do centro, tendo uma menor variância. Finalmente, a média dos valores obtidos deve se mover na direção desejada; a média de custo, por exemplo, deve diminuir. Assim, podemos observar que, com a adoção do programa de melhoria de processo as estimativas efetuadas pela organização melhoram significativamente comparando-se a situação antes da implantação de melhores práticas de engenharia de software e depois de 4 ciclos de melhoria contínua.

Enquanto técnicas deste tipo são muito úteis para rastrear as mudanças através do tempo no desempenho de um processo de software, elas não especificam o que está causando as mudanças. Isto pode ocorrer pela melhoria do processo, ou pelo aumento da experiência dos projetistas, novas pessoas, novas análises ou métodos de *design*, novas ferramentas, mudanças na complexidade das aplicações, *hardware* mais rápido, etc. A identificação das causas das

mudanças no desempenho introduz uma nova série de resultados, esta identificação requer técnicas adicionais para a obtenção do quanto a melhoria no desempenho foi ocasionada pela mudança do processo.

A terceira análise de impacto deve estabelecer uma ligação causal entre as mudanças no processo de software e a medição do sucesso. Isto é importante porque o desenvolvimento de sistemas de software é um processo dinâmico em que muitas coisas em torno do processo de software estão mudando. Para medir os benefícios de um programa de melhoria de processos de software deve-se isolar os efeitos da melhoria do processo dos efeitos de outros fatores. Infelizmente, considerações práticas usualmente impõem sérias limitações na habilidade de uma organização para determinar as causas de mudanças no desempenho dos processos de software. Uma das mais comuns e sérias é simplesmente a falha em coletar qualquer dado de maneira confiável. E onde a coleta de dados é iniciada, ela está muitas vezes no contexto de um projeto piloto utilizando um processo novo, tal que muitas vezes não há dados de projetos anteriores ou outros projetos semelhantes que permita a comparação entre os dados obtidos. É impossível, desta maneira, determinar se a mudança de desempenho tenha ocorrido como resultado de um processo alterado. Onde dados de muitos projetos são coletados, a interpretação destes dados é extremamente difícil de ser feita pela falha na captura de características importantes do projeto que contribuem para os resultados.

#### **4.4 Ciclos de Melhoria**

Para cada um dos quatro ciclos de melhoria de processo serão apresentados qual o foco de cada ciclo de melhoria em termos das dimensões do processo de software e quais as atividades e processos a serem implementados para atingir os objetivos de melhoria propostos para o ciclo de melhoria considerado. Serão avaliados os impactos da introdução destas atividades, procedimentos e processos nos projetos desenvolvidos pela organização. E, para cada ciclo de melhoria do método integrado será apresentado o questionário de maturidade integrado que verifica se tais processos de melhoria foram devidamente institucionalizados pela organização. No capítulo 5 será apresentado um estudo de caso da implementação do primeiro ciclo de melhoria de processos que é proposto na seção 4.4.1.

#### 4.4.1 Primeiro Ciclo de Melhoria

Uma organização imatura normalmente não fornece um ambiente estável para o desenvolvimento e a manutenção de software. Os benefícios de boas práticas de engenharia de software são minados por planejamentos ineficazes e pelo comprometimento com reações induzidas pela falta de um bom gerenciamento.

Durante uma crise, os projetos em desenvolvimento na organização abandonam os procedimentos planejados e reverterem para codificação e teste. O sucesso depende inteiramente em ter um gerente excepcional e uma equipe de software afinada e eficaz. Ocasionalmente, gerentes de software capazes e vigorosos podem resistir as pressões para tomar atalhos no processo de software, mas quando eles deixam o projeto, sua influência de estabilização vão embora com eles. Mesmo um forte processo de engenharia não pode superar a instabilidade criada pela ausência das práticas de gerenciamento.

A capacidade do processo de software de uma organização imatura é imprevisível porque o processo de software é constantemente alterado ou modificado quando o trabalho avança. Cronogramas, orçamentos, funcionalidade e qualidade de produto são geralmente imprevisíveis. O desempenho depende da capacidade dos indivíduos e varia com suas habilidades inatas, conhecimentos e motivações. Há poucos processos de software em evidência e que se repetem de um projeto para outro e o desempenho é previsto apenas por indivíduos experientes ao invés de ser uma capacidade organizacional.

Neste primeiro ciclo de melhoria do método integrado a organização irá adquirir experiência com as tarefas de melhoria de processo e qualidade de software.

Esta experiência é importante para que os ciclos seguintes sejam mais eficazes e para que a organização verifique os ganhos obtidos com a melhoria do processo de software. A partir destas observações a organização selecionará as abordagens de melhoria a serem adotadas nos ciclos subsequentes e selecionará as prioridades de melhoria.

#### 4.4.1.1 Práticas a serem Implementadas

No primeiro ciclo de melhoria do processo de software a organização deve focar a melhoria da infra estrutura organizacional, aumentando a maturidade do processo de software, implementando práticas que aumentam a capacidade de desenvolvimento de software e que melhoram o controle e o gerenciamento do processo de software. Com o aumento da maturidade do processo de software diminuem os riscos associados ao desenvolvimento de software devidos a problemas internos à organização. Esta melhoria inicial do processo de software fornecerá as condições mínimas para se adotar as melhorias provenientes dos ciclos posteriores.

Com base na comparação entre os modelos efetuada na seção 4.2 verifica-se a necessidade de implementar práticas chave que estão presentes isoladamente em cada modelo. A obtenção destas práticas e sua implementação conjunta permite a integração entre os diferentes modelos de melhoria de processo.

Para iniciar a implementação das atividades de melhoria de processo de software a organização deve definir primeiramente o seu processo padrão. O processo padrão de desenvolvimento da organização deve possuir a definição do ciclo de vida de desenvolvimento de produtos. Tal ciclo de vida deve ser flexível ao ponto de permitir o desenvolvimento de produtos para parceiros e contratantes que possuam diferentes processos de desenvolvimento. Deve-se considerar que a organização deve apresentar seus resultados parciais de desenvolvimento de forma diferenciada para cada contratante. Tal flexibilidade deve ser levada em conta na definição do processo padrão da organização.

As atividades para a definição do processo estão descritas na área chave de processo Definição do Processo da Organização do nível 3 do CMM nas suas atividades 1 e 2 [38]. Porém, esta área chave abrange somente as atividades do ciclo de vida de desenvolvimento de software como descrito em sua atividade 3: "As descrições dos ciclos de vida de software que são aprovadas para utilização pelos projetos são documentados e mantidos". As atividades do ciclo de vida de software estão definidas na área chave do processo Engenharia de Produto de Software, como apresentado na Tabela 8.

Para estender o ciclo de vida de desenvolvimento para o caso em que o produto é um sistema, deve-se adotar as práticas fornecidas pelo SPICE. Neste modelo, tal prática é estabelecida na característica comum 3.1 “Definindo um Processo Padrão” das práticas genéricas do nível 3. A prática básica ORG.2 “Definir o Processo” da categoria de processo organizacional apresenta uma seqüência de passos para a definição do processo padrão da organização. Os 13 passos apresentados na prática básica ORG.2 [44] são:

- Definir as metas;
- Identificar as atividades, funções e responsabilidades atuais;
- Identificar entradas e saídas;
- Definir critérios de entrada e saída;
- Definir pontos de controle;
- Identificar interfaces externas;
- Identificar interfaces internas;
- Definir registros de qualidade;
- Definir medidas de processo;
- Documentar o processo padrão;
- Estabelecer um plano de ação;
- Estabelecer expectativas de desempenho;
- Disseminar o processo.

O processo padrão deve ser implementado levando-se em consideração as 7 práticas básicas da categoria de processo de engenharia do modelo SPICE e as práticas chave da área chave de processo “Engenharia de Produto de Software” do CMM, às quais estão definidas nas atividades 1 a 8 [38]. Cada fase do ciclo de vida de desenvolvimento deve ser: desenvolvida, mantida, documentada e verificada sistematicamente – como estabelecido pelo modelo CMM. As 7 práticas que compõem as atividades básicas do ciclo de vida de desenvolvimento de sistemas são:

- Requisitos e *design* do sistema;
- Requisitos de software;
- *Design* do software;

- Implementação do software;
- Integração e teste do software;
- Integração e teste de sistema;
- Manutenção e documentação do sistema.

Entre as atividades definidas no processo padrão da organização deve-se implementar as revisões de *design* e código. Estas revisões, ao invés de serem efetuadas como definido pela área chave do processo Revisões pelos Pares do nível 3 do CMM devem ser efetuadas individualmente pelos engenheiros da organização. A descrição das revisões de *design* e código estão no modelo PSP [15]. O objetivo principal de aplicar estas revisões é antecipar a detecção de erros, efetuando-as nas fases iniciais do projeto. Estas revisões podem ser implementadas através de um *checklist* de verificação, onde o desenvolvedor deve se basear para identificar os pontos mais comuns de defeitos a serem removidos. Porém, diferentemente do proposto pelo modelo PSP a revisão de código deve ser efetuada após a compilação. O objetivo desta revisão de código é obter os erros de lógica inseridos pelo desenvolvedor, os erros semânticos são detectados pela compilação.

Juntamente com a definição do processo padrão da organização, devem ser implementadas as áreas chave do processo do nível 2 do CMM[37]:

1. Gerenciamento de Requisitos
2. Planejamento do Projeto de Software
3. Rastreamento e Supervisão do Projeto de Software
4. Garantia de Qualidade de Software
5. Gerenciamento da Configuração de Software

Estas áreas chave devem ser implementadas tomando-se como base o processo de desenvolvimento definido. Para cada fase do ciclo de vida do processo de desenvolvimento da organização devem ser implementadas as práticas que garantam estas áreas chave de processo.

As práticas chave existentes nas áreas chave do processo do nível 2 do CMM tratam do gerenciamento e controle dos projetos. Os projetos devem seguir o processo padrão da

organização e as atividades descritas no nível 2 devem ser implementadas sobre os processos padronizados. Por exemplo, o gerenciamento de requisitos deve ser considerado sobre cada um dos processos do ciclo de vida definido pelo processo padrão, mesmo nos casos não previstos pelo CMM, como nas atividades específicas do desenvolvimento de HW ou componentes mecânicos.

A área chave de processo Gerenciamento do Subcontrato de Software deve ser implementada se a organização terceiriza partes do desenvolvimento dos sistemas. Neste caso os projetos subcontratados devem ser planejados e controlados segundo as práticas definidas nesta área chave. Porém, a organização sob estudo não possui terceirização no desenvolvimento de produtos ou de parte dos projetos. Logo, tal processo não será implementado no estudo de caso.

Em conjunto com a área chave de processo Gerenciamento de Requisitos deve-se implementar as práticas básicas CUS.3 - Identificar as Necessidades do Cliente - definidas pela categoria de processo Cliente-Fornecedor (CUS)[44], já que no CMM o Gerenciamento de Requisitos não abrange as etapas iniciais de obtenção dos requisitos junto ao cliente.

A atividade 5 da área chave de processo Planejamento do Projeto de Software declara: “Um ciclo de vida de software com estágios pré definidos de tamanhos gerenciáveis são identificados ou definidos”. Tal atividade depende claramente da definição do processo da organização justificando, assim, a sua realização antecipada em relação ao recomendado pelo CMM.

As outras áreas chave de processo do nível 2 a serem implementadas permitem o rastreamento do projeto de software, a verificação da qualidade do produto e do processo de desenvolvimento e o controle das versões do sistema. Estas áreas chave do processo são plenamente compatíveis com o modelo SPICE, segundo verificado na Tabela 7 na página 55.

Outras práticas do modelo SPICE que devem ser implementadas são: CUS.1 – Aquisição de Produto e/ou Serviço de Software e ORG.7 – Fornecer Instalações de Trabalho. Tais práticas preenchem uma lacuna existente no modelo CMM, como ilustrado pela Figura 21 na página 56, quanto a relação com os clientes/fornecedores e do ambiente organizacional. As práticas a serem

implementadas no processo CUS.1 são semelhantes ao definido pela área chave de processo Gerenciamento de Subcontrato de Software do CMM, porém o escopo definido no modelo SPICE é mais apropriado para o caso em que há avaliação de ferramentas de software que podem ser adquiridas ou desenvolvidas pela organização. O escopo adotado pelo CMM abrange a relação com um subcontratado que desenvolve parcial ou integralmente um sistema de software para a organização. No caso estudado, as práticas definidas pelo SPICE são mais convenientes com o realizado pela organização.

Para organizações que desenvolvem sistemas baseados em alto conteúdo tecnológico e com alta incidência de mudança tecnológica deve-se implementar algumas atividades da área chave de processo Gerenciamento de Mudança Tecnológica do nível 5 do CMM. Tais atividades fornecerão à organização maior agilidade na implantação de novas tecnologias, métodos e ferramentas no desenvolvimento de seus produtos. As atividades 2, 3, 4, 5 e 6 podem ser implementadas integralmente por uma organização que está iniciando um programa de melhoria de processos, pois não dependem de áreas chave do processo não plenamente realizadas [38].

A Tabela 11 apresenta a quantidade de práticas/atividades a serem implementadas pela organização neste primeiro ciclo de melhoria de processos, conforme apresentado na Tabela 10. Este ciclo de melhoria de processos perfaz um total de 97 práticas ou atividades que devem ser implementadas.

Modelos	No. de Atividades
CMM Nível 2	49
CMM Nível 3	11
CMM Nível 5	5
SPICE ENG	4
SPICE ORG	18
SPICE CUS	8
PSP	2
Total:	97

**Tabela 11** – Quantidade de práticas/atividades implementadas no primeiro ciclo.

#### 4.4.1.2 Impacto no Desempenho da Organização

Ao final do primeiro ciclo de melhoria a organização deve possuir estabelecidas as políticas de gerenciamento de projeto de software e os procedimentos para implementar aquelas políticas. Planejamento e gerenciamento de novos projetos são baseados na experiência com projetos similares. O objetivo a ser alcançado neste primeiro ciclo é a institucionalização de processos efetivos de gerenciamento nos projetos de software, que permita à organização repetir práticas desenvolvidas com pleno sucesso em projetos anteriores, embora os processos específicos implementados pelos projetos possam ser diferentes. Um processo eficaz pode ser caracterizado como praticado, documentado, reforçado, treinado, medido e habilitado para melhoria contínua. Isto é possível pela definição do processo padrão da organização. Tal definição permite a eliminação dos processos ineficientes ou a introdução de novos processos.

Projetos em organizações que concluem este primeiro ciclo tem instalado controles básicos de gerenciamento de software. Comprometimentos realistas de projeto são baseados nos resultados observados em projetos prévios e nos requerimentos do projeto atual. Os gerentes de software localizam no projeto: custos, cronogramas e funcionalidade do software; problemas de comprometimentos são identificados quando eles surgem. Requisitos de software e os produtos de trabalho desenvolvidos para satisfazê-los são sincronizados e sua integridade é controlada. Modelos de projeto de software são definidos e a organização assegura que eles são fielmente seguidos. O projeto de software trabalha com seus subcontratados, qualquer um, para estabelecer uma forte relação cliente-fornecedor. A introdução de novas tecnologias é, também, gerenciada e controlada.

A capacidade do processo pode ser sumariada como disciplinada porque o planejamento e o rastreamento do projeto de software é estável e os sucessos anteriores podem ser repetidos. O processo do projeto está sob o controle efetivo de um sistema de gerenciamento de projeto, seguindo planos realistas baseados no desempenho de projetos prévios. E as atividades que devem ser cumpridas durante o ciclo de vida de desenvolvimento estão definidas. Neste ciclo de vida de desenvolvimento devem estar contidas a definição de requisitos, o *design*, as revisões de *design* e código, a implementação e os testes. A introdução destas atividades torna o ciclo de vida

nais longo, já que grande parte das organizações iniciam seus projetos de software diretamente através da codificação e corrigem-no através de testes cíclicos.

Ao final da introdução deste primeiro ciclo de melhoria espera-se que o desempenho dos projetos da organização sejam melhores, embora maiores com relação aos quesitos de custo e tempo de desenvolvimento, já que a quantidade de tarefas a serem executadas aumenta significativamente. Porém as estimativas efetuadas pela organização melhoram significativamente pelo aumento na sua capacidade de estimativa de desenvolvimento, advindo das atividades de gerenciamento e controle.

#### 4.4.1.3 Questionário Integrado para o Primeiro Ciclo

O questionário a seguir serve de *checklist* para a verificação da implementação das atividades do primeiro ciclo de melhoria segundo o método integrado aqui proposto. Para que o primeiro ciclo esteja completo as questões devem ser respondidas afirmativamente e evidências concretas da sua execução na organização devem ser demonstradas.

1. Há um procedimento escrito que define as atividades de aquisição de um produto e/ou serviço de software?
2. Há uma política organizacional para a seleção de um fornecedor de produto e/ou serviço de software (p.ex. análise de risco, estratégia de aceitação e definição dos requisitos)?
3. A obtenção dos requisitos do sistema junto ao cliente segue um procedimento escrito?
4. Os requisitos do sistema são revisados juntamente com o cliente e os usuários?
5. Há um procedimento documentado para manter os clientes informados sobre o status dos requisitos de sistema?
6. Os requisitos de sistema são alocados para serem utilizados pelo software para estabelecer uma linha básica para o gerenciamento e a engenharia de software?
7. Como os requisitos de sistema alocados ao software mudam, são feitos ajustes necessários nos planos de software, produtos de trabalho e nas atividades?
8. Os projetos seguem uma política organizacional escrita para o gerenciamento dos requisitos de sistema alocados ao software?

9. As pessoas no projeto que são responsáveis pelo gerenciamento dos requisitos alocados são treinadas nos procedimentos de gerenciamento de requisitos?
10. Medições são utilizadas para determinar o status das atividades executadas pelo gerenciamento dos requisitos alocados (p.ex., o número total de requisitos alterados que são propostos, abertos, aprovados e incorporados dentro do projeto)?
11. As atividades de gerenciamento dos requisitos alocados ao projeto são sujeitas a revisão da Garantia da Qualidade de Software?
12. Estimativas (p.ex., tamanho, custo e cronograma) são documentadas para uso no planejamento e no rastreamento do projeto de software?
13. O plano de software documenta as atividades a serem executadas e os comprometimentos feitos pelo projeto de software?
14. Todos os grupos e indivíduos afetados concordam com seus comprometimentos relacionados ao projeto de software?
15. O projeto segue uma política organizacional escrita para o planejamento do projeto de software?
16. Recursos adequados são fornecidos para o planejamento do projeto de software (p.ex., financiamento e indivíduos experientes)?
17. Medições são utilizadas para determinar o status das atividades de planejamento do projeto de software (p.ex., conclusão dos marcos das atividades de planejamento de projeto quando comparadas ao plano)?
18. O gerente de projeto revisa as atividades de planejamento de projeto de software em bases periódicas e em eventos dirigidos?
19. O projeto segue uma política organizacional escrita para o rastreamento e o controle de suas atividades de desenvolvimento de software?
20. Alguém no projeto é assinalado com as responsabilidades específicas de rastreamento dos produtos de trabalho e das atividades de software (p.ex., esforço, cronograma e orçamento)?
21. Medições são utilizadas para determinar o status das atividades de rastreamento e supervisão de software (p.ex., esforço total expendido na execução das atividades de rastreamento e supervisão)?

22. As atividades de rastreamento e supervisão do projeto de software são revisadas periodicamente pelo gerente sênior (p.ex., desempenho do projeto, resultados abertos, riscos e itens de ação)?
23. As atividades de Garantia da Qualidade de Software são planejadas?
24. A Garantia da Qualidade de Software fornece uma verificação objetiva que os produtos e atividades de software se aderem as normas, procedimentos e requisitos aplicáveis?
25. Os resultados das revisões e auditorias de Garantia da Qualidade de Software são fornecidos aos grupos e indivíduos afetados (p.ex., aqueles que executaram o trabalho e aqueles que são responsáveis pelo trabalho)?
26. Resultados não concordantes que não são resolvidos dentro do projeto de software são endereçados pelo gerente sênior (p.ex., desvios dos padrões aplicáveis)?
27. O projeto segue uma política organizacional escrita para a implementação da Garantia da Qualidade de Software?
28. Recursos adequados são fornecidos para a execução das atividades de Garantia da Qualidade de Software?
29. Medições são utilizadas para determinar o custo e o status das atividades do cronograma executadas pela Garantia da Qualidade de Software (p.ex., trabalho completado, esforço e financiamento despendido comparado ao plano)?
30. As atividades de Garantia de Qualidade de Software são revisadas pelo gerente sênior periodicamente?
31. As atividades de gerenciamento de configuração de software são planejadas pelo projeto?
32. O projeto tem identificado, controlado e tornado disponível os produtos de trabalho de software através do uso do gerenciamento de configuração?
33. O projeto segue um procedimento documentado para controlar as mudanças na configuração dos itens/unidades?
34. Relatórios padrão das linhas básicas do software (minutas da gerência de controle de configuração de software, sumário das requisições de mudanças e relatórios de status) são distribuídas aos grupos e indivíduos afetados?
35. O projeto segue uma política organizacional escrita para a implementação das atividades de gerenciamento da configuração de software?

36. As pessoas do projeto são treinadas para executar as atividades de gerenciamento de configuração às quais elas são responsáveis?
37. Medições são utilizadas para determinar o status das atividades de gerenciamento de configuração de software (p.ex., esforços e financiamento gasto pelas atividades de gerenciamento de configuração de software)?
38. Auditorias periódicas são executadas para verificar que as linhas básicas do software estão conforme a documentação que as define (p.ex., pelo grupo de gerenciamento de configuração de software)?
39. A organização tem desenvolvido e mantido um processo de software padrão?
40. A organização segue uma política escrita para o desenvolvimento e manutenção do seu processo de software padrão e itens relacionados ao processo (p.ex., descrições dos ciclos de vida de software aprovado)?
41. Indivíduos que desenvolvem e mantêm o processo de software padrão da organização recebem o treinamento requerido para executar estas atividades?
42. As atividades e produtos de trabalho são produzidos de acordo com o processo de software definido pelo projeto?
43. As atividades e produtos de trabalho desenvolvidos e mantidos pelo processo de software padrão da organização são sujeitos a revisão e auditoria de Garantia da Qualidade de Software?
44. Os produtos de trabalho de software são produzidos de acordo com o processo de software definido pelo projeto?
45. O projeto segue uma política organizacional escrita para a execução das atividades de engenharia de software (p.ex., uma política que requer o uso de métodos e ferramentas apropriadas para construção e manutenção dos produtos de software)?
46. Recursos adequados são fornecidos para a execução das tarefas de engenharia de software (p.ex. financiamento, indivíduos habilitados e ferramentas apropriadas)?
47. As atividades e produtos de trabalho de engenharia de software estão sujeitas às revisões e auditorias de Garantia da Qualidade de Software (p.ex. o teste requerido é executado, os requisitos alocados são rastreados através dos requisitos, *design*, codificação e casos de teste)?
48. A revisão de *design* e código seguem uma política organizacional escrita?

49. Os engenheiros de software da organização possuem um *checklist* para revisão de *design* e código?
50. Há uma política organizacional para fornecer um ambiente organizacional seguro e confiável (p.ex. equipamentos necessários, garantir a integridade dos dados, fornecer *backup* dos dados, acesso a instalações que forneçam suporte às atividades de projeto, fornecer acesso remoto ao ambiente de trabalho quando apropriado)?
51. Novas tecnologias são avaliadas para determinar seus efeitos na qualidade ou produtividade?
52. O gerente sênior é o responsável pelas atividades da organização pelo gerenciamento de mudança de tecnologia (p.ex. pelo estabelecimento de planos e comprometerimentos de longo prazo para o financiamento, pessoas e outros recursos)?
53. As atividades da organização para o gerenciamento da mudança de tecnologia são revisados pelo gerente sênior periodicamente?

#### 4.4.2 Segundo Ciclo de Melhoria

O objetivo do segundo ciclo de melhoria é aprimorar o processo de software individual dos engenheiros de software da organização, introduzir o gerenciamento do processo de software e melhorar a interface com os clientes da organização.

##### 4.4.2.1 Práticas a serem Implementadas

No primeiro ciclo de melhoria de processos definiu-se o processo padrão da organização. Tal processo possui as fases do ciclo de vida de desenvolvimento de sistemas, cada fase deste ciclo de vida deve ser aprimorado pelos desenvolvedores através da introdução do modelo PSP.

O PSP fornece ao engenheiro de software os seguintes benefícios:

- Uma melhor percepção de seus talentos e habilidades;
- Um estímulo para um fluxo quase ilimitado de idéias de melhoria;
- Um “framework” para a melhoria pessoal;

- Um maior grau de controle do seu trabalho;
- O sentimento de orgulho e realização pessoal;
- Uma melhor eficiência no trabalho em equipe;
- A convicção de fazer o trabalho da maneira que ele sabe que deveria fazer.

A adoção do PSP fornecerá ao engenheiro de software a habilidade de estruturar suas tarefas em uma escala menor, a desenvolver um “framework” para medir estas tarefas e um aprimoramento na melhoria de processos, já que tal tarefa foi introduzida no primeiro ciclo, porém a nível organizacional. Com o modelo PSP, os processos definidos no primeiro ciclo serão medidos pelos desenvolvedores, obtendo-se o esforço gasto em cada fase do desenvolvimento.

O modelo PSP pode ser introduzido através de um curso de 2 semanas que deve ser ministrado aos desenvolvedores em duas etapas. Na primeira etapa devem ser apresentados os fundamentos do PSP e as técnicas de obtenção dos dados de tempo de implementação e tipos de defeito padrão. Após uma semana de curso sobre estas técnicas os desenvolvedores devem aplicar tais conceitos durante uma semana obtendo-se os dados sobre o seu processo individual de desenvolvimento e efetuar o planejamento pessoal do processo.

Na terceira semana a segunda etapa do curso deve apresentar aos desenvolvedores como estimar o tempo de implementação, elaborar um relatório de testes, elaborar revisões de *design* e codificação e o desenvolvimento cíclico de melhoria de processo. Tais métodos possibilitarão efetuar o gerenciamento da qualidade pessoal e a melhoria contínua do processo pessoal de desenvolvimento. Após o final do curso os desenvolvedores devem ser capazes de detectar os defeitos nas fases iniciais do processo de desenvolvimento, através dos procedimentos de revisão, diminuindo-se o alto custo de retrabalho devido a correção de defeitos que são detectados somente em testes de sistema, ao final do ciclo de vida de desenvolvimento.

Como ilustrado na Figura 17 na página 43, durante o curso de PSP são executados uma seqüência de 10 programas (pequenos softwares com aproximadamente 300 linhas de código), os dados

apresentados na Figura 23, na Figura 24 e na Figura 25 são resultados do impacto do PSP no desempenho de 298 engenheiros de software [12].

Talvez a mais impressionante mudança com o PSP seja na maneira como os engenheiros de software gastam seu tempo. No PSP0 eles gastavam menos tempo projetando seus programas do que com outras tarefas. Gastavam mais tempo compilando do que projetando. Ao final do curso, eles gastavam mais esforço projetando do que qualquer outra atividade técnica. Este resultado é ilustrado na Figura 23.

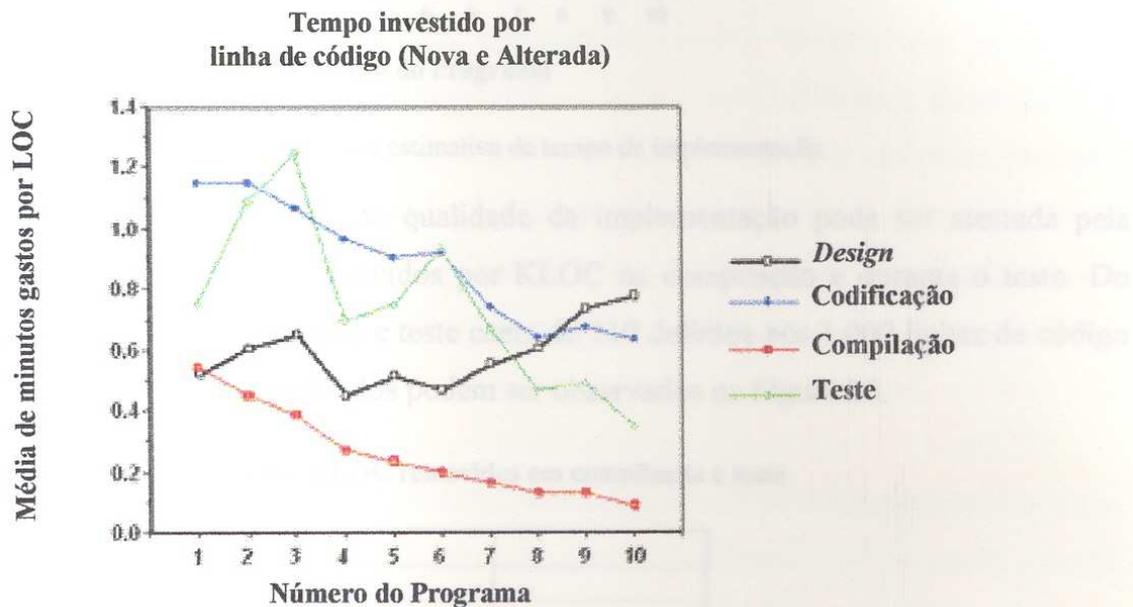


Figura 23 – Evolução do tempo médio em cada fase do desenvolvimento através do PSP.

Através do PSP a média de erros em estimativa de tamanho de software é reduzida de 63% com o PSP0 para 40% com o PSP2 e PSP3, devido a adoção do método PROBE no PSP1. Para estimativa de tempo o erro é reduzido de 55% no PSP0 para 27% no PSP3, esta melhora pode ser verificada na Figura 24.

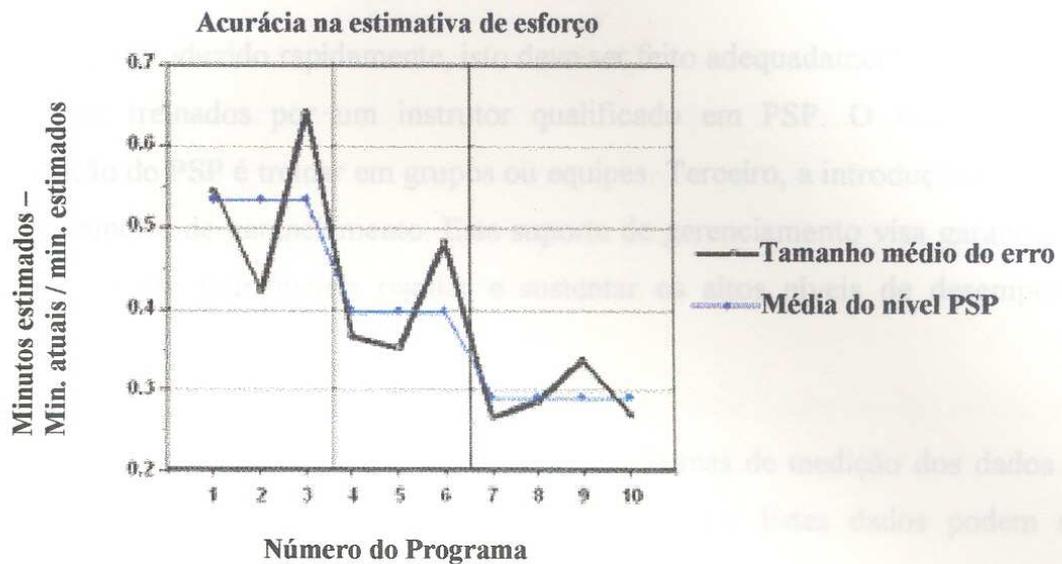


Figura 24 – Melhora na estimativa de tempo de implementação.

Os resultados obtidos com a melhora na qualidade da implementação pode ser atestada pela diminuição do número de defeitos removidos por KLOC na compilação e durante o teste. Do PSP0 ao PSP3 os defeitos de compilação e teste caem de 110 defeitos por 1.000 linhas de código para 20 defeitos por KLOC. Tais resultados podem ser observados na Figura 25.

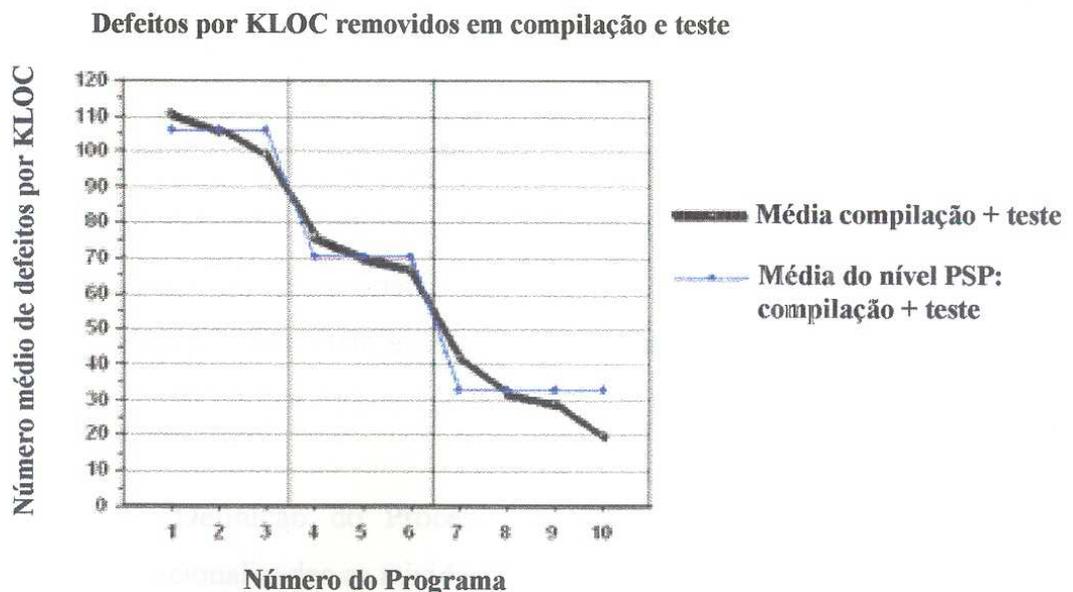


Figura 25 – Resultados obtidos quanto a melhora na qualidade da implementação

Mesmo com um enfoque no planejamento e na qualidade do desempenho a produtividade em termos de linhas de código é mais ou menos constante, sendo de 24 LOC/hora no PSP0, caindo para 23 LOC/hora no PSP1 e aumentando para 26 LOC/hora no PSP2 e PSP3.

Embora o PSP possa ser introduzido rapidamente, isto deve ser feito adequadamente. Primeiro, os engenheiros devem ser treinados por um instrutor qualificado em PSP. O segundo passo importante na introdução do PSP é treinar em grupos ou equipes. Terceiro, a introdução eficaz do PSP requer um forte suporte de gerenciamento. Este suporte de gerenciamento visa garantir aos engenheiros de software um treinamento regular e sustentar os altos níveis de desempenho pessoal adquiridos com o PSP [26].

A partir da introdução do PSP a organização deve introduzir formas de medição dos dados de desempenho em cada fase do ciclo de vida de desenvolvimento. Estes dados podem ser levantados, por exemplo, durante as fases de revisão e teste registrando-se a quantidade de defeitos encontrados e o tempo gasto em cada fase, a atividade 9 da área chave de processo Engenharia de Produto de Software estabelece que deve ser definida uma prática de coleta e análise dos dados de defeitos identificados nas fases de revisão pelos pares – a ser estabelecida neste ciclo - e testes. Estes dados serão muito úteis para a implementação do gerenciamento qualitativo a ser implementado no terceiro ciclo de melhoria.

Juntamente com a introdução do PSP a organização deve implementar as práticas chave das áreas chave do processo do nível 3 do CMM. A organização deve implementar neste segundo ciclo de melhoria os processos de *Peer Reviews*, Programa de Treinamento, Gerenciamento de Software Integrado, Coordenação Intergrupo e Foco no Processo da Organização [37]. As áreas chave do processo Definição do Processo Organizacional e Engenharia de Produto de Software foram implementadas parcialmente no primeiro ciclo e devem ser plenamente atendidos neste segundo ciclo.

A área chave do processo “Definição do Processo Organizacional” já foi implementada parcialmente, tendo sido institucionalizadas as atividades 1 a 3. A atividade 5 define a criação de um banco de dados de processo que deve conter e tornar disponível justamente àqueles dados como defeitos encontrados, tempo de implementação e outros dados importantes para o gerenciamento e controle dos projetos, tais como, estimativas e dados atuais de esforço e custo, medidas de confiabilidade, cobertura e eficiência dos testes, etc.

O processo padrão da organização deve ser ajustado às características e procedimentos do ciclo de vida de cada projeto. Este ajuste deve seguir um critério estabelecido atendendo a atividade 4 da área de processo “Definição do Processo Organizacional”. Como previsto na atividade 6 desta mesma área de processo toda a documentação relacionada ao processo, incluindo-se os processos ajustados ao projeto devem ser mantidas em uma biblioteca.

Nas atividades de *peer review* pode-se adotar o mesmo *checklist* adotado para revisão de *design* e código definidos no primeiro ciclo, porém tais revisões aqui devem ser feitas por outros desenvolvedores. As atividades de *peer review* devem ser planejadas e seguir um procedimento documentado, por exemplo através da utilização dos *checklists*. Os dados de defeitos encontrados devem ser armazenados no banco de dados de processo. Este procedimento satisfaz a atividade 9 da área de processo chave “Engenharia de Produto de Software” que pode ser implementada somente após a adoção do *peer review*.

O Programa de Treinamento visa comprometer a organização com o planejamento das atividades de treinamento e o fornecimento das habilidades e conhecimentos necessários para o desempenho das funções dentro da organização.

A atividade 10 da área chave de processo Engenharia de Produto de Software pode ser implementada neste segundo ciclo de melhoria pois ela depende da área chave de processo Planejamento do Projeto de Software e da definição do processo padrão da organização, ambos institucionalizados no primeiro ciclo de melhoria. A atividade 10 visa manter a consistência entre os produtos de trabalho, tais como planos de projetos, descrições dos processos, requisitos alocados, requisitos de software, *design*, código, planos de teste e procedimentos de teste.

A Coordenação Intergrupo tem como metas gerenciar e planejar as interações entre os grupos de desenvolvimento para garantir a qualidade e a integridade do sistema inteiro, tais atividades estão descritas nas atividades 2 a 7 [38]. A atividade 1 desta área chave do processo já foi implementado no primeiro ciclo de melhoria através do CUS.3 - “Identificar as Necessidades do Cliente” do modelo SPICE [44].

O propósito do Gerenciamento de Software Integrado é integrar as atividades de engenharia de software e de gerenciamento em um processo de software coerente e definido que é ajustado do processo de software padrão da organização. Todas as atividades e tarefas do processo de software padrão da organização devem ser ajustadas para o gerenciamento do desenvolvimento do projeto de software.

O estabelecimento do Foco no Processo da Organização permite que as atividades de desenvolvimento e melhoria do processo de software sejam coordenadas através da organização. Os pontos fortes e fracos dos processos de software são identificados e relacionados ao processo padrão. As atividades de desenvolvimento e melhoria do processo a nível organizacional são planejadas e coordenadas.

A área chave do processo Gerenciamento de Mudança de Tecnologia deve ter sua implementação concluída. As atividades 2 a 6 já foram institucionalizadas no primeiro ciclo de melhoria [38]. A organização deve desenvolver e manter um plano para o gerenciamento de mudança de tecnologia, novas tecnologias devem ser incorporadas dentro do processo de software padrão da organização e dos processos ajustados para os projetos de acordo com um procedimento documentado. Estas atividades estão descritas nas atividades 1, 7 e 8 desta área chave do processo.

Neste segundo ciclo de melhoria devem ser implementadas as práticas do modelo SPICE que tratam do relacionamento com o cliente. Tais práticas são[44]:

- CUS.5 – “Empacotar, Entregar e Instalar o Software”: Nesta categoria de processos devem ser identificados os requisitos para instalação e preparados os locais para a instalação. O software deve ser rotulado identificando-se suas novidades e suas alterações, instalado no cliente e procedimentos devem ser fornecidos para a sua utilização.
- CUS.7 – “Fornecer Serviços ao Cliente”: Nesta categoria de processos devem ser estabelecidos e mantidos procedimentos para o treinamento do cliente, o suporte ao produto, a monitoração do desempenho do software e a instalação de atualizações do produto.

- CUS.8 – “Avaliar a Satisfação do Cliente”: Nesta categoria de processos devem ser avaliadas o nível de satisfação do cliente, efetuada a comparação com competidores e a comunicação dos dados da satisfação do cliente através da organização.

Na melhoria do ambiente organizacional deve ser implementada as práticas definidas na ORG.5 – “Facilitar o Reuso” do modelo SPICE. O propósito deste processo é maximizar o reuso dos componentes existentes do sistema e do software. Para implementar as práticas de reuso deve-se determinar uma estratégia organizacional de reuso, identificar os componentes reutilizáveis, desenvolver componentes reutilizáveis, estabelecer uma biblioteca de reuso, certificar os componentes reutilizáveis, integrar o reuso dentro do ciclo de vida de desenvolvimento e propagar a mudança cuidadosamente – avaliando o impacto da mudança.

A Tabela 12 apresenta a quantidade de práticas/atividades implementadas neste segundo ciclo de melhoria de processos conforme apresentado na Tabela 10, perfazendo um total de 66.

Modelos	No. de Atividades
CMM Nível 3	38
CMM Nível 5	3
SPICE CUS	14
PSP	11
Total:	66

Tabela 12 – Quantidade de práticas/atividades implementadas neste segundo ciclo.

#### 4.4.2.2 Impacto no Desempenho da Organização

A partir de um processo plenamente definido e da introdução do PSP o tempo de desenvolvimento da organização deve diminuir significativamente. O PSP permite obter uma melhoria significativa também na estimativa de tempo e tamanho dos produtos. Esta melhora permite que a organização faça um plano de projeto mais realista.

A introdução de processos de *peer review* antecipa a detecção de defeitos das fases de teste para as fases iniciais do projeto, permitindo que a implementação ocorra corretamente já da primeira

vez e, além disso, a habilitação do reuso de produtos anteriores evita o retrabalho. Isto é fundamental para organizações que trabalham com projetos de alto grau tecnológico em que os produtos são avaliados pelo *time-to-money*, onde o produto deve ser levado ao mercado rapidamente para se antecipar o retorno do investimento.

Além do ganho obtido com a diminuição no tempo de desenvolvimento a organização melhora significativamente a relação com o cliente, através da institucionalização de práticas de interface entre a organização e o cliente.

Em relação ao segundo ciclo de melhoria a organização deve obter uma melhor relação entre a estimativa de custo e tempo de desenvolvimento e o efetivamente realizado. Deve, também, diminuir a variância nesta estimativa. A qualidade do produto deve melhorar pela introdução das atividades de *peer review*, garantindo uma menor quantidade de defeitos não verificados durante as fases iniciais do ciclo de vida de desenvolvimento.

#### 4.4.2.3 Questionário Integrado para o Segundo Ciclo

O questionário a seguir serve de *checklist* para a verificação da implementação das atividades do segundo ciclo de melhoria segundo o método integrado aqui proposto. Para que o segundo ciclo esteja completo as questões devem ser respondidas afirmativamente e evidências concretas da sua execução na organização devem ser demonstradas.

1. As atividades para o desenvolvimento e a melhoria dos processos de software da organização e dos projetos são coordenados através da organização (p.ex. via um grupo de processo de engenharia de software)?
2. O processo de software da organização é avaliado periodicamente?
3. A organização segue um plano documentado para o desenvolvimento e melhoria de seus processos de software?
4. O gerente sênior é responsável pelas atividades de desenvolvimento e melhoria do processo de software da organização (p.ex. pelo estabelecimento de planos de longo prazo e pelo comprometimento de recursos e financiamento)?

5. Um ou mais indivíduos tem responsabilidade por tempo integral ou parcial pelas atividades do processo de software da organização (p.ex. o grupo de processo de engenharia de software)?
6. Medições são utilizadas para determinar o status das atividades executadas para desenvolver e melhorar o processo de software da organização (p.ex. esforço gasto na avaliação e melhoria do processo de software)?
7. As atividades executadas pelo desenvolvimento e melhoria do processo de software são revisados periodicamente pelo gerente sênior?
8. A organização coleta, revisa e torna disponível as informações relacionadas ao uso do processo de software padrão da organização (p.ex. dados estimados e atuais do tamanho, esforço e custo do software; dados de produtividade; e medições de qualidade)?
9. Medições são utilizadas para determinar o status das atividades executadas para definir e manter o processo de software padrão da organização (p.ex. status dos marcos do cronograma e o custo das atividades da definição do processo)?
10. As atividades de treinamento são planejadas?
11. Treinamento é fornecido para o desenvolvimento das habilidades e do conhecimento necessário para executar os papéis de gerenciamento e técnico?
12. Membros do grupo de engenharia de software e outros grupos relacionados recebem o treinamento necessário para desempenhar seus papéis?
13. A organização segue uma política organizacional escrita para atingir suas necessidades de treinamento?
14. Recursos adequados são fornecidos para implementar o programa de treinamento da organização (p.ex. financiamento, ferramentas de software, instalações apropriadas para treinamento)?
15. Medições são utilizadas para determinar a qualidade do programa de treinamento?
16. As atividades do programa de treinamento são revisadas pelo gerente sênior periodicamente?
17. O processo de software definido pelo projeto foi desenvolvido pelo ajuste do processo de software padrão da organização?
18. O projeto planejado e gerenciado está de acordo com o processo de software definido pelo projeto?

19. Há uma política organizacional requerendo que o projeto de software seja planejado e gerenciado utilizando-se o processo de software padrão da organização?
20. Treinamento é fornecido para os indivíduos com a tarefa de ajustar o processo de software padrão da organização para definir um processo de software para um novo projeto?
21. Medições são utilizadas para determinar a eficiência das atividades do gerenciamento integrado de software (p.ex. frequência, causas e magnitude dos esforços de replanejamento)?
22. Atividades e produtos de trabalho são utilizadas para gerenciar o projeto de software sujeitas à revisão e auditoria de Garantia de Qualidade de Software?
23. A consistência é mantida através dos produtos de trabalho de software (p.ex. a documentação é mantida rastreada dos requisitos alocados através dos requisitos de software, *design*, código e casos de teste)?
24. Medições são utilizadas para determinar a funcionalidade e qualidade dos produtos de software (p.ex. números, tipos e severidade dos defeitos identificados)?
25. Os grupos de engenharia concordam com os comprometerimentos como representado no plano de projeto global?
26. Os grupos de engenharia identificam, rastreiam e resolvem questões intergrupo (p.ex. cronogramas incompatíveis, riscos técnicos ou problemas a nível de sistema)?
27. Há uma política organizacional escrita que guia o estabelecimento de equipes de engenharia interdisciplinares?
28. Ferramentas de suporte utilizados pelos diferentes grupos de engenharia habilitam a comunicação e a coordenação efetiva?
29. Medições são utilizadas para determinar o status das atividades de coordenação intergrupo (p.ex. esforço gasto pelo grupo de engenharia de software para suportar outros grupos)?
30. As atividades de coordenação intergrupo são revisadas pelo gerente de projeto periodicamente ou por eventos dirigidos?
31. As revisões pelos pares são revisadas?
32. Ações associadas com defeitos que são identificadas durante as revisões pelos pares são rastreadas até que elas sejam resolvidas?
33. O projeto segue uma política organizacional escrita para executar as revisões pelos pares?
34. Participantes de revisão pelos pares recebem o treinamento requerido para desempenhar seus papéis?

35. Medições são utilizadas para determinar o status das atividades de revisão pelos pares (p.ex. número de revisões pelos pares executadas, esforço gasto em revisão pelos pares e número de produtos de trabalho revisado comparado ao planejado)?
36. Atividades de revisão pelos pares e produtos de trabalho estão sujeitas à revisão e auditoria de Garantia da Qualidade de Software (p.ex. revisões planejadas são conduzidas e ações contínuas são rastreadas)?
37. A organização segue um plano para o gerenciamento de mudança de tecnologia?
38. A organização segue um procedimento documentado por incorporação de novas tecnologias dentro do processo de software padrão da organização?
39. Dados de processo existem para socorrer na seleção de nova tecnologia?
40. Medições são utilizadas para determinar o status das atividades da organização para o gerenciamento de mudança de tecnologia (p.ex. a eficiência da implementação da mudança de tecnologia)?
41. Há um procedimento documentado para identificar os requisitos de instalação do software no cliente (p.ex. empacotamento, tipo de media, documentação, *copyright* e licenças)?
42. Há um rótulo que identifique o software e a documentação que o acompanha (p.ex. identificação do conteúdo e das características adicionadas ou modificadas)?
43. Há a verificação da correta entrega do software ao cliente?
44. Há um processo organizacional escrito para fornecer serviços ao cliente (p.ex. treinamento, suporte, monitoração de desempenho, atualização do produto)?
45. Há um processo organizacional escrito para avaliar a satisfação do cliente (p.ex. determinar o nível de satisfação do cliente, compará-lo com os competidores e comunicá-lo a toda a organização)?
46. Há uma estratégia organizacional documentada de reuso de componentes (p.ex. identificação de componentes reutilizáveis, desenvolver componentes reutilizáveis e estabelecer uma biblioteca de reuso)?
47. O processo padrão da organização integra o reuso no seu ciclo de vida?
48. Os engenheiros de software da organização seguem um padrão de design e codificação definido?
49. Há a medição e a estimativa do tamanho do software pelos engenheiros de software?

50. Há o registro do tempo de desenvolvimento e padrão de defeitos pelos engenheiros de software?

51. Os engenheiros de software da organização possuem um procedimento documentado para melhorar o processo de software individual?

### 4.4.3 Terceiro Ciclo de Melhoria

Após a adoção do PSP no ciclo anterior, a organização deve focar agora a melhoria do processo de software das equipes de projeto. Para tal, o TSP fornece um método para guiar as equipes de desenvolvimento de software em como trabalhar como membros de uma equipe. O objetivo principal do TSP é mostrar ao engenheiros treinados em PSP como unir os diferentes processos individuais de forma a permitir ao projeto decorrer baseado em uma equipe homogênea. Além disso, devem ser adotados métodos de gerenciamento estatísticos para o gerenciamento dos processos, o gerenciamento da qualidade do produto de software e o suporte contínuo ao cliente.

#### 4.4.3.1 Práticas a serem Implementadas

O TSP mostra às equipes como planejar seu trabalho e fornecer a elas os dados necessários para defender seus planos. Assim, as equipes treinadas em TSP serão capazes de convencer a gerência que seus planos são agressivos mas atingíveis. Com o TSP, sob uma severa pressão de cronograma, os engenheiros de software geralmente se concentrarão naquelas tarefas em que eles sabem como fazer, mesmo que outras sejam mais importantes. Quando é fornecida uma série de tarefas claramente estruturadas e um processo definido para conduzi-los, os engenheiros geralmente seguirão o processo. Eles assumem que sabem como seguir um processo definido.

A partir dos dados obtidos pelos engenheiros de software com a adoção do PSP, o TSP permite a obtenção do perfil de qualidade de um dado projeto (como apresentado no capítulo 3.4). O exemplo a seguir apresenta como este perfil de qualidade pode ser obtido:

- Considere-se os seguintes dados levantados por um projeto qualquer:

Linhas de Código(LOC): 121.000

Tempo de *Design*: 4.970 horas-homem

Tempo de Revisão de *Design*: 1.450 horas-homem

Tempo de Codificação: 5.628 horas-homem

Tempo de Revisão de Código: 2.550 horas-homem

Defeitos na Compilação: 1.626 defeitos

Defeitos no Teste de Unidade: 502 defeitos

Cálculo das dimensões do perfil de qualidade:

Tempo de *Design*/Tempo de Codificação =  $0.88 < 1.0$ ,

então o perfil do *Design* =  $0.88/1.0 = 0.88$

Tempo de Revisão de *Design*/Tempo de *Design* =  $0.29 < 0.5$ ,

então o perfil da Revisão de *Design* =  $0.29/0.5 = 0.58$

Tempo de Revisão de Código/Tempo de Codificação =  $0.45 < 0.5$ ,

então o perfil da Revisão de Código =  $0.45/0.5 = 0.9$

Defeitos na Compilação/Linhas de Código(KLOC) =  $13.44 > 10.0$ ,

então o perfil da Compilação =  $2*10.0/(13.44+10) = 0.85$

Defeitos no Teste de Unidade/Linhas de Código(KLOC) =  $4.15 < 5.0$ ,

então o perfil do Teste de Unidade =  $1.0$

A Figura 26 ilustra o perfil de qualidade do exemplo apresentado. Para um projeto com este perfil, embora o tempo de revisão do *design* seja menor do que o considerado ideal, a codificação e a revisão de código foi muito eficaz, obtendo-se uma melhora considerável na quantidade de defeitos encontrados na compilação e no teste de unidade. Há poucos defeitos a serem detectados no teste de integração e no teste de sistema, devido a baixa densidade de defeitos na compilação e no teste de unidade.

Com os dados do TSP, os engenheiros de software podem determinar quais componentes são os que mais comumente tem defeitos antes de eles iniciarem o teste de integração e sistema. A partir desta conclusão a gerência de projeto pode diminuir substancialmente o tempo a ser despendido com os testes de integração e sistema e antecipar a liberação de um produto com maior qualidade.

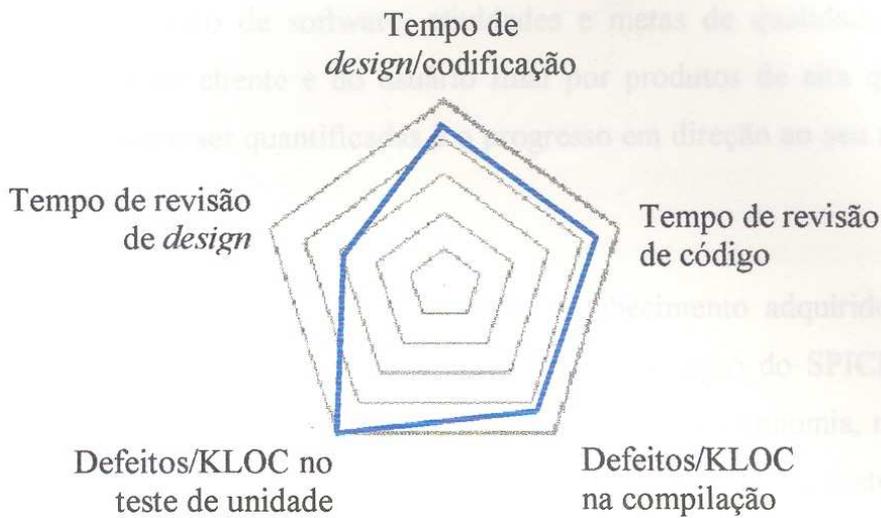


Figura 26 – Perfil de qualidade do projeto analisado.

O TSP permite que a organização se prepare para a adoção do controle estatístico do processo de software. A introdução do TSP irá facilitar a implementação das áreas chave do processo do nível 4 do CMM, que são o Gerenciamento Quantitativo do Processo e o Gerenciamento da Qualidade de Software.

O propósito do Gerenciamento Quantitativo do Processo é controlar o desempenho do processo do projeto de software quantitativamente. O desempenho do processo de software representa os resultados adquiridos pela utilização do processo plenamente definido no ciclo anterior. Para o controle estatístico do processo deve-se obter medições do desempenho dos processos, como efetuado pelo TSP, análise destas medições e fazer ajustes para manter o desempenho dos processos dentro de limites aceitáveis. A organização deve coletar os dados de desempenho do processo dos projetos e utilizar estes dados para caracterizar a capacidade do processo padrão da organização. Assim, a capacidade do processo é conhecida em termos quantitativos. A coleta dos dados, sua análise e as medidas de ajuste devem ser um procedimento documentado e regularmente revisado.

O propósito do Gerenciamento da Qualidade de Software é desenvolver um entendimento quantitativo da qualidade dos produtos e atingir metas de qualidade específicas. O Gerenciamento da Qualidade de Software envolve a definição de metas de qualidade para os produtos de

software, o estabelecimento de planos para atingir estas metas e a monitoração e ajuste dos planos, produtos de trabalho de software, atividades e metas de qualidade para satisfazer as necessidades e desejos do cliente e do usuário final por produtos de alta qualidade. Todas as metas estabelecidas devem ser quantificadas e o progresso em direção ao seu atendimento devem ser gerenciadas.

Para melhor disseminar, em toda a organização, o conhecimento adquirido pelas equipes de projeto com o TSP, deve-se implementar a categoria de processo do SPICE PRO.3 – Formar equipes de projeto. O propósito deste processo é definir, fornecer autonomia, manter a interação e gerenciar os resultados das equipes de projeto da organização. Através deste processo todas as equipes de projeto da organização devem ter definida suas regras de operação, os conhecimentos e habilidades requeridas, o entendimento de seu trabalho, mecanismos apropriados ou recursos para a comunicação e o trabalho e identificar, rastrear e resolver conflitos que afetam o seu progresso.

Para garantir que a melhoria da qualidade do produto de software seja repassada ao cliente a organização deve implementar a categoria de processo CUS.6 – “Suportar a Operação de Software” do SPICE. O propósito deste processo é suportar a operação correta e eficaz do software durante a sua operação intencionada. Este processo deve identificar os riscos operacionais, testar o desempenho operacional do software, operar o software no ambiente intencionado, resolver problemas operacionais, tratar as requisições do usuário, documentar os trabalhos temporários necessários para manter o sistema operando até que uma solução permanente para um problema possa ser achada e monitorar a capacidade e o serviço do sistema.

A Tabela 13 apresenta a quantidade de práticas/atividades implementadas neste terceiro ciclo de melhoria de processos conforme apresentado na Tabela 10, perfazendo um total de 46.

Modelos	No. de Atividades
CMM Nível 4	12
SPICE PRO	4
SPICE CUS	7
TSP	23
Total:	46

**Tabela 13** – Quantidade de práticas/atividades implementadas neste terceiro ciclo.

#### 4.4.3.2 Impacto no Desempenho da Organização

Ao final deste ciclo de melhoria a organização fixa metas de qualidade quantitativas para os produtos e processos de software. A produtividade e a qualidade são medidas nas atividades do processo de software através de todos os projetos como parte de um programa organizacional de medição. Um banco de dados de processo de software a nível organizacional é utilizado para coletar e analisar os dados disponíveis dos processos de software definidos. Estas medições são utilizadas para efetuar o controle estatístico dos processos de software. Variações significativas no desempenho do processo podem ser distinguidas de variações aleatórias (ruído), ele é previsível porque o processo é medido e opera dentro de limites mensuráveis. Isto permite a uma organização prever tendências no processo e na qualidade do produto dentro de contornos quantitativos. Quando limites estabelecidos são excedidos, uma ação é tomada para corrigir a situação. Os produtos de software são de qualidade previsivelmente alta.

As equipes de projeto obtém, através do TSP, a habilidade e o conhecimento necessários para integrar o processo de software individual de cada membro do projeto. Através da análise dos dados levantados durante as etapas de desenvolvimento as equipes de projeto podem obter o perfil de qualidade do projeto e avaliar o tempo a ser despendido nas fases de teste do produto.

Neste ciclo de melhoria a organização também introduz uma melhor interface com o cliente ou usuário final dos seus produtos avaliando a utilização do produto no seu ambiente operacional e manter o funcionamento do produto quando problemas são detectados.

#### 4.4.3.3 Questionário Integrado para o Terceiro Ciclo

O questionário a seguir serve de *checklist* para a verificação da implementação das atividades do terceiro ciclo de melhoria segundo o método integrado aqui proposto. Para que o terceiro ciclo esteja completo as questões devem ser respondidas afirmativamente e evidências concretas da sua execução na organização devem ser demonstradas.

1. As equipes de projeto seguem um procedimento para integrar os diferentes processos de software individuais?
2. As atividades a serem efetuadas pelas equipes de projeto em cada fase são antes avaliadas em um *workshop*?
3. As equipes de projeto, quando sob severa pressão de cronograma, concentram-se nas tarefas em que elas sabem como fazer?
4. São calculados e analisados os perfis de qualidade dos projetos da organização?
5. O tempo despendido nos testes de integração e sistema são fixados de acordo com o perfil de qualidade do projeto?
6. O projeto segue um plano documentado para condução do gerenciamento quantitativo do processo?
7. O desempenho do processo de software definido para o projeto é quantitativamente controlado?
8. A capacidade do processo de software padrão da organização é conhecida em termos quantitativos?
9. O projeto segue uma política organizacional escrita para a medição e o controle do desempenho do processo de software definido para o projeto?
10. Recursos adequados são fornecidos para as atividades de gerenciamento de processo quantitativo?
11. Medições são utilizadas para determinar o status das atividades de gerenciamento quantitativo do processo?
12. As atividades do gerenciamento quantitativo do processo são revisadas pelo gerente de projeto periodicamente ou em eventos dirigidos?
13. As atividades de gerenciamento da qualidade de software são planejadas para o projeto?

14. O projeto utiliza metas mensuráveis e priorizadas para o gerenciamento da qualidade de seus produtos de software?
15. Medições de qualidade são comparadas às metas de qualidade dos produtos de software para determinar se as metas de qualidade estão satisfeitas?
16. O projeto segue uma política organizacional escrita para o gerenciamento da qualidade de software?
17. Membros do grupo de engenharia de software e outros grupos relacionados recebem treinamento requerido em gerenciamento de qualidade de software?
18. Medições são utilizadas para determinar o status das atividades de gerenciamento da qualidade de software?
19. As atividades executadas pelo gerenciamento da qualidade de software são revisadas pela gerência sênior periodicamente?
20. São definidas as funções, regras de operação e a estrutura das equipes de projeto?
21. As equipes de projeto tem autonomia para executar o seu trabalho?
22. Problemas entre as equipes de projeto são gerenciados através da identificação, rastreamento e resolução de questões que afetem o seu progresso?
23. Os riscos de operação do produto de software são identificados?
24. Testes de desempenho operacional são efetuados para todas as versões de software liberadas?
25. Problemas operacionais são identificados e resolvidos de acordo com o ambiente do usuário?
26. As soluções provisórias na resolução de problemas com o sistema são documentadas?
27. A capacidade e os serviços operacionais do sistema são monitoradas?

#### **4.4.4 Quarto Ciclo de Melhoria**

O objetivo do quarto ciclo de melhoria é habilitar a organização na implementação da melhoria contínua dos processos pela realimentação quantitativa do processo, a partir do gerenciamento quantitativo introduzido no ciclo anterior.

#### 4.4.4.1 Práticas a serem Implementadas

Para implantar a melhoria contínua dos processos a organização deve implementar as áreas chave do processo Prevenção de Defeitos e Gerenciamento da Mudança do Processo, ambos do nível 5 do CMM.

O propósito da Prevenção de Defeitos é identificar as causas de defeitos e prevenir-se de sua repetição. A Prevenção de Defeitos envolve a análise dos defeitos que foram encontrados no passado e tomar ações específicas para prevenir a ocorrência daqueles tipos de defeitos no futuro. Os defeitos podem ter sido identificados em outros projetos tanto quanto nos estágios iniciais ou tarefas do projeto corrente. As atividades de prevenção de defeitos são também um mecanismo para propagar as lições aprendidas entre os projetos. Através da identificação dos tipos de defeitos mais encontrados, as causas de origem dos defeitos e as implicações dos defeitos nas atividades futuras são determinadas. A prevenção de defeitos permite à organização agir pró-ativamente, detectando as causas de problemas e tomando ações para evitá-las.

O propósito do Gerenciamento da Mudança no Processo é melhorar continuamente os processos de software utilizados na organização com a intenção de melhorar a qualidade de software, aumentando a produtividade e diminuindo o tempo do ciclo de vida de desenvolvimento dos produtos.

O Gerenciamento da Mudança no Processo envolve a definição das metas de melhoria de processo e, com o patrocínio do gerente sênior da organização, pró-ativamente e sistematicamente identificando, avaliando e implementando melhorias no processo de software padrão da organização e nos processos de software definidos para os projetos de maneira contínua.

Treinamento e programas de incentivo são estabelecidos para habilitar e encorajar todos na organização a participar nas atividades de melhoria de processo. Oportunidades de melhoria são identificadas e avaliadas pelo potencial de retorno para a organização. As mudanças no processo são avaliadas através de uma aplicação piloto antes de serem incorporadas como uma prática normal. As atividades de melhoria contínua do processo são planejadas e há um procedimento

documentado para a manipulação das propostas de melhoria. Este procedimento deve habilitar a organização a introduzir novos modelos de melhoria de processos [38].

#### 4.4.4.2 Impacto no Desempenho da Organização

Neste ciclo de melhoria a organização inteira está enfocada na melhoria contínua do processo. A organização tem os meios para identificar pró ativamente as fraquezas e os pontos fortes do processo, com o objetivo de prevenir a ocorrência de defeitos. Os dados da execução do processo de software são utilizados na análise do desempenho do custo benefício das mudanças propostas para o processo de software da organização. Inovações que exploram as melhores práticas de engenharia de software são identificadas e transferidas para toda a organização.

As equipes de projeto de software analisam os defeitos para determinar suas causas. Os processos de software são avaliados para prevenir a recorrência de tipos conhecidos de defeitos, e lições aprendidas são disseminadas para outros projetos. A melhoria ocorre pelo avanço incremental no processo existente e pelas inovações utilizando novos métodos e modelos.

#### 4.4.4.3 Questionário Integrado para o Quarto Ciclo

O questionário a seguir serve de *checklist* para a verificação da implementação das atividades do quarto ciclo de melhoria segundo o método integrado aqui proposto. Para que o quarto ciclo esteja completo as questões devem ser respondidas afirmativamente e evidências concretas da sua execução na organização devem ser demonstradas.

1. As atividades de prevenção de defeitos são planejadas?
2. O projeto conduz reuniões para análise causal para identificar causas comuns de defeitos?
3. Uma vez identificadas, as causas comuns de defeitos são priorizadas e sistematicamente eliminadas?
4. O projeto segue uma política organizacional escrita para as atividades de prevenção de defeitos?

5. Membros do grupo de engenharia de software e outros grupos relacionados recebem o treinamento requerido para executar suas atividades de prevenção de defeitos?
6. Medições são utilizadas para determinar o status das atividades de prevenção de defeitos?
7. As atividades e produtos de trabalho para a prevenção de defeitos estão sujeitas a revisão e auditoria da Garantia da Qualidade de Software?
8. A organização segue um procedimento documentado para o desenvolvimento e a manutenção de planos de melhoria do processo de software?
9. As pessoas da organização participam das atividades de melhoria do processo de software?
10. As melhorias são feitas no processo de software padrão da organização e nos processos de software definidos para os projetos continuamente?
11. A organização segue uma política escrita para implementação das melhorias no processo de software?
12. Treinamento em melhoria do processo de software é requerido aos gerentes e ao corpo técnico?
13. Medições são efetuadas para determinar o status das atividades de melhoria do processo de software?
14. Esforços de melhoria do processo de software são revisados pelo gerente sênior periodicamente?
15. Novos modelos de melhoria de processos de software são avaliados através de um procedimento documentado?

#### **4.5 Resumo do Método Integrado**

O método integrado se caracteriza pela existência de 4 ciclos de melhoria de processos, cada um destes ciclos têm o objetivo de aumentar o nível de maturidade da organização e implementar a melhoria do processo de desenvolvimento de software dos indivíduos e das equipes de projeto.

No primeiro ciclo de melhoria deve ser implementada as atividades básicas de suporte aos projetos, tais como gerenciamento de projeto, gerenciamento de configuração, garantia da qualidade, etc. Devem ser também iniciado a definição do processo padrão da organização, identificando-se os processos chave a serem implementados pela organização. Tais atividades

garantem que o processo se repete de um projeto para outro. Neste primeiro ciclo são utilizados os modelos CMM, SPICE e PSP.

No segundo ciclo de melhoria deve ser implementado as atividades de definição das tarefas de engenharia de software e de suporte ao desenvolvimento propriamente dito. Além disso deve ser introduzido o PSP, a fim de aumentar a maturidade individual dos engenheiros de software da organização. Neste segundo ciclo de melhoria são utilizados os modelos CMM, SPICE e PSP.

No terceiro ciclo de melhoria deve ser implementado o gerenciamento qualitativo e quantitativo dos produtos e processos, baseando-se no histórico de métricas obtidos pela organização nos ciclos anteriores. Após a introdução do PSP no segundo ciclo de melhoria, deve ser adotado pela equipes de projeto da organização o TSP, fornecendo a estas a capacidade necessária para atuar com alto desempenho. Neste terceiro ciclo são utilizados os modelos CMM, SPICE e TSP.

O quarto ciclo de melhoria habilita a organização a antecipar a detecção de defeito e gerenciar a mudança de seus processos de forma contínua. Neste quarto ciclo é utilizado o modelo CMM.

No próximo capítulo será analisado um estudo de caso da implementação do primeiro ciclo de melhoria de processos em uma unidade de desenvolvimento de sistemas de telecomunicações da Siemens Ltda.

## 5 Estudo de Caso

Para verificar a viabilidade deste método de melhoria de processo e como ele pode ser implementado será apresentado um estudo de caso em uma unidade de negócios da Siemens Ltda. Em tal estudo de caso serão analisadas as motivações para se adotar um programa de melhoria de processos, as características do método de avaliação de processos utilizado, a estratégia de implementação das melhorias e os benefícios obtidos ao final do primeiro ciclo de melhoria de processos.

Na seção 5.1 será apresentada uma pequena história que descreve a evolução de um grupo de desenvolvimento de produtos e a importância de se focar o gerenciamento de processos. Tal história serviu de justificativa para se compreender a importância, pela alta direção da organização, da melhoria de processos.

Na seção 5.2 estão relacionados os aspectos que serviram de motivação para a adoção do método integrado de melhoria de processos.

A seção 5.3 apresenta a estratégia de introdução do método integrado de melhoria de processos proposto e como tal método foi implementado na organização. São analisados as dificuldades encontradas e as correções que foram necessárias para a implementação das atividades e dos procedimentos.

Na seção 5.4 são apresentados os processos introduzidos ao final do primeiro ciclo de melhoria. Tais processos estão descritos no *SyDeP – System Development Process Handbook*, que possui a definição das atividades e dos procedimentos seguidos pela organização.

E finalmente, na seção 5.5 são localizados os processos propostos pelo primeiro ciclo de melhoria de processos do método integrado no SyDeP, verificados os resultados obtidos com a introdução do método integrado de melhoria de processos na organização e dos esforços gastos com a adoção deste método.

## 5.1 O Processo de Engenharia

Para definir o que é um processo de engenharia, imagine uma companhia que não está dividida em departamentos e possui um grande número de funcionários de diferentes especialidades e com diferentes aptidões. Imagine que o primeiro desafio desta companhia seja construir algum produto que atenda as necessidades de um mercado específico, e tais necessidades sejam conhecidas por todos os funcionários. Se observarmos esta companhia, veremos que aparecerá um grande número de idéias e suposições sobre como o produto poderia ser. Depois de algum tempo de discussões e quando o movimento se tornar incontrolável, aparecerá um líder tentando organizar e disciplinar o grupo. Passado mais algum tempo, o grupo terá formado uma idéia mais ou menos comum sobre o produto a ser desenvolvido. Neste momento iniciará as discussões sobre como será desenvolvido o produto, culminando na separação do grupo em diferentes áreas de especialização e definindo o que cada área fará no desenvolvimento do produto. No decorrer das atividades serão aplicados diferentes métodos para se desenvolver diferentes atividades, o que provavelmente será discutido com o líder do grupo. Cada grupo, entretanto permanece “sob os olhares” do líder dividindo a atenção das tarefas entre os sub-líderes. Novas interfaces e necessidades serão identificadas durante os trabalhos e certamente serão discutidas com o líder para definir quem faz o que, ou para criar um grupo responsável pela nova atividade. Então rapidamente se concluirá o desenvolvimento do produto, o que ocorrerá antes do que se poderia imaginar. Por quê?

É simples: foco no processo. Desde que o grupo nunca tenha feito este tipo de trabalho antes, e ninguém do grupo sabia qual eram as suas atribuições e responsabilidades, não existem pré-conceitos sobre as responsabilidades e funções de cada um. Existe apenas um e único objetivo comum a todos da companhia: vencer o desafio. Por este motivo, esta companhia imediatamente focalizou-se no processo de desenvolvimento, concentrando, na figura do líder, o conhecimento das inter-relações e responsabilidades de cada um. O líder do grupo não apenas gerenciou o desenvolvimento do produto, mas serviu também como um sustentáculo para a formalização do processo de desenvolvimento. Não importa que as técnicas e métodos escolhidos pelo líder não tenham sido as melhores, importa é que todos “sabiam” quem devia fazer o que e como. É muito

provável, ao contrário do que seria óbvio, que na próxima vez o sucesso seja minado, já que os funcionários não concentrarão com o líder todo o conhecimento. A iniciativa de cada funcionário em executar suas funções, e dos outros em tentar otimizar o processo sozinho certamente criará lacunas e superposições de tarefas, tornando o processo incompleto e ineficiente.

Este conhecimento das tarefas e funções é chamado de processo. Obviamente que em grandes organizações ele não esteja apenas na mente dos líderes, mas colocado formalmente em documentos explicando a maneira de como fazer as coisas. Mas há organizações cujo processo faz parte apenas da chamada “cultura” da companhia, onde é comum ouvir “aqui as coisas são feitas desta maneira ...”. Estas organizações possuem totais condições de sucesso, entretanto deixam-na na mão (ou melhor, nas mentes) de poucos funcionários que se tornam os proprietários do processo e a qualquer momento eles podem se tornar indisponíveis e tornar o processo instável, pela natureza humana de querer sempre fazer melhor o que já foi feito.

A necessidade de se atuar nos processos de engenharia, como a forma mais clara e eficaz de se obter significativas melhorias na qualidade e principalmente no “time-to-market” no desenvolvimento de produtos, evidencia cada vez mais a obrigatoriedade das empresas em investirem na gerência dos seus processos.

As pequenas empresas geralmente apresentam excelentes resultados financeiros e excelentes produtos para o mercado, desde que bem administradas e com as pessoas certas nas atividades corretas. Por outro lado, em uma grande organização, mesmo que sua força tarefa seja de altíssimo nível, muitos esforços são neutralizados e a eficiência total é sensivelmente prejudicada. Cada líder “formal ou não” tende a direcionar os esforços individuais em uma determinada área de influência (que podem ir além ou ficar aquém do seu grupo), na direção das suas convicções. De forma simples e informal, poderia ser dito que: “O resultado dos esforços de uma organização é a soma vetorial dos esforços individuais de seus funcionários.” A Figura 27 ilustra estes esforços em pequenas e grandes organizações.

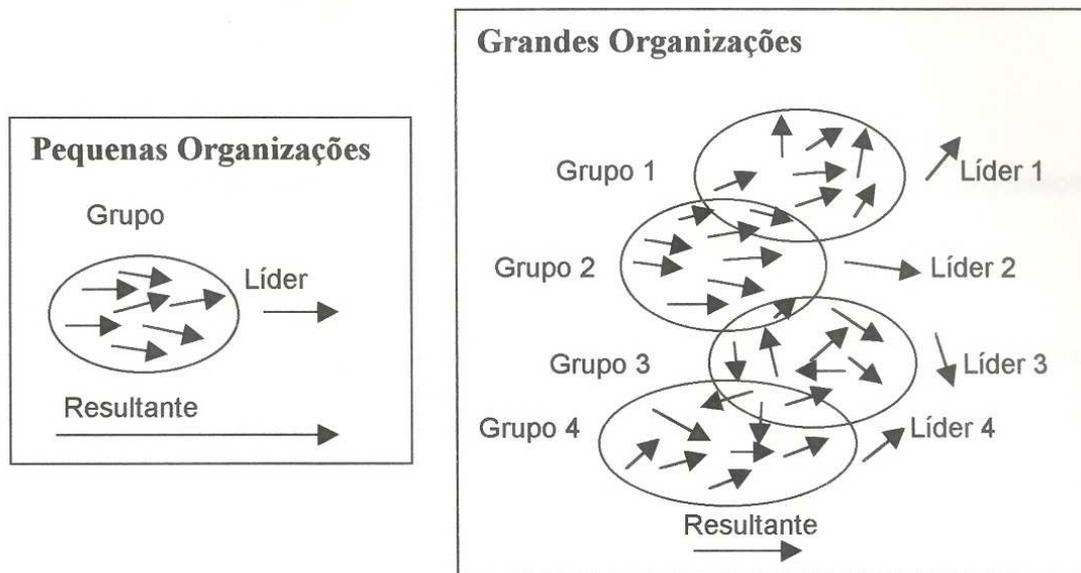


Figura 27 – Esforços em diferentes organizações

Assim sendo, a gerência do processo de engenharia se propõe a:

- Garantir o patrimônio maior da empresa – o conhecimento de como fazer – documentando os processos de engenharia praticados;
- Estabelecer pontos de medição e grandezas para viabilizar o controle do processo;
- Estabelecer os padrões de qualidade referentes ao processo de engenharia;
- Orientar as convicções das lideranças e dos indivíduos usuários e participantes do processo em questão;
- Viabilizar a utilização consciente e eficaz do processo;
- Avaliar sistematicamente e ajustar ou modificar o processo de engenharia para aumentar seu desempenho;
- Avaliar novas sistemáticas e métodos para modernizar o processo;
- Promover a introdução controlada de novas sistemáticas e métodos para o processo;
- Promover a sinergia de atividades e produtos como parte integrante do dia a dia do processo de engenharia;
- Identificar as oportunidades de melhoria do processo;
- Integrar as áreas da empresa em função dos objetivos dos processos.

Tendo-se em vista a função do gerenciamento do processo em engenharia verifica-se a necessidade de buscar a motivação para se adotar um programa de melhoria de processos.

## 5.2 Motivação

Por que uma unidade da Siemens deve se preocupar com o gerenciamento de processos e com a melhoria de processos?

A principal razão para se adotar um programa de melhoria de processos é atingir um ganho significativo de produtividade e obter o controle dos processos utilizados pela organização. Se a organização não possui o controle do processo ao nível adequado, isto se reflete em perda de produtividade e qualidade, atrasos de cronograma, aumento imprevisto nos custos, etc. Apesar da existência, na Siemens, de um processo formal, documentado e aplicado durante o projeto de introdução de novos produtos, este processo contudo, não atende plenamente as necessidades dos desenvolvedores. Este processo, chamado INP – Introdução de Novos Produtos - abrange todas as unidades funcionais da empresa, isto é, vendas, comercial, marketing, desenvolvimento, fábrica e serviços, permitindo o sincronismo entre estas diversas unidades. Porém, a unidade de desenvolvimento de produtos propriamente dita é vista como uma caixa-preta, ao nível da INP, onde tal unidade recebe e gera produtos para e de outras unidades. A fim de “abrir” esta caixa-preta e visualizar os processos e produtos gerados internamente, deve-se adotar um programa de gerenciamento de processos, que tenha como objetivo definir e buscar melhorias nestes processos que são executados de maneira informal.

Há que se obter, assim um processo padronizado que permita o rastreamento e o controle das etapas de desenvolvimento e que garanta os padrões de qualidade especificados em cada etapa do projeto. Tal processo, além de padronizado, deve ser seguido e melhorado continuamente por todos os responsáveis pelo desenvolvimento dos produtos dentro da organização.

A melhoria de processos já está sendo utilizada por um número grande de empresas [3]. Como exemplo pode-se citar alguns casos:

- A Raytheon, que dobrou sua produtividade e obteve um retorno sobre o investimento de 7.7 para 1.
- A Hughes Aircraft que computou uma razão de retorno de 5 para 1, baseado nas mudanças no seu índice de custo-desempenho.
- A Motorola GED que obteve um retorno sobre o investimento total de 7.77 para 1 quando calculado sobre a redução no número de defeitos por MAELOC – Mega Assembly-Equivalent Lines of Code [6].

### 5.3 A Implementação do Programa de Melhoria pela Siemens

A organização alvo desta implementação de melhoria de processo tem 70 desenvolvedores, sendo que 20 destes foram diretamente envolvidos com a melhoria de processo.

Após a obtenção do comprometimento da diretoria da unidade de negócio da organização que implementará um programa de melhoria, os seguintes passos foram iniciados para se efetuar um programa de melhoria de processos [33]:

- Efetuar uma avaliação do processo para identificar os pontos fortes e fracos;
- Criar um EPG – *Engineering Process Group*;
- Estruturar as recomendações seguindo os potenciais identificados;
- Construir grupos de atividades gerenciáveis;
- Definir metas gerenciáveis, realistas e motivantes para o programa de melhoria;
- Fornecer um suporte forte e um patrocínio visível pelo gerente;
- Iniciar o programa de melhoria através de uma reunião de “partida”.

A condução da avaliação do processo na Siemens seguiu a seqüência ilustrada na Figura 28, que é baseada na seqüência apresentada na Figura 7.

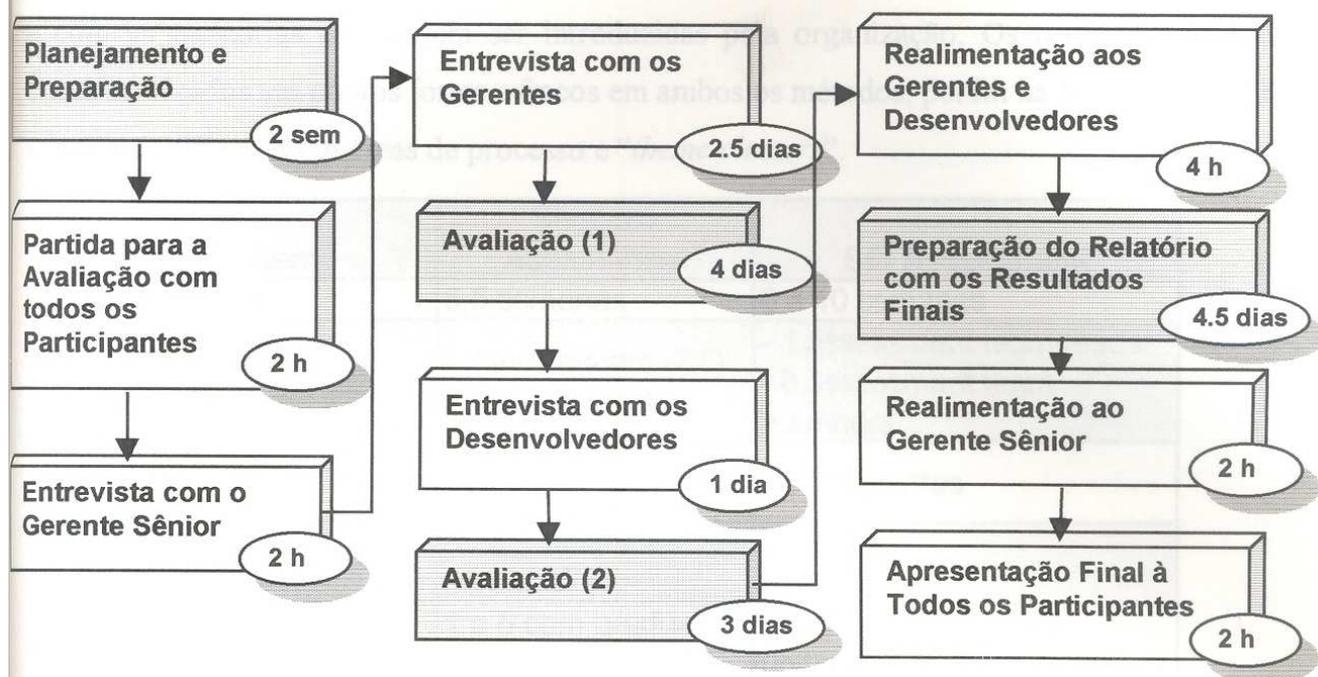


Figura 28 – Seqüência adotada na Siemens para a condução da avaliação do processo.

Como resultado da avaliação a organização adquire as seguintes informações:

- Um catálogo detalhado de observações e recomendações;
- Um sumário quantitativo dos resultados classificados em perfis de pontos fortes e fracos;
- O nível de maturidade da organização e de cada projeto avaliado;
- O catálogo de medidas de melhoria com prioridade, esforço e tempo para a implementação;
- O agrupamento das medidas em grupos de ação priorizados de acordo com o perfil da organização.

A Tabela 14 apresenta a comparação entre o método utilizado pela Siemens para avaliação do processo com o método utilizado pela SEI (CBA IPI, descrito na seção 2.2.2). O método de avaliação utilizado na Siemens foi desenvolvido na sua Central Técnica na Alemanha e é aplicado em todas as suas unidades de desenvolvimento através do mundo. A avaliação adotada pela Siemens possui uma duração menor que o proposto pela SEI; o número total de avaliadores é menor, pois exige-se apenas um coordenador local e entrevistas curtas com os gerentes e desenvolvedores. O questionário de maturidade possui uma maior quantidade de perguntas, permitindo maior granularidade na classificação das organizações avaliadas. Ao final da avaliação pelo método Siemens é produzido um catálogo de medidas com os achados verificados

e com as melhorias que devem ser introduzidas pela organização. Os resultados obtidos são contextualizados em pontos fortes e fracos em ambos os métodos, porém na SEI para cada KPA e no método Siemens em áreas de processo e “*theme clusters*”.

Ítem	Siemens Assessment	SEI Assessment
<b>Duração</b>	5,5 semanas	6 a 10 semanas
<b>Número de avaliadores</b>	- 2 avaliadores (ZT) - 1 coord. local	- 1 assessment team leader - 8 assessment team members
<b>Questionário de maturidade</b>	339 perguntas	124 perguntas
<b>Catálogo de medidas</b>	50 a 80 medidas para melhoria	Não há
<b>Classificação</b>	1 a 5 com graduação de 0,25	1 a 5 com graduação unitária
<b>Contextualização dos resultados</b>	Fortes e fracos da org. em áreas de processo e “ <i>theme clusters</i> ”	Fortes e fracos da org. em cada KPA

**Tabela 14** – Tabela comparativa entre o método de avaliação da Siemens e da SEI.

Após a avaliação do processo o projeto de melhoria deve efetuar as seguintes tarefas:

- Personificar as responsabilidades para a realização das medidas resultantes da avaliação do processo;
- Planejar a realização das medidas;
- Efetuar uma aplicação piloto para medidas complexas;
- Introdução das medidas de modo disseminado em todos os projetos;
- Rastrear o progresso dos projetos;
- Comunicar os sucessos verificados aos envolvidos com o programa de melhoria.

O programa de melhoria de processo deve ser encarado pela organização como um projeto a ser desenvolvido. A Figura 29 apresenta as 4 fases usualmente seguidas para a realização de um programa de melhoria:

- Planejamento do projeto;

- Definição das metas e das aplicações piloto;
- Pilotar os projetos selecionados;
- Disseminar os procedimentos estabelecidos que são introduzidos na aplicação piloto selecionada através de toda a organização.



- |   |   |   |   |
|---|---|---|---|
| <ul style="list-style-type: none"> <li>• Definir as metas do projeto de melhoria</li> <li>• Iniciar o projeto de melhoria</li> <li>• Estabelecer a equipe de melhoria de processo</li> <li>• Planejar os pacotes de trabalho em detalhes</li> </ul> | <ul style="list-style-type: none"> <li>• Mostrar o status quo</li> <li>• Modificar os procedimentos e torná-los disponíveis</li> <li>• Selecionar os projetos pilotos</li> <li>• Planejar as aplicações piloto</li> </ul> | <ul style="list-style-type: none"> <li>• Treinar</li> <li>• Realizar e instruir procedimentos em projetos pilotos</li> <li>• Medir os resultados, dar realimentação e otimizar os procedimentos</li> <li>• Preparar a disseminação</li> </ul> | <ul style="list-style-type: none"> <li>• Modificar a documentação do processo</li> <li>• Treinar</li> <li>• Realizar procedimentos otimizados nos departamentos</li> <li>• Medir os resultados e controlar o progresso</li> </ul> |
|---|---|---|---|

**Figura 29** – As 4 fases de um projeto de melhoria na Siemens.

O primeiro passo para implementar e implantar as medidas resultantes da avaliação foi criar um EPG – *Engineering Process Group* que é composto por um gerente de processo e 2 engenheiros de processo em tempo integral e 2 em tempo parcial. Foram criados 4 PITs – *Process Improvement Teams* compostos por desenvolvedores com reconhecida experiência e um componente do EPG para orientar as discussões de como implementar as medidas geradas pela avaliação. Para a aprovação das medidas propostas pelos PITs foi estabelecido o MSG – *Managing Steering Group* composto pelos membros da gerência da organização. A Figura 30 resume a seqüência de como implementar um programa de melhoria de processos e a responsabilidade dos PITs e do MSG.

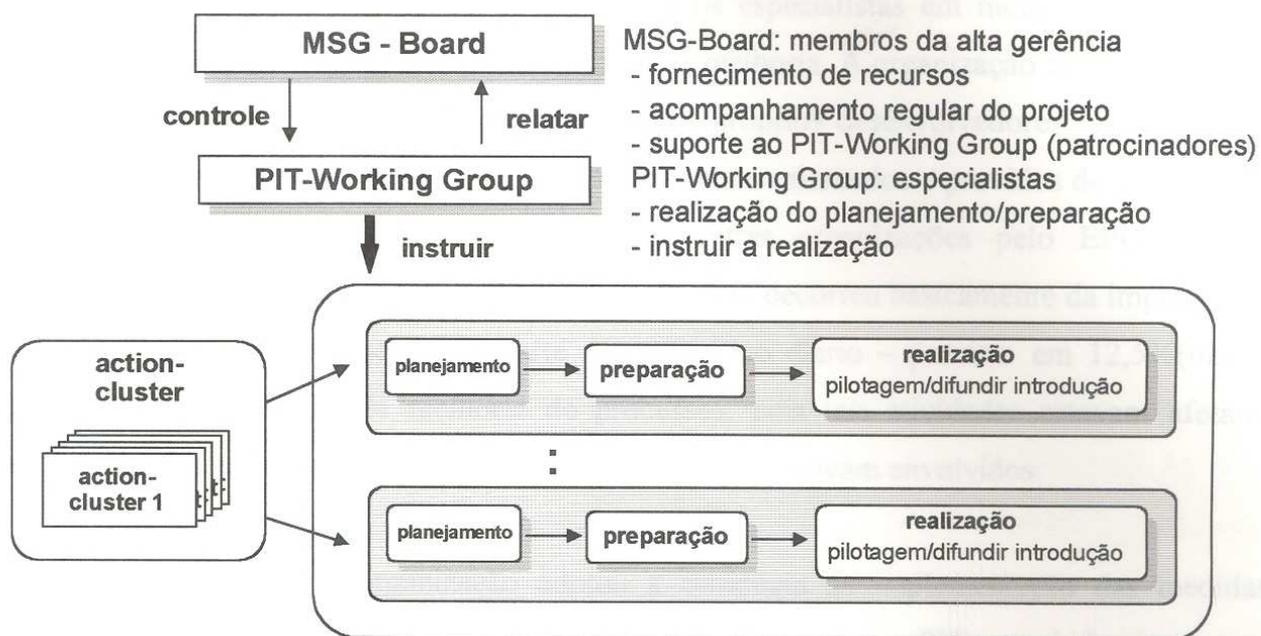


Figura 30 – Responsabilidades do PITs e do MSG.

Depois da definição de como as medidas serão implementadas, imediatamente elas serão colocadas em prática. Mas todos os desenvolvedores aceitarão mudar o seu modo de trabalhar? A resposta a esta questão está justamente na escolha dos membros dos PITs. Os membros escolhidos tem influência expressiva sobre os outros desenvolvedores, contribuindo com o EPG no convencimento de que tais mudanças são necessárias. Além disso, os próprios desenvolvedores criam os procedimentos de acordo com as análises de seu modo de trabalho. Assim todos os desenvolvedores tem consciência de que tais mudanças são importantes, tanto para a organização quanto para eles mesmos – através do aumento da produtividade individual – foi enfatizado o treinamento nos assuntos sobre processos.

Com o objetivo de desviar a resistência dos gerentes em ceder seus funcionários para estas atividades de implementação das melhorias, os gerentes foram envolvidos no programa de melhoria de processo pela criação de um comitê de análise e aprovação dos procedimentos – o MSG.

Através do estabelecimento de metas e de discussões preliminares entre os membros dos PITs, que são os desenvolvedores mais representativos de cada área de desenvolvimento da

organização, e os orientadores do EPG que são os especialistas em modelos de melhoria de processos, devem ser implementadas as medidas de melhoria. A organização adotou, inicialmente a estratégia de implementação das melhorias pelos próprios desenvolvedores que conhecem os problemas no processo de desenvolvimento e estes eram orientados e providos de informações de novos métodos e experiências realizadas em outras organizações pelo EPG. Porém tal procedimento se mostrou muito ineficaz. Esta ineficácia decorreu basicamente da impossibilidade dos membros dos PITs reservarem parte de seu tempo diário – previsto em 12,5%(uma hora diária) - para as tarefas de melhoria de processos, pois tais atividades estavam afetando o rendimento dos projetos aos quais estes desenvolvedores estavam envolvidos.

Em vista destes fatos a organização adotou a estratégia de implementação das medidas de melhoria pelos próprios membros do EPG e os membros dos PITs servindo apenas como assessores, consultores e avaliadores das implementações propostas. Todas as medidas de melhoria de processos, depois de aprovada pelos PITs e por todos os membros do EPG devem ser aprovadas pelos membros do MSG que é composto pelos gerentes da organização. Estes gerentes verificam se tais medidas de melhoria estão de acordo com os objetivos de negócio da organização.

O acompanhamento da introdução das medidas é feito através do rastreamento do progresso dos projetos e pela verificação de seu progresso e comparação com o histórico dos projetos anteriores (quando possível, como descrito anteriormente no impacto sobre o desempenho no item 4.3 na página 61).

#### **5.4 Processos Implementados**

O Processo de Desenvolvimento de Sistemas (SyDeP – *System Development Process*) [31][32] é definido como o processo padrão da organização, ele regulamenta os procedimentos, o fluxo, as interfaces e os produtos do desenvolvimento de sistemas da organização, mediante a determinação inequívoca e obrigatória de atividades e produtos que devem ser, respectivamente, realizadas e gerados pelo desenvolvimento. Além disso, deve estabelecer as funções operacionais e os recursos necessários para a execução dos processos definidos. A partir da definição dos

processos com o estabelecimento da arquitetura de alto nível a organização obtém a capacidade de entender as responsabilidades e os papéis dentro do processo de desenvolvimento e como as atividades interagem entre si com um nível apropriado de detalhes. São também estabelecidas e determinadas pelo SyDeP as atividades e os procedimentos de gerenciamento de projeto, gerenciamento de qualidade, gerenciamento de configuração, gerenciamento de processos, definição de produtos, engenharia e treinamento.

O SyDeP é de aplicação obrigatória a todos os projetos em desenvolvimento na organização e está organizado em categorias de processo que são compostas de áreas de processo. As áreas de processo identificam um grupo de atividades relacionadas - práticas-chave - que, quando executadas coletivamente, atingem uma série de metas consideradas importantes para a capacidade do processo de desenvolvimento de produtos. As áreas de processo atuam no ciclo de vida de desenvolvimento de produtos, fornecendo suporte, gerenciamento e controle das atividades de desenvolvimento.

Para a definição do SyDeP foram identificadas as áreas funcionais com as quais o SyDeP troca produtos. Estas áreas, para a organização estudada são:

⇒ **Contratante**

O SyDeP desenvolve produtos que sejam requeridos pelo departamento de produtos da Siemens Ltda., da ICN WN BS da Siemens AG (Alemanha) e da Siemens BCS (Estados Unidos).

⇒ **Treinamento**

O SyDeP é responsável pelo fornecimento das especificações técnicas dos produtos para que o treinamento desenvolva os manuais necessários para o produto.

⇒ **Fábrica**

O SyDeP deve fornecer à fábrica a documentação necessária para a manufatura do produto.

⇒ **Service**

Deve ser fornecido ao *service* as especificações técnicas do produto para preparação das atividades de *service* e interface para manutenção aos produtos.

⇒ **Suporte Técnico**

Para a aquisição de componentes não homologados pela fábrica, teste de qualificação de componentes e de produto e armazenagem de documentação técnica.

⇒ **Recursos Humanos (RH)**

Para garantir e manter o treinamento das habilidades pessoais deve ser mantido com o RH um sistema que gerencia um programa de treinamento para o SyDeP.

⇒ **Alta Direção**

Fornecer o Plano Estratégico de Negócios para a definição das ações futuras.

Para cada um destes parceiros deve ser identificado os produtos e as informações que são compartilhadas pelo desenvolvimento de produtos. Os produtos devem ser descritos e seu conteúdo deve ser claramente definido. Os produtos podem ser gerados pelo desenvolvimento de produtos ou recebidos por ele. Por exemplo: o desenvolvimento de produtos deve fornecer ao contratante um plano de projeto, que é composto da Especificação de Requisitos de Projeto, Especificação de Arquitetura do Produto, Plano de Realização, Plano de Gerenciamento de Custo e Plano de Gerenciamento de Risco de Projeto. Por outro lado, o desenvolvimento de produtos deve receber do service Notificações de Falha Sistêmica, onde estão descritos as falhas relatadas por clientes durante a utilização do produto.

Identificados os produtos que o desenvolvimento de produtos deve manipular – gerando ou interpretando-os – podem ser definidas as categorias de processo e as áreas chave de processo. Para facilitar o estabelecimento deste processo foram tomados como referência o modelo CMM e a futura norma ISO15504 (SPICE).

A Figura 31 ilustra as categorias de processo (nas caixas de texto em amarelo) e áreas de processo (nas caixas de texto brancas) que compõe o SyDeP.

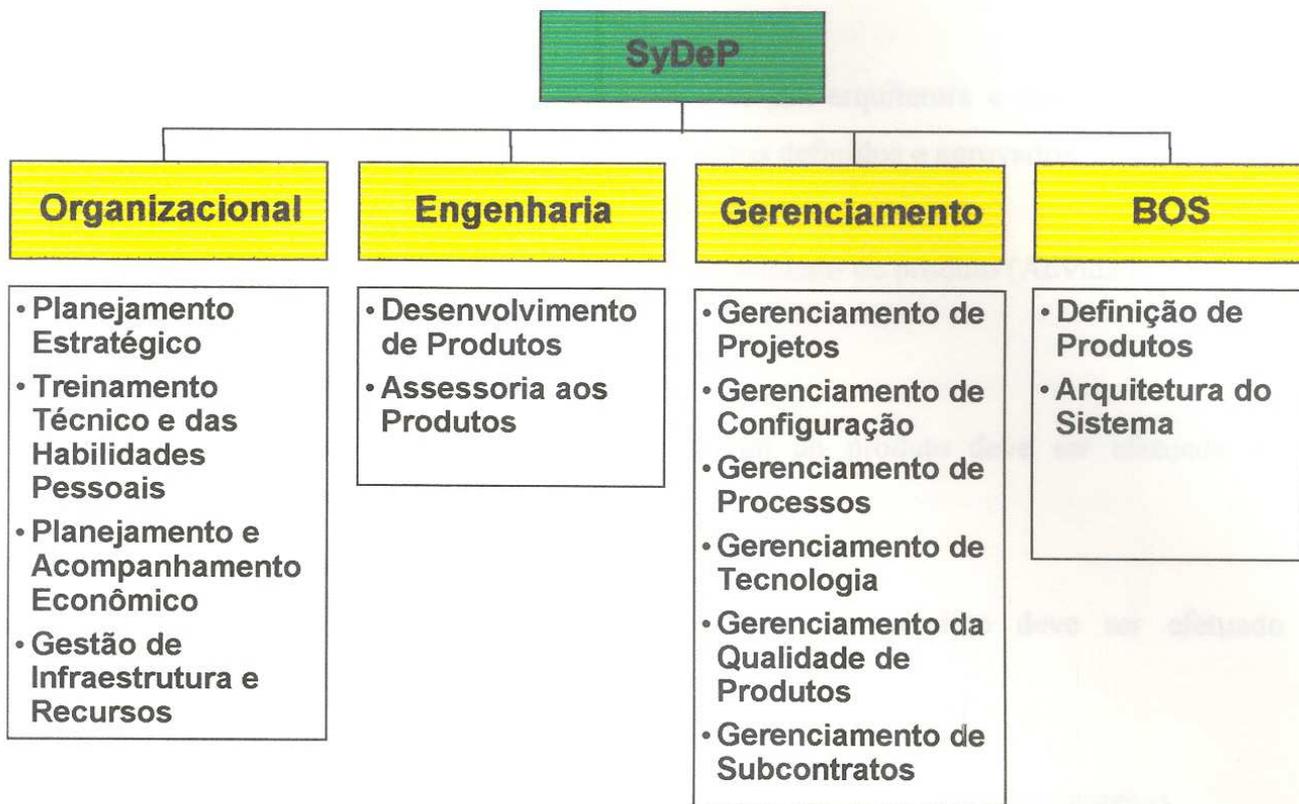


Figura 31 - Estrutura das Categorias e Áreas de Processo do SyDeP.

O SyDeP possui todos os processos a serem implementados em todos os ciclos de melhoria do método integrado. Porém até o momento foram estabelecidas somente as atividades do primeiro ciclo de melhoria. A seguir são verificados as práticas e processos contidos no SyDeP e onde os processos propostos pelo primeiro ciclo de melhoria do método integrado estão contidos:

#### ⇒ Definição de Produtos

- Definição dos requisitos de produtos

Os requisitos técnicos atendidos pelos produtos desenvolvidos pelo SyDeP devem ser definidos e aprovados em conjunto com o contratante.

- Gestão dos requisitos de produtos

Os requisitos aprovados dos produtos devem ser gerenciados ao longo das etapas do ciclo de vida de desenvolvimento, garantindo-se assim a sua rastreabilidade.

Processos do primeiro ciclo do método integrado atendidos:

- Identificação das Necessidades do Cliente
- Definição da necessidade de aquisição de produto e/ou serviço
- Procedimento para o Gerenciamento de Requisitos

### ⇒ **Arquitetura do Sistema**

Os produtos desenvolvidos pelo SyDeP devem ter sua arquitetura e design definidas pelas equipes de projeto e estar de acordo com os requisitos definidos e aprovados.

Processo do primeiro ciclo do método integrado atendido:

- Definição das práticas do ciclo de vida de desenvolvimento do produto (Atividade de Design)

### ⇒ **Desenvolvimento de Produtos**

- **Implementação de HW e SW**

A partir da definição da arquitetura e do design do produto deve ser efetuada a sua implementação de HW e SW.

- **Desenvolvimento construtivo e mecânico**

A partir da definição da arquitetura e do design do produto deve ser efetuado o desenvolvimento construtivo e mecânico.

- **Elaboração de layout**

A partir da implementação de HW deve ser elaborado o layout de circuito impresso.

- **Documentação técnica de desenvolvimento**

A equipe de desenvolvimento deve elaborar a documentação técnica do produto para fabricação e manutenção.

- **Qualificação técnica de OEM**

Equipamentos OEM devem ser qualificados a fim de serem comercializados.

- **Verificação e Validação de Produto**

Devem ser efetuados testes de sistema e de campo a fim de garantir a qualidade dos produtos liberados pelo departamento.

Processos do primeiro ciclo do método integrado atendidos:

- Definição das práticas do ciclo de vida de desenvolvimento do produto (Atividades de Implementação, Integração e Teste de Sistema)
- Parte do processo de Garantia da Qualidade (relativos à Qualidade de Produto)

### ⇒ **Gerenciamento de Projetos**

- **Proposta de Projeto**

Consultas efetuadas pelo departamento de produtos devem ser analisadas pelo SyDeP para verificação e validação de sua viabilidade técnica e planejamento preliminar de custos e prazos ou oferta de desenvolvimento.

- **Planejamento de projetos**

Os projetos devem possuir um plano de projeto que deve definir as tarefas, recursos, orçamento, tempo, risco e qualidade envolvidos na sua execução.

- **Controle dos projetos**

Os projetos em andamento no departamento devem ser acompanhados, relatórios de status devem ser disponibilizados e os desvios no plano de projeto devem ser corrigidos.

Processos do primeiro ciclo do método integrado atendidos:

- Planejamento de Projeto
- Rastreamento e Supervisão de Projeto

⇒ **Gerenciamento de Configuração**

Os produtos de trabalho gerados pelo SyDeP devem ser colocados sob gerência de configuração a fim de se controlar as suas versões e as mudanças efetuadas ao longo do andamento do projeto a fim de sustentar, entre outros, o controle do projeto.

Processo do primeiro ciclo do método integrado atendido:

- Gerenciamento de Configuração

⇒ **Gerenciamento de Processos**

- **Definição dos processos**

Os processos utilizados, seguidos ou introduzidos devem ser definidos utilizando-se de alguma modelagem de definição de processos, treinamento e coaching.

- **Garantia de qualidade de processos**

Um plano de qualidade deve ser elaborado e métodos devem ser estabelecidos para que a qualidade especificada seja alcançada.

- **Gerenciamento quantitativo dos processos**

Os processos definidos pelo departamento devem ser monitorizados e estabelecido o seu controle estatístico.

- **Gerenciamento da mudança dos processos**

As atividades de melhoria contínua dos processos deve ser planejada e metas de melhoria de processos devem ser definidas.

Processo do primeiro ciclo do método integrado atendido:

- Parte do processo de Garantia da Qualidade (relativos à Qualidade de Processo)
- Definição dos Processos da Organização

⇒ **Gerenciamento de Tecnologia**

Pesquisa e introdução de novas tecnologias devem ser gerenciadas e controladas a fim de garantir sua adequação e eficácia.

Processo do primeiro ciclo do método integrado atendido:

- Práticas de Gerenciamento de Mudança de Tecnologia

⇒ **Gerenciamento da Qualidade de Produtos**

Os produtos desenvolvidos devem ser avaliados e medidos quanto a sua qualidade e o gerenciamento quantitativo deve ser efetuado.

⇒ **Assessoria aos Produtos**

Aos produtos desenvolvidos deve ser fornecida assessoria após o término do projeto e início do período de venda do produto.

⇒ **Treinamento Técnico e das Habilidades Pessoais**

O SyDeP deve manter um programa de treinamento que garanta às pessoas as habilidades pessoais para o cumprimento das funções a elas designadas.

⇒ **Planejamento Estratégico**

O SyDeP deve possuir um planejamento estratégico de longo prazo.

⇒ **Planejamento e Acompanhamento Econômico**

Deve haver o planejamento e o acompanhamento econômico dos custos do SyDeP de acordo com seu planejamento estratégico.

⇒ **Gestão de Infraestrutura e Recursos**

Devem ser gerenciadas a infraestrutura e os recursos necessários para que o SyDeP cumpra o seu planejamento estratégico.

Processo do primeiro ciclo do método integrado atendido:

- Fornecimento de instalações de trabalho adequadas

⇒ **Gerenciamento de Subcontratos**

As oportunidades de subcontratação de parte ou de todo o desenvolvimento de produto devem ser avaliadas e gerenciadas.

O relacionamento entre as áreas chave de processo se dá através da definição das suas interfaces. Estas interfaces são efetuadas pelo compartilhamento de produtos que são gerados por uma área chave do processo e utilizados por outra área chave. A Figura 32 ilustra um exemplo da relação entre as áreas chave de processo Gerenciamento de Projeto, Gerenciamento de Configuração, Garantia da Qualidade de Produtos e Desenvolvimento de Produtos.

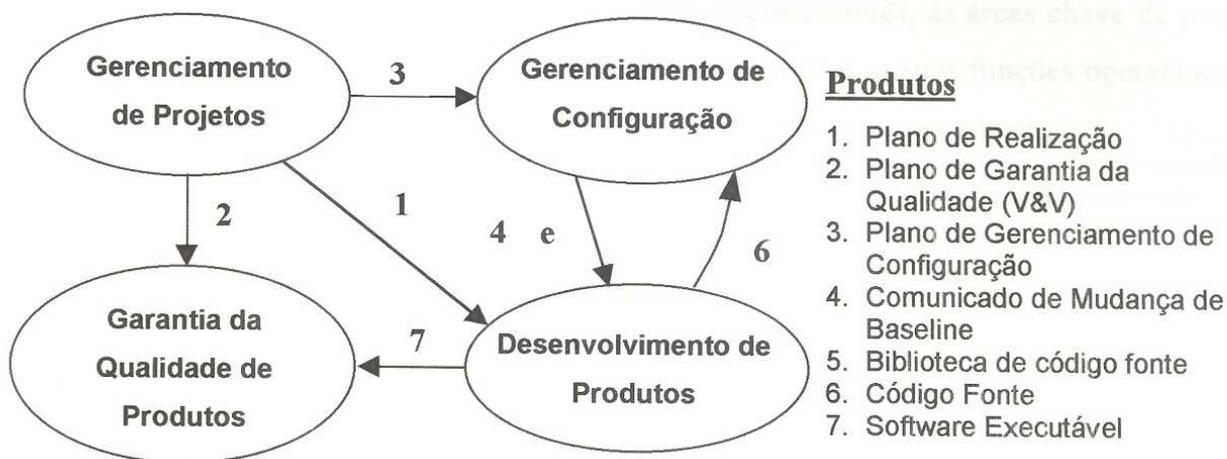


Figura 32 – Descrição reduzida do modelo de processos.

Os processos geram os produtos de forma assíncrona, ou seja, não há uma seqüência pré definida para a ocorrência do modelo apresentado na Figura 32. A geração dos produtos por cada processo ocorre de acordo com o ciclo de vida de desenvolvimento adotado por cada projeto. Logo, a operação da “máquina” de desenvolvimento de produtos varia de acordo com o projeto em andamento. O tipo de ciclo de vida de desenvolvimento varia com a tecnologia utilizada, quantidade de trabalho envolvida, complexidade do produto, etc.

Após a definição dos processos-chave devem ser estabelecidas as funções necessárias que irão desempenhá-los. Em outras palavras, estando definido os componentes e as condições de funcionamento da “máquina” de desenvolvimento de produtos devem ser selecionadas as habilidades dos “operários” que irão manipulá-la. Devido a complexidade de cada um dos “equipamentos” – ou processos-chave – que compõe a “máquina”, os “operários” – ou funções

operacionais - são agrupados em comitês executivos. Cada comitê é, assim, responsável pela execução das atividades dos processos-chave. Cada comitê possui um *process owner* que deve reportar ao Grupo de Gerenciamento de Processo suas experiências e observações do uso dos processos definidos. Este Grupo tem o objetivo de conhecer, medir e analisar os processos e propor quais as melhorias que devem ser feitas. Além disso, cada process owner deve reportar-se diretamente ao gerente sênior da organização sobre as atividades desempenhadas pelas áreas chave de processo afetadas pelo seu comitê.

A Tabela 15 apresenta um exemplo da relação entre alguns comitês, as áreas chave de processo definidas pela organização sobre a abrangência destes comitês e as suas funções operacionais. A primeira função operacional de cada comitê é intitulada como o *process owner*.

Comitês	Áreas Chave de Processo	Funções Operacionais
<i>Change Control Boards</i>	<ul style="list-style-type: none"> <li>• Gerenciamento de Configuração</li> <li>• Definição de Produtos</li> <li>• Arquitetura do Sistema</li> </ul>	<ul style="list-style-type: none"> <li>• Líder Técnico de Projeto</li> <li>• Consultor de Sistemas</li> <li>• Engenheiro de Sistemas</li> <li>• Gerente de Projeto</li> <li>• Líder de Projeto de HW/SW/Teste</li> </ul>
Gerenciamento da Qualidade de Produtos	<ul style="list-style-type: none"> <li>• Gerenciamento da Qualidade de Produtos</li> </ul>	<ul style="list-style-type: none"> <li>• Gerente de Qualidade</li> <li>• Engenheiro de Teste</li> </ul>
Grupo de Desenvolvimento de Sistemas	<ul style="list-style-type: none"> <li>• Desenvolvimento de Produtos</li> <li>• Assessoria aos Produtos</li> </ul>	<ul style="list-style-type: none"> <li>• Líder Técnico de Projeto</li> <li>• Gerente de Projeto</li> <li>• Líder de HW/SW/Teste</li> </ul>
Gerenciamento de Projetos	<ul style="list-style-type: none"> <li>• Gerenciamento de Projetos</li> <li>• Gerenciamento de Subcontratos</li> </ul>	<ul style="list-style-type: none"> <li>• Gerente de Projetos</li> </ul>

**Tabela 15** – Exemplo de relacionamento entre comitês, áreas chave de processo e funções operacionais.

Cada uma das funções operacionais devem ser descritas e as suas responsabilidades definidas. Por exemplo:

“O Líder Técnico de Projeto é o responsável geral pelo projeto. É responsável pela coordenação técnica de todas as partes envolvidas no projeto, zelando pelas interfaces externas e internas das atividades da categoria de processos de Engenharia. É também responsabilidade do líder técnico de projeto a convocação do CCB – *Change Control Board* para a análise de alterações no *baseline* do projeto.”

A identificação das áreas chave de processo e das funções operacionais necessárias para a sua execução não estão necessariamente vinculadas ao organograma da organização. A organização pode ter, por exemplo, seu modelo organizacional orientado ao tipo de produtos desenvolvidos. Resultando assim numa descrição de cargos completamente diferente do modelo orientado a processo aqui descrito. Portanto, as funções operacionais devem ser ocupadas pelas pessoas que estão alocadas em uma organização cujo modelo organizacional não é orientado a processos. Esta alocação pode ser efetuada por projeto, por atividade, etc. Esta alocação dinâmica dos recursos pessoais permite que uma única pessoa desempenhe várias funções em um determinado período do ciclo de vida de desenvolvimento de produtos ou atue em funções diferentes de um ciclo de vida para outro.

Esta alocação dinâmica para o desempenho das funções caracteriza uma adequação de um modelo organizacional não orientado a processos ao modelo orientado a processos. Esta adequação e a introdução ou alteração de atividades advindas do foco no processo da organização caracteriza uma melhoria contínua do processo. Outra possibilidade é a reengenharia completa da organização, alterando-se o seu organograma interno para torná-lo igual ao modelo aqui apresentado. Esta alteração se torna viável e útil quando o tamanho da organização justifica a existência de pessoas ocupando todos os cargos descritos no modelo orientado a processos, não existindo sobreposição de funções ou esta sendo pequena.

## **5.5 Resultados Obtidos**

Com a adoção do programa de melhoria de processos a organização obteve um maior controle dos seus processos de desenvolvimento de software, melhorando significativamente o planejamento e acompanhamento dos projetos de software. Através da definição do processo de engenharia de software, houve uma melhor identificação dos papéis ou funções no ciclo de vida de desenvolvimento dos produtos. Um indicativo importante da melhoria obtida com a adoção de melhores práticas decorrente do programa de melhoria de processos é apresentado na Tabela 16. Para a obtenção dos resultados apresentados nesta tabela foram analisados os cronogramas dos projetos nela citados. Destes cronogramas foram obtidos os períodos de realização e os esforços necessários para a sua realização estimados e efetivamente obtidos. Além disso, foram coletados

os dados históricos das falhas identificadas em teste de sistema de cada um destes projetos analisados e calculado o número médio de esforço realizado por falha identificada. Este último dado indica o grau de qualidade do desenvolvimento do produto até o seu teste de sistema.

Como conclusões, observou-se que o esforço estimado para os projetos é muito semelhante ao realizado. O erro médio obtido é de 2,5%, com uma variação de -56,5% no projeto A a +19% no projeto E. Esta precisão no erro médio é resultado da implementação dos processos de planejamento e rastreamento de projeto, tendo sido pior nos projetos iniciais (A e B) e melhor nos posteriores, onde os procedimentos estavam melhor consolidados.

Projeto	Esforo planejado (horas-homem)	Esforo realizado (horas-homem)	Período de realização estimado (meses)	Período de realização obtido (meses)	Falhas em teste de sistema	Esforo por falha (horas-homem)
A	2.960	1.891	8	8	2	945
B	3.360	2.652	9	9	2	1.326
C	2.880	2.713	8	8	2	1.356
D	6.048	6.032	12	13	0	-
E	17.040	21.056	11	11	141	149
F	38.208	37.920	13	15	166	228

Tabela 16 – Dados obtidos após o primeiro ciclo de melhoria.

A média de esforço gasto por falha é alto para pequenos projetos, porém muito pequeno para projetos que envolvam maior esforço de desenvolvimento. Esta diferença decorre da insuficiência das práticas de garantia da qualidade implementadas durante o primeiro ciclo de melhoria de processos. A equivalência entre o índice de falhas para ambos os projetos deve ser obtida no segundo ciclo de melhoria com a introdução do processo de *peer review* e do PSP. Nos estudos realizados com o PSP [22] demonstram que o índice médio de falhas antes da adoção do PSP é de 96 horas-homem/falha, após a introdução do PSP este índice aumenta para 1057 horas-homem/falha. Logo, verifica-se que para pequenos projetos o índice obtido pela organização é muito semelhante (sendo inclusive acima da média) ao verificado com o uso do PSP, porém para grandes projetos este índice é significativamente pior. O projeto D obteve 0 falhas em teste de sistema indicando que os erros foram obtidos nas fases anteriores do processo de

desenvolvimento. Tal projeto entretanto, caracterizou-se pela adaptação de um software já desenvolvido a fim de atender a normas de telecomunicações nacionais, facilitando desta forma a sua revisão e antecipação de defeitos.

O período de realização dos projetos foi de 10,6 meses com uma variação de 8 a 15 meses. O erro médio na previsão do tempo de duração do projeto foi de 4,7%, com uma variação máxima de 13%. Para o segundo ciclo de melhoria de processos tem-se como meta a redução do ciclo de vida de desenvolvimento dos projetos para 9 meses.

A definição do processo padrão da organização permitiu o refinamento da medição dos processos da organização, viabilizando a obtenção dos esforços gastos em cada um destes processos que são executados pelo desenvolvimento de produtos. A partir desta definição dos processos e da obtenção de medições do seu desempenho pode-se obter uma melhor localização de onde as falhas ocorrem durante o período de desenvolvimento dos produtos. Pode-se verificar também onde se localizam os maiores esforços e os benefícios obtidos por eles.

A implementação do primeiro ciclo de melhoria de processos viabilizou a adoção de uma ferramenta de “workflow” que efetue o gerenciamento de processos e facilite a troca de produtos de trabalho gerados durante o desenvolvimento do produto final.

O resultado mais importante e significativo obtido com a adoção do programa de melhoria de processos foi a contribuição dada pela sua existência na organização para a conquista do Prêmio Nacional da Qualidade de 1998. É interessante citar que os avaliadores da Fundação PNQ inquiriram e atestaram a existência, na organização, de um programa de melhoria contínua dos processos de software. O comprometimento da alta diretoria da organização com as metas gerais do programa de melhoria do processo de software está sendo significativa para que ela obtenha e continue obtendo o sucesso verificado até aqui.

## 6 Conclusão

A melhoria do processo de software nas suas três dimensões só é possível através da integração de diferentes modelos. Para isto, o desenvolvimento de um método integrado permite o relacionamento entre estes diferentes modelos e a definição de quais atividades devem ser introduzidas pela organização para melhorar o processo de software ao nível organizacional, individual e de suas equipes de projeto.

Para melhorar as três dimensões do processo de software a organização deve adotar um programa de melhoria do processo e adotar uma abordagem equivalente ao modelo IDEAL. Tal abordagem serve de guia de como colocar em prática tal programa de melhoria do processo. Esta abordagem é composta de 5 fases, onde deve ser adquirido o comprometimento necessário para que o programa de melhoria tenha sucesso; efetuado uma avaliação do processo atual, verificando-se os pontos fortes e fracos existentes; o estabelecimento de um plano de ação e de metas mensuráveis para monitorar o progresso do programa de melhoria; as soluções criadas devem ser pilotadas e disseminadas através da organização; e finalmente, as lições aprendidas e as métricas de desempenho das fases precedentes devem ser coletadas e analisadas.

Para a adoção das melhorias de processo, o programa de melhoria do método integrado aqui apresentado possui 4 ciclos de melhoria. Cada ciclo representa uma passagem pelas 5 fases do modelo IDEAL. Cada ciclo possui atividades selecionadas que devem ser postas em prática pela organização. A integração entre os modelos de melhoria de processo ocorre gradualmente na implementação das atividades selecionadas em cada ciclo de melhoria.

Este método integrado foi desenvolvido a partir da realidade de uma organização que desenvolve e mantém produtos com alto conteúdo tecnológico. Os produtos são desenvolvidos para vários parceiros ou contratantes que possuem processos definidos, porém distintos, necessitando um alto grau de adaptabilidade da organização, dos engenheiros de software e das equipes.

Para a melhoria do processo da organização foram utilizados os modelos CMM e SPICE (norma ISO 15504). Estes modelos de referência descrevem as atividades que uma organização que desenvolve software deve realizar para aumentar a sua capacidade de desenvolvimento e a sua maturidade do processo.

Para a melhoria do processo individual dos engenheiros de software foi utilizado o modelo PSP, que fornece um fluxo de trabalho onde a execução de uma seqüência de exercícios, análise de dados permite ao programador entender o que está errado e melhorar o seu próprio processo de trabalho.

Para a melhoria do processo das equipes de projeto de software foi utilizado o modelo TSP, que guia os engenheiros de software e os gerentes de projeto na construção de equipes auto controladas, que sustentem picos de desempenho e produzam produtos com qualidade mesmo com cronogramas agressivos.

A relação entre as práticas dos modelos de melhoria permitiu a seleção de quais práticas devem ser implementadas em cada um dos 4 ciclos de melhoria do processo. Através da verificação da dependência entre estas práticas foi possível estabelecer a seqüência com que elas devem ser implementadas pela organização. Em cada ciclo foi estabelecido um questionário de maturidade para verificar a implantação das práticas e o impacto esperado no desempenho da organização ao final da introdução destas práticas.

Para verificar como implementar o método integrado foi apresentado um estudo de caso na Siemens Ltda. Neste estudo de caso foram apresentadas as justificativas que convenceram a alta direção da organização a adotar a gerencia de processos e os aspectos que serviram de motivação para a implantação de um programa de melhoria de processos. Foi apresentada a estratégia de introdução do método integrado e analisadas as dificuldades encontradas para a implementação das práticas de melhoria de processos. Foram descritos os processos que compõe o SyDeP e os processos introduzidos e definidos no primeiro ciclo de melhoria de processo. E finalmente, verificados os resultados obtidos pela organização com a introdução do método integrado de melhoria de processo e os esforços que foram necessários para a adoção deste método.

A adoção do método integrado de melhoria de processos está sendo muito importante para a organização manter a sua posição de *global player*, ou seja, um centro de competência mundial no desenvolvimento de sistemas de comunicações. Tal posição exige o aprimoramento contínuo dos processos da organização e a adaptação destes processos aos processos exigidos e/ou utilizados pelos seus contratantes.

Como continuidade dos estudos efetuados nesta dissertação podem ser efetuadas as seguintes pesquisas futuras.

A partir da definição dos processos da organização podem ser avaliadas e implementadas as métricas de processos a serem utilizadas. Obtendo-se métricas de processos pode-se estudar o relacionamento entre tais medições e as metas de negócio da organização através do “Balanced Scorecard” [24] [8]. Tal método permite fornecer um equilíbrio entre os quatro itens básicos que influenciam o desempenho da organização, sendo: treinamento e crescimento pessoal, processos, foco no cliente e retorno financeiro.

A introdução do PSP pode ser avaliado quanto aos seguintes aspectos: custos envolvidos, impactos esperados, ferramentas necessárias e o seu relacionamento com o P-CMM (Personal Capability Maturity Model) [5].

A introdução de uma ferramenta de workflow que permita o gerenciamento de processos pode ser avaliada e implementada [25]. Tal ferramenta deve conter uma integração com uma ferramenta que gere um “trace” com o modelo de processos da organização customizado para cada um dos projetos a serem executados. Além disso, deve permitir a obtenção das métricas dos processos e produtos, o gerenciamento quantitativo dos processos e indicar as oportunidades de melhoria.

## Referências Bibliográficas

- [1] ANSI/IEEE Std. 610.12-1990: "IEEE Standard Glossary of Software Engineering Terminology", February 1991.
- [2] Bell Canada, "The Trillium Model", 1994.
- [3] Brodman, J.G. e Johnson, D.: "Return on Investment from Software Process Improvement as Measured by U.S. Industry", Crosstalk, Apr. 1996, pp.23-28.
- [4] Byrnes, Paul e Phillips, Mike: "Software Capability Evaluation Version 3.0 Method Description", Technical Report SEI/CMU-96-TR-002, 1996.
- [5] Curtis, Bill; Hefley, William E., and Miller, Sally A. : "People Capability Maturity Model, Version 1.0", September 1995, Pittsburgh, PA, Software Engineering Institute, CMU/SEI-95-MM-002.
- [6] Diaz, Michael e Sligo, Joseph: "How Software Process Improvement Helped Motorola", IEEE Software, Sept/Oct 1997, pp. 75-80.
- [7] Dunaway, Donna K. e Masters, Steve: "CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description", Technical Report CMU/SEI-96-TR-007, 1996.
- [8] Ferguson, P. "Connecting Process Improvement Goals to Business Objectives with a Balanced Scorecard." Proceedings of SEPG 98: The 1998 Software Engineering Process Group Conference, Chicago, Illinois, 1998, March 9.
- [9] Garcia, Suzanne M.: "Evolving Improvement Paradigms: Capability Maturity Model & ISO/IEC 15504 (PDTR)", SEI/CMU Software Process Improvement and Practice, vol. 3, issue 1, 1998.
- [10] Gremba, Jennifer e Myers, Chuck: "The IDEAL Model: A Practical Guide for Improvement", SEI/CMU Bridge publication, issue three, 1997.
- [11] Hadden, Rita: "How Scalable are CMM Key Practices?", CrossTalk, Apr. 1998.
- [12] Hayes, Will e Over, James W.: "The Personal Software Process(PSP): An Empirical Study of the Impact of PSP on Individual Engineers", Technical Report CMU/SEI-97-TR-001, 1997.
- [13] Herbsleb, James et al., "Benefits of CMM-Based Software Process Improvement: Initial

Results”, Technical Report SEI/CMU-94-TR-013, 1994.

- [14] Humphrey, Watts S.: “Three Dimensions of Process Improvement – Part I: Process Maturity”, *Crosstalk*, STSC, Hill Air Force Base, Utah, February 1998.
- [15] Humphrey, Watt S.: “A Discipline for Software Engineering”, Addison-Wesley, SEI Series on Software Engineering, 1995.
- [16] Humphrey, Watt S.: “Introduction to the Personal Software Process”, Addison-Wesley, SEI Series on Software Engineering, 1997.
- [17] Humphrey, Watts S.: “Three Dimensions of Process Improvement – Part II: The Personal Process”, *Crosstalk*, STSC, Hill Air Force Base, Utah, March 1998.
- [18] Humphrey, Watts S.: “Three Dimensions of Process Improvement – Part III: The Team Process”, *Crosstalk*, STSC, Hill Air Force Base, Utah, April 1998.
- [19] Humphrey, Watts S.: “The Team Software Process”, *Anais do IX CITS: Qualidade de Software*, 1998.
- [20] Humphrey, Watts S.: “Managing the Software Process”, Addison-Wesley, SEI Series on Software Engineering, 1989.
- [21] Humphrey, Watts S.: “Why Don’t They Practice What We Preach?”, SEI Publications, May 1998.
- [22] Humphrey, Watts S.: “The Software Quality Profile”, SEI Publications, July 1998.
- [23] Ibrahim, Linda: “Smart Buying with the Federal Aviation Administration’s Integrated Capability Maturity Model”, *CrossTalk*, Nov. 1998.
- [24] Kaplan, Robert: “The Balanced Scorecard : Translating Strategy into Action”, Harvard Business School Press, Sept. 1996.
- [25] Kelner, Marc I. et al.: “Process Guides: Effective Guidance for Process Participants”, *Proceedings of the 5th International Conference on the Software Process: Computer Supported Organizational Work*, International Software Process Association, New Jersey, 1998.
- [26] Kness, Steven P. e Satake, Mark S.: “A Level 5 Organization Looks at the Personal Software Process”, *CrossTalk*, Oct. 1997.
- [27] Konrad, Michael D., Paulk, Mark C. e Graydon, Allan W.: “An Overview of SPICE’s Model for Process Management” – *Proceedings of the Fifth International Conference on Software Quality*, Austin, TX, 23-26 October, 1995.

- [28] Masters, Steve e Bothwell, Carol: "CMM Appraisal Framework, Version 1.0", Technical Report CMU/SEI-95-TR-001, 1995.
- [29] McFeeley, Bob: "IDEAL: A User's Guide for Software Process Improvement", Handbook CMU/SEI-96-HB-001, 1996.
- [30] Melhoretto, Sandro M. e Betini, Roberto C.: "CMM em Telecomunicações", Anais II Workshop de Teses em Engenharia de Software, XI SBES, Brasil, 1997.
- [31] Melhoretto, Sandro M., Betini, Roberto C. e Stark, Paulo R.: "Why and How to Do a Software Process Improvement Program", Anales SoST'98, IX CLAIO, Argentina, 1998.
- [32] Melhoretto, Sandro M. e Betini, Roberto C.: "Por quê e como melhorar o processo de software", Anais IX Workshop de Qualidade de Software, XII SBES, Brasil, 1998.
- [33] Messer, Tilo: "The Siemens Improvement Method – From Assessment to Improvement Strategies", ISCN Newsletter, <http://www.iscn.ie/news/sp96/t.messer.html>, 1996.
- [34] Pall, Gabriel A.: "Quality Process Management". Englewood Cliffs, N.J.: Prentice Hall, 1987.
- [35] Paulk, Mark C., Konrad, Michael D. e Garcia, Suzanne M.: "CMM Versus SPICE Architectures" – Software Process Newsletter IEEE Computer Society Technical Council on Software Engineering, No. 3, Spring 1995.
- [36] Paulk et al., Mark: "The Capability Maturity Model: Guidelines for Improving The Software Process", SEI Series in Software Engineering, Addison-Wesley, 1995.
- [37] Paulk, Mark C. et al.: "Capability Maturity Model for Software, Version 1.1", Technical Report CMU/SEI-93-TR-24, 1993.
- [38] Paulk, Mark C. et al.: "Key Practices of the Capability Maturity Model, Version 1.1", Technical Report CMU/SEI-93-TR-25, 1993.
- [39] Pressman, Roger S.: "A Manager's Guide to Software Engineering", McGraw-Hill, 1993.
- [40] Rout, Terence P.: "Issues in the Development of an International Standard for Software Process Assessment", IEEE Software Process Newsletter No. 10, Fall 1997.
- [41] Sage, Andrew P.: "Systems Management for Information Technology and Software Engineering", John Wiley & Sons Inc., 1995.
- [42] Software Engineering Institute: "Software Maturity Profile of the Software Community 1997 Update", October 1997.
- [43] SPICE Project: "SPICE - ISO/IEC Software Process Assessment – Part 1: Concepts and

Introductory Guide (Working Draft V 1.00)” – 1995.

[44] SPICE Project: “SPICE - ISO/IEC Software Process Assessment – Part 2: A model for process management (Working Draft V 1.00)” – 1995.

[45] Zubrow, Dave et al.: “Maturity Questionnaire”, Special Report SEI/CMU-94-SR-7, 1994.