

Universidade Federal De Santa Catarina – UFSC
Pró-Reitoria de Pós-Graduação e Pesquisa
Programa de Pós-Graduação em Engenharia de Produção

Tereza Tiemi Nakajima

**AVALIAÇÃO DO COMPORTAMENTO DAS EMPRESAS EM RELAÇÃO À
QUALIDADE NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE**

Dissertação de Mestrado

Florianópolis
2004

Tereza Tiemi Nakajima

**AVALIAÇÃO DO COMPORTAMENTO DAS EMPRESAS EM RELAÇÃO À
QUALIDADE NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito parcial para obtenção do título de Mestre em Engenharia de Produção.

Orientador: Prof. Pedro Felipe de Abreu, Ph.D.

Florianópolis

2004

Tereza Tiemi Nakajima

**AVALIAÇÃO DO COMPORTAMENTO DAS EMPRESAS EM RELAÇÃO À
QUALIDADE NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE**

Esta dissertação foi julgada adequada para obtenção do grau de **Mestre em Engenharia de Produção** no Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

Florianópolis, 17 de Agosto de 2004.

Prof. Edson Pacheco Paladini, Dr.
Coordenador do Programa

Banca Examinadora

Prof. Pedro Felipe de Abreu, Ph.D.
Orientador

Profa. Ana Elizabeth Moiseichyk, Dra.

Prof. Manoel Agrasso Neto, MS
Co-Orientador

Prof. Marcello Thiry Comicholi da Costa, Dr.

À minha família que sempre me apoiou com amor e carinho em todos os momentos deste trabalho.

AGRADECIMENTOS

Ao professor Pedro Felipe de Abreu, pelas críticas e orientação deste trabalho.

Ao co-orientador professor Manoel Agrasso Neto pelo apoio e sugestões objetivas prestadas.

A toda equipe do IGTI, pela atenção dispensada.

Em especial ao professor e amigo Nelson Suga pela compreensão, incentivo, aprendizado e pelas valiosas contribuições à realização deste trabalho.

Ao professor Robert Burnett pelas observações.

Às empresas que participaram da pesquisa fornecendo informações utilizadas no presente trabalho.

À UFSC e aos professores do programa de Pós Graduação em Engenharia de Produção pela participação na construção do meu saber.

Aos amigos Maria Júlia Trevisan, Ivone Morimoto, Telma Picheti e Luiz Fernando Ortolani pela colaboração.

Aos colegas de mestrado que partilharam amizade, estudo e experiência a favor do aprendizado.

A todos que, de forma direta ou indireta, contribuíram para a realização deste trabalho.

À minha família, que sempre esteve a meu lado e que, em todos os desafios da minha vida, representou verdadeira força e riqueza.

SUMÁRIO

LISTA DE FIGURAS	ix
LISTA DE QUADROS	x
LISTA DE TABELAS	xi
LISTA DE ABREVIATURAS.....	xii
RESUMO	xiv
ABSTRACT	xv
1 INTRODUÇÃO	16
1.1 Origem do Trabalho	16
1.2 Objetivos	17
1.2.1 Objetivo Geral.....	17
1.2.2 Objetivos Específicos	18
1.3 Justificativa e Relevância da Pesquisa.....	18
1.4 Estrutura do Trabalho	19
2. FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 Qualidade de Software	20
2.1.1 Qualidade	21
2.1.2 Software.....	24
2.1.3 Os Diversos Conceitos de Qualidade de Software	26
2.1.4 Fatores Intervenientes na Qualidade de Software	30
2.2 Qualidade no Processo de Desenvolvimento de Software	32
2.2.1 Qualidade de Processo e de Produto	32
2.2.2 Processo de Desenvolvimento de Software	33
2.2.3 Estágios na Melhoria de Processo.....	38
2.3 Modelos de Melhoria de Processos	39
2.3.1 ISO 9000	40
2.3.1.1 ISO 9001:2000	41
2.3.2 QIP – <i>Quality Improvement Paradigm</i>	43
2.3.3 GQM – <i>Goal/Question/Metric</i>	44
2.3.3.1 Processo GQM.....	45
2.3.4 SPICE - Melhoria do Processo de Software e Determinação da Capacidade .	47
2.3.4.1 Objetivos da ISO/IEC 15504 (SPICE).....	47

2.3.5 SW-CMM – Modelo de Maturidade da Capacidade de Software	56
2.3.5.1 Histórico CMM – <i>Capability Maturity Model</i>	56
2.3.5.2 O Modelo SW-CMM	57
2.4 Estudo Comparativo dos Processos de Melhoria de Software	63
2.4.1 Comparando Estruturas de Melhoria de Processo de Software	63
2.4.2 Classificação	64
2.4.3 Comparando ISO 9001 e o SW-CMM	68
2.4.4 Comparando o SPICE e o SW-CMM	68
2.4.5 Considerações sobre o SW-CMM, ISO 9000 e SPICE	71
2.4.5.1 Considerações sobre o SW-CMM	71
2.4.5.2 Considerações sobre o SPICE	75
2.4.5.3 Considerações sobre a ISO 9000	76
2.4.5.4 Outras Considerações	76
2.4.6 Conclusão do Estudo Comparativo	77
3 METODOLOGIA	79
3.1 Caracterização da Pesquisa	79
3.2 Sujeitos da Pesquisa	79
3.3 Instrumentos de Coleta de Dados	80
3.4 Tratamento dos Dados	81
4 CONSOLIDAÇÃO, ANÁLISE E INTERPRETAÇÃO DOS DADOS DE PESQUISA	82
4.1 Análise dos Dados Relacionados às Empresas Situadas no Nível 1 - Inicial.....	82
4.2 Análise dos Dados Relacionados às Empresas Situadas no Nível 2 - Repetível	82
4.3 Análise dos Dados Relacionados às Empresas Situadas no Nível 3 - Definido .	83
4.4 Análise dos Dados Relacionados às Empresas Situadas no Nível 4 - Gerenciado.....	85
4.5 Análise dos Dados Relacionados às Empresas Situadas no Nível 5 - Otimizante	88
4.6 Análise dos Dados Relacionados às Questões Complementares	90
4.7 Nível de Maturidade das Empresas Pesquisadas	91
4.8 Considerações Finais	95
5 CONCLUSÕES E RECOMENDAÇÕES.....	99

5.1 Conclusões.....	99
5.2 Recomendações para Futuros Trabalhos	101
REFERÊNCIAS	102
APÊNDICES	109
Apêndice A - Carta de Apresentação da Pesquisa	110
Apêndice B - Carta de Apresentação do Pesquisador	112
Apêndice C - Questionário da Pesquisa	114

LISTA DE FIGURAS

Figura 1: Curvas de falhas idealizada e real para o software	25
Figura 2: Fatores de qualidade de software de McCall	31
Figura 3: Principais fatores da qualidade de produtos de software.....	33
Figura 4: Processo de software	36
Figura 5: Uso da ISO 15504 para melhoria de processos (ISO,1998)	48
Figura 6: Uso da ISO 15504 para determinação da capacidade (ISO,1998).....	49
Figura 7: Inter-relação dos elementos do SW-CMM.....	59
Figura 8: Características da progressão do processo de qualidade	66
Figura 9: Espectro de Planejamento	74
Figura 10: Freqüência de utilização do processo de supervisão e acompanhamento de projetos	83
Figura 11: Freqüência de utilização do processo de engenharia de produto de software.....	85
Figura 12: Freqüência de utilização do processo de gestão quantitativa dos processos.....	87
Figura 13: Freqüência de utilização do processo de gestão da qualidade de software.....	87
Figura 14: Freqüência de utilização do processo de prevenção de defeitos	89
Figura 15: Níveis encontrados na pesquisa em janeiro 2004	91
Figura 16: Índice Médio de Importância, por Nível no SW-CMM e Questões Complementares (Qc).....	96
Figura 17: Percentuais Médios de Respostas Positivas na Utilização dos Níveis no SW-CMM e Questões Complementares (Qc).....	97
Figura 18: Correlações entre Variáveis.....	98

LISTA DE QUADROS

Quadro 1: Características e subcaracterísticas - norma ISO / IEC 9126 / NBR 13596	23
Quadro 2: Atributos essenciais de um bom software.....	26
Quadro 3: Definições de qualidade de software	30
Quadro 4: Principais cláusulas da ISO 9001:2000.....	42
Quadro 5: Níveis da Capacidade SPICE	51
Quadro 6: Práticas Genéricas para Nível 1 : Desempenhado Informalmente.....	52
Quadro 7: Práticas Genéricas para Nível 2 : Planejado e Rastreado.....	52
Quadro 8: Práticas Genéricas para Nível 3 : Bem Definido	53
Quadro 9: Práticas Genéricas para Nível 4: Quantitativamente Controlado	53
Quadro 10: Práticas Genéricas para Nível 5: Melhorando Continuamente	53
Quadro 11: Descrição de categorias do processo	54
Quadro 12: Níveis do SW-CMM	58
Quadro 13: Descrição das Características Comuns	60
Quadro 14: Classificação aplicada em cinco estruturas MPS.....	65
Quadro 15: Comparação ISO 9001 e o SW-CMM	68
Quadro 16: Comparando ISO/IEC 15504 SPICE e o SW-CMM.....	70
Quadro 17: Receita do sucesso	77

LISTA DE TABELAS

Tabela 1: Ganhos obtidos em implementações do SW-CMM.....	73
Tabela 2: Escala Tipo LIKERT	80
Tabela 3: Nível 2 – Repetível.....	82
Tabela 4: Nível 3 – Definido	84
Tabela 5: Nível 4 – Gerenciado	86
Tabela 6: Nível 5 – Otimizante	88
Tabela 7: Questões complementares – SPICE	90

LISTA DE ABREVIATURAS

ACP	Áreas Chaves de Processo
ASSESPRO	Associação das Empresas Brasileiras de Software e Serviço de Informática
CASE	Engenharia de Software com o Auxílio de Computador / <i>Computer-Aided Software Engineering</i>
CMM	Modelo de Maturidade da Capacidade / <i>Capability Maturity Model</i>
DOD	Departamento de Defesa / <i>Department of Defense</i>
EUA	Estados Unidos da América
GQM	<i>Goal/Question/Metric</i>
HELP DESK	Equipe de Suporte ao Atendimento
IEC	Comissão de Eletrotécnica Internacional / <i>International Electrotechnical Commission</i>
ISO	Organização de Padronização Internacional / <i>International Organization for Standardization</i>
KLOC	Milhares de Linhas de Código / <i>Kilo Lines of Code</i>
KPAs	Áreas-Chave de Processo / <i>Key Process Areas</i>
MPS	Melhoria de Processo de Software
NASA	Administração do Espaço Aeronáutico Nacional / <i>National Aeronautics and Space Administration</i>
NBR	Normas Brasileiras
NIST	Instituto Nacional de Ciência e Tecnologia / <i>National Institute of Science and Technology</i>
QC	Questões Complementares (SPICE)
QIP	Paradigma de Melhoria da Qualidade / <i>Quality Improvement Paradigm</i>
Q&P	Qualidade e Produtividade
SEL	Laboratório de Engenharia de Software / <i>Software Engineering Laboratory</i>

SEI	Instituto de Engenharia de Software / <i>Software Engineering Institute</i>
SPICE	Melhoria do Processo de Software e Determinação da Capacidade / <i>Software Process Improvement and Capability determination</i>
STAKEHOLDERS	Grupos de Interesse, Especialistas de Domínio
SW-CMM	Modelo de Maturidade da Capacidade de Software / <i>Software Capability Maturity Model</i>
XP	Programação Extrema / <i>Extreme Programming</i>

RESUMO

NAKAJIMA, Tereza Tiemi. **Avaliação do Comportamento das Empresas em Relação à Qualidade no Processo de Desenvolvimento de Software**. 2004. 121f. Dissertação (Mestrado em Engenharia de Produção). Programa de Pós Graduação em Engenharia de Produção, UFSC, Florianópolis.

Esta pesquisa teve por objetivo avaliar o comportamento de empresas em relação à qualidade no processo de desenvolvimento de software. Para tanto procurou-se identificar, em literatura, modelos de qualidade aplicáveis no processo de desenvolvimento de software, realizando um estudo comparativo dos modelos de qualidade aplicáveis no processo de desenvolvimento de software identificados; selecionar, com base nos resultados da análise comparativa, modelos para avaliação das empresas produtoras de software; levantar, junto às empresas componentes da amostra, os dados requeridos pelos modelos selecionados. Os resultados indicam que: os modelos SW-CMM - *Software Capability Maturity Model* e o modelo SPICE - *Software Process Improvement and Capability Determination* foram selecionados pelo fato de que os dois modelos se complementam e permitem classificar tais empresas segundo seu nível de maturidade em relação à qualidade de software. Os resultados da pesquisa identificaram que 87,9% das empresas pesquisadas se encontram no Nível 1 do SW-CMM, 6,1% das empresas estão no Nível 2, 3% no Nível 3, 3% no Nível 4 e nenhuma empresa pesquisada está no Nível 5. Existe uma alta correlação entre as variáveis indicando maiores avaliações de importância aos conceitos de gestão mais utilizados. Os processos do modelo SPICE apresentaram um alto índice de utilização. No modelo SW-CMM, os processos do Nível 3 são os mais utilizados pelas empresas, estando entre eles a engenharia do processo de software.

Palavras-chave: Indústria de software; Software - controle de qualidade; Software - qualidade - avaliação.

ABSTRACT

NAKAJIMA, Tereza Tiemi. **Avaliação do Comportamento das Empresas em Relação à Qualidade no Processo de Desenvolvimento de Software**. 2004. 121f. Dissertação (Mestrado em Engenharia de Produção). Programa de Pós Graduação em Engenharia de Produção, UFSC, Florianópolis.

This research has the objective of assessing the behavior of Software manufacturing companies in relation to the quality in the process of software development. It aims at: identifying in the literature models of quality that could be applied in the process of software development; comparing the applicable models of quality in the identified software development process; selecting, based on the results of the comparative analysis, models for the assessment of the software manufacturing companies; defining the sample of companies to be assessed; collecting, within the sample of companies, the required data for the selected models; tabulating and analyzing, based on the selected models, the data collected in the sample companies. The results indicate the models SW-CMM - Software Capability Maturity Model and the model SPICE - Software Process Improvement Capability Determination - were selected because both of them are complementary and allow the classification of the companies according to their level of maturity in relation to software quality. The results of the research identified that 87.9% of the researched companies are in level 1 of the SW-CMM, 6.1% are in level 2, 3% in level 3, 3% in level 4 and none of the researched companies is in level 5. There is a high correlation between the variables indicating higher levels of importance given to the most used management concepts. The processes of the model SPICE presented a high rate of use. In the model SW-CMM, level 3 processes are the most used by the companies, being software process engineering among them.

Key Words: Software Industry; Software – quality control; Software – quality – assessment.

1 INTRODUÇÃO

Neste capítulo, são apresentados o tema da pesquisa, seus objetivos, sua justificativa e relevância, e finalmente a estrutura do trabalho.

1.1 Origem do Trabalho

Nos últimos 20 anos, a utilização do software tem conquistado um papel crítico e essencial em nossa sociedade. Depende-se cada vez mais dos serviços oferecidos por meio de sistemas computadorizados. Quase todo produto moderno ou serviço utiliza e/ou explora alguma parte do software. Porém, aplicações de software são produtos complexos que são difíceis de desenvolver e testar. Muito freqüentemente, o software manifesta um comportamento inesperado e indesejado que pode causar problemas graves e prejuízos (FUGGETTA, 2000).

Pesquisas realizadas em 2002 mostram que 34% dos projetos foram finalizados no tempo e custo previstos com todas funções e características originalmente especificadas. 51% dos projetos de software foram finalizados, mas acima do custo e tempo previstos e estavam fora da funcionalidade proposta originalmente. 15% dos projetos de software foram cancelados antes de serem finalizados ou nunca foram implementados (BROCK *et al*, 2003).

Observadores da indústria, nas décadas de 60 e 70, caracterizavam os problemas associados com o desenvolvimento do software como uma “crise”. Autores como Glass e Yourdon (*apud* PRESSMAN, 2002) relataram o impacto de algumas falhas de software que ocorreram ao longo da década passada. Entretanto, o sucesso alcançado pela indústria de software levou muitos autores a questionar se o termo crise de software ainda é apropriado.

Para Pressman (2002) o que ocorre realmente poderia ser caracterizado como uma aflição crônica. A palavra “crônica” tem por definição “de longa duração ou que volta freqüentemente, continuando indefinidamente”. O conjunto de problemas encontrados no desenvolvimento de software, independentemente da denominação dada, não é limitado ao software que “não funciona adequadamente”. A aflição inclui problemas associados com a maneira como se desenvolve software, como se

suporta volume dos softwares existentes e como podemos enfrentar uma crescente demanda por mais software.

Essa aflição gera desconfortos que estimulam mudanças em busca da melhoria permanente da qualidade do software. A busca da qualidade é uma jornada ilimitada e o alcance do processo de melhoria contínua é difícil e necessita de grandes esforços.

Há, por outro lado um entendimento que atualmente existe uma crise de qualidade de software. Um relatório preparado em maio 2002 pelo *National Institute of Science and Technology* (NIST) mostra que o custo anual dos defeitos de software nos Estados Unidos da América (EUA) é de 59,5 bilhões de dólares (HALLEM; PARK; ENGLER, 2003).

O objetivo básico do desenvolvimento de software é liberar um produto que possua os requisitos de qualidade necessários ao atendimento das expectativas e necessidades dos seus clientes (VIANA, 1994). Para que o objetivo seja alcançado é necessário que o foco do processo de desenvolvimento seja redirecionado para a satisfação do cliente.

As atividades do desenvolvimento precisam ser planejadas, padronizadas e seus resultados verificados (BASILI e CALDIERA, 1993) e para tal, é necessário que as responsabilidades sejam definidas e os membros das equipes estejam integrados, motivados e comprometidos com o processo.

As exigências cada vez maiores dos clientes e a globalização da economia têm influenciado as empresas produtoras e prestadoras de serviço de software a procurar alcançar o patamar de qualidade e produtividade internacional.

Neste contexto, esta pesquisa tem por objetivo avaliar as empresas, em relação à qualidade no desenvolvimento de software.

1.2 Objetivos

1.2.1 Objetivo Geral

Avaliar o comportamento de empresas em relação à qualidade no processo de desenvolvimento de software.

1.2.2 Objetivos Específicos

- ? Identificar, na literatura, modelos de qualidade aplicáveis no processo de desenvolvimento de software;
- ? Realizar análise comparativa dos modelos de qualidade aplicáveis no processo de desenvolvimento de software;
- ? Selecionar, com base nos resultados da análise comparativa, modelos para avaliação das empresas produtoras de software;
- ? Definir a amostra de empresas a serem avaliadas;
- ? Levantar, junto às empresas componentes da amostra, os dados requeridos pelos modelos selecionados;
- ? Tabular e analisar, com base nos modelos selecionados, os dados levantados junto às empresas componentes da amostra;
- ? Relacionar os resultados da pesquisa de campo a referenciais teóricos.

1.3 Justificativa e Relevância da Pesquisa

O cenário das empresas que atuam na área de software vem se alterando rapidamente nos últimos anos. O mercado globalizado está mais exigente e competitivo, fazendo com que as empresas procurem alternativas para uma melhoria na qualidade do processo e dos seus produtos.

Neste cenário, onde o computador e mais especificamente o software têm um papel significativo no processo produtivo, a relação custo-benefício e a qualidade do software podem se tornar vitais para as empresas produtoras de software.

Assim, se torna oportuno identificar, na literatura, modelos de qualidade aplicáveis à indústria de software que possibilitem avaliar tais empresas e/ou que sirvam de referência na busca pela qualidade e por uma boa relação custo-benefício.

A motivação para esta pesquisa está relacionada a motivos acadêmicos e práticos. Trata-se de uma área pouco absorvida pelas empresas, apresenta uma demanda significativa e busca a aplicação prática dos resultados da pesquisa.

1.4 Estrutura do Trabalho

O primeiro capítulo trata da introdução, do tema da pesquisa, seus objetivos, sua justificativa e relevância, e finalmente a estrutura do trabalho.

O segundo capítulo é apresentado o referencial teórico que engloba os temas pertinentes ao trabalho, ou seja: qualidade, software, processos, modelos de melhoria de processos, e análise comparativa dos modelos de melhoria de processos.

O terceiro capítulo refere-se aos materiais e procedimentos utilizados para o levantamento de dados, como a caracterização da amostra e o método utilizado.

No quarto capítulo contempla a apresentação, análise e discussão dos dados da pesquisa.

No último capítulo são apresentadas as conclusões e recomendações.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Qualidade de Software

A qualidade deveria dirigir o desenvolvimento do sistema como um todo. A falta de qualidade normalmente vai se tornando um problema real para muitas empresas. As conseqüências são a perda do negócio, perda do pessoal, que se recusa a fazer parte de operações de pouca qualidade, e responsabilidade legal por pobre desempenho ou prejuízo devido ao fato de os produtos carecerem de qualidade. A oportunidade seria clara: produto de alta qualidade faz bem no mercado (FREEMAN, 1987).

Hoje, a qualidade tem grande influência em todas áreas da atividade humana. Geralmente busca-se fornecer e receber produtos e serviços com qualidade. Mas o que é qualidade?

Utiliza-se para tal a definição de Robert Pirsig em “Zen e a arte da manutenção de motocicletas”, citada por Tsukumo *et all* (1997, p. 1).

Qualidade... a gente sabe o que é, e, ao mesmo tempo, não sabe. Isto é contraditório. Mas algumas coisas são melhores que outras, ou seja, têm mais qualidade. Porém se a gente tenta definir qualidade, isolando-as das coisas que possuem, então *puf* - já não tem mais o que falar.

Embora se possa concordar com esta afirmação, ao desenvolver atividades e para melhor administrar, é interessante ter medidas objetivas sobre qualidade. Os diversos modelos apresentados a seguir sugerem maneiras mais práticas e operacionais sobre como entender e administrar a qualidade.

2.1.1 Qualidade

Boa qualidade significa usuário feliz (HALVROSEN e CONRADI, 2002). Para Kitchenham (*apud* HALVROSEN e CONRADI, 2002) a qualidade “é difícil definir, impossível medir, fácil de reconhecer”.

Para Juran e Gryna (1993), “qualidade é a adequação ao uso sob ponto de vista do cliente”.

Na opinião de Campos (1992), “um produto ou serviço de qualidade é aquele que atende perfeitamente, de forma confiável, de forma acessível, de forma segura e no tempo certo às necessidades dos clientes”.

Segundo Deming (1990), “a qualidade é a perseguição às necessidades dos clientes e homogeneidade dos resultados do processo”.

De acordo com Fernandes (1995), “qualidade é a totalidade de requisitos e características de um produto ou serviço que estabelecem a capacidade de satisfazer as necessidades implícitas e explícitas”.

Para Crosby (1985), “a qualidade é vista como conformidade com os requisitos”. Weinberg (1993), diz que a “qualidade é a conformidade às exigências de alguma pessoa”.

Ishikawa (1993), “define a qualidade de maneira restrita como qualidade de produto ou de maneira mais ampla como qualidade de trabalho, qualidade de serviço, qualidade de informação, qualidade de processo, qualidade de pessoal, qualidade de sistema, qualidade de empresa, qualidade de objetivos”.

Na opinião de Paladini (1997), “a qualidade fornece bases para a Qualidade Total”. E os três modelos básicos de estruturação ambiental da produção da qualidade podem explicar melhor este conceito.

a) qualidade *in line*: refere-se à ênfase da qualidade no processo produtivo. Está relacionada com ausência de defeitos e especificações de projeto. As restrições do modelo *in-line* ocorrem quando nem sempre o cliente é considerado e quando enfatiza algumas atividades da empresa em detrimento das demais.

b) qualidade *off-line*: não ligadas diretamente ao processo produtivo, porém relevantes para a adequação ao uso do produto.

c) qualidade *on-line*: relacionada às preferências do cliente em adequá-las à fabricação do bem ou serviço. Opera primeiro com a qualidade de projeto do produto em seguida em função da dinâmica do mercado, reajustando o processo produtivo.

Stylianou e Kumar (2000) descrevem como diferentes *stakeholders* - especialista de domínio -, normalmente, têm percepções diferentes da importância das várias dimensões e atributos da qualidade. Por exemplo, os analistas de sistemas estão preocupados principalmente com a infraestrutura, software e qualidade dos dados, os usuários do sistema estão preocupados com os efeitos do sistema sobre as tarefas e processos de negócio. Gerente sênior está preocupado com a qualidade administrativa, do plano de processo em termos do alinhamento com a estratégia e desempenho organizacional.

Como o processo tem maior efeito sobre Q&P (Qualidade e Produtividade) liberada por uma organização, para Jalote (1999), um caminho para melhorar a Q&P é melhorar o processo usado pela organização. Esta relação conhecida como triângulo da qualidade, depende de três fatores: processo, pessoas e tecnologia.

A norma ISO 8402, citado por Tsukumo *et all* (1997), define Qualidade como “a totalidade de características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas”.

Necessidades explícitas são aquelas expressas na definição de requisitos propostos pelo produtor. Esses requisitos definem as condições em que o produto deve ser utilizado, seus objetivos, funções e o desempenho esperado.

As necessidades implícitas são aquelas que, embora não expressas nos documentos do produtor, são necessários para o usuário. Estão incluídos, nesta classe, tanto requisitos que não precisam ser declarados por serem óbvios (p. ex. “a caneta escreve”) como aqueles requisitos que não são percebidos como necessários no momento em que o produto foi desenvolvido, mas que pela gravidade de suas conseqüências devem ser atendidos (p. ex. mesmo em condições não previstas de erro ou má operação, um sistema de administração hospitalar não pode provocar a morte de pacientes).

A norma que representa a atual padronização mundial para a qualidade de produtos de software foi publicada em 1991 e chama-se ISO / IEC 9126. Ela é uma das mais antigas da área de qualidade de software e possui sua tradução para o Brasil como NBR 13596, publicada em agosto de 1996 (INTHURN, 2001).

A ISO / IEC 9126 / NBR 13596 lista o conjunto de características que devem ser verificadas em um software para que ele seja considerado um “software de qualidade”. Esta norma abrange seis grandes grupos de características. Além disso, cada grupo de característica pode ser dividido em algumas subcaracterísticas conforme Quadro 1.

Quadro 1: Características e subcaracterísticas - norma ISO / IEC 9126 / NBR 13596

Característica	Subcaracterística	Pergunta chave para a Subcaracterística
1 Funcionalidade - Satisfaz às necessidades?	Adequação	1.1. Faz o que é apropriado?
	Acurácia (Precisão)	1.2 Faz o que foi proposto da melhor forma?
	Interoperabilidade	1.3 Interage com outros sistemas?
	Conformidade	1.4 Esta de acordo com as normas, leis, etc?
	Segurança de acesso	1.5 Permite acesso não autorizado aos dados?
2 Confiabilidade – É imune a falhas?	Maturidade	2.1 Apresenta frequentemente falhas?
	Tolerância a falhas	2.2 Ocorrendo falhas, reage de forma apropriada?
	Recuperabilidade	2.3 Recupera dados em caso de falha?
3 Usabilidade – É fácil de usar?	Inteligibilidade	3.1 É fácil entender o conceito e a aplicação?
	Apreensibilidade (aprendizagem)	3.2 É fácil aprender e usar?
	Operacionalidade	3.3 É fácil de operar e controlar?
4 Eficiência – É rápido?	Tempo	4.1 O tempo de resposta é rápido?
	Recursos	4.2 Utiliza poucos recursos?
5 Manutenibilidade – É fácil de modificar?	Analisabilidade (Legibilidade)	5.1 É fácil de encontrar uma falha, quando ocorre?
	Modificabilidade	5.2 É fácil modificar e adaptar?
	Estabilidade	5.3 Há grande risco quando se faz alterações?
	Testabilidade	5.4 É fácil testar quando faz alterações?
6 Portabilidade – É fácil de usar em outro ambiente?	Adaptabilidade	6.1 É fácil adaptar a outros ambientes?
	Capacidade para ser instalado	6.2 É fácil instalar em outros ambientes?
	Conformidade	6.3 Está de acordo com padrões de portabilidade?
	Capacidade para substituir?	6.4 É fácil usar para substituir outro?

Fonte: Inthurn, 2001

2.1.2 Software

Em 1970, menos de 1% das pessoas poderiam descrever inteligentemente o que significava “programa de computadores”. Hoje, a grande maioria dos profissionais e o público em geral entendem o que seja software (PRESSMAN, 2002).

Para Pressman (2002), software pode ser definido como:

- ? Instruções que são “programas de computador” que quando executadas, produzem a função e o desempenho desejados.
- ? estruturas de dados que possibilitem que os programas manipulem a informação adequadamente.
- ? documentos que descrevem a operação e o uso dos programas.

Sommerville (2003) define software como programas e documentação associada. Produtos de software podem ser desenvolvidos para um cliente específico ou para o mercado.

O software é um elemento de sistema lógico, e não físico. Portanto o software tem características que são consideravelmente diferentes do hardware.

As características principais do software segundo Pressman (2002) são:

- ? O software é desenvolvido, ou passa por um processo de engenharia, não é manufaturado no sentido clássico. A alta qualidade é alcançada mediante um bom projeto. Os custos do software são concentrados na engenharia, isto significa que projetos de software não podem ser gerenciados como se fossem projetos de fabricação.
- ? O software não se “desgasta”. O software não é sensível aos males ambientais que causam o desgaste do hardware.
- ? Apesar da indústria estar se movendo em direção a montagem baseada em componentes, a maior parte de software continua a ser construída sob encomenda. Um componente de software deve ser projetado e implementado de forma que possa ser reusado em muitos programas diferentes.

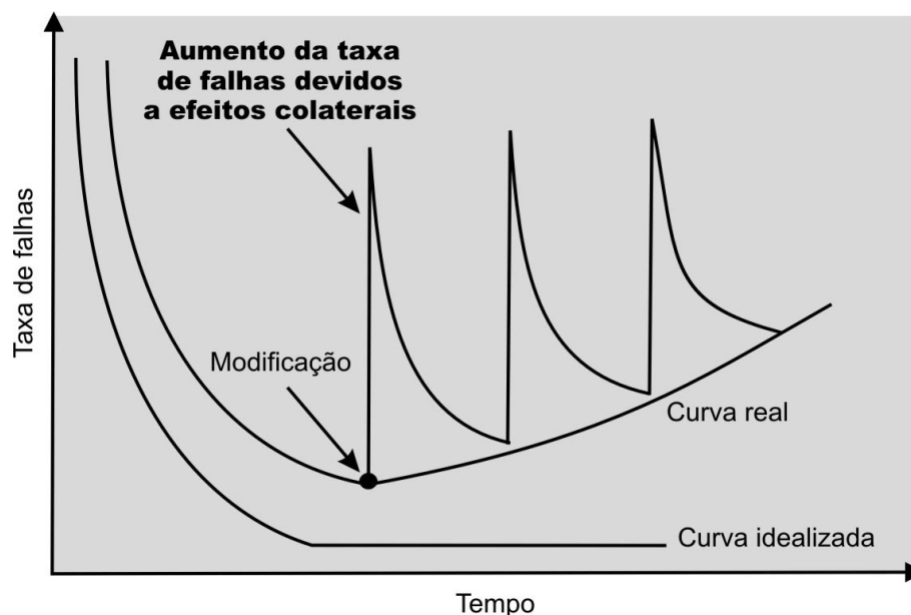


Figura 1: Curvas de falhas idealizada e real para o software

Fonte: Pressman, 2002

De acordo com Basili e Caldiera (1993) o produto software apresenta quatro características principais:

- a) O software é uma combinação lógica de partes invisíveis: a qualidade bem como a combinação de partes depende de uma adequada estruturação lógica sob uma documentação precisa e de fácil entendimento desta estrutura.
- b) O software é desenhado para uso de aplicações, das quais é esperado um a evolução contínua: a qualidade da aplicação do software depende de um entendimento conceitual preciso das necessidades do usuário.
- c) O software é desenvolvido e não produzido: cada produto de software é como um protótipo.
- d) O software é a tecnologia baseada no ser humano: a qualidade do produto de software é dependente do envolvimento dos indivíduos, o uso apropriado das habilidades, a satisfação individual e a motivação são as chaves para o melhoramento da qualidade e produtividade.

Para Sommerville (2003) características de um software bem projetado são apresentados no Quadro 2.

Quadro 2: Atributos essenciais de um bom software

Característica do produto	Descrição
Facilidade de manutenção	O software deve ser escrito de modo que possa evoluir para atender às necessidades mutáveis dos clientes. Esse é um atributo crucial, porque as modificações em um software são uma consequência inevitável de um ambiente de negócios em constante mutação.
Nível de confiança	O nível de confiança do software tem uma gama de características que incluem confiabilidade, proteção e segurança. O software confiável não deve ocasionar danos físicos ou econômicos, no caso de um defeito no sistema.
Eficiência	O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. A eficiência, portanto, inclui a rapidez de resposta, o tempo de processamento a utilização da memória, entre outros.
Facilidade de uso	O software deve ser utilizável, sem esforços indevidos, pelo tipo de usuário para quem foi projetado. Isso significa que ele deve dispor de uma interface apropriada com o usuário e de documentação adequada.

Fonte: Sommerville, 2003, p. 12

Os produtos de software, assim como os serviços que eles fornecem, têm uma série de outros atributos associados que refletem a qualidade do software. Esses atributos não são diretamente relacionados ao que o software faz, eles refletem o seu comportamento quando em funcionamento, a estrutura e a organização do programa fonte e a documentação associada. Como exemplo desses atributos temos o tempo de resposta do software à consulta de um usuário e a facilidade de compreensão do código de programa. O conjunto específico de atributos que se pode esperar de um sistema de software depende de sua aplicação. Portanto, um sistema bancário tem que ser seguro, um jogo iterativo deve ter resposta rápida um sistema de telefonia precisa ser confiável e assim sucessivamente.

2.1.3 Os Diversos Conceitos de Qualidade de Software

Nossa vida é afetada pela qualidade de software, ela foi criada porque a viabilidade da organização depende muitas vezes dela. Então devemos questionar o que os autores de qualidade e os clientes pensam sobre a qualidade de software.

A maioria das pessoas acredita que qualidade é importante e que pode ser melhorada. Companhias e países continuam a investir muito tempo, dinheiro e esforço em melhorias da qualidade de software. Mas, nós deveríamos determinar se

essas iniciativas têm afetado diretamente e melhorado a qualidade do software (KITCHENHAM e PFLEEGER, 1996).

Pressman (2002), define qualidade de software como: “satisfação aos requisitos funcionais e a desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente desenvolvido”. Para Pressman (2002, p. 500), a definição serve para enfatizar três pontos importantes:

- a) Os requisitos de software são a base a partir da qual a qualidade é medida. A falta de conformidade aos requisitos significa falta de qualidade.
- b) Padrões especificados definem um conjunto de critérios de desenvolvimento que orientam a maneira segundo a qual o software é construído. Se os critérios não forem seguidos, provavelmente resultará em falta de qualidade.
- c) Há um conjunto de “requisitos implícitos” que freqüentemente não são mencionados (p. ex., o desejo de facilidade de uso). Se o software satisfaz seus requisitos explícitos, mas deixar de cumprir seus requisitos implícitos, a qualidade de software será suspeita.

A qualidade de software é uma combinação complexa de fatores que variarão de acordo com diferentes aplicações e clientes que as solicitam (PRESSMAN, 2002).

Na opinião de Fournier (1994) qualidade de software não somente significa “conformidade com os requisitos”, mas abrange uma definição muito mais ampla. Um dos méritos dessa definição reside no fato das questões de qualidade não estarem mais arbitrariamente restritas ao próprio produto de software. Qualidade, então, engloba todo pacote de sistema que está sendo criado, incluindo os processos de elementos de suporte, como documentação de sistemas, treinamento da equipe e até mesmo o suporte pós-instalação – em outras palavras, todos os itens que atendam às expectativas de nossos clientes, quaisquer que sejam, além do próprio produto de software. Desta forma a qualidade implica satisfazer o cliente e não somente atender aos requisitos.

A Qualidade Total de Sistemas de Informações, de acordo com Stylianou e Kumar (2000), é um conceito com várias dimensões como:

- a) Qualidade de Infraestrutura A qualidade de infraestrutura (hardware e software – apropriado a um hardware específico) que é software instalado e mantido

pelo Sistema de Informação – inclui, por exemplo, a qualidade das redes e sistemas de software.

- b) Qualidade software A qualidade de aplicações de software construído, mantido ou apoiado pelo Sistema de Informação.
- c) Qualidade Dados A qualidade de dados que entram em vários sistemas de informações.
- d) Qualidade de Informação Qualidade da saída resultante do sistema de informações. Em muitos casos, a saída de um sistema é entrada de um outro sistema. Neste aspecto, qualidade da informação está relacionada com qualidade de dados.
- e) Qualidade da Administração A qualidade da gestão da função do sistema de informação, inclui a qualidade de orçamento, planejamento e programação de atividades.
- f) Qualidade do Serviço O componente da qualidade de serviço da função do sistema de informação, inclui a qualidade de processos de apoio ao cliente tais como aquelas relacionadas a *help desk* (equipe de suporte ao atendimento).

Freeman (*apud* ROCHA; MALDONADO; WEBER, 2001), faz uma distinção entre qualidade básica e qualidade extra. Em qualidade básica ele relaciona: funcionalidade, confiabilidade, facilidade de uso, economia e segurança de uso. Em qualidade extra, flexibilidade, facilidade de reparo, adaptabilidade, facilidade de entendimento, boa documentação e facilidade de adicionar melhorias. Mas é importante salientar que dar prioridade a essas qualidades depende de cada caso e o custo de cada uma dessas qualidades.

A escolha dos métodos que serão utilizados no processo de produção é uma escolha gerencial ligada à qualidade, pois a qualidade resultante será função dos métodos de produção escolhidos e seguidos. Outro fator importante é que a produção de software é um processo que envolve, como parte fundamental, seres humanos. As tecnologias de produção de software, nas quais estão incluídas as tecnologias de gerência, têm por objetivo automatizar o máximo a produção de software, mas ainda são fundamentalmente dependentes da qualidade das equipes.

Basili e Caldiera (1993) identificaram que o melhoramento da qualidade de software consiste em seis passos:

1. Caracterização do meio ambiente;
2. Conjunto de metas;
3. Escolha dos processos;
4. Execução dos processos;
5. Análise dos dados dos processos;
6. Pacote de experiências.

David Garvin estudou como a qualidade é percebida em vários domínios, incluindo a filosofia, economia, marketing e operações de gerenciamento. Garvin concluiu que a “qualidade é um conceito complexo e multifacetado” que pode ser descrito em cinco diferentes perspectivas (PFLEEGER, 2001):

- ? Visão transcendental pela qual qualidade é algo que se reconhece, mas não se define;
- ? Visão do usuário pela qual qualidade é atingir os objetivos;
- ? Visão da manufatura pela qual qualidade está relacionada a características do produto;
- ? Visão do produto pela qual a qualidade está relacionada a características inerentes do produto;
- ? Visão com base no valor pela qual qualidade depende do valor que o cliente está disposto a pagar por ela.

Portanto, qualidade depende do contexto do ambiente de negócios em que o software será utilizado, do domínio da aplicação.

A complexidade de se definir o que é qualidade de software vai ao encontro da(s) pessoa(s) que a define(m). Alguns exemplos seriam:

Quadro 3: Definições de qualidade de software

Definições de Alta Qualidade	Para Quem
Defeito zero	Usuários e gerentes de projeto
Ter um grande número de funções	Usuários (que poderão tirar proveito destas funções) e distribuidores (que acreditam que funções vendem produto).
Codificação elegante	Desenvolvedores e professores de ciências da computação.
Alto desempenho	Usuários e pessoal de vendas (que desejam submeter seus produtos a <i>benchmark</i>).
Baixo custo de desenvolvimento	Usuários (que terão que comprar muitas cópias) e gerentes de projeto.
Desenvolvimento rápido	Usuários (que dependerão do software) e distribuidores (que querem chegar ao mercado antes de seus concorrentes)
Facilidade para o usuário	Usuários (esquecem de detalhes de interface, passam muito tempo ao dia utilizando o software).

Fonte: Weinberg, 1993

É comum associar qualidade a produtos finais, mas a qualidade está associada ao processo pelo qual os produtos ou serviços são realizados. De acordo com Fuggetta (2000) se o processo for bom o produto provavelmente será bom. Portanto a qualidade deve estar presente não só nos produtos produzidos, como também nos processos utilizados para gerar esses produtos.

2.1.4 Fatores Intervenientes na Qualidade de Software

Os fatores que afetam a qualidade do software podem ser categorizados em dois amplos grupos, de acordo com Pressman (2002, p. 501):

- a) fatores que podem ser medidos diretamente (p. ex., defeitos por ponto por função).
- b) fatores que podem ser medidos apenas indiretamente (p. ex., usabilidade ou manutenibilidade). Em cada caso medições devem ocorrer. Devemos comparar o software (documentos, programas e dados) a algum valor e chegar a uma indicação da qualidade.

McCall; Richards e Walters (*apud* PRESSMAN, 2002) propõem uma categorização útil de fatores que afetam a qualidade de software, eles concentraram em três aspectos importantes:

- ? características operacionais;
- ? habilidade de passar por modificações;

? adaptabilidade a novos ambientes;



Figura 2: Fatores de qualidade de software de McCall

Fonte: Pressman, 2002

- ? Correção Quanto um programa satisfaz sua especificação e preenche os objetivos da missão do cliente.
- ? Confiabilidade Quanto se pode esperar que um programa realize a função pretendida com a precisão exigida.
- ? Eficiência Quantidade de recursos de computação e código necessários para um programa realizar sua função.
- ? Integridade Quanto do acesso ao software ou dados por pessoas não autorizadas pode ser controlado.
- ? Utilização O esforço necessário para aprender, operar, preparar entradas e interpretar saídas de um programa.
- ? Manutenibilidade O esforço necessário para localizar e consertar um erro num programa.
- ? Flexibilidade O esforço necessário para modificar um programa em operação.
- ? Testabilidade Esforço necessário para testar um programa, a fim que ele realize a função esperada.
- ? Portabilidade Esforço necessário para transferir o programa de um ambiente de hardware ou software para outro.

- ? Reutilização Quanto de um programa (ou partes deles) pode ser reusado em outras aplicações – relativo ao empacotamento e escopo das funções que o programa realiza.
- ? Interoperabilidade Esforço necessário para acoplar um sistema a outro.

2.2 Qualidade no Processo de Desenvolvimento de Software

Processo de software, para Pressman (2002), pode ser entendido como um roteiro, uma série de passos previsíveis, que ajuda a criar a tempo um resultado de qualidade, fornecendo para uma atividade, estabilidade e controle à organização. Quando deixada sem controle torna-se bastante caótica.

2.2.1 Qualidade de Processo e de Produto

A melhoria da qualidade focaliza a identificação de produtos de boa qualidade, examinando os processos utilizados para desenvolver esses produtos e então, generalizando esses processos de forma que eles possam ser aplicados em diversos projetos.

Melhorar um processo de modo que os defeitos sejam evitados levará a produção de melhores produtos, na manufatura, onde a relação processo/produto é óbvia. Segundo Sommerville (2003) esse vínculo é menos óbvio quando o produto é intangível e dependente, até certo ponto, de processos intelectuais, que não podem ser automatizados. A qualidade do software é dependente do processo de projeto em que considerações individuais humanas são importantes. O processo utilizado, em algumas classes de produtos, é o determinante mais importante da qualidade do produto.

Na tese, Cockburn (2003) diz que características pessoais têm mais efeito, principalmente em aplicações inovadoras, na trajetória do projeto que metodologias ou processos usados. Se a equipe é composta somente de duas ou três pessoas então, a metodologia como um tópico distinto pode ser sem muita importância. Mas

se o tamanho da equipe cresce, então, questão de coordenação torna-se significativa e, portanto mais consideração deve ser dada à metodologia.

Se a equipe for inexperiente e menos capacitada, um bom processo poderá limitar o dano, mas, por si só, não produzirá software de alta qualidade (SOMMERVILLE, 2003).

Existem quatro fatores que podem afetar a qualidade como mostrados na Figura 3, mas a influência deles depende do tamanho e do tipo do projeto.

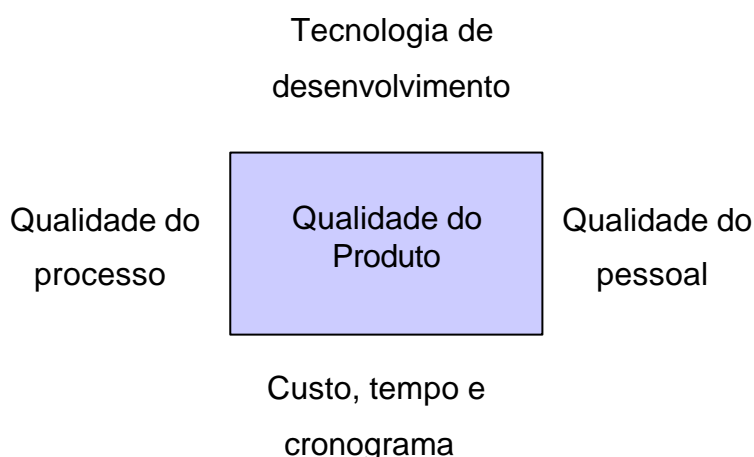


Figura 3: Principais fatores da qualidade de produtos de software

Fonte: Sommerville, 2003, p. 480

A relação entre a qualidade de processos e de produtos de software é complexa. A modificação do processo nem sempre conduz à melhoria da qualidade.

2.2.2 Processo de Desenvolvimento de Software

Para Fuggetta (2000), processo de software pode ser definido como um coerente conjunto de políticas, estrutura organizacional, tecnologias, procedimentos e artefatos que são necessários para conceber, desenvolver, dispor e manter um produto de software. Então, um processo de software explora um número de contribuições e conceitos como:

- a) Tecnologia de desenvolvimento de software: suporte tecnológico usado no processo. Certamente, para concluir as atividades de desenvolvimento

precisamos de ferramentas, infraestrutura e meio ambiente. É preciso tecnologia própria que torna possível e economicamente viável para criar um produto de software complexo necessário a nossa sociedade.

b) Métodos e Técnicas de desenvolvimento de software: orientação sobre como usar a tecnologia e concluir atividades do desenvolvimento de software. O apoio metodológico é essencial para explorar a tecnologia eficientemente.

c) Comportamento organizacional: a ciência das pessoas e organizações. Em geral, o desenvolvimento de software é realizado pelas equipes que têm que ser coordenadas e gerenciadas dentro de uma estrutura organizacional.

d) Economia e Marketing. Desenvolvimento de software não é um esforço auto-contido. Como qualquer outro produto, software deve endereçar a real necessidade do usuário dentro de um específico cenário de mercado. Conseqüentemente diferentes estágios de desenvolvimento de software (p. ex., especificação de requisitos e desenvolvimento) devem ser moldados dentro de um caminho que leve em conta o contexto onde software será vendido ou usado.

Do ponto de vista técnico, Pressman (2002) define o processo de software como um arcabouço para as tarefas que são necessárias para construir software de alta qualidade.

Abordagens importantes como normas ISO 9000 e a ISO IEC/12207 o modelo SW-CMM e o SPICE entre outros sugerem que melhorando o processo de software, pode-se melhorar a qualidade dos produtos segundo as considerações de Pfleeger (*apud* ROCHA; MALDONADO; WEBER, 2001). O objetivo principal destes modelos é auxiliar as empresas em melhorias organizacionais para o desenvolvimento de software, principalmente os gerenciais.

O processo de software é toda a atividade realizada durante as fases do ciclo de vida do software. É através do controle de cada uma destas atividades que se obtém um completo domínio, uma gerência eficaz e um planejamento adequado das ações tomadas, e a serem tomadas, durante todo o processo de desenvolvimento. A definição do processo de software é a descrição dessas atividades (ALBUQUERQUE, 1997).

Aplicar um processo de software não é uma tarefa fácil. Geralmente, as pessoas oferecem uma resistência natural à mudança, e implantar um novo processo de software numa organização demanda custo e tempo. Há alguns conceitos mal

interpretados que favorecem a resistência à adoção de um novo processo (HUMPHREY, 1995):

- ? “Só começamos com os requisitos totalmente definidos”. Existe uma falsa cultura de que análise de requisitos é tarefa do cliente e que o desenvolvimento só deve começar com todos os requisitos definidos.
- ? “Se passar nos testes, então está OK”. Esta é uma visão minimalista dos requisitos de qualidade a serem satisfeitos.
- ? “Os problemas são de ordem técnica”. Existem muitos problemas técnicos devido a ferramentas e linguagens de programação.
- ? “Qualidade de software não pode ser medida”. Até que se prove o contrário, esta afirmação é errada. O que existe são parâmetros difíceis de serem medidos, alguns já estão bastante consolidados (GRADY, 1992).

De acordo com Humphrey e Paulk (*apud* ROCHA; MALDONADO; WEBER, 2001), o processo de software envolve métodos, ferramentas e pessoas para um funcionamento satisfatório do processo, ele deve possuir:

- ? procedimentos e métodos que descrevam a relação entre as tarefas;
- ? ferramentas e equipamentos que dêem suporte a realização das tarefas, simplificando e automatizando o trabalho;
- ? pessoas com perfis adequados, treinadas nos métodos e nas ferramentas para poderem realizar as atividades adequadamente.

Para funcionar de maneira eficaz, esse conjunto deve estar integrado harmoniosamente, como mostrado na Figura 4.

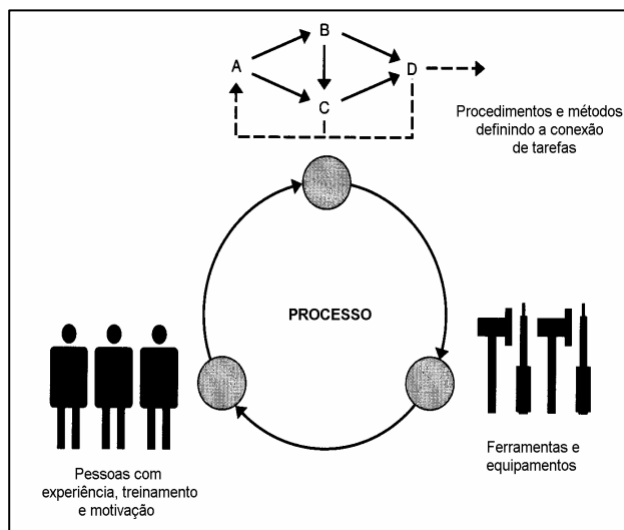


Figura 4: Processo de software

Fonte: adaptado de Paulk, 1995c

Para Paulk (2002) pessoas, tecnologias, modelos e padrões são importantes:

- a) Pessoas são a grande vantagem - reter bons profissionais que a empresa possui e contratar as melhores pessoas que você pode.
- b) Tecnologia que suporta eficaz e eficientemente engenheiros de software - ferramenta eficaz e tecnologia apropriada disponível para pessoas que as necessitam.
- c) A vantagem principal para construir capacidade hoje - modelos e padrões para o controle da qualidade e melhoria do processo.

Alguns dos modelos como o SW-CMM e SPICE são específicos para o software outros, como a ISO 9000 são aplicados aos mais variados tipos de produtos, pois visam o processo da organização (ALBUQUERQUE, 1997).

Em face dessas análises, conclui-se que as empresas que desenvolvem software precisam estar continuamente em mudança para que se adaptem a novos cenários advindos da crescente complexidade do software a ser construído. A competição entre concorrentes é cada vez mais dura, o que faz com que as mudanças devam ser rápidas para acompanhar o mercado.

Segundo Vidotti (2003) essas mudanças só serão possíveis, se a organização conhecer detalhadamente o seu método de desenvolver software para que possa buscar a melhoria contínua nos seus processos. Isso levará as empresas a se manterem competitivas e duradouras.

Diferentes tipos de processo podem ser identificados e aplicados a uma variedade maior de empresas e projetos (SOMMERVILLE, 2003, p. 490).

a) Processos informais São processos em que não há um modelo estritamente definido. O processo utilizado é escolhido pela equipe de desenvolvimento. Os processos informais podem utilizar procedimentos formais, como o gerenciamento de configuração, mas os procedimentos a serem utilizados e as relações entre eles não são predefinidos.

b) Processos gerenciados São processos em que há um modelo definido de processo implantado. Eles são utilizados para orientar o processo de desenvolvimento. O modelo de processo define os procedimentos utilizados, os cronogramas e as relações entre os procedimentos.

c) Processos metódicos São processos que têm objetivos de melhoria inerentes. Há um orçamento específico para as melhorias de processo e os procedimentos em andamento para introduzir essas melhorias. Medições quantitativas de processo podem ser introduzidas, como parte dessas melhorias.

Essas classificações são úteis, pois servem de base para a melhoria de processo multidimensional (p. ex. um processo pode ser classificado como informal, metódico e de melhoria), ajudando as empresas a escolher um processo apropriado para diferentes tipos de desenvolvimento de produtos. Contudo, a classificação de processo reconhece que ela afeta a qualidade do produto, mas não supõe que ele seja sempre o fator dominante. Ele fornece base para a melhoria de diferentes tipos de processo (SOMMERVILLE, 2003). Hoje em dia, a maioria dos processos de software tem algum apoio de ferramentas CASE, portanto são processos apoiados.

Os processos de software são inerentemente complexos e envolvem um número grande de atividades, os processos também têm atributos ou características, como os produtos.

Característica do processo de acordo com Sommerville (2003) são as que seguem:

a) Facilidade de compreensão Até que ponto o processo está explicitamente definido e com que facilidade se pode compreender a definição do processo?

- b) Visibilidade As atividades de processo culminam em resultados nítidos, de modo que o progresso do processo seja externamente visível?
- c) Facilidade de suporte Até que ponto as atividades do processo podem ser apoiadas por ferramentas CASE?
- d) Aceitabilidade O processo definido é aceitável e utilizável pelos engenheiros responsáveis pela produção do produto de software?
- e) Confiabilidade O processo está projetado de tal maneira que seus erros sejam evitados ou identificados antes que resultem em erros no produto?
- f) Robustez O processo pode continuar, mesmo que surjam problemas inesperados?
- g) Facilidade de manutenção O processo pode evoluir para refletir os requisitos mutáveis da organização ou melhorias de processo identificadas?
- h) Rapidez Com que rapidez pode ser concluído o processo de entrega de um sistema, a partir de uma determinada especificação?

2.2.3 Estágios na Melhoria de Processo

De acordo com Sommerville (2003), existem importantes estágios na melhoria de processo como:

- a) Análise de processo envolve examinar os processos existentes e produzir um modelo específico para documentar e compreender o processo. A análise quantitativa antes e depois das mudanças terem sido introduzidas permite uma avaliação objetiva dos benefícios da mudança de processo.
- b) Identificação de melhoria identificar gargalos com relação a prazo, à qualidade e custo. A melhoria de processo deve focar a eliminação desses gargalos, propondo novos procedimentos, métodos e ferramentas para corrigir os problemas.
- c) Introdução de mudança de processo Implantar novos procedimentos métodos e ferramentas e integrá-los com outras atividades, devem ser compatíveis com outras atividades de processo e com os procedimentos e padrões organizacionais.

d) Treinamento em mudanças de processo Não é possível obter plenos benefícios das mudanças de processo, sem treinamento. É comum mudanças serem impostas (os gerentes e engenheiros podem rejeitá-las) sem treinamento os efeitos dessas mudanças resultem em degradação.

e) Ajuste de mudanças As mudanças não serão eficientes de imediato. Há necessidade de uma fase de ajuste, em que problemas menores sejam descobertos e modificações sejam introduzidas.

Segundo Sommerville (2003), é impossível introduzir muitas mudanças ao mesmo tempo. Além dos problemas de treinamento, torna-se impossível avaliar o efeito de cada uma delas no processo.

2.3 Modelos de Melhoria de Processos

A melhoria do processo não significa adotar métodos ou ferramentas específicos ou algum modelo de processo que tenha sido utilizado em outro lugar. Há fatores locais organizacionais, procedimentos e padrões que influenciam o processo, embora empresas que desenvolvem o mesmo tipo de software claramente tenham muito em comum. As melhorias de processos devem ser sempre vistas como uma atividade específica de uma organização ou uma parte de uma organização maior (SOMMERVILLE, 2003).

Shiller (1992, p. 26) faz uma citação de John Von Neumann para definição de modelos.

As ciências não tentam explicar, e até mesmo dificilmente tentam interpretar; elas principalmente constroem modelos. Entendemos por modelo uma construção matemática que, com a adição de certas interpretações verbais, descreve o fenômeno observado. A justificativa de tal construção matemática é somente e precisamente que se espera que ela funcione.

No tópico a seguir descrevem-se “conceitos” de alguns modelos de melhoria de processo no nível organizacional como a ISO 9000, SPICE, SW-CMM, GQM e QIP o qual servirá de base para uma posterior comparação, e para a consolidação da pesquisa, relacionando os processos ao modelo SW-CMM.

2.3.1 ISO 9000

A norma ISO 9000 (ANTONIONI, 1995) (DE CICCO, 2000) (ROCHA; MALDONADO; WEBER, 2001) (PAULK, 2002) é uma série de padrões de qualidade internacionais, desenvolvida pela ISO - *International Organization for Standardization*.

Estabelece um conjunto de requisitos necessários para assegurar que nos processos estabelecidos em uma organização resultem produtos que satisfaçam as expectativas de seus clientes. Os padrões ISO 9000 são escritos genericamente e provêm modelos ao invés de especificações. Eles descrevem o que deve ser realizado e não como deve ser feito. As principais normas são apresentadas a seguir:

- ? ISO 9001: Modelo para garantia da qualidade em projeto, desenvolvimento, produção, instalação e assistência técnica (serviços).
- ? ISO 9002: Modelo para garantia da qualidade em produção e instalação.
- ? ISO 9003: Modelo para garantia da qualidade em inspeção e ensaios finais.
- ? ISO 9000-1: Diretrizes para escolher entre as normas ISO 9001, 9002 e 9003.
- ? ISO 9000-3: Orientação para a aplicação da ISO 9001 em software.

A norma ISO 9000-3 traz os roteiros para aplicar a ISO 9001 especificamente na área de desenvolvimento, fornecimento e manutenção de software. Todas as orientações giram em torno de uma “situação contratual”.

O processo de certificação de uma empresa de software segundo as normas ISO 9001/9000-3 segue um conjunto de passos bem definidos, onde a empresa estabelece o seu sistema de qualidade e realiza um processo de certificação junto a um órgão certificador credenciado. Posteriormente o certificado é mantido através de auditorias periódicas.

Inicialmente, a revisão 2000 das normas da série ISO 9000 baseou-se nos resultados de pesquisas realizadas pelo comitê técnico, ISO/TC 176, com usuários de 40 países, que resultou em oito princípios (PRESSMAN, 2002), quais sejam:

1. foco no cliente;
2. liderança;

3. envolvimento das pessoas;
4. abordagem de processos;
5. abordagem de sistemas;
6. melhoria contínua;
7. decisão com base em fatos; e
8. relação benéfica cliente/fornecedor.

2.3.1.1 ISO 9001:2000

O padrão especificado na série ISO 9000 concernente às empresas de software é a ISO 9001 (ROCHA; MALDONADO; WEBER, 2001). A grande modificação introduzida pela versão 2000 da ISO 9001 de acordo com Xavier (*apud* ROCHA; MALDONADO; WEBER, 2001), está na mudança do objetivo dos sistemas de gestão da qualidade ISO 9000 da garantia da qualidade, ou seja, do atendimento aos requisitos especificados pelo cliente, para satisfação do cliente, para o atendimento não somente dos requisitos explícitos, especificados pelo cliente, mas dos requisitos implícitos.

O desafio atual para as empresas, inclusive as que detêm certificados, é de evoluir o seu sistema de gestão da qualidade ISO 9000 para atender aos novos requisitos introduzidos na versão 2000 orientados para a satisfação do cliente conforme enuncia Xavier (*apud* ROCHA; MALDONADO; WEBER, 2001).

As principais mudanças que ocorreram da ISO de 1994 para 2000 (PAULK, 2002):

- a) estão suspensos a ISO 9002 e 9003
- b) nova arquitetura (passou de 20 cláusulas para 23)
- c) promover a adoção da abordagem de processo
- d) explicitar pedido e melhorar continuamente para satisfação do cliente e
- e) simplificação da terminologia (p. ex. Subcontrato para fornecedor, inspeção e teste para verificação e validação do produto etc).

Quadro 4: Principais cláusulas da ISO 9001:2000

<p>1 Responsabilidade da administração</p> <p>1.1 Comprometimento da administração</p> <p>1.2 Foco no cliente</p> <p>1.3 Política da qualidade</p> <p>1.4 Planejamento</p> <p>1.4.1 Objetivos da qualidade</p> <p>1.4.2 Planejamento da qualidade</p> <p>1.5 Administração</p> <p>1.5.1 Generalidades</p> <p>1.5.2 Responsabilidade e autoridade</p> <p>1.5.3 Representante da administração</p> <p>1.5.4 Comunicação interna</p> <p>1.5.5 Manual da qualidade</p> <p>1.5.6 Controle de documentos</p> <p>1.5.7 Controle de registros</p> <p>1.6 Análise crítica do sistema</p>	<p>2 Realização do produto</p> <p>2.1 Planejamento dos processos de realização</p> <p>2.2 Processos relativos ao cliente</p> <p>2.3 Projeto e/ou desenvolvimento</p> <p>2.4 Aquisição</p> <p>2.5 Operações de produção e serviços</p> <p>2.5.1 Controle das operações</p> <p>2.5.2 Identificação e rastreabilidade</p> <p>2.5.3 Propriedade do cliente</p> <p>2.5.4 Preservação do produto</p> <p>2.5.5 Validação de processos</p> <p>2.6 Controle de dispositivos de medição e monitoramento.</p>
<p>3 Gerenciamento dos recursos</p> <p>3.1 Provisão de recursos</p> <p>3.2 Recursos humanos</p> <p>3.2.1 Designação de pessoal</p> <p>3.2.2 Treinamento, conscientização e competência</p> <p>3.3 Infra-estrutura</p> <p>3.4 Ambiente de trabalho</p>	<p>3 Medição, análise e melhoria</p> <p>4.1 Planejamento</p> <p>4.2 Medição e monitoramento</p> <p>4.2.1 Satisfação do cliente</p> <p>4.2.2 Auditorias internas</p> <p>4.2.3 Medição e monitoramento dos processos</p> <p>4.2.4 Medição do produto</p> <p>4.3 Controle de não conformidade</p> <p>4.4 Análise de dados</p> <p>4.5 Melhoria</p> <p>4.5.1 Planejamento para melhoria contínua</p> <p>4.5.2 Ação corretiva</p> <p>4.5.3 Ação preventiva</p>

Fonte: Paulk, 2002 e Rocha *et al*, 2001

Desde a versão original de 1987, um ponto forte das normas ISO 9000 foi a padronização dos requisitos mínimos da garantia da qualidade, que devem ser atendidos pelos fornecedores de produtos e serviços. Outro ponto forte do modelo ISO 9000 é a sua flexibilidade, permitindo ser adaptado ou complementado (p. ex.: CMM e SPICE) para atender aos requisitos setoriais específicos. Enfim, qualquer que seja a abordagem utilizada para a melhoria dos processos de software, as empresas de software no Brasil precisam, segundo Weber (*apud* ROCHA; MALDONADO; WEBER, 2001):

- ? investir em métodos de prevenção de defeitos;
- ? cultivar o hábito de medir os seus processos de software;
- ? aprender a identificar as causas dos problemas ou defeitos;

? saber agir corretiva e preventivamente para eliminar esses problemas ou defeitos e, principalmente, as suas causas.

Isso contribuirá para o desenvolvimento de uma indústria de software de classe mundial no Brasil.

2.3.2 QIP – *Quality Improvement Paradigm*

O QIP – *Quality Improvement Paradigm* (BASILI; CALDIERA; ROMBACH, 1994) foi desenvolvido na Universidade de Maryland (EUA) em cooperação com o SEL - *Software Engineering Laboratory* da NASA – *National Aeronautics and Space Administration*, visando o melhoramento contínuo e a execução sistemática dos projetos de software.

O princípio básico é a necessidade de identificação, através de mensuração, dos problemas particulares e dos pontos fortes e fracos dos produtos e processos de software estabelecidos em uma organização. Baseado nesse conhecimento, áreas potenciais para a melhoria são identificadas e são estabelecidas ações de melhoria. As modificações são acompanhadas por mensuração, avaliando os resultados na qualidade dos produtos e processos de software da organização.

O QIP objetiva, também, a modelagem do processo de software, a captura de experiências adquiridas no processo de melhoria da qualidade e a reutilização destas experiências em projetos futuros.

O QIP é um processo iterativo e contínuo, que inclui o planejamento, execução e avaliação de melhorias em organizações de software, baseado em mensuração. Identifica os produtos e processos de software, e estabelece ações de melhoria cujos resultados são acompanhados e avaliados.

a) **Caracterização:** Consiste no entendimento dos processos e metas de negócio estabelecidos na organização, conforme os fatores que determinam a situação. Caracteriza o contexto da organização e dos projetos através de modelos, dados, intuição, entre outros.

b) **Definição de objetivos:** Estabelece um conjunto de objetivos quantitativos para a melhoria do desempenho dos projetos e da organização, dentro de expectativas razoáveis conforme a caracterização.

- c) Escolha do processo: Define os processos apropriados para melhoria, conforme a caracterização e definição de objetivos. Escolhe os métodos, técnicos e ferramentas usadas no projeto conforme as metas e características.
- d) Execução: Executa o processo de construção de produtos, e em paralelo realiza a realimentação do processo através da coleta e análise de dados referentes ao alcance dos objetivos.
- e) Análise: No fim do projeto, avalia os dados e informações coletadas durante a realização do projeto para a verificação das práticas realizadas, problemas encontrados e registros elaborados. Definem recomendações para a melhoria de futuros projetos.
- f) Acondicionamento: Consolida as experiências adquiridas no projeto na forma de novos modelos ou na forma de conhecimento estruturado para armazenamento e disponibilização em futuros projetos.

O ciclo QIP implementa dois caminhos de realimentação para o software:

- a) Ciclo de controle: Realimentação para o projeto do software durante a fase de execução, com objetivo de controlar o projeto e prevenir e solucionar problemas.
- b) Ciclo de capitalização: Realimentação para a organização de software, providenciando informações analíticas da performance de projeto e acumulando experiências reutilizáveis em futuros projetos.

A aplicação de melhoria de qualidade QIP requer uma apropriada determinação de linhas base de modelos e objetivos para os processos e produtos de software, e da avaliação sob as diversas perspectivas do contexto de negócios da organização através de mensuração.

2.3.3 GQM – *Goal/Question/Metric*

A mensuração é uma tecnologia chave do programa de melhoramento. Proporciona a definição e implementação de objetivos operacionais mensuráveis de melhoria, e especifica precisamente e explicitamente a tarefa de análise de medidas através de objetivos.

O GQM, em direção de cima para baixo, define um programa de mensuração através de objetivos, questões, modelos e medidas, e, em direção de baixo para cima, realiza a análise e interpretação dos dados de mensuração.

O processo de mensuração permite a organização alcançar os seguintes resultados:

- a) Compreensão de produtos e processos de software, que é crucial para planejar, controlar e melhorar o desenvolvimento e a manutenção de software. Permite o estabelecimento de linha-base quantitativa com respeito aos produtos e processos de software da organização e definição de modelos de qualidade para o exame de fatores de contexto que tem impacto sobre produtos e processos.
- b) Gerenciamento de projetos de software, com planejamento de novos projetos (p.ex., por modelos de distribuição de esforços), controle do projeto através de comparação de valores estimados com valores coletados e validação de modelos organizacionais.
- c) Melhoramento contínuo, realizado através da modificação dos processos gerenciais ou técnicos e pela introdução de novas tecnologias. Permite a identificação de pontos fortes e fracos e de áreas de melhoramento potencial, para ações de melhoria selecionadas, executadas e avaliadas quantitativamente.

2.3.3.1 Processo GQM

Estabelecendo um programa de mensuração GQM (FUGGETTA; LAVAZZA; MORASCA, 1998), realizando a medição, coletando e analisando dados é um processo complexo. GQM sugere um processo específico para guiar os engenheiros de software na criação e operação de um programa de mensuração. A estrutura do processo GQM reflete as fases do QIP, desde a criação e operação do programa de mensuração são componentes essenciais para qualquer atividade de melhoria.

Estudo prévio: É a coleta de informações pertinentes a um programa de mensuração na organização, com objetivo principal de suportar o planejamento de programa de mensuração e a captura e contextualização de experiências adquiridas.

Identificação de metas GQM: Definição das metas de mensuração baseadas nas metas de negócios e estratégicas da organização, bem como das metas organizacionais de melhoramento de problemas conhecidos.

Desenvolvimento de plano GQM: Identifica informações necessárias para atingir as metas de mensuração. Refinar metas de mensuração em perguntas, modelos e medidas para cada meta GQM.

Desenvolvimento do plano de mensuração: Integra apropriadamente medições no plano de projeto de software. Determinar quando, como e por quem as medidas estabelecidas no plano GQM podem ser coletadas de acordo com o processo de software. Instrumentos de coleta de dados são desenvolvidos.

Coleta e validação de dados: Na execução do programa de mensuração os dados são coletados conforme com o plano de mensuração. Os dados são validados e armazenados num banco de mensuração.

Análise dos dados: Os dados coletados são analisados no contexto do plano GQM para identificar padrões e relações entre atributos, permitindo o estabelecimento de linhas-base e identificação de problemas.

Captura de experiências: É a captura de experiências de forma explícita para reutilização em projetos futuros. Envolve a elaboração de políticas e guias de gerência de software, padrões e procedimentos, material de treinamento, ferramentas de suporte e relatórios de estudo.

Apesar do processo GQM aqui apresentado como processo linear seqüencial de atividades, na prática, ciclos são possíveis para refinar e revisar os resultados das atividades previamente completadas de acordo com as novas informações que foram identificadas.

O planejamento de um programa de mensuração baseado em GQM é complexo, necessita muito conhecimento e é inclinado a erros. Muitos problemas críticos podem ocorrer durante sua aplicação na prática. O esforço relacionado é alto. Embora seja um método bem conhecido uma orientação detalhada para o estabelecimento de um programa em um ambiente industrial é ainda limitado. Também, existem poucos relatórios de experiências sobre a aplicação do GQM em casos industriais. Finalmente o suporte tecnológico para o GQM ainda é inadequado conforme enuncia Fuggetta *et al* (1998).

2.3.4 SPICE - Melhoria do Processo de Software e Determinação da Capacidade

SPICE (SALVIANO; CUNHA; CÔRTEZ; OLIVEIRA, 2001) (ISO, 2003) é o nome do projeto de elaboração norma ISO/IEC 15504 para avaliação de processos de software. 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.

O projeto SPICE originou-se da pressão do aparecimento de diversos modelos de qualidade específicos para o desenvolvimento de software. Dentre eles o SW-CMM e a ISO/IEC 12207. Da ISO/IEC 12207 herdou a arquitetura dos processos do ciclo de vida do software e do SW-CMM herdou o conceito de níveis de maturidade de processos. Diferentemente do SW-CMM, que associa áreas de processos aos níveis de maturidade, a ISO 15504 tem uma abordagem de duas dimensões, que permite associar níveis de maturidade a qualquer processo do ciclo de vida de software.

A abordagem de duas dimensões, de processo e capacidade de processo, é muito flexível e influenciou decisivamente na concepção de outros modelos criados posteriormente.

2.3.4.1 Objetivos da ISO/IEC 15504 (SPICE)

A ISO/IEC 15504 (SALVIANO; CUNHA; CÔRTEZ; OLIVEIRA, 2001) presta-se à realização de avaliações de processos de software com dois objetivos principais:

- a) a melhoria dos processos.
- b) determinação da capacidade de processos de uma organização.

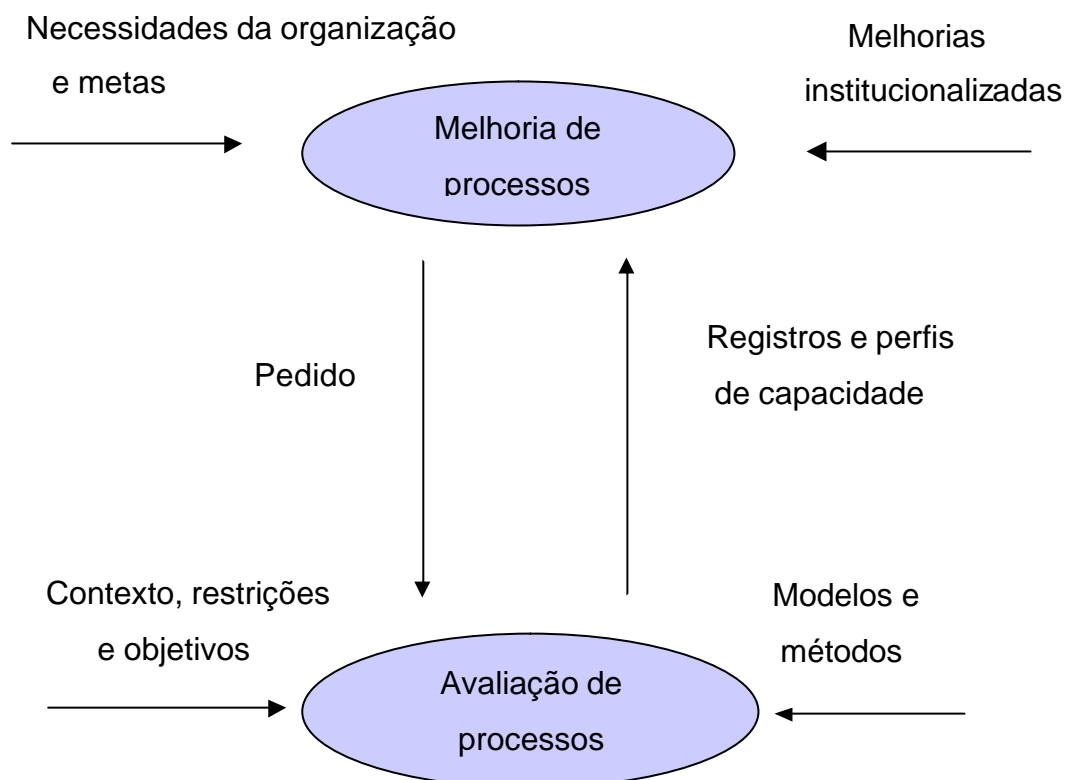


Figura 5: Uso da ISO 15504 para melhoria de processos (ISO,1998)

Fonte: Rocha *et al*, 2001

Se o objetivo for melhoria dos processos, a organização pode realizar a avaliação gerando um perfil dos processos que serão usados para a elaboração de um plano de melhorias (Figura 5). A organização deve definir os objetivos e o contexto, bem como escolher o modelo e o método para a avaliação e definir objetivos de melhoria.

No segundo caso, a empresa tem o objetivo de avaliar um fornecedor em potencial, obtendo o seu perfil de capacidade (Figura 6) para isso ele deve definir os objetivos e o contexto da avaliação, os modelos e métodos de avaliação e os requisitos esperados. O perfil de capacidade permite ao contratante estimar o risco associado à contratação daquele fornecedor em potencial para auxiliar na tomada de decisão de contratá-lo ou não.

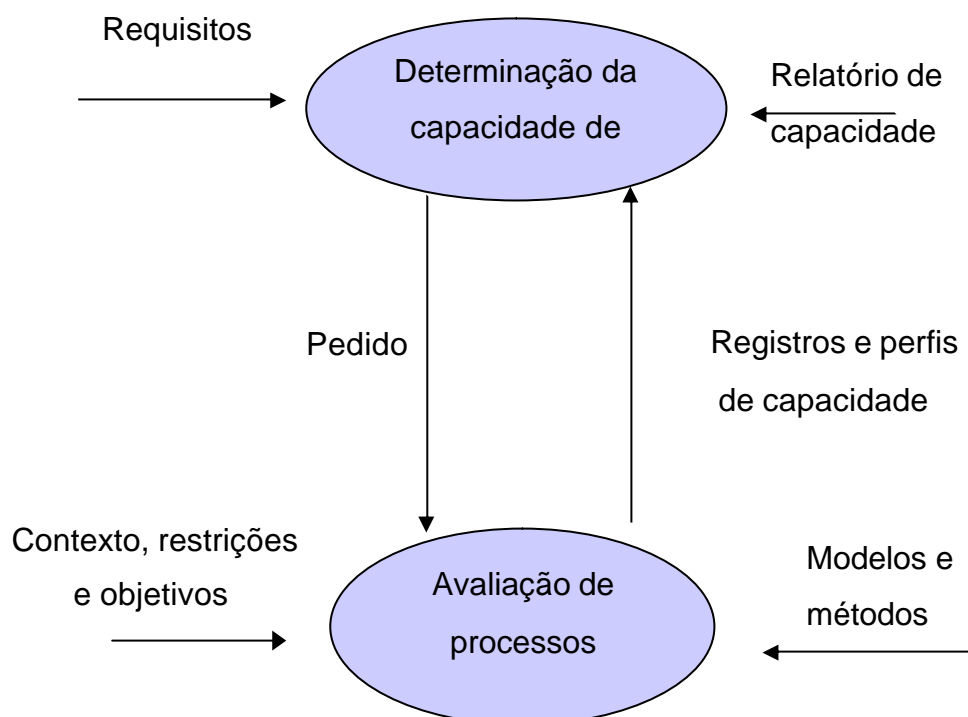


Figura 6: Uso da ISO 15504 para determinação da capacidade (ISO,1998)

Fonte: Rocha *et al*, 2001

O modelo de referência estabelecido na norma 15504 define duas dimensões: a dimensão de processo, que corresponde definição de um conjunto de processos considerados universais e fundamentais para a boa prática da engenharia de software, e a dimensão de capacidade, que corresponde à definição de um modelo de medição com base na identificação de um conjunto de atributos que permite determinar a capacidade de um processo para atingir seus propósitos gerando produtos de trabalho e os resultados estabelecidos conforme descreve Moreau (*apud* ROCHA; MALDONADO; WEBER, 2001).

A arquitetura apresenta as práticas de dois modos diferentes (ISO/IEC – parte 2):

Práticas genéricas aplicáveis a qualquer processo, que representa as atividades necessárias para gerenciar um processo e melhorar sua capacidade de desempenho.

Práticas básicas, que são atividades essenciais de um processo específico agrupados dentro de processos e categorias de processos por tipos de atividades.

Na decomposição por tipo de atividade 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.

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 atividades que funcionam em conjunto para fornecer um aumento na capacidade de desempenho de um processo. Cada nível fornece um aumento maior na capacidade daquele fornecido pelo seu predecessor no desempenho de um processo. Eles constituem um meio racional de progredir através de práticas genéricas.

Os níveis de capacidade fornecem dois benefícios: eles reconhecem dependências entre as práticas de um processo e ajudam uma organização a identificar quais melhorias podem desempenhar primeiro.

A dimensão de capacidade estabelece a graduação dessa escala em seis níveis com a seguinte caracterização:

Quadro 5: Níveis da Capacidade SPICE

<p>Nível 0 Não Desempenhado</p> <p>Descrição</p> <ul style="list-style-type: none"> ? Falha geral para desempenhar práticas básicas no processo. ? Identificação difícil dos produtos ou saídas do processo.
<p>Nível 1 Desempenhado Informalmente</p> <p>Descrição</p> <ul style="list-style-type: none"> ? Práticas básicas do processo são geralmente desempenhadas. ? O cumprimento das práticas básicas geralmente não é planejado e rastreado. ? O cumprimento das práticas básicas depende do conhecimento e esforço individual. ? Os produtos comprovam o cumprimento do processo. ? Há produtos identificáveis pelo processo.
<p>Nível 2 Planejado e Rastreado</p> <p>Descrição</p> <ul style="list-style-type: none"> ? O cumprimento das práticas básicas é planejado e rastreado. ? O cumprimento de acordo com procedimentos especificados é verificado. ? Funções dos produtos estão conforme especificado em padrões e requisitos. <p>A distinção básica em relação ao Nível 1, Desempenhado informalmente, é que o desempenho do processo é planejado e gerenciado e progride em direção a um processo bem definido.</p>
<p>Nível 3 Bem Definido</p> <p>Descrição</p> <ul style="list-style-type: none"> ? Práticas básicas são desempenhadas de acordo com um processo bem definido, utilizando padrões e processos documentados. <p>A distinção básica em relação ao Nível 2, Planejado e Rastreado, é que o processo deste nível é planejado e gerenciado utilizando um processo padrão em toda organização.</p>
<p>Nível 4 Quantitativamente Controlado</p> <p>Descrição</p> <ul style="list-style-type: none"> ? Medidas detalhadas do desempenho são analisadas e coletadas. ? Entendimento quantitativo da capacidade do processo e uma melhoria na habilidade de prever o desempenho. ? Desempenho gerenciado objetivamente. ? Qualidade dos produtos conhecida quantitativamente. <p>A distinção básica em relação ao Nível 3, Bem Definido, é que o processo definido é quantitativamente entendido e controlado.</p>
<p>Nível 5 Melhorando Continuamente</p> <p>Descrição</p> <ul style="list-style-type: none"> ? Metas de eficácia e eficiência quantitativas do processo para o desempenho são estabelecidas, baseadas na metas de negócio da organização. ? Melhoria contínua do processo. ? Realimentação quantitativa da execução de processos. ? Adoção de idéias e tecnologia Inovadoras. <p>A distinção básica em relação ao Nível 4, Quantitativamente Controlado, é que o processo definido e o processo padrão suportam refinamentos e melhorias contínuas, baseado no entendimento quantitativo do impacto das mudanças destes processos.</p>

Fonte: ISO/IEC (Parte 2)

Na dimensão de processo, os processos são agrupados em cinco categorias de acordo com o tipo de atividades que executam. Cada processo é descrito em termos de um propósito que exprime um único objetivo funcional. Satisfazer esse propósito é o primeiro passo na efetivação do mesmo. O modelo não define como, ou em que ordem, os propósitos de determinado processo podem ser alcançados. Esses

propósitos são avaliados pela constatação da execução das atividades tarefas e práticas necessárias para gerar os produtos de trabalho, pertinentes ao propósito estabelecido. As tarefas, atividades e práticas, bem como as características dos produtos de trabalho, são definidas como indicadores que demonstram se determinado processo é adequadamente praticado em um determinado nível de capacidade. Esses indicadores, que representam características do processo, são classificados de acordo com uma escala que expressa justamente o seu nível de capacidade.

Quadro 6: Práticas Genéricas para Nível 1 : Desempenhado Informalmente

	Característica Comum 1.1: Desempenhando Práticas Básicas
1.1.1	Desempenhar o processo. Desempenhar as práticas básicas do processo para prover produto do trabalho e/ou serviços ao cliente. Nota: O cliente do processo pode ser interno ou externo da organização.

Fonte: ISO/IEC (Parte 2)

Quadro 7: Práticas Genéricas para Nível 2 : Planejado e Rastreado

	Característica Comum 2.1: Planejando o desempenho
2.1.1	Alocar recursos. Alocar recursos adequados (incluindo pessoas) para execução do processo.
2.1.2	Assinalar responsabilidades. Assinalar responsabilidades para o desenvolvimento do produto e/ou provendo serviços para o processo.
2.1.3	Documentar o processo. Documentar a execução do processo dentro de padrões e/ou procedimentos.
2.1.4	Fornecer ferramentas. Fornecer ferramentas apropriadas para dar suporte na execução do processo.
2.1.5	Garantir treinamento. Garantir que realizações de processo individuais são apropriadamente treinado em como realizar o processo.
2.1.6	Planejar o processo. Planejar a execução do processo
	Característica Comum 2.2: Desempenho Disciplinado
2.2.1	Usar planos, padrões e procedimentos. Use planos, padrões documentados, e/ou procedimentos na implementação do processo.
2.2.2	Fazer gerenciamento de configuração. Por os produtos de trabalho dos processos de acordo com o controle de versão ou gerenciamento de configuração como apropriado.
	Característica Comum 2.3: Verificando o Desempenho
2.3.1	Verificar a concordância com o processo . Verificar a concordância com o processo com aplicação de padrões e/ou procedimentos.
2.3.2	Auditar os produtos Verificar a concordância dos produtos com aplicação de padrões e /ou requisitos.
	Característica Comum 2.4: Rastreado o Desempenho
2.4.1	Rastrear com medição. Rastrear o status do processo contra o plano usando medição.
2.4.2	Tomar ações corretivas. Tomar ações corretivas como apropriado quando o progresso variar significativamente do planejado.

Fonte: ISO/IEC (Parte 2)

Quadro 8: Práticas Genéricas para Nível 3 : Bem Definido

	Característica Comum 3.1: Definindo um Processo Padrão
3.1.1	Padronizar o Processo. Documentar processos padrões ou família de processo para a organização os quais descreve como implementar as práticas básicas do processo.
3.1.2	Ajustar o processo padrão. Ajustar os processos padrões organizacionais para criar uma definição de processo as quais endereça uma necessidade particular para um uso específico.
	Característica Comum 3.2: Desempenhando o Processo Definido
3.2.1	Usar um processo bem definido. Usar um processo bem definido na implementação do processo.
3.2.2	Desempenhar revisões pontuais. Desempenhar revisões por pares apropriado para processo dos produtos de trabalho.
3.2.3	Usar dados bem definidos. Usar dados executando o processo definido para gerenciar a definição do processo.

Fonte: ISO/IEC (Parte 2)

Quadro 9: Práticas Genéricas para Nível 4: Quantitativamente Controlado

	Característica Comum 4.1: Estabelecendo Metas de Qualidade Mensuráveis
4.1.1	Estabelecer Metas de Qualidade. Estabelecer metas de medição de qualidade para os produtos de trabalho padrão da organização.
	Característica Comum 4.2: Gerenciando objetivamente o desempenho
4.2.1	Determinar a capacidade do processo. Determinar a capacidade do processo quantitativamente
4.2.2	Usar a capacidade do processo. Tomar ações corretivas apropriadas quando o processo não executou dentro da capacidade do processo.

Fonte: ISO/IEC (Parte 2)

Quadro 10: Práticas Genéricas para Nível 5: Melhorando Continuamente

	Característica Comum 5.1: Melhoria da Capacidade Organizacional
5.1.1	Estabelecer Metas Eficazes do Processo. Estabelecer metas quantitativas para a melhoria efetiva do processo padrão, baseado nas metas de negócio da organização e da capacidade corrente do processo.
5.1.2	Melhorar continuamente o processo padrão. Melhorar continuamente o processo pela troca de família de processos padrão da organização e aumentar sua eficácia.
	Característica Comum 5.2: Melhorando a Eficácia do Processo
5.2.1	Executar análises causais. Executar análises causais de defeitos.
5.2.2	Eliminar causas de defeitos. Eliminar as causas de defeitos do processo definido, seletivamente.
5.2.3	Melhorar continuamente o processo definido. Melhorar continuamente o processo trocando o processo definido para aumentar sua eficácia.

Fonte: ISO/IEC (Parte 2)

As práticas básicas são agrupadas em cinco categorias de processo. O Quadro 11 descreve essas categorias.

Quadro 11: Descrição de categorias do processo

Categoria de processo	Descrição
Cliente-Fornecedor (CUS)	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.
Engenharia (ENG)	A categoria de processo de Engenharia consiste dos processos que diretamente especificam, implementam ou mantém um sistema e os produtos de software e a documentação do usuário.
Projeto (PRO)	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.
Suporte (SUP)	A categoria de processo Suporte consistem de processo que habilitam e suportam o desempenho de outros processos em um projeto.
Organização (ORG)	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.

Fonte: ISO/IEC (Parte 2)

A Categoria de Processo Cliente-Fornecedor compõe-se dos seguintes processos:

- ? 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 de Engenharia compõe-se dos seguintes processos:

- ? desenvolver os requisitos e o design do sistema
- ? desenvolver os requisitos do software
- ? desenvolver o design do software
- ? implementar o design do software
- ? integrar e testar software
- ? integrar e testar o sistema
- ? manter o sistema e o software

A Categoria de Projeto compõe-se dos seguintes processos:

- ? 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 os recursos e cronograma
- ? gerenciar subcontratados

A Categoria de Processo Suporte compõe-se dos seguintes processos:

- ? desenvolver a documentação
- ? executar o gerenciamento de configuração
- ? executar a garantia de qualidade
- ? executar a resolução de problemas
- ? executar a revisão pelos pares

A Categoria de Processo Organizacional compõe-se dos seguintes processos:

- ? engenhar o negócio
- ? definir o processo
- ? melhorar o processo
- ? executar treinamento
- ? facilitar reuso

- ? fornecer um ambiente de engenharia de software
- ? fornecer facilidades para o trabalho

2.3.5 SW-CMM – Modelo de Maturidade da Capacidade de Software

2.3.5.1 Histórico CMM – *Capability Maturity Model*

Análises realizadas pelo governo dos EUA sobre diversos contratos de software do DOD - *Department of Defense*, revelaram problemas relacionados com o software. Atrasos em relação a cronogramas, orçamentos ultrapassados, sistemas resultantes com desempenhos insatisfatórios e projetos cancelados significavam o desperdício de enormes quantias nos sistemas de software.

Para dirigir uma transição da tecnologia da engenharia de software, o Congresso americano criou, em 1984, o SEI – *Software Engineering Institute*, integrado a Universidade *Carnegie Mellon* e patrocinado pela Agência de Projetos de Pesquisa de Defesa Avançada dos EUA. A missão do SEI foi definida para prover liderança no progresso do estado da prática da engenharia de software, com objetivo de melhorar a qualidade dos sistemas que dependem de software.

O Governo Federal dos EUA solicitou ao SEI um método de avaliação da capacidade de seus fornecedores. Em 1984 o SEI iniciou o desenvolvimento de uma estrutura de melhoria do processo de desenvolvimento de software das empresas.

Watts Humphrey, experiente gerente de projetos de engenharia de hardware e software da IBM, liderou a criação, evolução e divulgação da estrutura, que foi desdobrada no CMM, que teve sua primeira versão pública lançada no ano de 1991.

No trabalho de Humphrey, foram realizados um forte programa de avaliações de processos de software, com a participação do governo e da indústria. Foram reunidos registros de engenharia de software e observadas as melhores práticas estabelecidas.

O CMM (PAULK; CURTIS; CHRISSIS; WEBER, 1993) (ROCHA; MALDONADO; WEBER, 2001) (WINTERS, 1997) é um modelo para a melhoria do processo de software, que descreve um caminho evolutivo para as empresas de desenvolvimento

de software. O modelo CMM segundo Paulk *et al* (1995c), baseou-se nos conceitos de qualidade total estabelecido por Philip Crosby. Crosby mostrou que a implantação de sistemas da qualidade em empresas segue um amadurecimento gradativo em patamares que o autor denominou incerteza, despertar, esclarecimento, sabedoria e certeza.

2.3.5.2 O Modelo SW-CMM

? Bases

Um programa de melhoria de processos não pode ser desenhado de forma intuitiva. Ele deve refletir o conjunto de experiência dos profissionais e organizações da área.

Um modelo de capacitação serve de referência para avaliar a maturidade dos processos de uma organização. O SW-CMM (PAULA, 2003) teve uma grande aceitação na indústria de software americana e considerável influência no resto do mundo. Originalmente conhecido apenas como CMM, recebeu a denominação atual para diferenciar de modelos de capacitação em outras áreas, que foram posteriormente formuladas pela SEI.

O SW-CMM (PAULA, 2003) é um modelo de capacitação específico para área de software. Estão fora de seu escopo outras áreas importantes para a sobrevivência de uma organização produtora de software, como marketing, finanças e administração. Mesmo áreas importantes da informática, como hardware e banco de dados, estão fora do escopo do SW-CMM.

? Níveis de Maturidade

O SW-CMM (PAULA, 2003) é um exemplo de modelo de capacitação de arquitetura em estágios. As práticas que ele descreve ou recomenda são agrupadas em níveis de maturidade. Esses níveis são selecionados de acordo com critérios estabelecidos como:

- ? representar fases históricas razoáveis na vida de organizações típicas;
- ? prover degraus intermediários de maturidade, em seqüência razoável;
- ? sugerir medidas de progresso e objetivos intermediários;
- ? definir, para cada estágio, prioridade de melhoria.

Para Paula (2003) a arquitetura em estágios reconhece a dificuldade de que uma organização melhore todos os seus processos simultaneamente. Por isso, oferece uma seqüência de estados intermediários que possam ser atingidos em tempo relativamente curto.

O Quadro 12, resume os níveis dos SW-CMM, destacando as características mais marcantes de cada nível.

Quadro 12: Níveis do SW-CMM

Número do Nível	Nome do Nível	Característica da Organização	Característica dos processos
Nível 1	Inicial	Não segue rotinas	Processos caóticos
Nível 2	Repetitivo	Segue rotinas	Processos disciplinados
Nível 3	Definido	Escolher rotinas	Processos padronizados
Nível 4	Gerido	Cria e aperfeiçoa rotinas	Processos previsíveis
Nível 5	Otimizante	Otimiza rotinas	Processos em melhoria contínua

Fonte: Paula, 2003, p. 66

? Áreas Chaves

Cada nível do SW-CMM (PAULA, 2003) exceto o Nível 1 é composto de várias áreas chaves de processo (KPAs – *Key Process Area*). Cada área chave identifica um grupo de atividades correlatas que realizam um conjunto de metas consideradas importantes quando executadas em conjunto. As áreas chaves de um nível identificam as questões que devem ser resolvidas para atingir esse nível.

No SW-CMM, cada área chave reside em um único nível de maturidade. Para ser classificada em um determinado nível, uma organização tem de ter implementado completamente as áreas chaves desse nível e todos os níveis inferiores.

Com exceção da área chave de Gestão da Subcontratação (Nível 2), não é aplicável a organizações que não subcontratam o desenvolvimento de software.

? Práticas Chaves

Cada área chave define um conjunto de metas, que representam o estado atingido por uma organização que domine a área chave. Para atingir as metas da área chave, a organização deve implementar um conjunto de práticas chaves. Essas práticas descrevem procedimentos gerenciais técnicos descritos no SW-CMM com grau de detalhe variável.

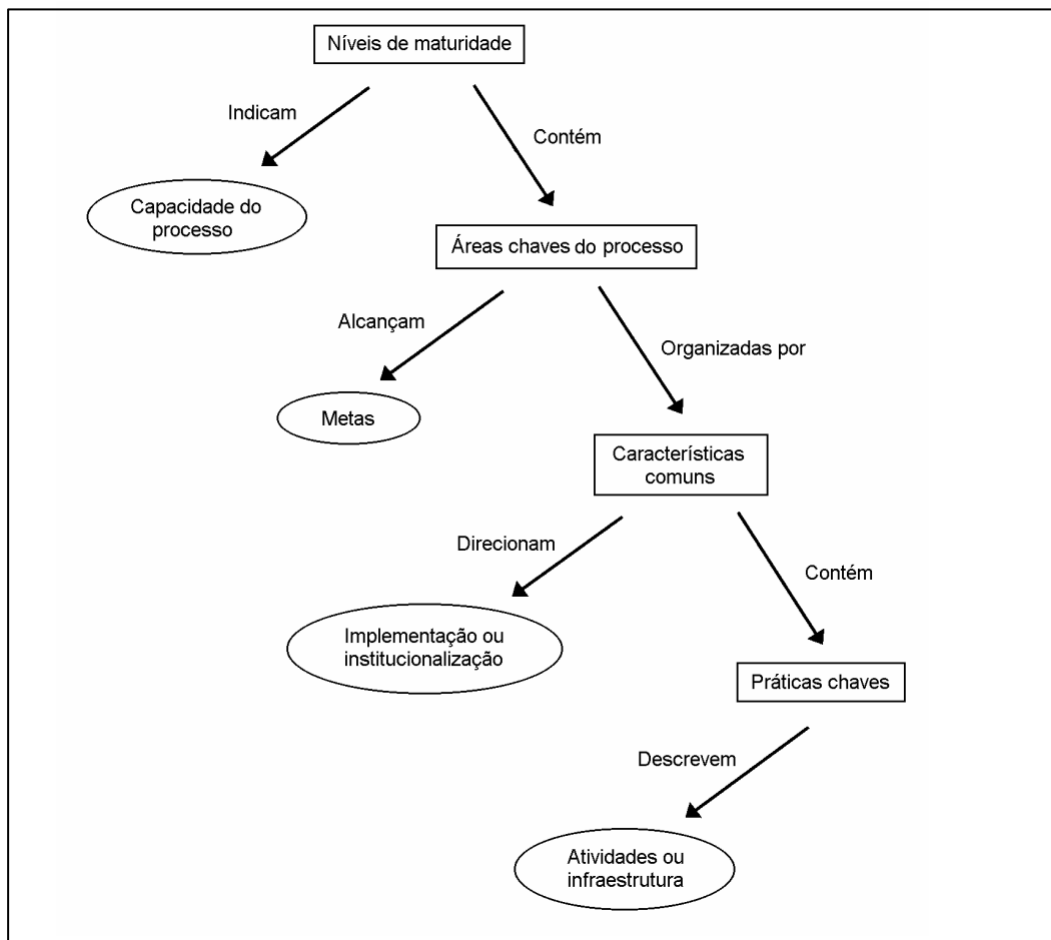


Figura 7: Inter-relação dos elementos do SW-CMM

Fonte: Paula, 2003

O grupo principal de práticas chaves é chamado de “Atividades a executar”. Essas práticas estão diretamente relacionadas com as metas que a área chave pretende atingir.

Um tema central do SW-CMM é a estabilidade da melhoria conseguida. O SW-CMM definiu, em cada área chave, vários grupos de atividades de institucionalização, essas atividades formam uma espécie de travas organizacionais que dificultam o retrocesso nas atividades de implementação. As atividades de institucionalização são divididas conforme Quadro 13.

Quadro 13: Descrição das Características Comuns

Característica Comum	Descrição	Subcaracterísticas
Comprometimento em executar	Condições que garantem a permanência da melhoria dos processos, geralmente associadas à existência de patrocinadores para a melhoria da área e à existência de políticas documentadas.	Políticas organizacionais. Envolvimento direto da gerência sênior. Definição das Responsabilidades.
Capacitação para executar	Pré-condições que devem existir no projeto ou na organização para implementar adequadamente o processo de software.	Recursos. Estrutura da organização. Delegação de responsabilidades. Treinamento. Coordenação.
Medição e Análise	Medições básicas necessárias para avaliar o status da área chave.	Medições em geral.
Verificação da implementação	Ações que garantem a conformidade das demais atividades com os processos estabelecidos.	Revisões e auditorias realizadas nos diversos níveis de controle: gerência sênior, gerentes dos projetos e grupo independente de Garantia de Qualidade.

Fonte: adaptado de Fiorini , 2001 e Paula, 2003

As organizações que situam no Nível 1, 2, 3, 4 e 5, respectivamente têm as seguintes características:

A organização Nível 1 geralmente não é capaz de fazer estimativas de custo ou planos de projeto; se fizer, não é capaz de cumpri-los. As ferramentas não são integradas com os processos, e não são aplicadas com uniformidade pelos projetos. Geralmente a codificação é a única fase dos processos de desenvolvimento que merece atenção. Engenharia de requisitos e desenho são fracos ou inexistentes; mudanças de requisitos e de outros artefatos ocorrem sem controle. A instalação e a manutenção costumam ser deficientes, sendo tratadas como atividades de pouca importância (PAULA, 2003).

Para Paula (2003) os gerentes dessas organizações geralmente não entendem os verdadeiros problemas, por falta de processos que lhes dêem visibilidade real em relação ao progresso dos projetos.

O forte da organização Nível 2 é a capacidade de cumprir compromissos, neste nível, uma organização é capaz de assumir compromissos referentes a requisitos, prazos e custos com alta probabilidade de ser capaz de cumpri-los. Isto requer o domínio das seguintes áreas chaves (PAULA, 2003):

- ? Gestão de requisitos permite definição e controle dos requisitos em que se baseiam os compromissos;

- ? Planejamento de Projetos prevê prazos e custos para cumprimento dos compromissos, como bases técnicas e não apenas intuitiva;
- ? Supervisão e Acompanhamento de Projetos conferem o atendimento dos compromissos, comparando o conseguido com o planejado e adicionando providências corretivas sempre que haja desvios significativos em relação aos compromissos;
- ? Gestão da Subcontratação cobra de organizações subcontratadas, para desenvolver partes do software, os mesmos padrões de qualidade que a organização principal oferece aos seus clientes;
- ? Garantia da Qualidade confere o cumprimento dos compromissos, de forma independente em relação aos projetos;
- ? Gestão de Configurações garante a consistência permanente dos resultados dos projetos, entre si e com os requisitos, ao longo do projeto, mesmo quando ocorrem alterações nos compromissos.

A organização Nível 2 é disciplinada em nível dos projetos. Por isso, ela sabe estimar e controlar projetos semelhantes a projetos anteriores bem-sucedidos. No entanto, ela corre riscos diante de vários tipos de mudanças a que as organizações estão sujeitas como (PAULA, 2003).

- ? Mudanças de ferramentas e métodos, trazidas pela evolução de tecnologia;
- ? Mudanças de tipos de produto, causadas por variações dos mercados;
- ? Mudanças de estrutura organizacional, causadas por diversos fatores da dinâmica das organizações.

Já a organização Nível 3, não repete os sucessos de projetos anteriores, mas estabelece uma infra-estrutura de processos que permite a adaptação a vários tipos de mudanças. O Nível 3 conduz da gestão de projetos à engenharia de produtos. A organização, neste nível, requer o domínio das seguintes áreas chaves (PAULA, 2003):

- ? Estabelecimento formal de um grupo de engenharia de processos de software responsável pelas atividades de desenvolvimento, melhoria e manutenção de processos de software.
- ? Estabelecimento de um processo padrão de software no nível da organização, a partir do qual devem ser derivados os processos definidos para os projetos.

- ? Estabelecimento de um programa de treinamento em processos de software, no nível da organização.
- ? Gestão Integrada dos Projetos, baseada nos processos definidos para os projetos, com uso de procedimentos documentados para gestão de tamanho, esforços, prazos e riscos;
- ? Coordenação entre Grupos que participam de projetos de sistemas, no nível da organização;
- ? Coordenação de revisões no nível da organização.

A organização Nível 3 mesmo durante as crises, sabe manter-se dentro do processo. As ferramentas passam a serem aplicadas de forma sistemática, padronizada e coerente com os processos. Portanto passam a colaborar para a melhoria da produtividade e da qualidade. Neste nível o conhecimento dos processos por parte da organização é basicamente qualitativo. Existe uma base de dados de processos, com os dados recolhidos dos projetos; esta base é utilizada para gestão dos projetos, mas não é aplicada de maneira sistemática e no nível da organização, para atingir metas quantitativas de desempenho de processo e de qualidade de produto.

Na organização Nível 4, o domínio dos processos de software evolui para uma forma quantitativa. Isso não quer dizer que apenas organizações desse nível devam coletar métricas de processo. Todas as áreas chaves do SW-CMM contêm pelo menos uma prática de medição e análise que sugerem métricas adequadas para medir o sucesso da implantação da respectiva área. A organização Nível 3 constrói e mantém uma base de dados de processos.

A organização neste nível é proficiente em coletar métricas e gerir base de dados de processo, que é povoada e analisada por profissionais treinados. Além disso, sabe intervir nos processos para atingir metas de qualidade dos produtos. Este nível tem duas áreas chaves (PAULA, 2003).

- ? A gestão quantitativa dos processos controla quantitativamente o desempenho dos processos usados pelos projetos;
- ? A gestão da qualidade de software promove o entendimento quantitativo da qualidade dos produtos de software, permitindo atingir metas quantitativas desejadas.

A organização Nível 4 passa da engenharia de produtos à qualidade de processos e produtos. Ela tem elementos para decidir, qual deve ser a fração de recursos dos projetos destinada à garantia da qualidade, considerando o nível máximo de defeitos, que se quer admitir nos produtos. Esse domínio quantitativo dos processos é necessário para atingir um estado de melhoria contínua.

No Nível 5, a organização atinge um estado em que os processos estão em melhoria contínua, sendo otimizados para as necessidades de cada momento. As seguintes áreas chaves são executadas Paula (2003):

- ? prevenção dos defeitos, através da identificação e remoção de suas causas;
- ? gestão da evolução tecnológica, com procedimentos sistemáticos de identificação, análise e introdução de tecnologia apropriada;
- ? uso dos dados de processo para gestão das mudanças de processos, colocando-os em melhoria contínua.

Em 2002, 85 empresas no mundo tinham certificação SW-CMM no Nível 5. Dessas, 44 empresas estavam na Índia. Ou seja, a Índia tinha mais de 50% das empresas deste nível do mundo (ISSAC; RAJENDRAN; ANANTHARAMAN, 2004).

2.4 Estudo Comparativo dos Processos de Melhoria de Software

2.4.1 Comparando Estruturas de Melhoria de Processo de Software

Um objetivo de comparar as estruturas de MPS (Melhoria de processo de software) é prover entendimento em suas similaridades e diferenças. O tipo de entendimento necessário, no entanto, é altamente dependente do contexto. Tingey (*apud* HALVORSEN e CONRADI, 2002) analisa dois cenários organizacionais com diferentes necessidades imposta sobre o método de comparação. A diferença essencial está no nível de detalhe e do ponto de vista.

Primeiro existe organizações sem uma estratégia MPS que gostaria aprender sobre diferentes abordagens por causa da pressão competitiva ou necessidades de certificação. Normalmente, essa organização tem um conhecimento geral limitado.

Segundo existe organizações com uma estratégia MPS organizado e sistemático que gostaria de aprender sobre outras estruturas. Muitas organizações se encontram numa situação onde tem que usar mais do que uma abordagem MPS. Como uma ajuda no processo de aprendizado sobre outras estruturas que obviamente nós gostaríamos usar com a existência do conhecimento do MPS. A discussão das estruturas menos conhecidos do ponto de vista de uma estrutura bem conhecida pode assim ser beneficiada.

2.4.2 Classificação

As características devem ser objetivas, medida e comparada. Ser objetivo é quase impossível no sentido estrito, ser medido é freqüentemente uma questão de julgamento humano, e ser comparado significa que um praticante será beneficiado na comparação conforme enuncia Halvorsen e Conradi (2002).

Quadro 14: Classificação aplicada em cinco estruturas MPS

Categoria	Característica	CMM v.1.1	ISO 9000	ISO/IEC 15504 (SPICE)	QUIP/GQM
C1. Geral	Geográfico	EUA/ mundial	Europa/Mundial	mundial/ mundial	EUA/ mundial
	Origem/extensão				
	Origem científica	QT,SPC	- ¹	CMM, Bootstrap Trillium,SPQA	Em parte QT/TQM
	Desenvolvimento/ estabelecimento	Desde 1986	Desde 1987	Em desenvolvimento	Desde 1976
	Popularidade	Topo (espec. nos EUA)	Alta (espec. na Europa)	Crescendo	Médio
	software espec.	Sim	Não	Sim	Sim
	Receita/descritivo	Ambos	Ambos	Ambos	Descritivo
C2. Processo	Adaptabilidade	Limitado	Sim	Sim	Sim
	Avaliação	Organização madura	Processo	Maturidade do processo	Nenhum
	Avaliador	Interno e Externo	Externo	Interno e Externo	NA ²
	Método de melhoria do Processo.	IDEAL	Nenhum	SPICE Doc. Parte 7	QIP
	Iniciação de Melhoria	Top-down	NA ²	Exemplo de Processo	Iterativo Bottom-up
	Foco de melhoria	Processo de gerenciamento	Processo de gerenciamento	Experiência de reuso	Experiência de reuso
	Técnicas de análise	Questionários de avaliação	Orientação ISO e Checklists.	Vários (manual e automatizado). Requerido	GQM
C3. Organização	Atores/regras/ Stakeholders	Gerenciamento	Cientes, fornecedores	Gerenciamento	Fábrica de experiência Projeto da Organização
	Tamanho da Organização	Grande	Grande	Todos	Todos
	Coerência	Interna	Interna e limitado Externa	Interna	Interna
C4. Qualidade	Perspectiva da Qualidade	Gerenciamento.	Cientes	Gerenciamento	todos
	Progressão	Estágio	Sem variação	Contínua (na Instância do nível do processo)	Contínua
	Relação causal	F1'(área do processo chave) ? F2'(nível maturidade) ? Q(processo) ? Q(prodoto)	F1'(elementos) ? F2' (certificação) ? Q(processo) ? Q(prodoto)	F1'(atributos do processo) ? F2' (nível capacidade) ? Q(processo) ? Q(prodoto)	F'(experiência de reuso) ? Q (processo) ? Q(prodoto)
	Comparativo	Sim, nível de maturidade	Sim, certificação	Sim, perfil de maturidade	Não
C5. Resultados	Objetivos	Melhoria do processo determinação da capacidade do fornecedor.	Estabelece o centro do gerenciamento do processo	Avaliação do processo	Organização específica
	Artefatos do processo	Documentação Processo, resultado da avaliação	Documentação do processo, certificado	Perfil do processo, registro da avaliação	Pacotes de experiência modelos GQM
	Certificação	Não	Sim	Não	Não
	Custo de implementação	- ¹	- ¹	- ¹	- ¹
	Validação	Inspeção e estudo de caso	Inspeção	Revisão de documento, teste (estudo de caso e inspeção)	Experimentos e estudo de caso

Fonte: adaptado Halvorsen e Conradi, 2002

- ¹ Ainda Não Aplicado

- ² Não Aplicável

Análises técnicas: algumas estruturas usam técnicas quantitativas, outras qualitativas. Exemplos de técnicas de análise são questionários, controle do processo estatístico e várias medições.

Categoria Organizacional: esta categoria está diretamente ligada aos atributos da organização e o meio ambiente nas quais a estrutura MPS é usada.

Atores / regras / *stakeholders*: cada estrutura tem seus atores primários, os quais são as pessoas, grupos afetados por ou fazendo parte do processo de melhorias. Geralmente, cada ator / *stakeholders* tem regras pré-definidas na melhoria de processos.

Tamanho da organização: esta característica indica o tamanho da organização para as quais o MPS é conveniente.

Coerência: é a conexão lógica entre fatores de engenharia e fatores relatados para o negócio ou organização.

? Coerência interna: tecnologia, recursos humanos, estratégia e cultura.

? Coerência externa: mercado, clientes e competidores.

Categoria de qualidade: características relatadas para a dimensão da qualidade, tais como a progressão é medida, sob qual perspectiva a qualidade é trabalhada.

Perspectiva da qualidade: o conceito de qualidade naturalmente será diferente dependendo de quem está perguntando. Por exemplo, gerentes terão uma perspectiva diferente da qualidade comparado com o que o cliente poderia ter.

Progressão: a progressão pode ser caracterizada como “sem variação” “por estágios” ou “contínua”, conforme Figura 8.

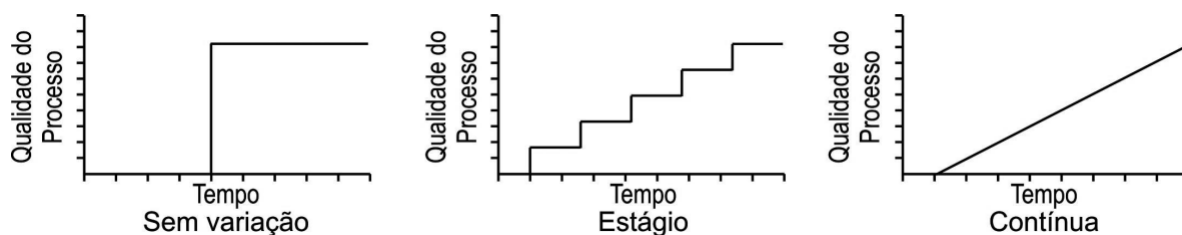


Figura 8: Características da progressão do processo de qualidade

Fonte: Halvorsen e Conradi, 2002

Relação causal: estrutura de MPS (melhoria de processos de software) assume que produto de qualidade é afetado pela qualidade do processo usado para produzi-lo.

F(Indicador de qualidade) ? **Q**ualidade (Processo) ? **Q**ualidade (Produto)

Comparação: algumas estruturas são comparáveis em natura, significando que eles podem ser usados para comparar certos aspectos de diferentes unidades organizacionais, internamente ou externamente.

Resultado da categoria: descreve os resultados do emprego da estrutura MPS, mas também os custos para alcançar estes resultados e os métodos usados para validá-los.

Objetivo: a estrutura é um objetivo primário ou o resultado final.

Artefatos de processo: são as “coisas” criadas como um resultado da adoção da estrutura MPS. Os artefatos podem ser tangíveis, tais como documento contendo modelos de processo, planos de melhorias, dados de experiências e outros. No entanto, os artefatos podem ser intangíveis, tais como aumento de conhecimento na organização.

Certificação: a ISO - *International Organization for Standardization*, é um variado grupo de padronização nacional têm desenvolvido padrões para os quais uma organização pode ser certificada.

Custo de implementação: é uma importante característica, porque freqüentemente dita se a abordagem para qualidade é financeiramente viável. Há custo diferente entre as estruturas, sendo impossível para organizações pequenas adotá-las.

Validação: é necessário para avaliar se o esforço da melhoria tem realmente funcionado. Tal validação deveria distinguir entre melhorias resultante do esforço MPS e melhoria resultante de outros fatores (contexto). Por exemplo, o maior fator de sucesso pode ser a tecnologia atual utilizada (p. ex., orientação ao objeto, reuso, inspeções etc). Outras melhorias podem ser atribuídas para a eliminação de processos deficientes.

2.4.3 Comparando ISO 9001 e o SW-CMM

Há algum julgamento envolvido em decidir a correspondência exata, dada os níveis diferentes de abstração.

No Quadro 15, as duas últimas colunas têm o seguinte significado:

(1) - Parcialmente endereçado pela ISO 9001.

(2) - Largamente (não completamente) endereçado pela ISO 9001

Quadro 15: Comparação ISO 9001 e o SW-CMM

Nível	Processos endereçados no SW-CMM	(1)	(2)
2	Gestão de Requisitos		X
	Planejamento de Projetos	X	
	Supervisão e Acompanhamento de Projetos	X	
	Gestão da Subcontratação		X
	Garantia da Qualidade		X
	Gestão de Configurações		X
3	Focalização dos Processos da Organização		X
	Definição dos Processos da Organização		X
	Programa de Treinamento		X
	Gestão Integrada de Software	X	
	Engenharia de Produtos de Software		X
	Coordenação entre Grupos	X	
4	Revisões	X	
	Gestão Quantitativa dos Processos	X	
	Gestão da Qualidade de Software		X
5	Prevenção de Defeitos	X	
	Gestão das Mudanças de Tecnologia		
	Gestão das Mudanças de Processos	X	

Fonte: Paulk, 2002

Observa-se nesta comparação que os processos da ISO 9001 não cobrem amplamente os processos do modelo SW-CMM. Com isso pode-se concluir que o SW-CMM é mais completo para o desenvolvimento de software.

2.4.4 Comparando o SPICE e o SW-CMM

O SW-CMM e a ISO/IEC 15504 (SPICE) (PAULK, 1999b) (VARKOI e MAKINEN, 1998) (PAULK; KONRAD; GARCIA, 1995a) são dois importantes modelos de avaliação de melhoria do processo de software e determinação da capacidade de

processo de uma organização.

Os dois modelos têm arquitetura e foco diferentes. SPICE separa processos e determinação de capacidade em duas dimensões enquanto o SW-CMM trabalha em uma dimensão. O SW-CMM foca a capacidade da organização e o SPICE foca a capacidade de um único processo. A comparação da capacidade do processo parece ser possível, mas requer consideração porque ambos contêm estruturas diferentes em seus modelos.

Quadro 16: Comparando ISO/IEC 15504 SPICE e o SW-CMM

ISO/IEC 15504	Software CMM v1.1
CUS.1 Aquisição	Gestão da Subcontratação
CUS.1.1 Preparação da Aquisição	<i>Gestão da Subcontratação, Atividade 1</i>
CUS.1.2 Seleção do Fornecedor	Gestão da Subcontratação, Atividade 2
CUS.1.3 Seleção do Fornecedor	<i>Gestão da subcontratação, Atividades 5 e 7-11</i>
CUS.1.4 Aceitação do Cliente	<i>Gestão da subcontratação, Atividade 12</i>
CUS.2 Fornecer ²	(Planejamento de Projetos; Supervisão e acompanhamento de Projetos; Eng. de Produtos de SW)
CUS.3 Evidenciar Requisitos	
CUS.4 Operação	
CUS.4.1 Uso Operacional	
CUS.4.2 Apoio ao cliente	
ENG.1 Desenvolvimento	Engenharia de Produtos de SW
ENG.1.1 Análise de Requisitos e Design do Sistema	
ENG.1.2 Análise de requisitos do SW	<i>Engenharia de Produtos SW, Atividade 2</i>
ENG.1.3 Design do Software	<i>Engenharia de Produtos SW, Atividade 3</i>
ENG.1.4 Construção do Software	<i>Engenharia de Produtos SW, Atividade 4</i>
ENG.1.5 Integração do Software	<i>Engenharia de Produtos SW, Atividade 6</i>
ENG.1.6 Teste de Software	<i>Engenharia de Produtos SW, Atividade 7</i>
ENG.1.7 Integração e Teste do Sistema	<i>(Engenharia de Produtos SW, Atividade 6 e 7)</i>
ENG.2 Manutenção de Sistema e de Software	
SUP.1 Documentação	<i>Engenharia de Produtos SW, Atividade 8</i>
SUP.2 Gerenciamento de Configuração	Gestão de Configurações
SUP.3 Garantia da Qualidade	Garantia da Qualidade
SUP.4 Verificação	(Revisões; Engenharia de Produtos SW atividade 5 e 6)
SUP.5 Validação	<i>Produto de Engenharia SW, Atividade 5</i>
SUP.6 Revisão da Ligação	<i>Supervisão e Acompanhamento de Projetos; Atividade 13</i>
SUP.7 Auditoria	(Garantia da Qualidade do SW) ⁴
SUP.8 Solução do Problema	<i>Gestão de Configurações, Atividade 5</i>
MAN.1 Gerenciamento ⁵	(Planejamento de Projetos; Supervisão e Acompanhamento de Projetos, Gestão Integrada de SW)
MAN.2 Gestão de Projeto	Planejamento de Projetos; Supervisão e Acompanhamento de Projetos; Gestão Integrada de SW
MAN.3 Gestão da Qualidade	Gestão de Configurações
MAN.4 Gestão de Risco	<i>Planejamento de Projetos, atividade 13; Supervisão e Acompanhamento de Projetos, atividade 10; Gestão Integrada de SW, atividade 10</i>
ORG.1 Alinhamento Organizacional ⁶	
ORG.4 Infraestrutura	Definição dos Processos da Organização
ORG.2 Melhorias	Definição dos Processos da Organização
ORG.2.1 Estabelecimento de Processos	Definição dos Processos da Organização
ORG.2.2 Avaliação de Processos	<i>Focalização dos Processos da Organização, Ativ. 1</i>
ORG.2.3 Melhoria dos Processos	Focalização dos Processos da Organização; (Gestão das Mudanças de Processos)
ORG.3 Gerenciamento de Recursos Humanos	Programa de Treinamento
ORG.4 Infraestrutura	
ORG.5 Medição	<i>Análise e Medição (característica comum)</i>
ORG.6 Reuso	
	Gestão de Requisitos
	Coordenação entre Grupos
	Revisões ⁷
	Gestão Quantitativa dos Processos ⁸
	Prevenção de Defeitos
	Gestão das Mudanças de Tecnologia
	Gestão das Mudanças de Processos

Fonte: Paulk, 1999b

⁷ Indiretamente coberta por SUP.4 Verificação do Processo

⁸ Indiretamente coberta por ORG.5 Medição do Processo

O Quadro 16 mostra que existe uma relação equivalente entre os tópicos na ISO/IEC 15504 e SW-CMM. Os tópicos não têm a mesma forma, mas são altamente correlacionados. Qualquer um adequadamente implementando, por exemplo, Gerenciamento de Configuração de Processo em ISO/IEC 15504 poderia razoavelmente satisfazer o processo no Gerenciamento de Configuração de Processo no SW-CMM. Os tópicos não são isomórficos porque as extensões podem ter sido adicionadas ou diferentes níveis de abstração podem ter sido escolhidos. Por exemplo, o Desenvolvimento de processo na ISO/IEC 12207 enfoca o mesmo tipo de conceito que o Produto de Engenharia de Software no CMM. Subprocessos e atividades no Quadro 16 estão em itálico para enfatizar que eles são componentes de uma estrutura maior. Onde a relação é indireta, o componente do software no CMM está em parênteses para destacar a diferença no escopo.

2.4.5 Considerações sobre o SW-CMM, ISO 9000 e SPICE

2.4.5.1 Considerações sobre o SW-CMM

O criador do SW-CMM foi Watts Humphrey. O SW-CMM foi desenvolvido para contemplar questões de gerenciamento de software e engenharia de processo. Visa o aumento do nível de maturidade nas organizações de software no nível corporativo ou organizacional, baseia-se em disciplina, e é a base para que se construa um conjunto de ferramentas. Essas ferramentas de controle são usadas por times de controle de processo e de avaliação de capacidade de software para identificar pontos fortes, fracos e riscos do processo de software de uma organização.

Durante o desenvolvimento de um sistema, a solução técnica é muito importante, pois influenciará diretamente o custo total do projeto. Assim, a escolha de metodologia é igualmente importante, pois a estruturação melhora a probabilidade de sucesso de um empreendimento, principalmente daqueles de grande porte.

Mas o objetivo maior da equipe de desenvolvimento é atender às necessidades dos usuários de forma rápida, objetiva e direta, com a menor margem de erro possível.

No SW-CMM cada fase provê a base para a seguinte, permitindo a identificação de deficiências na organização. Cada nível de maturidade compreende um conjunto de objetivos de processo que, quando satisfeitos, estabilizam um componente importante do processo de software, resultando em um aumento da capacidade de processo da organização.

O objetivo principal do SW-CMM foi estabelecer um método para evolução da capacidade para liberar software de qualidade dos contratos do DOD. O SW-CMM sugere que a maturidade do processo de uma organização tem que ser desenvolvido do Nível 1 para o Nível 5 progressivamente através dos níveis intermediários sem a possibilidade de cortes. Define áreas de processos chaves de baixo nível para aquele de alto nível, por exemplo, definição de processo (Nível 3) sendo pré-requisito para a medição de processo (Nível 4) a melhorias (Nível 5) (GHOSH, 1994).

Para Paulk (1996) o SW-CMM representa um “sentido comum entre os engenheiros” na abordagem para melhoria de processo de software. Os níveis de maturidade, as áreas chaves dos processos, os objetivos e as práticas chaves tem sido bastante discutido e examinado dentro da comunidade de software. Enquanto o SW-CMM, segundo Curtis e Bate (*apud* PAULK, 1996), nem é completo nem compreensível, representa um amplo consenso na comunidade de software é e uma ferramenta útil guiando os esforços para a melhoria de processos de software.

Um estudo realizado por Herbsleb e citado por Paula (2003), resumiu avaliações de um conjunto de organizações produtoras de software, chegando aos resultados apresentados na Tabela 1.

Tabela 1: Ganhos obtidos em implementações do SW-CMM

	Faixa	Mediana
Duração do programa de melhoria de processos	1 a 9 anos	3,5 anos
Custo anual total	\$49K a \$1.202K	\$245K
Custo anual para o engenheiro de software	\$490 a \$2.004	\$1.375
Ganho anual de produtividade	9% a 67%	35%
Redução anual de tempo até o mercado	15% a 23%	19%
Ganho anual de detecção precoce de defeitos	6% a 25%	22%
Redução anual em defeitos achados em operação	10% a 94%	39%
Retorno do investimento	400% a 880%	500%

Fonte: Paula, 2003, p. 73

O custo do processo para uma empresa brasileira de software obter certificação do SW-CMM no Brasil está em média de R\$ 200 mil (O ESTADO de SÃO PAULO, 2004).

Foram relatados também com a implantação do SW-CMM os seguintes benefícios (PAULA, 2003, p. 73).

- ? melhoria do moral;
- ? melhoria da qualidade de vida;
- ? diminuição de horas extras;
- ? maior estabilidade do ambiente de trabalho;
- ? menor rotatividade da equipe;
- ? melhoria da comunicação;
- ? melhoria da qualidade percebida pelos clientes.

O Instituto de Engenharia de software (SEI) preocupado com melhoria do gerenciamento de pessoal criou o P-CMM, que é uma ferramenta prática que pode ser utilizado como uma estrutura para melhorar a maneira pela qual uma organização gerência seu patrimônio humano como exemplo: reter pessoas com conhecimento e habilidades fundamentais, adequar a motivação dos indivíduos à organização, assegurar que a capacidade de desenvolvimento não seja somente para alguns indivíduos e melhorar a capacidade da organização pelo aumento da capacidade de sua força de trabalho (SOMMERVILLE, 2003).

Na opinião de Paulk; Curtis; Chrissis e Weber (1993) a organização imatura é reativa e os gerentes estão normalmente tentando resolver crises imediatas. Prazos e orçamentos estão rotineiramente excedidos porque eles não são baseados em

estimativas realistas. Quando prazos apertados são impostos funcionalidade e a qualidade do produto, são freqüentemente comprometidas. O nível de confiança para Sommerville (2003) depende do propósito do sistema, ele mostra que a qualidade depende de vários fatores, um dos quais é o mercado, e nem por isso seja imatura neste contexto a questão não é de imaturidade, mas de decisão econômica.

Por outro lado, a organização de software madura o processo de software é corretamente comunicados para todos e atividade de trabalho são executados de acordo com o planejamento do processo. Estes processos definidos são atualizados quando necessários, e melhorias são desenvolvidos através de testes pilotos controlados e análises de custos e benefícios. Papéis e responsabilidades estão claros na definição do processo e em toda parte do projeto através da organização (PAULK; CURTIS; CHRISSIS; WEBER, 1993).

A organização madura mede e prevê quanto tempo levará para alcançar seus objetivos (PAULK, 1999a).

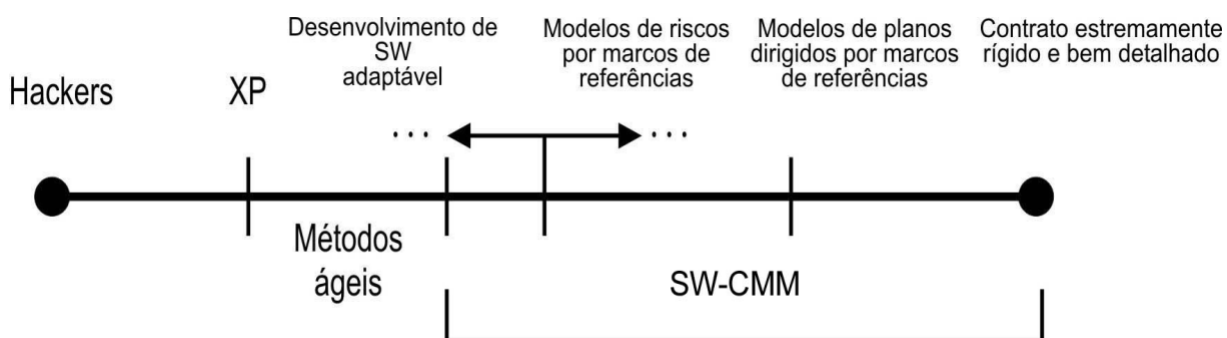


Figura 9: Espectro de Planejamento

Fonte: adaptado de Boehm, 2002

Segundo Boehm (2002) os métodos baseados em planos são mais necessários em software de alta garantia. Outro conjunto maior de objetivos para os métodos tradicionais de planejamento, e para o SW-CMM, tem sido previsibilidade, repetição e otimização. No mundo em que envolve mudanças radicais e freqüentes, tendo processos como repetível e otimizado é como um dinossauro e pode não ser um bom objetivo.

O que faz do SW-CMM um modelo respeitado é o fato de compilar e organizar de forma extremamente coerente, uma série de boas idéias antigas, que estavam soltas, sem uma estrutura que as conectasse. Um dos pontos fundamentais deste método é o hábito de revisar o código antes de compilá-lo. Humphrey provou estatisticamente através de seus estudos que, essa revisão de código gera, para a maioria dos programadores, um aumento substancial da produtividade e uma redução ainda maior de defeitos no produto final.

2.4.5.2 Considerações sobre o SPICE

O método de Melhoria do processo de software e determinação da capacidade - SPICE, que é derivado na norma ISO/IEC 15504, serve para se fazer uma avaliação dos processos de softwares, do ponto de vista da melhoria e da determinação da capacidade dos processos.

O método SPICE é útil na fase de se determinar e contratar um fornecedor adequado para o desenvolvimento de sistemas específicos para as organizações. Tem como uma vantagem o fato de ter uma grande diversidade de formatos de apresentação dos resultados, superando dessa forma o SW-CMM que possui apenas um número para representar o nível de maturidade dos processos da organização.

O SW-CMM associa áreas de processos aos níveis de maturidade, enquanto o SPICE ou ISO 15504 tem uma abordagem de duas dimensões, permitindo associar-se níveis de maturidade a qualquer processo do ciclo de vida do software.

O SPICE segundo Paulk; Konrad e Garcia (1995a), mede a evolução de cada processo separadamente de outros processos, o que permite a uma equipe de melhoria de processo a visualizar seus aspectos independentemente de outros processos e fornecer maior confiabilidade de medida e análise.

2.4.5.3 Considerações sobre a ISO 9000

ISO 9000 é uma série de padrão que é aplicada para todo tipo de indústria, há uma notável terminologia para o setor de fabricação (GHOSH, 1994).

ISO 9000 não fornece passos específicos para estabelecer um sistema de qualidade, o SW-CMM recomenda um mecanismo progressivo passo a passo através de sucessivos níveis de maturidade.

Segundo Weber (*apud* ROCHA; MALDONADO; WEBER, 2001) a certificação ISO 9000 é reconhecida por todos os países e setores, não só pelo setor de software. Para uma empresa, a conquista da certificação ISO 9000 significa alcançar padrão internacional de qualidade em seus processos. Entretanto, mesmo no âmbito de um determinado setor, não é possível diferenciar o nível de maturidade de uma empresa em relação a outra, em um conjunto de empresas que detenham certificação ISO 9000, a não ser pelo escopo de certificação, pela qualidade do certificador e pelo tempo pelo qual a certificação vem sendo mantida.

Um ponto forte das normas ISO 9000 foi padronização dos requisitos mínimos da garantia da qualidade, que devem ser atendidos pelos fornecedores de produtos e serviços. Outro ponto forte do modelo ISO 9000 é a sua flexibilidade, permitindo ser adaptado ou complementado (p. ex. CMM e SPICE), para atender aos requisitos setoriais específicos (ROCHA; MALDONADO; WEBER, 2001).

2.4.5.4 Outras Considerações

Segundo Rezende e Abreu (2000), a tecnologia de informação não deve ser trabalhada e estudada de forma isolada. É necessário envolver e discutir questões conceituais dos negócios e das atividades empresariais, que não podem ser organizadas e resolvidas simplesmente com os computadores e recursos de software, por mais tecnologia que se tenham. Em conseqüência aparecem as questões comportamentais necessárias para a utilização efetiva dessas tecnologias.

Para Meira (2002), 70% da receita de sucesso para os projetos de software inclui os fatores conforme apresentado no Quadro 17.

Quadro 17: Receita do sucesso

Envolvimento e apoio dos executivos e usuários Gerentes de projetos experientes Objetivos de negócio claro Escopo reduzido	70%
Infraestrutura padrão de software Requisitos básicos firme Metodologia formal Estimativas confiáveis Outros	30%

Fonte: Meira, 2002

Para Kitchenham e Pfleeger (1996) as medições deixam nos conhecer se nossas técnicas realmente melhoram nossos softwares, bem como se a qualidade do processo pode gerar produto de qualidade. Nós também precisamos conhecer como a qualidade que construímos pode afetar o uso do produto depois de liberados e se o investimento de tempo e recursos para garantir alta qualidade tem algum retorno. Em outras palavras, nós queremos saber se um bom software é um bom negócio.

Meira (2003) lembra que as empresas que sobrevivem na economia de software são as que combinam infra-estrutura, métodos, processos e pessoas com vendas e serviços, parte de uma complexa receita onde se deve equilibrar qualidade, produtividade, custo e preço. Desenvolvimento de software depende de educação e prática, é um tipo de desenvolvimento moderno que pode, com relativa facilidade, ser contratado em qualquer lugar do mundo, desde que administre, os processos envolvidos.

2.4.6 Conclusão do Estudo Comparativo

O estudo comparativo serviu de embasamento para a escolha dos modelos de qualidade que iriam compor a pesquisa. O motivo de escolher o SW-CMM e o modelo SPICE é que o SW-CMM, em relação a ISO 9001, tem uma cobertura mais abrangente no que se refere ao desenvolvimento de software, e, com relação ao modelo SPICE, é que este tem uma cobertura superior ao SW-CMM. Então,

selecionaram-se do modelo SPICE os processos que o SW-CMM não estava atendendo para compor o questionário de pesquisa.

3 METODOLOGIA

Para que um conhecimento possa ser considerado científico, torna-se necessário identificar as operações mentais e técnicas que possibilitarão a sua verificação (Gil, 1991). Pode-se definir pesquisa como o processo formal e sistemático de desenvolvimento do método científico. O objetivo fundamental da pesquisa é descobrir respostas para problemas mediante o emprego de procedimentos científicos. Neste capítulo, descrevem-se os procedimentos metodológicos utilizados para a elaboração desta pesquisa.

3.1 Caracterização da Pesquisa

A pesquisa realizada neste trabalho é classificada segundo Silva e Menezes (2001), do ponto de vista da sua natureza, como aplicada porque objetiva gerar conhecimentos para aplicação prática, dirigidos à solução de um problema específico. Quanto à abordagem do problema ela é uma pesquisa quantitativa, pois pode traduzir em números opiniões e informações para classificá-las e analisá-las. Do ponto de vista de seus objetivos ela é descritiva, pois visa descrever determinada população ou fenômeno.

3.2 Sujeitos da Pesquisa

A população a ser avaliada engloba 91 empresas produtoras de software. Esta pesquisa utilizou uma amostragem não probabilística com utilização da técnica de amostragem intencional. Participaram da entrevista empresas de grande, médio e pequeno porte, associadas a ASSESPRO – Associação das Empresas Brasileira de Software e Serviço de Informática - Regional do Paraná.

3.3 Instrumentos de Coleta de Dados

Esta pesquisa utiliza técnicas semi-formais de coleta de dados, como, questionário para o levantamento de dados.

Com base na revisão da literatura foi elaborado um questionário para levantar o posicionamento atual das empresas produtoras de software em relação à qualidade no desenvolvimento do software, conforme o Apêndice C.

O questionário é composto por 28 questões com perguntas abertas e fechadas. Foi solicitado que cada participante classificasse cada resposta conforme o grau de importância variando de (1) **Influência Muito Baixa** a (5) **Influência Muito Alta**.

Tabela 2: Escala Tipo LIKERT

Influência Muito Baixa	Influência Baixa	Influência Média	Influência Alta	Influência Muito Alta
(1)	(2)	(3)	(4)	(5)

Fonte: Martins e Lintz, 2000

O questionário é objetivo, a linguagem utiliza termos técnicos conhecidos na literatura, e é acessível ao entendimento da população estudada. A formulação das perguntas evita a possibilidade de interpretação dúbia.

A escolha do questionário como instrumento de pesquisa é justificada pela facilidade de sua distribuição e recolhimento, além de permitir obter informações de um número considerável de empresas.

A opção pelo anonimato (opcional) permitiu que os pesquisados se sentissem livres para expressar suas opiniões. O tempo oferecido, quatro semanas, foi suficiente para proporcionar respostas mais refletidas.

A pesquisa foi realizada do dia **09 de janeiro 2004 a 06 de fevereiro 2004**. O questionário de pesquisa foi enviado via e-mail. A pesquisa visou explorar um grupo de empresas que possuem um departamento, ou uma unidade organizacional para o desenvolvimento de software, ou empresas que construam software como negócio (indústria de software/*softhouse*).

Foram respondidos 33 questionários de um total de 91 questionários enviados para a pesquisa.

3.4 Tratamento dos Dados

Uma vez coletado o material, efetuou-se uma verificação crítica, a fim de detectar falhas ou erros, evitando informações confusas, distorcidas e incompletas, sem detalhes suficientes. Pois, conforme Marconi e Lakatos (2002), este é o processo da seleção.

A codificação é a técnica utilizada para categorizar os dados que se relacionam, agrupando-os em determinadas categorias. Mediante a codificação, os dados são transformados em símbolos, podendo ser tabelados e contados. A tabulação é disposição dos dados em tabelas, possibilitando maior facilidade na verificação das inter-relações entre eles. Esta parte do processo permite sintetizar os dados de observação pelas diferentes categorias e representá-los graficamente, possibilitando uma rápida compreensão e interpretação. Utilizou-se, para a tabulação dos dados, o software *Statística 6.0* da *StatSoft*.

4 CONSOLIDAÇÃO, ANÁLISE E INTERPRETAÇÃO DOS DADOS DE PESQUISA

Neste capítulo, faz-se necessário interpretar e analisar os dados colhidos. Para melhor visualização, serão montados gráficos e tabelas buscando relacionar os resultados obtidos ao modelo de qualidade. A análise dos resultados deve atender aos objetivos da pesquisa.

4.1 Análise dos Dados Relacionados às Empresas Situadas no Nível 1 - Inicial

Nesta pesquisa 87,9% das empresas situam-se no Nível 1, isto quer dizer que, de acordo com o SW-CMM, o processo de desenvolvimento de software não existe ou se existe não está documentado. Essas empresas não fornecem um ambiente sólido para a manutenção e desenvolvimento de software. No item 4.7 descreveremos mais detalhadamente sobre este nível.

4.2 Análise dos Dados Relacionados às Empresas Situadas no Nível 2 - Repetível

A seguir serão apresentadas as respostas das empresas envolvidas na pesquisa referente ao Nível 2 do Modelo SW-CMM.

Tabela 3: Nível 2 – Repetível

Nível 2 - Conceitos de Gestão	Índice de Importância	Respostas Positivas	fr
10. Supervisão e Acompanhamento de Projetos	4,35	27	82%
9. Planejamento de Projetos	4,26	23	70%
8. Gestão de Requisitos	3,93	18	55%
13. Gestão de Configurações	3,97	16	48%
11. Gestão da Subcontratação	3,88	14	42%
12. Garantia da Qualidade	3,46	8	24%

A preferência das empresas para os processos do Nível 2 quanto ao grau de importância dos processos para a melhoria da qualidade do software coincide com os processos utilizados pelas empresas para o desenvolvimento do software.

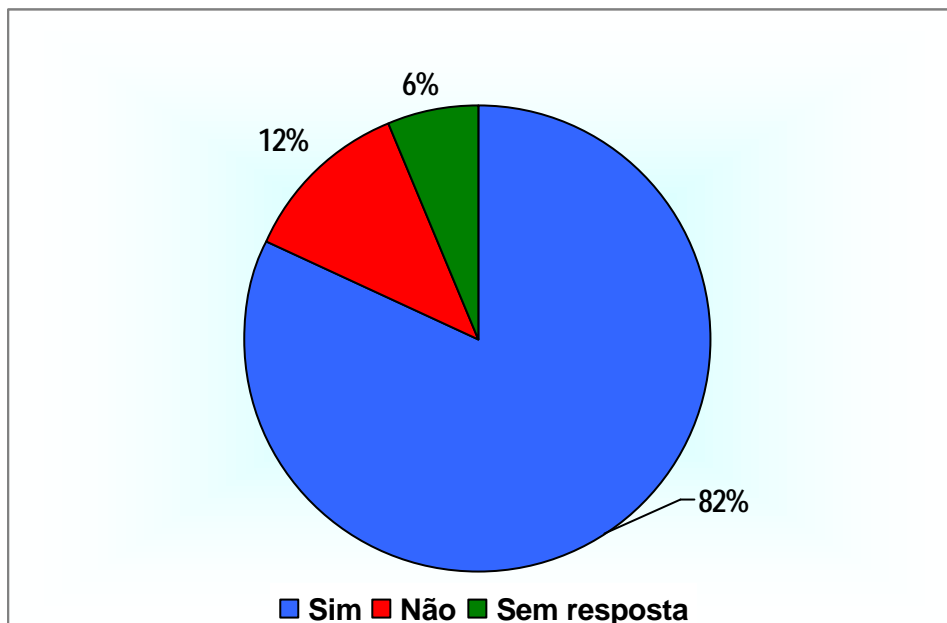


Figura 10: Frequência de utilização do processo de supervisão e acompanhamento de projetos

Dentro do Nível 2 o processo Supervisão e Acompanhamento de Projetos é o considerado mais importante e o mais utilizado pelas empresas correspondendo a 82% de aplicação desse processo e, contém também, o maior índice de importância de 4,35, na opinião dos entrevistados, considerado neste nível.

A finalidade do processo de Supervisão e Acompanhamento de Projetos de software é fornecer uma visão realista do efetivo progresso do projeto, permitindo que a gerência de desenvolvimento possa tomar ações eficazes quando o desempenho do projeto desviar-se de forma significativa dos planos de software.

4.3 Análise dos Dados Relacionados às Empresas Situadas no Nível 3 - Definido

Para o Nível 3, pode-se identificar o processo de Engenharia de Produtos de Software que é composto pelas atividades:

- a) desenvolve Requisitos e Projeto (Design) do Sistema;
- b) desenvolve Requisitos do Software;
- c) desenvolve o Design do Software;

- d) implementa o Design do Software;
- e) integração e Teste do Software;
- f) integração e Teste do Sistema; e
- g) manutenção do Sistema e do Software.

Tabela 4: Nível 3 – Definido

Nível 3 - Conceitos de Gestão	Índice de Importância	Respostas Positivas	fr
2. Desenvolve Requisitos do Software	4,516	24	73%
1. Desenvolve Requisitos e Projeto (Design) do Sistema	4,438	24	73%
6. Integração e Teste do Sistema	4,394	29	88%
5. Integração e Teste do Software	4,313	29	88%
15. Definição dos Processos da Organização	4,276	17	52%
16. Programa de Treinamento	4,233	17	52%
4. Implementa o Design do Software	4,200	26	79%
7. Manutenção do Sistema e do Software	4,188	27	82%
17. Gestão Integrada dos Projetos	4,133	17	52%
3. Desenvolve o Design do Software	4,000	27	82%
18. Coordenação entre Grupos	3,750	19	58%
14. Focalização dos Processos da Organização	3,655	12	36%
19. Revisões	3,643	10	30%

Nota = A base é o total de empresas entrevistadas (33).

As atividades do processo de Engenharia de Produtos de Software são as mais utilizadas pelas empresas que desenvolvem software e têm o índice de importância mais significativo na opinião dos entrevistados, representados pelas empresas neste nível, seguidos dos processos: Definição dos Processos da Organização, 4,276; Programa de Treinamento, 4,233; Gestão Integrada dos Projetos, 4,133; Coordenação entre Grupos, 3,750; Focalização dos Processos da Organização, 3,655 e Revisões, 3,643.

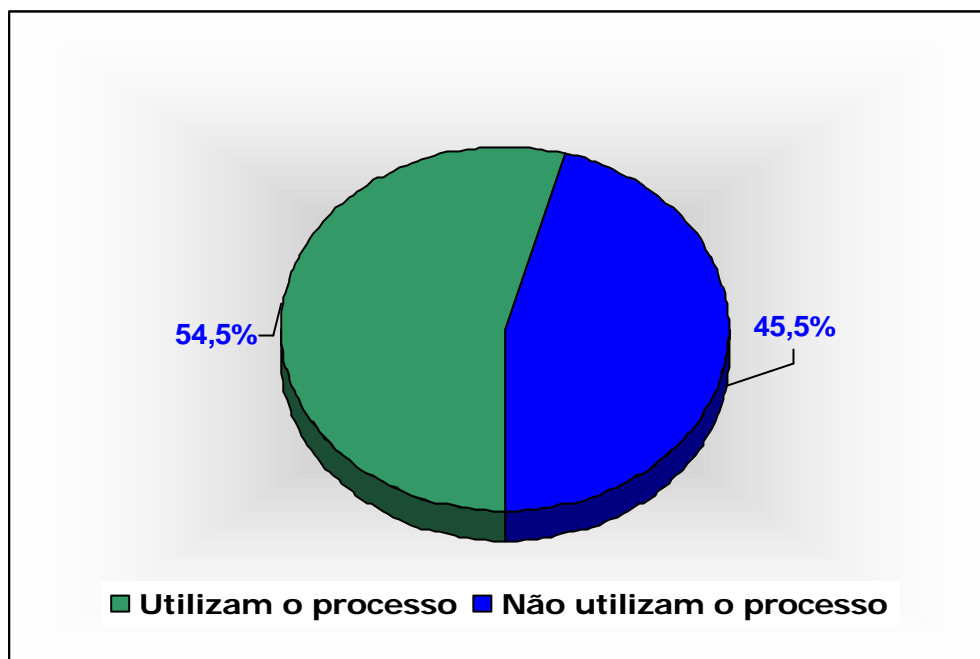


Figura 11: Frequência de utilização do processo de engenharia de produto de software

A padronização no nível da organização do processo de Engenharia de Produtos de Software abrange engenharia de requisitos, testes, desenho, codificação e documentação de uso.

A finalidade deste processo é executar de forma consistente um processo de engenharia bem definido, que integre todas atividades de engenharia de software, para produzir de maneira eficaz e eficiente, produtos de software corretos e consistentes.

4.4 Análise dos Dados Relacionados às Empresas Situadas no Nível 4 - Gerenciado

Pode-se comprovar, no Nível 4, que as empresas em geral não utilizam processos para medição da qualidade de software. Quanto mais alto o nível de maturidade, menor é o grau de utilização dos processos de Gestão Quantitativa dos Processos e Gestão da Qualidade de software pelas empresas que desenvolvem software. Da mesma forma, o índice de importância dada a esses dois processos do

Nível 4 também é menor, comparando-os com os processos do Nível 2 ou 3, que apresentam grau de importância maior.

Tabela 5: Nível 4 – Gerenciado

Nível 4 - Conceitos de Gestão	Índice de Importância	Respostas Positivas	fr
20. Gestão Quantitativa dos Processos	3,48	8	24%
21. Gestão da Qualidade de Software	3,68	8	24%

Humphrey (EICKELMANN; HAYES, 2004) lembra que os desenvolvedores devem:

- 1) definir o que deve ser feito a cada trabalho e insistir exatamente neste caminho;
- 2) fazer planejamento e cronogramas realistas;
- 3) medir e administrar a qualidade;
- 4) estabelecer meta corporativa, para que todo produto tenha melhor qualidade que seu predecessor; e
- 5) planejar para fazer trabalho de qualidade.

A Figura 12 representa a freqüência de utilização do Processo Gestão Quantitativa dos Processos, que é de 73%.

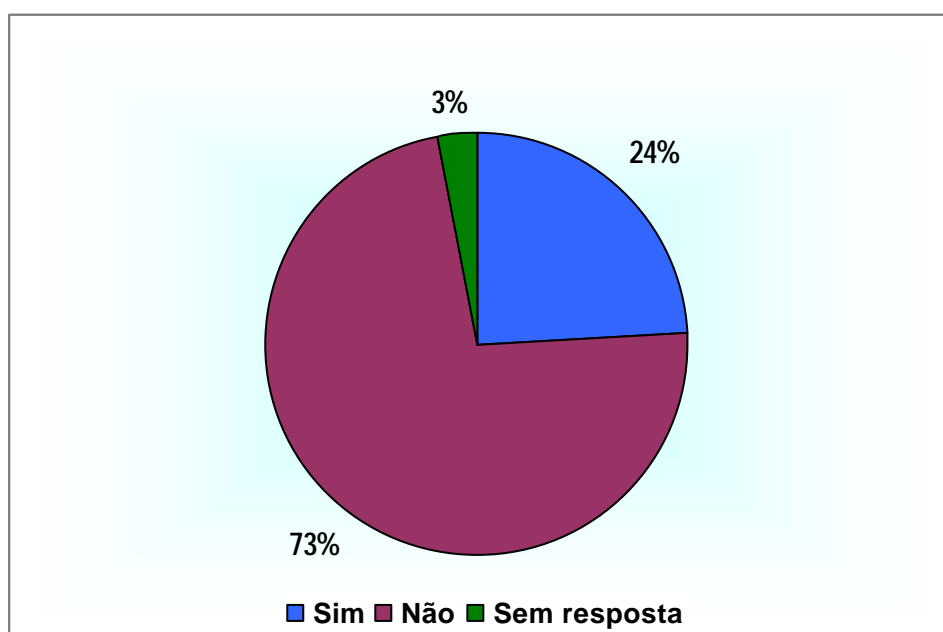


Figura 12: Freqüência de utilização do processo de gestão quantitativa dos processos

O propósito da Gestão Quantitativa dos Processos é controlar quantitativamente o desempenho do processo do projeto de software. O desempenho do processo de software representa os resultados reais alcançados ao seguir um processo de software estabelecido.

A Gestão Quantitativa dos Processos envolve estabelecer metas para o desempenho do processo de software definido do projeto, obter medições do desempenho do processo, analisar estas medições e efetuar ajustes para manter o desempenho de processo dentro de limites aceitáveis.

A organização coleta dados de desempenho dos processos a partir dos projetos de software e usa esses dados para caracterizar a capacitação (isto é, o desempenho que um novo projeto pode alcançar) do processo de software padrão da organização. A capacitação do processo descreve a faixa de resultados esperados ao seguir um processo de software (ou seja, os resultados mais prováveis e esperados de um novo projeto de software que venha a ser empreendido pela organização).

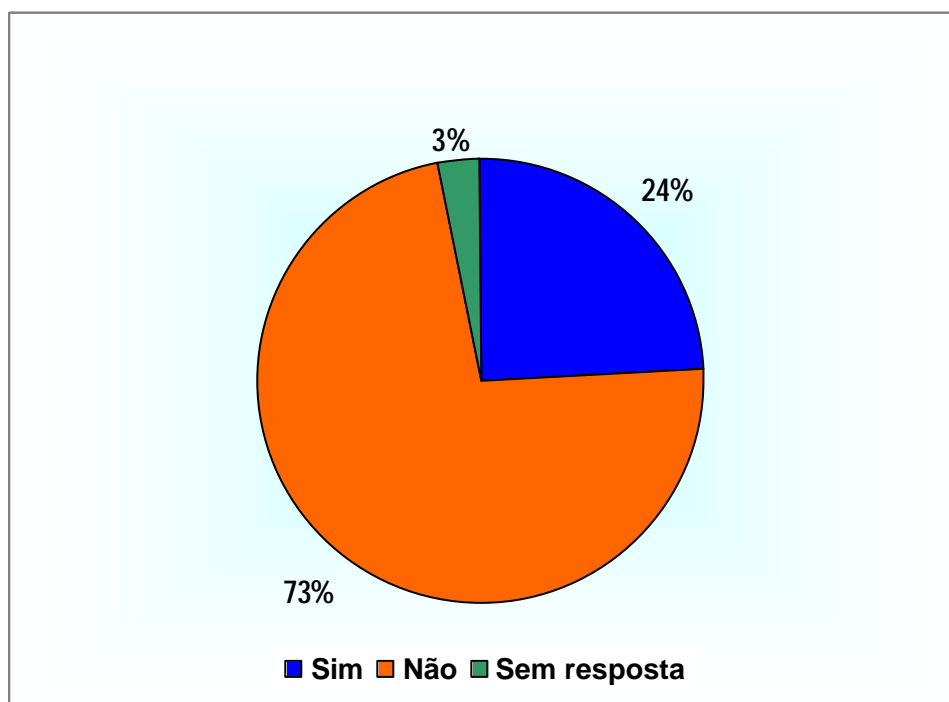


Figura 13: Freqüência de utilização do processo de gestão da qualidade de software

A Gestão da Qualidade de Software envolve definir metas de qualidade para os produtos de software, estabelecer planos para atingir essas metas, monitorar e ajustar os planos de software, artefatos de software, atividades e metas de qualidade de modo que estejam capacitados a satisfazer as necessidades e os anseios por produtos de alta qualidade por parte do cliente e dos usuários finais.

A metas quantitativas são estabelecidas para os artefatos de software com base na necessidade da organização, do cliente e dos usuários finais. Para que essas metas possam ser atingidas, a organização estabelece estratégias e planos e o projeto adapta seu processo de software definido, de modo que venha a alcançar essas metas de qualidade.

4.5 Análise dos Dados Relacionados às Empresas Situadas no Nível 5 - Otimizante

Neste nível a organização atinge um estado em que os processos estão em melhoria contínua, sendo otimizada para as necessidades de cada momento, conforme apresentado na Tabela 6.

Tabela 6: Nível 5 – Otimizante

Nível 5 - Conceitos de Gestão	Índice de Importância	Respostas Positivas	fr
22. Prevenção de Defeitos	4,16	24	73%
23. Gestão das Mudanças de Tecnologias	3,60	16	48%
24. Gestão das Mudanças de Processos	3,66	16	48%

No Nível 5, o processo Prevenção de Defeitos é o mais importante no quesito do grau de importância para desenvolvimento de software e é o mais utilizado pelas empresas conforme frequência e índice apresentado na Tabela 6.

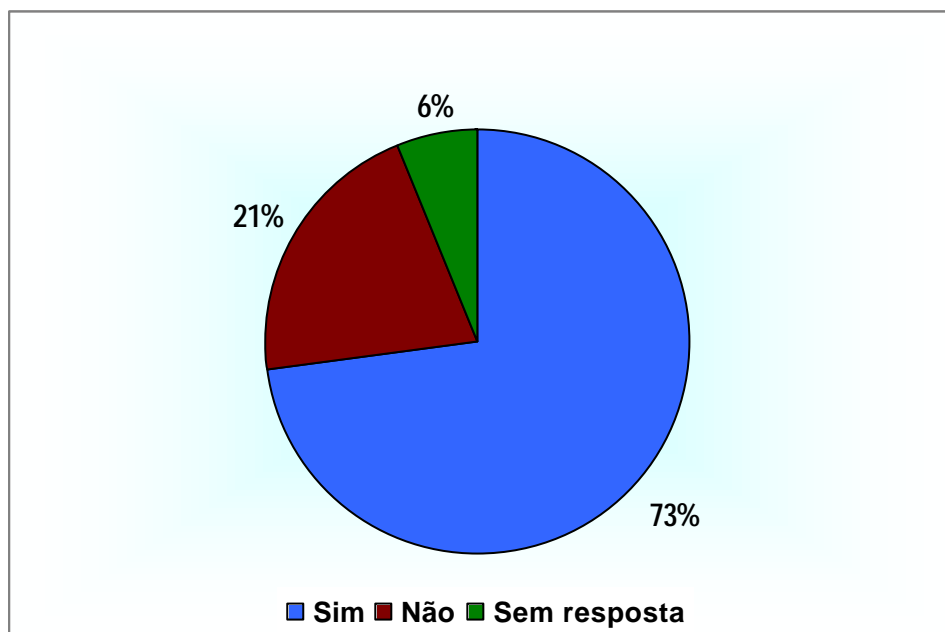


Figura 14: Frequência de utilização do processo de prevenção de defeitos

A finalidade do processo de Prevenção de Defeitos é reduzir o volume de defeitos acidentalmente inseridos nos artefatos de software. Isto contribui para a redução de retrabalho e custos associados e, ainda, para a melhoria da produtividade e da qualidade efetiva dos produtos desenvolvidos. A Prevenção de Defeitos envolve analisar defeitos que foram encontrados no passado e tomar ações específicas para prevenir ocorrências deste tipo de defeitos no futuro. Os defeitos podem ter sido identificados em outros projetos, bem como em estágios anteriores ou até em tarefas do projeto atual. Essas atividades são também um mecanismo para disseminar lições aprendidas entre os projetos. Difunde-se, assim, o *know-how* pela organização, fazendo com que este se torne um ativo da organização e não somente das pessoas nela atuam.

Basili (EICKELMANN; HAYES, 2004) lembra que organizações de software deveriam observar e analisar seus processos e produtos de desenvolvimento de software. Eles deveriam sintetizar e armazenar o que eles têm aprendido. Isto permitiria entender o que deve ser feito para melhorar seus esforços de desenvolvimento de software. Um ingrediente necessário para isso seria implementar um mecanismo efetivo de *feedback* do aprendizado dos projetos da organização.

No item Gestão das Mudanças de Tecnologias a organização define um grupo (tal como um grupo de engenharia de software, ou um grupo de suporte de tecnologia), que trabalha com projetos de software para introduzir e avaliar novas tecnologias, e gerenciar alterações nas tecnologias existentes. O objetivo deste processo é melhorar a qualidade do software, aumentar a produtividade e diminuir o ciclo de tempo requerido para o desenvolvimento de produto, ou para concluir um passo de evolução (manutenção) do produto.

A Gestão das Mudanças de Processos envolve definir metas de melhoria de processos e, com o incentivo da gerência sênior, identificar, avaliar e implementar melhorias no processo de software padrão da organização e nos processos de software definidos dos projetos. São estabelecidos programas de incentivo e treinamento para capacitar e encorajar qualquer pessoa na organização a participar nas atividades de melhoria de processo. As oportunidades de melhorias são identificadas e avaliadas quanto ao seu potencial de retorno (benefícios) para a organização. Os testes pilotos são realizados para avaliar as alterações de processo antes delas serem incorporadas na rotina da organização.

4.6 Análise dos Dados Relacionados às Questões Complementares

Os processos relacionados ao modelo SPICE e não cobertos pelo modelo SW-CMM foram selecionados e enviados aos entrevistados. Obteve-se como resposta alta frequência de utilização destes processos conforme apresentado na Tabela 7.

Tabela 7: Questões complementares – SPICE

Questões Complementares de Gestão	Respostas Positivas	fr
28. Fornecer Facilidades para o Trabalho	25	76%
27. Ambiente Integrado de Eng. de Software	22	67%
26. Facilitar o Reuso	21	64%
25. Engenhar o Negócio	16	48%

O processo Fornecer Facilidades para o Trabalho, cuja finalidade é fornecer um ambiente seguro e confiável no qual atividades de projeto de software possam ser executadas, é o mais utilizado pelas empresas que fizeram parte da pesquisa. Do total de 33 empresas que responderam o questionário, 25 delas usam este processo. Essa facilidade pode vir da utilização da ferramenta CASE.

4.7 Nível de Maturidade das Empresas Pesquisadas

Com base na análise dos dados da pesquisa as empresas, componentes da amostra, foram classificadas segundo seu nível de maturidade, conforme sugere o modelo SW-CMM.

O SW-CMM é uma arquitetura em estágios. Lembrando segundo (GHOSH, 1994) que o SW-CMM sugere que a maturidade do processo de uma organização tem que ser desenvolvido do Nível 1 para o Nível 5 progressivamente através dos níveis intermediários sem a possibilidade de cortes. De acordo com essas características obtivemos o seguinte resultado conforme apresentado na Figura 15.

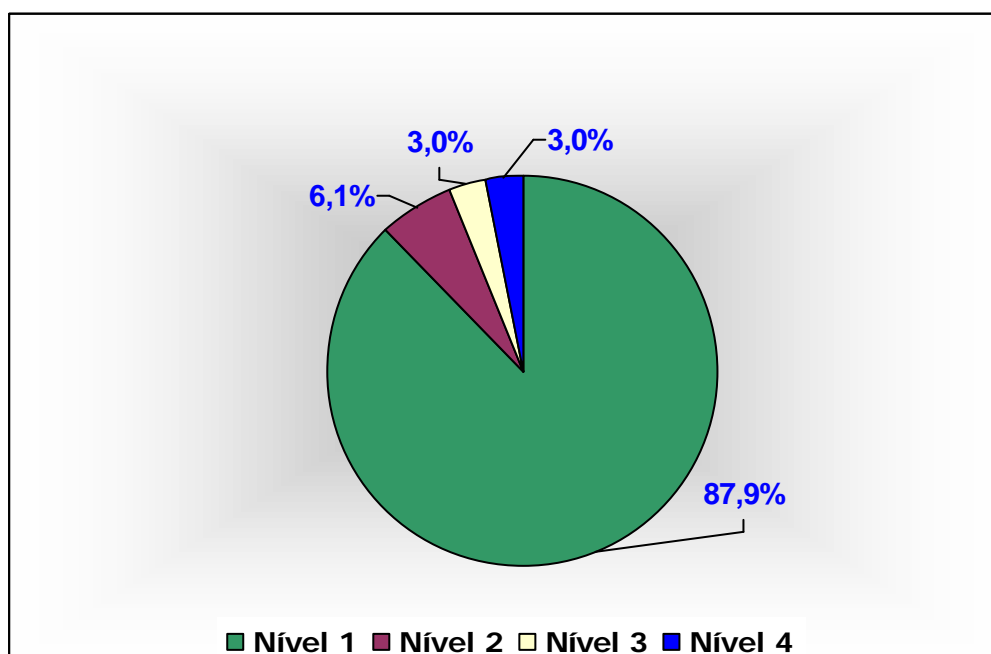


Figura 15: Níveis encontrados na pesquisa em janeiro 2004

Conforme levantamento da literatura as empresas que classificaram no **Nível Inicial ou Nível 1** o processo de software não está documentado e, usualmente, sequer existe. A organização não propicia uma ambiente estável para o desenvolvimento e manutenção de software, geralmente não é capaz de fazer estimativas de custo ou planos de projeto; se o faz, não é capaz de cumprí-los. Durante uma crise, os projetos abandonam os procedimentos que foram planejados e partem para a codificação e teste, a codificação é a única fase do processo de desenvolvimento que merece atenção. Mudanças de requisitos e de outros artefatos ocorrem sem controle. A instalação e a manutenção costumam ser deficientes, sendo tratadas como atividades de pouca importância. Poucos processos estáveis existem ou são usados. Ela utiliza processos informais e métodos *ad hoc* ou caóticos. Apesar disso, muitas dessas organizações são bem sucedidas, já que o mercado de software ainda é tolerante em relação à má qualidade dos produtos. Muitas vezes, a qualidade do marketing pode ocultar deficiências técnicas, e existe pouca competição em muitos setores deste mercado (PAULA, 2003). No Nível 1 o sucesso depende de esforços individuais e gerentes de softwares capazes.

A caracterização das empresas no SW-CMM, segundo Fiorini (1998), é fornecida conforme o nível de maturidade em que as empresas se situam.

Caracterização (Fiorini, 1998):

- ? Padrões não existem, ou tendem a ser ignorados.
- ? Ferramentas são usadas ao acaso, muitas vezes por iniciativa pessoal.
- ? Metodologias são praticadas informalmente.
- ? Resultados não são previsíveis.
- ? Coleta e análise de dados é *ad hoc*.

Tipo de capacitação: Processo *ad hoc*.

Paula (2003) comenta que os gerentes dessas organizações geralmente não entendem os verdadeiros problemas, por falta de processos que lhes dêem visibilidade real em relação ao progresso dos projetos.

Buscando apoio em Sommerville (2003), o modelo da SEI é apropriado para grandes sistemas desenvolvidos por grandes equipes de engenheiros. Ele não deve ser aplicado, sem adaptação, em circunstâncias locais. Há muitos outros tipos de projetos e organizações de software em que essa visão de maturidade de processo

não deve ser aplicada. Se a equipe for inexperiente e menos capacitada, um bom processo poderá limitar o dano.

Segundo Cameron (2002), os processos devem ser adaptados para serem usados. Abordagens de processos mais flexíveis e configuráveis são necessárias, devido a diversidades de projetos de Tecnologia de Informação. Um dos caminhos é moldar o processo para a necessidade de cada projeto em particular.

Nelson (2003) relata que os métodos para serem eficazes devem ser ajustados aos times. Os gerentes devem usar os processos que equilibrem eficientemente o desenvolvimento de métodos com talentos e necessidades individuais.

Para Cockburn (2003), características pessoais têm mais efeito, principalmente em aplicações inovadoras, na trajetória do projeto que metodologias ou processos usados. Se a equipe é composta somente de duas ou três pessoas então, a metodologia como um tópico distinto pode ser sem muita importância. Mas se o tamanho da equipe cresce, então, questão de coordenação torna-se significativa e, portanto mais consideração deve ser dada à metodologia.

As empresas que sobrevivem na economia de software são as que combinam infra-estrutura, métodos, processos e pessoas com vendas e serviços, parte de uma complexa receita onde se deve equilibrar qualidade, produtividade, custo e preço. Desenvolvimento de software depende de educação e prática, é um tipo de desenvolvimento moderno que pode, com relativa facilidade, ser contratado em qualquer lugar do mundo, desde que administre, os processos envolvidos conforme enuncia Meira (2003).

O **Nível Repetível ou Nível 2** é disciplinado em nível dos projetos. O planejamento e a gestão de novos projetos são baseados em projetos semelhantes a projetos anteriores bem-sucedidos. Uma organização neste nível é capaz de assumir compromissos referentes a requisitos, prazos e custos com alta probabilidade de ser capaz de cumpri-los. Estão estabelecidos processos básicos de gerência de projeto os gerentes de projetos acompanham custos, prazos/cronogramas e funcionalidades do software. Compromissos são firmados e gerenciados, sucessos podem ser repetidos.

O projeto de software trabalha com os seus subcontratados, se existirem, para estabelecer uma forte relação cliente fornecedor.

Caracterização (Fiorini, 1998):

- ? Gerência de projetos estabelecida.
- ? Processo organizado.
- ? Alguns procedimentos técnicos escritos.
- ? Acompanhamento da qualidade.
- ? Gerência de configuração inicial.
- ? O sucesso depende do gerente de projeto

Tipo de capacitação: Processo Disciplinado

O **Nível Definido ou Nível 3** o processo de software é documentado padronizado e integrado num processo único, chamado de Processo de Software Padrão da Organização tanto para as atividades de gerência básica como para as de engenharia de software.

Neste nível, a organização não repete os sucessos de projetos anteriores, mas estabelece uma infra-estrutura de processos que permite a adaptação a vários tipos de mudanças. O Nível 3 conduz da gestão de projetos à engenharia de produtos.

A organização Nível 3 mesmo durante as crises, sabe manter-se dentro do processo. As ferramentas passam a ser aplicadas de forma sistemática, padronizada e coerente com os processos. Portanto passam a colaborar para a melhoria da produtividade e da qualidade. Neste nível o conhecimento dos processos por parte da organização é basicamente qualitativo. Existe uma base de dados de processos, com os dados recolhidos dos projetos; esta base é utilizada para gestão dos projetos, mas não é aplicada de maneira sistemática e no nível da organização, para atingir metas quantitativas de desempenho de processo e de qualidade de produto.

Caracterização (Fiorini, 1998):

- ? Processos gerenciais e técnicos básicos bem definidos.
- ? Possibilidade de avaliação do processo
- ? Ferramentas e metodologias padronizadas.
- ? Medições iniciais de desempenho.
- ? Inspeções e auditorias rotineiras.
- ? Testes padronizados.
- ? Gerência de configuração generalizada.
- ? Evolução controlada dos processos técnicos e gerenciais básicos.

Tipo de capacitação: Processo Padronizado e Consistente

O **Nível Gerenciado ou Nível 4** os processos de software evoluem para uma forma quantitativa. São coletadas medições detalhadas do processo de software e da qualidade dos produtos. São gerados relatórios estatísticos. O processo e os artefatos de software são avaliados quantitativamente e são, também, controlados.

A organização neste nível é proficiente em coletar métricas e gerir base de dados de processo, que é povoada e analisada por profissionais treinados. Além disso, sabe intervir nos processos para atingir metas de qualidade dos produtos (PAULA, 2003).

A organização Nível 4 passa da engenharia de produtos à qualidade de processos e produtos. Ela tem elementos para decidir, qual deve ser a fração de recursos dos projetos destinada à garantia da qualidade, considerando o nível máximo de defeitos que se pode admitir nos produtos. Esse domínio quantitativo dos processos é necessário para atingir um estado de melhoria contínua.

Caracterização (Fiorini, 1998):

- ? Está estabelecido e em uso rotineiro em programa de medição.
- ? Está estabelecido um grupo de garantia da qualidade.
- ? A qualidade é planejada.
- ? A qualidade é rotineiramente avaliada e aprimorada.

Tipo de capacitação: Processo Previsível

Nesta pesquisa nenhuma empresa foi classificada para o Nível 5 no modelo SW-CMM.

4.8 Considerações Finais

Pôde-se observar que as empresas pesquisadas, que desenvolvem algum produto de software, utilizam alguns processos na construção do software, mas a grande maioria não os utiliza baseando-se no processo ou não se baseiam nos níveis que o SW-CMM propõe para a maturidade da organização. A grande maioria (87,9%) das empresas pesquisadas se encontram no Nível 1 do SW-CMM, 6,1% das empresas estão no Nível 2, 3% no Nível 3, 3% no Nível 4 e nenhuma empresa pesquisada está no Nível 5 do modelo SW-CMM.

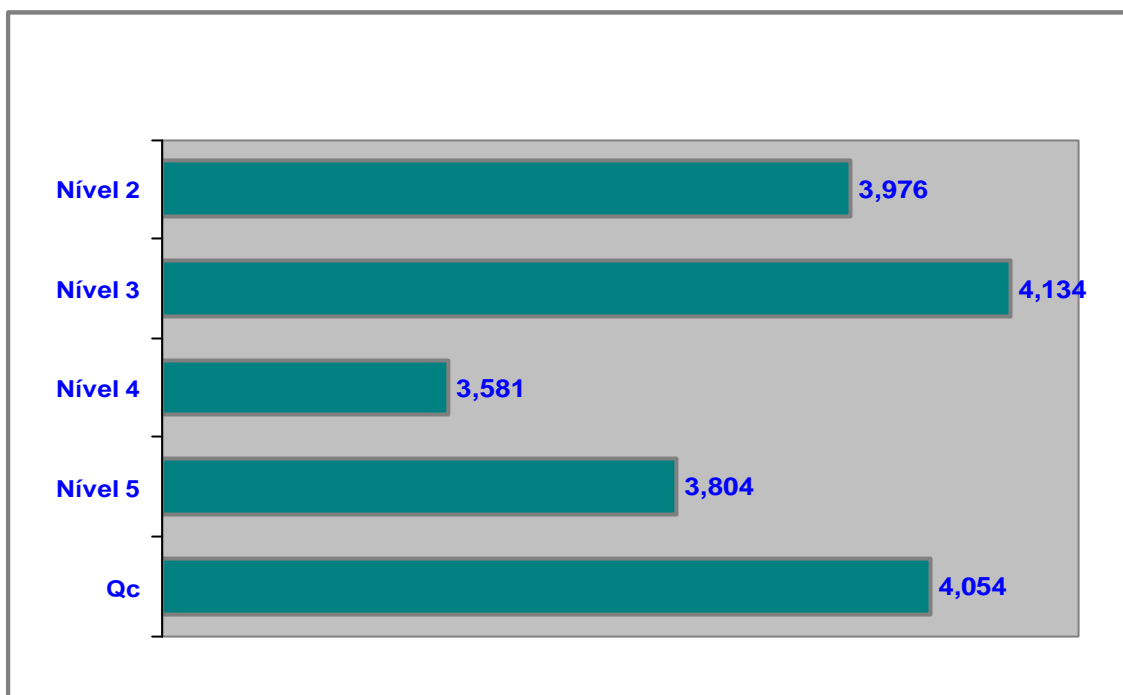


Figura 16: Índice Médio de Importância, por Nível no SW-CMM e Questões Complementares (Qc)

A Figura 16 apresenta o índice médio de importância para o desenvolvimento de software com qualidade e o peso maior está no Nível 3 e mais especificamente o mais utilizado é o “Processo de Engenharia de Produtos de Software”. Em segundo lugar, as questões complementares, processos pertencentes ao modelo SPICE, representa o índice médio de importância de 4,054, e em terceiro está o Nível 2 sendo o processo “Supervisão e Acompanhamento de Projetos” considerado o processo mais importante para construção do software com qualidade. O Nível 5 aparece em quarto lugar na opinião dos entrevistados com índice médio de importância de 3,804 e o processo com grau de importância mais alto neste nível é o processo “Prevenção de Defeitos”. E, finalmente, os processos que compõem o Nível 4 são considerados pelos entrevistados os menos importantes na construção de software.

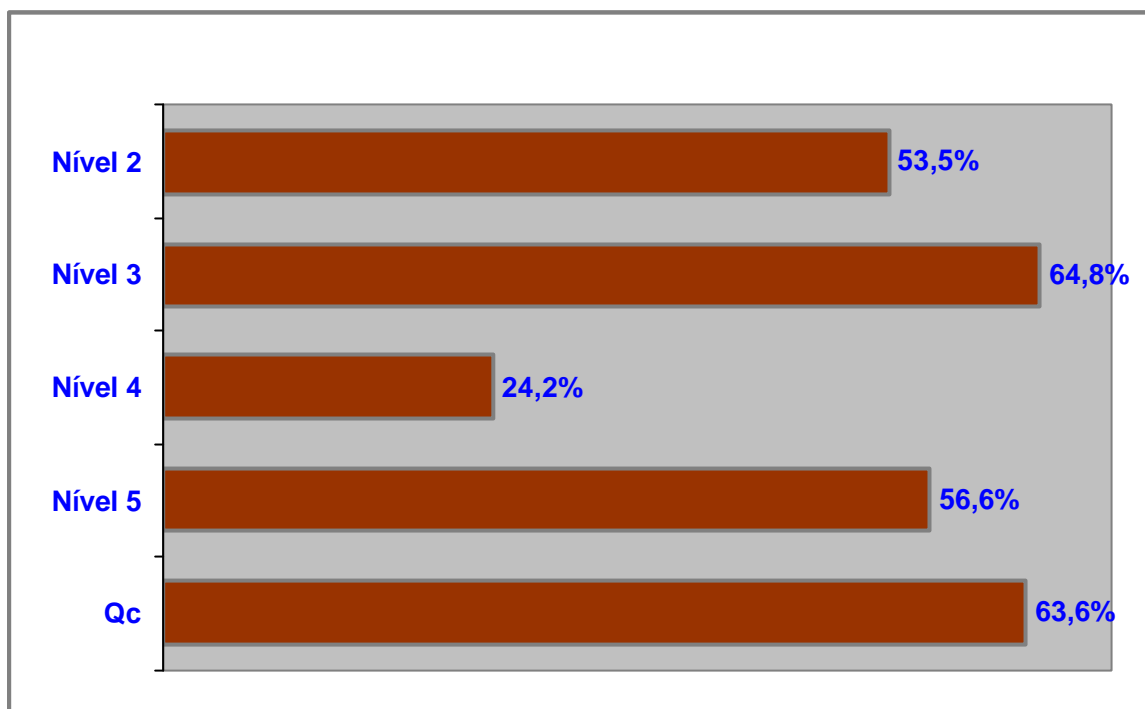


Figura 17: Percentuais Médios de Respostas Positivas na Utilização dos Níveis no SW-CMM e Questões Complementares (Qc)

Nota-se que os percentuais médios de respostas positivas, ou seja, os processos que são utilizados pelas empresas entrevistadas na construção do produto de software com qualidade, para o Nível 2, 3, 4, 5 e para questões complementares (processos do modelo SPICE), quase coincidem com o índice médio de importância da figura anterior, exceto para o Nível 2 e 5. Ficando na seqüência em ordem decrescente o Nível 3 com 64,8% em primeiro, em segundo questões complementares (processos do modelo SPICE) ficaram com a freqüência de 63,6%, em terceiro o Nível 5 aparece com percentual de freqüência de 56,6%, em quarto o Nível 2 com percentual de 53,5% e por último o Nível 4 com índice de respostas positivas de 24,2%.

CORRELAÇÃO ENTRE VARIÁVEIS: NÍVEL DE IMPORTÂNCIA E RESPOSTAS POSITIVAS (USO DO PROCESSO)	r= 0,808
--	----------

Com essas análises pôde-se observar que existe uma alta correlação entre as variáveis indicando maiores avaliações de importância aos conceitos de gestão mais utilizados (maior frequência) (correlação alta positiva).

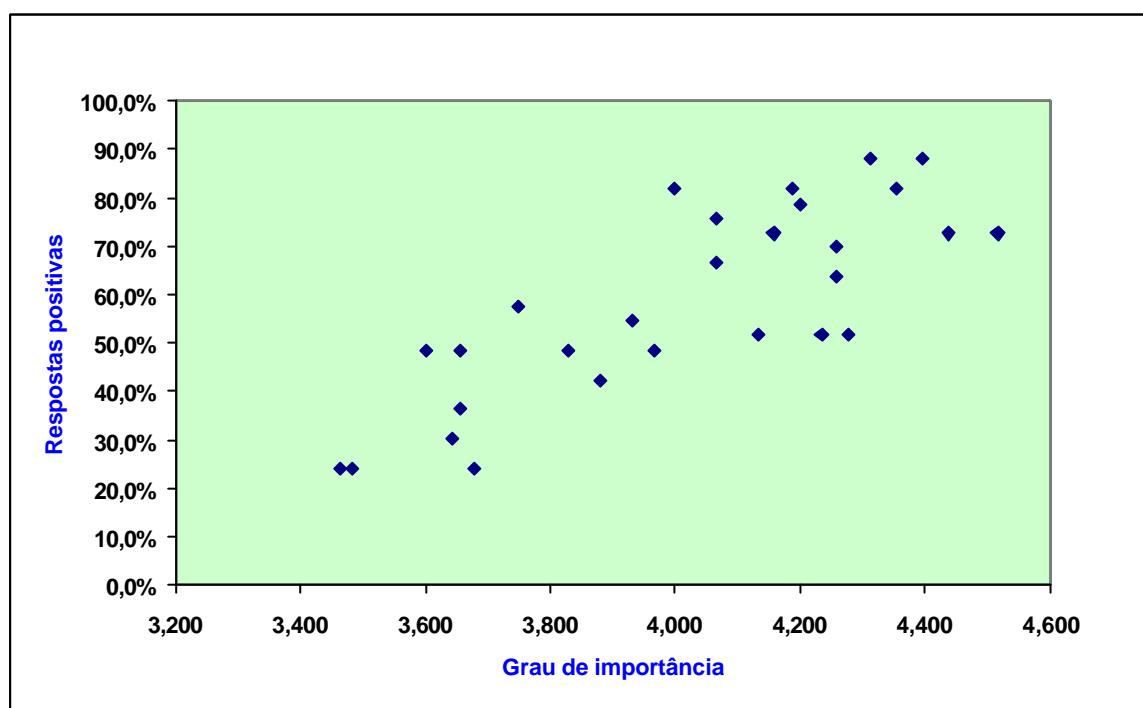


Figura 18: Correlações entre Variáveis

Na Figura 18, os processos que mais influenciaram para um alto grau de correlação entre as respostas positivas (processos que as empresas utilizam para o desenvolvimento de software) são: Supervisão e Acompanhamento de Projetos; Engenharia de Processo de Software; Prevenção de Defeitos; e Fornecer Facilidades para o Trabalho. E para o grau de importância, com pontuação mais elevada, estão os mesmos processos relacionados anteriormente e Facilitar o Reuso.

5 CONCLUSÕES E RECOMENDAÇÕES

Este capítulo apresenta as conclusões e recomendações relacionadas à presente pesquisa.

5.1 Conclusões

A principal razão para se adotar um programa de melhoria de processos é atingir um ganho significativo de qualidade e produtividade e obter o controle dos processos utilizados pela organização. Se a organização não possui controle do processo, isto se reflete em perda de produtividade e qualidade, atrasos de cronograma e aumento imprevisto nos custos.

Assim, há necessidade de se obter 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. Esse processo além de padronizado, deve ser melhorado continuamente por todos os responsáveis pelo desenvolvimento dos produtos dentro da organização.

Através desta pesquisa buscou-se diagnosticar os processos utilizados por empresas que desenvolvem software assim como o índice de importância desses processos para o desenvolvimento de software com qualidade. A ferramenta utilizada para a coleta de dados foi o questionário. A elaboração e a aplicação do questionário foi eficiente e pôde-se observar que a grande maioria das empresas utiliza algum processo para o desenvolvimento do software, não seguindo um modelo específico como o SW-CMM, mas utilizando também processos pertencentes a outro modelo, como o SPICE. A frequência de utilização deste último é de 63,6%. Também, verificou-se um alto grau de correlação entre o grau de importância que os processos têm na construção do software e a utilização prática desses processos pelas empresas, chegando a coeficiente de correlação $r=0,808$. Com isso foi possível relacionar os processos que mais contribuem para a melhoria da qualidade do software, servindo principalmente como base para aquelas empresas que ainda não utilizam nenhum processo ou utilizam parcialmente e

gostariam de avançar ou adotar essas práticas no desenvolvimento de software. O questionário serviu também como propósito de relacionar ou classificar as empresas entrevistadas aos níveis de maturidade do modelo SW-CMM - *Software Capability Maturity Model*.

No mundo globalizado, as empresas, para tentarem sobreviver em ambientes cada vez mais competitivos, investem esforços no sentido de buscarem tecnologias e soluções que empreendam e solidifiquem sua posição no mercado em que atuam. Para isso, é necessário que os sistemas de informações gerenciadas se baseiem em bases estruturais sólidas e de retorno rápido, sem falarmos da questão da qualidade no desenvolvimento do produto. Assim, as linguagens utilizadas pelos sistemas de qualidade, a engenharia de software, a arquitetura, enfim, o software, passa a ter um papel estratégico fundamental para o sucesso das empresas.

Como qualidade de software deve-se ter em mente e levar em consideração não só a engenharia em que o sistema foi erigido. É preciso, também, levar-se em consideração a integração deste software com a legislação em vigor, e com toda a legislação porventura existente do ambiente em que o software vai funcionar, etc.

Essa questão, a principal, é a qualidade dos softwares, na forma como vão se amoldar à cultura das empresas, como vão interagir e manipular a massa de dados de forma lógica, como vão trabalhar esses dados e arrumá-los de forma gerencial, visando dotar os administradores das visões gerenciais necessárias para a tomada de decisões estratégicas.

Então se pode concluir que a organização que desenvolve software pode usar vários padrões e modelos de processo de software como base para projetar e melhorar seu próprio processo.

Para melhoria do processo da organização podem ser utilizados os modelos SW-CMM e SPICE (norma ISO 15504). Estes modelos de referência descrevem as atividades que devem ser realizadas, pelas empresas que desenvolve software, para aumentar a capacidade de desenvolvimento e a maturidade do processo a nível organizacional.

O princípio básico para se sobressair no mercado de desenvolvimento de software é o comprometimento com a excelência do serviço prestado. Esse mercado é promissor, mas sem qualidade será impossível progredir e permanecer nele por muito tempo.

5.2 Recomendações para Futuros Trabalhos

Considerando os resultados da pesquisa, recomenda-se para pesquisas futuras os seguintes temas:

- ? Estudo comparativo entre os métodos ágeis - métodos leves - e métodos tradicionais - métodos dirigidos por planos - (ARMITAGE, 2004);
- ? Avaliação econômica dos ganhos com a introdução dos modelos SW-CMM e SPICE nas empresas;
- ? Proposta de modelo de qualidade de software simplificado para as empresas inseridas no Nível 1 do SW-CMM; e
- ? Metodologia para diagnóstico das necessidades organizacionais relacionadas à qualidade de software.

REFERÊNCIAS

ALBUQUERQUE, J. O. **PSP - JOA Uma abordagem orientada a Java**. Dissertação (Mestrado em Ciências da Computação) – Centro de Ciências Exatas e da Natureza, Universidade Federal de Pernambuco, Recife, 1997.

ANTONIONI, JOSÉ. **Qualidade em software: manual de aplicação da ISO-9000**. São Paulo: Makron Books, 1995.

ARMITAGE, JOHN. Are Agile Methods Good for Desing?. **Interactions**, New York: ACM Press, v. 11, issue 1, p. 14-23, Jan./Feb. 2004.

ATRASSO no setor de software. **O Estado de São Paulo**, S. Paulo, 22 de fev. 2004. p. A3, coluna 2.

BASIL, V. R.; CALDIERA, G. **Software Quality Improvement Strategy and Practice**. Institute for Advanced Computer Studies/Departament of Computer Science / University of Maryland / College Park, Maryland, Jul. 1993.

BOEHM, BARRY. Get Ready for Agile Methods with Care. **Computer**, v. 35, n. 1, p. 64-69, Jan. 2002.

BROCK, SUSAN; HENDRICKS, DANYAL; LINNELL, STEPHEN; SMITH, DEREK; . A balanced approach to IT project management. **ACM International Conference Proceeding Series**. South African Institute for Computer Scientists and Information Technologists. p. 2-10, 2003.

CAMERON, JOHN. Configurable Development Process. **Communications of the ACM**, v. 45, n. 3, p. 72-77, Mar. 2002.

CAMPOS, V. F. **TQC: Controle da Qualidade Total (no estilo japonês)**. 5. ed., Belo Horizonte: Fundação Christiano Ottoni, UFMG e Rio de Janeiro: Bloch, 1992.

COCKBURN, ALISTAIR. **People and Methodologies in Software Development**, Submitted as partial fulfillment of the degree Doctor Philosophiae at the Faculty of Mathematics and Natural Sciences, University of Oslo, Norway, 2003.

CROSBY, P. B. **Qualidade é Investimento**. Rio de Janeiro: José Olympio, 1985.

DE CICCIO, F. A versão final da ISO 9001:2000. São Paulo: **Banas Qualidade**, v. 10, n. 102, p. 32-25, nov. 2000.

DEMING, W. E. **Qualidade: a revolução na administração**. Rio de Janeiro: Marques Saraiva, 1990.

EICKELMANN, N.; HAYES, J. H. Quality time - New year's resolutions for software quality. **IEEE Software**, v. 21, issue 1, p. 12-13, Jan./Feb. 2004.

FERNANDES, A. A. **Gerência de software através de métricas: garantindo a qualidade do projeto, processo e produto**. São Paulo: Atlas, 1995.

FIORINI, SOELI T.; STAA, ARNDT VON; BATISTA, RENAN M. **Engenharia de Software com CMM**. Rio de Janeiro: Brasport, 1998.

FOURNIER, R. **Guia prático para desenvolvimento e manutenção de sistemas estruturados**. S. Paulo: Makron Books, 1994.

FREEMAN, P. **Software Perspectives The System is the Message**. Reading, MA: Addison-Wesley, 1987.

FUGGETTA, A. Software Process: A Roadmap. **Proceedings of the 22nd International Conference on the Software Engineering**. Limerick Ireland, p. 25-34, 2000.

FUGGETTA, ALFONSO; LAVAZZA, LUIGI; MORASCA, SANDRO; CINTI, STEFANO; OLDANO, GIANDOMENICO; ORAZI, ELENA. Applying GQM in an

Industrial Software Factory. **ACM Transactions on Software Engineering and Methodology (TOSEM)**, New York: ACM Press, v. 7, n. 4, p. 411-448, Oct. 1998.

GHOSH, H. A comparison of ISO 9000 and SEI/CMM for Software engineering organizations. First International Conference on Software Testing, Reliability and Quality Assurance, New Delhi India, p. 78-83, 21-22 Dec. 1994.

GIL, ANTONIO C. Como elaborar projetos de pesquisa. 3. ed., São Paulo: Atlas, 1991.

GRADY, ROBERT B. **Practical Software Metrics for Project Management and Process Improvement**. Upper Saddle River, NJ: Prentice Hall, 1992.

HALLEM, SETH; PARK, DAVID; ENGLER, DAWSON. Uprooting SOFTWARE DEFECTS at the Source. **QUEUE**, New York: ACM Press, v. 1, n. 8, p. 64-71, Nov. 2003.

HALVORSEN, CHRISTIAN PRINTZELL; CONRADI, REIDAR. A Taxonomy to Compare software Process Improvement Frameworks. **SQL - Software Quality Professional**, v. 4, n. 3, , p. 46-58, Jun. 2002.

HUMPHREY, WATTS S. A Discipline for Software Engineer. Reading, MA: Addison-Wesley, 1995.

INTHURN, C. **Qualidade & Teste de Software**. Florianópolis: Visual Books, Bookstore Livrarias Ltda, 2001.

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

ISSAC, G.; RAJENDRAN C.; ANANTBARAMAN, R. N. Significance of Quality Certification: The Case of the Software Industry in India. **Quality Management Journal**, v. 11, issue 1, 2004.

JALOTE, P. **CMM in practice: processes for executing software projects at Infosys**. Reading, MA: Addison-Wesley, 1999.

JURAN, J. M; GRZYNA, F. M. **Controle da qualidade: conceitos, políticas e filosofia da qualidade**, v. 1. São Paulo: Makron Books, 1993.

KITCHENHAM, B.; PFLEEGER, S. L. Software quality: the elusive target [special issues section]. **IEEE Software**, v. 13, issue 1, p. 12-21, Jan. 1996.

MARCONI, MARIANA DE ANDRADE; LAKATOS, EVA MARIA. **Técnicas de Pesquisa**. 5. ed., São Paulo: Atlas, 2002.

MARTINS, G. de A.; LINTZ, A. **Guia para Elaboração de Monografias e Trabalhos de Conclusão de Curso**. São Paulo: Atlas, 2000.

MEIRA, SILVIO. **Qualidade no Século de Software** - Palestra, 25 de Maio 2002, disponível em <http://www.meira.com>, acesso em: 21 de dez. de 2003.

MEIRA, SILVIO. Um Mundo Feito (Quase Completamente) de Software. **Revista da Sociedade Brasileira para o Progresso da Ciência**, ano 55, n. 2, p.24-28, abr./mai./jun. 2003.

NELSON, W. M.; Embraceable processes: adapting methods to fit the team. **IT Professional**, v. 5, issue 3, p. 42-47, May/Jun. 2003.

PALADINI, E. P. **Qualidade total na prática: implantação e avaliação de sistemas de qualidade**. São Paulo: Atlas, 1997.

PAULA FILHO, WILSON P. **Engenharia de Software - Fundamentos, Métodos e Padrões**. 2. ed., Rio de Janeiro: Editora LTC, 2003.

PAULK, MARK C. Analyzing the Conceptual Relationship Between ISO/IEC 15504 (Software Process Assessment) and the Capability Maturity Model for Software. **Proceedings of the Ninth International Conference on Software Quality**,

Cambridge, MA, p. 293-303, 4-6 Oct. 1999. PAULK (1999b). Disponível em: <<http://www.sei.cmu.edu/publications/articles/paulk/iso15504-cmm99.html>>. Acesso em: 28 jun. 2003.

PAULK, MARK C. Effective CMM-Based Process Improvement. **Proceedings of the 6th International Conference on Software Quality**, Ottawa Canada, p. 226-237, 28-31 Oct. 1996. Disponível em: <<http://www.sei.cmu.edu/publications/articles/paulk/effective-spi.html>> Acesso em: 21 jun. 2003.

PAULK, MARK C. How ISO 9001 compares with the CMM. **IEEE Software**, v. 12, issue 1 , p. 74-83, Jan. 1995. PAULK (1995b).

PAULK, MARK C. Practices of High Maturity Organizations. **The 11th Software Engineering Process Group (SEPG) Conference, Atlanta Georgia**, 8-11 Mar. 1999. PAULK (1999a). Disponível em: <<http://www.sei.cmu.edu/publications/articles/paulk/high.maturity.survey.html>>. Acesso em: 28 jun. 2003.

PAULK, MARK C. **Trends in Software Process and Quality**. Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA 15213-3890, Oct. 2002.

PAULK, MARK C. WEBER, C. V.; CURTIS, B. e CHRISSIS, M. B. **The Capability Maturity Model: Guidelines for Improving the Software Process/CMU/SEI**, ISBN 0-201-54664-7. Reading, MA: Addison-Wesley, 1995. PAULK (1995c).

PAULK, MARK C.; CURTIS, BILL; CHRISSIS, MARY BETH; and WEBER, CHARLES V. Capability Maturity Model, Version 1.1. **IEEE Software**, v. 10, n. 4, p. 18-27, Jul. 1993.

PAULK, MARK C.; KONRAD, MICHAEL D.; and GARCIA, SUZANE M. CMM Versus SPICE Architectures. **IEEE Computer Society Technical Council on Software Engineering**, n. 3, p. 7-11, Spring 1995. PAULK (1995a).

PFLIEGER, S. L. **Software Engineering - Theory and practice**. New Jersey: Prentice Hall Inc., 2001.

PRESSMAN, ROGER S. **Engenharia de Software**. 5. ed., Rio de Janeiro: McGraw-Hill, 2002.

REZENDE, D. A.; ABREU, A. F. **Tecnologia da Informação Aplicada a Sistemas de Informações Empresariais: o papel estratégico da informação e dos sistemas de informação nas empresas**. São Paulo: Atlas, 2000.

ROCHA, A. R. C.; MALDONADO, J. C.; WEBER, K. C. **Qualidade de Software**. São Paulo: Prentice Hall, 2001.

SALVIANO, CLENIO F.; CUNHA, MARIA ALEXANDRA V. C. da; CORTES, MARIO LÚCIO; OLIVEIRA, WILLIAM L. de. In: Rocha, Maldonado e Weber. *Qualidade de Software: Teoria e Prática*. São Paulo: Prentice Hall, p. 29-34, 2001.

SEI - **Software Engineering Institute**. Carnegie Mellon University, Disponível em <http://www.sei.cmu.edu/>. Acesso em: 08 de set. 2003.

SHILLER, LARRY. **Excelência em Software**. São Paulo: Makron Books, 1992.

SILVA, E. L. e MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. 3. ed. Florianópolis. Laboratório de Ensino a Distância da UFSC, 2001.

SOMMERVILLE, IAN. **Engenharia de Software**. São Paulo: Addison Wesley, 2003.

SPICE Project: “**SPICE**” - ISO/IEC Software Process Assessment – Part 1: Concepts and Introductory Guide (Working Draft v 1.00)” – 2002.

SPICE Project: “**SPICE**” - ISO/IEC Software Process Assessment – Part 2: A Model for Process Management (Working Draft v 1.00)” – 2002.

SPICE, disponível em <http://www.sqi.cit.gu.edu.au/spice/>, acesso em: 01 de out. 2003.

STARK, J. A.; CROCKER, R. Trends in software process: the PSP and agile methods. **IEEE Software**, v. 20, issue 3, p. 89-91, May/Jun. 2003.

STYLIANOU, ANTONIS C.; KUMAR, RAM L. An Integrative Framework for IS Quality Management. **Communications of the ACM**, v. 43, n. 9, p. 99-104, Sep. 2000.

TSUKUMO, ALFREDO N.; RÊGO, CLAUDETE M.; SLAVIANO, CLENIO F.; AZWEVEDO, GLAUCIA F.; MENEGHETTI, LUCIANO K.; COSTA, MÁRCIA C. C.; CARVALHO, MÁRIO BENTO DE; COLOMBO, REGINA M. T. **Qualidade de Software: Visões de Produto e Processo de Software**. Publicado na "II Escola Regional de Informática da Sociedade Brasileira de Computação Regional de São Paulo – II ERI da SBC" – Piracicaba, SP, p. 173-189, jun. 1997.

VIANA, A. C. H. **Satisfação do cliente como metas no desenvolvimento de software**. Simpósio Brasileiro de Engenharia de Software. Anais. Curitiba: CITS, 1994.

VIDOTTI, CLEBER APARECIDO. **Método Simplificado de Desenvolvimento de Software para Empresa de Pequeno e Médio Porte**: Uma Aplicação prática na WOPM Informática. Dissertação (Mestrado em Engenharia de Avaliação e Inovação Tecnológica) - UFSC, Florianópolis, 2003.

WEINBERG, G. M. **Software com Qualidade: pensando e idealizando sistemas**. São Paulo: Makron Books, 1993.

WINTERS, G. R. **Software Process Improvement Overview**. Carnegie Mellon University, 1997.

APÊNDICES

Apêndice A - Carta de Apresentação da Pesquisa



Universidade Federal de Santa Catarina - UFSC

Programa de Pós-Graduação em Engenharia de Produção - PPGEP

Núcleo de Estudos em Inovação, Gestão e Tecnologia da Informação – IGTI

Curitiba, 09 de janeiro 2004.

Prezado(a) Senhor(a)

Esta pesquisa busca identificar o comportamento das empresas produtoras de software em relação à qualidade no processo de desenvolvimento de produtos. Tem por finalidade a consecução da dissertação de mestrado de Tereza Tiemi Nakajima, no Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

Solicitamos vossa colaboração no sentido de fornecer as informações requeridas, para o presente estudo, respondendo o questionário anexo. Certo de vossa compreensão e disponibilidade aguardamos a devolução do questionário respondido até o dia **06 de fevereiro de 2004**.

Sendo o que se apresenta para o momento, agradecemos antecipadamente vossa colaboração e nos colocamos à vossa disposição para eventuais esclarecimentos que se façam necessários.

Atenciosamente,

Tereza Tiemi Nakajima

Mestranda da UFSC

Tel. (41)262-5951

Email: terezanakajima@netbank.com.br

Apêndice B - Carta de Apresentação do Pesquisador



Universidade Federal de Santa Catarina - UFSC

Programa de Pós-Graduação em Engenharia de Produção - PPGEPP

Núcleo de Estudos em Inovação, Gestão e Tecnologia da Informação – IGTI

Curitiba, 09 de janeiro 2004.

Esta carta visa apresentar a mestrandia Tereza Tiemi Nakajima e fornecer a visão geral da pesquisa proposta para a conclusão do trabalho de dissertação de Mestrado, no programa de Pós-Graduação em Engenharia de Produção, pela UFSC.

O objetivo desta pesquisa é verificar quais os processos que são utilizados pelas empresas no desenvolvimento de software e relacionar ao modelo de qualidade, Modelo de Maturidade da Capacidade do Software - SW-CMM e Melhoria do Processo de Software e Determinação da Capacidade - SPICE.

Dessa forma, o projeto de estudo visa explorar um grupo de empresas que possuem um departamento, ou uma unidade organizacional para o desenvolvimento de software, ou empresas que construam software como negócio (indústria de software/*softhouse*).

Para a coleta de dados é utilizado um questionário estruturado, se alguma identidade nos é revelada, será conservada em estrito sigilo.

Em nome de toda a equipe do Núcleo de Estudos em Inovação, Gestão e Tecnologia da Informação – (IGTI), desejo expressar nossa gratidão pela colaboração de todos e ressaltar a importância para os propósitos da pesquisa de se obter uma taxa de retorno dos questionários próxima a 100%.

Orientador:

Prof. Pedro Felipe de Abreu, Ph.D

Apêndice C - Questionário da Pesquisa

Pesquisa Sobre Processos Para o Desenvolvimento de Software

O objetivo é relacionar os processos mais utilizados pelas empresas para o desenvolvimento de software.

Nesta pesquisa, há um espaço livre para complementar com processos que não estejam listados, mas que você considera fundamental.

Empresa/Instituição/Organização (opcional): _____

Nome (opcional): _____

A sua empresa utiliza alguns dos processos relacionados abaixo? Para cada item assinale (marque com um X) a alternativa e apenas uma. Caso utilize o processo ou não o utilize, mas ache relevante, preencha a escala no “()” colocando o grau de importância para o desenvolvimento de um software com qualidade, conforme indicação na Tabela 1.

TABELA 1 – ESCALA TIPO LIKERT

Influência Muito Baixa	Influência Baixa	Influência Média	Influência Alta	Influência Muito Alta
(1)	(2)	(3)	(4)	(5)

<p>1. Desenvolve Requisitos e Projeto (Design) do Sistema: estabelece os requisitos e projeto do sistema, feito de forma documentada, identificando quais requisitos do nível de sistema poderiam ser alocado para quais elementos do projeto do sistema e para quais versões.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim</p> <p>() Não</p> <p>() Outra.</p> <p>Especifique</p> <hr style="width: 80%; margin-left: 0;"/> <p>()</p>
---	---

<p>2. Desenvolve Requisitos do Software: estabelece, analisa e refina, de forma documentada e revisada, os requisitos do software.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique</p> <hr/> <p>()</p>
<p>3. Desenvolve o Design do Software: estabelece um desenho de software que efetivamente acomoda os requisitos, no topo do nível identifica e os maiores componentes do software e refina em níveis menores os quais podem ser codificados, compilados e testados.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique</p> <hr/> <p>()</p>
<p>4. Implementa o Design do Software: produz código de software o qual implementa os componentes do design do software.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique</p> <hr/> <p>()</p>
<p>5. Integração e Teste do Software: integra e testa as unidades do software produzido com outros softwares produzidos ou já existentes que satisfarão os requisitos do software.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique</p> <hr/> <p>()</p>

<p>6. Integração e Teste do Sistema: integra e testa o software com operações manuais e elementos de hardware produzindo um sistema que satisfará os requisitos do sistema.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>
<p>7. Manutenção do Sistema e do Software: modifica um ou mais dos seguintes itens: o sistema aplicativo, seu hardware, sistema em rede, software aplicativo e a documentação associada em resposta aos pedidos dos usuários (ou não usuários), preservando a integridade do design do sistema.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>
<p>8. Gestão de Requisitos: permite definição e controle dos requisitos em que se baseiam os compromissos.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>
<p>9. Planejamento de Projetos: prevê prazos e custos para cumprimento dos compromissos, como bases técnicas e não apenas intuitiva.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>

<p>10. Supervisão e Acompanhamento de Projetos: conferem o atendimento dos compromissos, comparando o conseguido com o planejado e adicionando providências corretivas sempre que haja desvios significativos em relação aos compromissos.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>
<p>11. Gestão da Subcontratação: cobra de organizações subcontratadas, para desenvolver partes do software, os mesmos padrões de qualidade que a organização principal oferece aos seus clientes (se a empresa não subcontrata marque “Não” como resposta).</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>
<p>12. Garantia da Qualidade: existe um grupo de garantia de qualidade independente dos projetos, o GGQSw - Grupo de Garantia de Qualidade de Software, que confere o cumprimento dos compromissos, de forma independente em relação aos projetos.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>
<p>13. Gestão de Configurações: garante a consistência permanente dos resultados dos projetos, entre si e com os requisitos, ao longo do projeto, mesmo quando ocorrem alterações nos compromissos.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>
<p>14. Focalização dos Processos da Organização: estabelecimento formal de um grupo de engenharia de processos de software responsável pelas atividades de desenvolvimento, melhoria e manutenção de processos de software.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>

<p>15. Definição dos Processos da Organização: estabelecimento de um processo padrão de software no nível da organização, a partir do qual devem ser derivados os processos definidos para os projetos.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>
<p>16. Programa de Treinamento: estabelecimento de um <i>programa de treinamento</i> em processos de software, no nível da organização para os profissionais de informática.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>
<p>17. Gestão Integrada dos Projetos: baseada nos processos definidos para os projetos, com uso de procedimentos documentados para gestão de tamanho, esforços, prazos e riscos.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>
<p>18. Coordenação entre Grupos: que participam de projetos de sistemas, no nível da organização. Estabelecer mecanismos para o grupo de engenharia participar junto a outros grupos de engenharia. Por ex: grupos de engenharia de sistemas e de hardware.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>
<p>19. Revisões: coordenação de revisões no nível da organização (p. ex. revisão técnica, inspeção, walkthrough e revisão gerencial).</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p>() Sim () Não () Outra. Especifique _____ ()</p>

<p>20. Gestão Quantitativa dos Processos: controla quantitativamente o desempenho dos processos usados pelos projetos.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>
<p>21. Gestão da Qualidade de Software: promove o entendimento quantitativo da qualidade dos produtos de software, permitindo atingir metas quantitativas desejadas.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>
<p>22. Prevenção de Defeitos: identifica e remove as suas causas.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>
<p>23. Gestão das Mudanças de Tecnologias: com procedimentos sistemáticos de identificação, análise e introdução de tecnologia apropriada.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>
<p>24. Gestão das Mudanças de Processos: uso dos dados de processo para gestão das mudanças de processos, colocando os processos em melhoria contínua.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim</p> <p><input type="checkbox"/> Não</p> <p><input type="checkbox"/> Outra.</p> <p>Especifique</p> <hr/> <p><input type="checkbox"/></p>

<p>25.Engenhar o Negócio: fornece aos indivíduos visão e cultura dentro de uma organização e projetos que os autorizam a funcionar efetivamente.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Outra. Especifique _____ <input type="checkbox"/></p>
<p>26.Facilitar o Reuso: maximiza o reuso do sistema e os componentes do software existentes, guiando para um baixo custo de manutenção e desenvolvimento e alta qualidade do produto.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Outra. Especifique _____ <input type="checkbox"/></p>
<p>27.Fornecer um Ambiente Integrado de Engenharia de Software: fornece um conjunto integrado de ferramentas para o desenvolvimento de software para uso coerente dos projetos da organização, apoiado no processo padrão.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Outra. Especifique _____ <input type="checkbox"/></p>
<p>28.Fornecer Facilidades para o Trabalho: fornece um ambiente seguro e confiável no qual atividades de projeto de software possam ser executadas ou utiliza CASE.</p> <p>Grau de Importância do Processo conforme Tabela 1</p>	<p><input type="checkbox"/> Sim <input type="checkbox"/> Não <input type="checkbox"/> Outra. Especifique _____ <input type="checkbox"/></p>