



Universidade Metodista de Piracicaba

Faculdade de Ciências Exatas e da Natureza

Mestrado em Ciência da Computação

**Um Método para a Melhoria do Processo de
Desenvolvimento de Software Aplicando Conceitos de
CMM, SPICE e da Norma ISO 12207.**

Júlio César Navas

Banca examinadora: Prof. Dr. Rafael Ferreira Alves (orientador)

Profa. Dra. Tereza Gonçalves Kirner

Prof. Dr. Carlos Miguel Tobar Toledo

PIRACICABA - São Paulo - Brasil

Defesa: 20.abr.2006

ficha catalográfica

Navas, Júlio César

Um método para a melhoria do processo de desenvolvimento de software aplicando conceitos de CMM, SPICE e da norma ISO 12207. Piracicaba, 2006.

p.

Orientador - Prof. Dr. Rafael Ferreira Alves

Dissertação (mestrado) - Programa de Pós-Graduação em Ciência da Computação - Universidade Metodista de Piracicaba

1- Software - Engenharia de Sistemas. 2- Engenharia de Software - Qualidade.

“Toda vez que um artigo científico apresenta alguns dados, eles vêm acompanhados por uma margem de erro – um lembrete silencioso, mas insistente, de que nenhum conhecimento é completo ou perfeito. É uma calibração de nosso grau de confiança naquilo que pensamos conhecer. Se as margens de erro são pequenas, a acuidade de nosso conhecimento empírico é elevada; se são grandes, então é também enorme a incerteza de nosso conhecimento. Exceto na matemática pura (e, na verdade, nem mesmo nesse caso), não há certezas no conhecimento.”

Sagan C., O Mundo Assombrado pelos Demônios, p.41.

À

Joyce, companheira e parceira nesta jornada, que iluminou a pesquisa-ação, a análise de conteúdo e a análise qualitativa, e a quem, com amor, reconheço os méritos de educadora e conciliadora;

Aos meus filhos, Patrícia Helena, Soraia Cecília e Rafael Ângelo, tele-participantes na pesquisa-ação pelo sacrifício do convívio;

Ao Arthur e Giovana pela tranquilidade no convívio.

A todos, meu amor e dedicação.

AGRADECIMENTOS

À Secretaria da Receita Federal (SRF) especialmente aos colegas lotados na Delegacia em Piracicaba – SP, na pessoa do Sr. Delegado em exercício, Dr. Luis Antonio Arthuso; e aos demais colegas da Coordenação de Tecnologia (DF) e sua projeção em São Paulo (8ª região) pela compreensão da importância deste trabalho em melhorar continuamente a qualidade de sistemas e serviços de Tecnologia da Informação no âmbito de nossa Instituição.

Aos professores, pesquisadores, orientadores, colegas e alunos das instituições de ensino superior por onde passamos pela oportunidade de com eles aprender: IMES (Instituto Educacional de Ensino Superior de São Caetano do Sul); FGV-EASP-SP (Escola de Administração de Empresas de São Paulo); UnB (Universidade Nacional de Brasília); ETUC (ex Escritório Técnico de Informática da PUC-DF); ICAT-UDF (Instituto de Cooperação e Assessoria Técnica da Universidade do Distrito Federal); USF (University of South Florida) e UNIMEP - Universidade Metodista de Piracicaba – SP.

Aos colegas, executivos em TI, analistas de sistemas, programadores e demais colaboradores em qualquer função, das organizações onde tivemos o prazer de trabalhar, arriscar e implantar soluções em diversas funções e cargos: General Motors do Brasil S.A.; Volkswagen do Brasil S.A.; PRODAM-SP; COBAL (CPD-Min. Agricultura); Brastemp S.A / Whirlpool Corporation; Grupo Pão de Açúcar; ABAC (atual EAN-Brasil) – Associação Brasileira de Automação Comercial; Navas & Associados; AKZO - Div. Tintas (Br); ABC ALGAR S.A; Heublein S.A.

Aos colegas do SERPRO lotados em Piracicaba e da pequena equipe de Tecnologia e Segurança da Informação local, pelo apoio frente aos desafios diários do uso das TIs.

AGRADECIMENTOS ESPECIAIS

Aos profs. Dra. Tereza Gonçalves Kirner e Dr. Carlos Miguel Tobar Toledo, tanto por suas intervenções, re-orientações e sugestões preciosas que permitiram correções de rumo e aperfeiçoamentos significativos desde a concepção até a defesa em banca do trabalho proposto, principalmente no tocante à metodologia e adaptação às exigências de uma pesquisa acadêmica, como também, e em conjunto, à Profa. Dra. Marina Teresa Pires Vieira, pela viabilização do curso de pós-graduação em Ciência da Computação nesta Universidade Metodista de Piracicaba, conquista da maior importância a todos nós.

Ao orientador, prof. Dr. Rafael Ferreira Alves, pelo empenho, dedicação, incentivo, paciência e confiança neste trajeto, especialmente pela troca de idéias e experiências e pela colocação de desafios que, acreditamos, deram robustez ao método. Adicionalmente, pela compreensão e apoio que viabilizaram soluções face às inúmeras dificuldades surgidas.

Reconhecemos ter testado ao limite a paciência, compreensão e tolerância destes mestres pelas inúmeras falhas que cometemos no processo de compilação e montagem deste trabalho decorrentes de nossas limitações e escassa disponibilidade de tempo a enfrentar o que consideramos um enorme desafio: A Melhoria da Qualidade do Processo de Construção de Software. A eles nosso débito eterno como reconhecimento.

Ao prof. Dr. Kleber Xavier Sampaio de Souza cuja percepção, sensibilidade e visão abriu o caminho e nos orientou em produtivos encontros.

Aos profs. Dr. Luiz Camolesi Jr. e Dr. Márcio Merino Fernandes por suas contribuições e orientações por ocasião do Exame de Qualificação deste trabalho.

A todos os professores já citados com quem tivemos, adicionalmente, o prazer de assistir aulas ou receber alguma orientação, incluindo os profs. Dra. Angela Maria Cassavia Jorge Correa, Dr. Joinvile Batista Junior, Dr. Ricardo Luís de Freitas, Dr. Luis Eduardo Galvão Martins, Dr. Nivaldi Calonego Júnior e Dra. Maria Imaculada de Lima Montebelo, mestres das disciplinas obrigatórias e optativas do Programa de mestrado pela oportunidade de rever conceitos, atualizar conhecimentos, explorar novas idéias e confrontar opiniões. Suas atuações, influências e intervenções permitiram abrir o caminho a esta pesquisa.

Ao corpo de funcionários e aos ex-funcionários da UNIMEP, em especial à Rosa Maria Alves e à Solange Sabadin, que, em sua exemplar dedicação e generosidade, nos ajudaram em todas as tarefas que nos foram exigidas.

Ao prof. Ms. Vilmondes Gomes da Silva e amigos de Brasília, em especial Carlos Roberto Chamelete e João Carlos Angelini da Policentro e PiX Tecnologia, e João Serrano da CONSIST S.A. (São Paulo) pela atenção e disposição com que sempre nos atenderam oferecendo parte de seu tempo, conhecimentos e recursos.

Ao amigo Wagner Tadeu Rodrigues, empresário de sucesso, que sabe fazer acontecer, pela coragem de expor sua Organização à pesquisa, e permitindo tão intenso envolvimento com seus colaboradores, servindo, desinteressadamente, como um verdadeiro irmão.:

Aos colaboradores da STORE, em todos os níveis, verdadeiros heróis da qualidade no lidar com os processos de software, pela dedicação e oportunidade de aprendizado.

Todos são merecedores de nosso carinho e respeito, pois, graças aos momentos enriquecedores de convivência produtiva, acreditamos ter conseguido entender o tema, compreender sua importância, e, principalmente, produzir com mais qualidade.

O fruto desta experiência se materializa neste trabalho, cuja qualidade, infelizmente, não pôde atingir a excelência por exclusiva responsabilidade ou incapacidade nossa em “driblar” a gama de adversidades e o volume de detalhes e inter-relacionamentos que uma pesquisa e comprovação acadêmica de conteúdo exigem quando se empreende a pesquisa-ação.

A lista de agradecimentos é extensa – faltando, certamente, outros ainda a agradecer, aos quais registramos nossas desculpas - por uma simples razão: com todos os que interagimos temos a oportunidade de aprender o que é qualidade. Talvez seja difícil, a princípio definir e explicar perfeitamente o que qualidade significa, porém, é com as pessoas, em nossos diários relacionamentos, que passamos a compreender sua importância.

RESUMO

NAVAS, J.C, - **Um Método para a Melhoria do Processo de Desenvolvimento de Software Aplicando Conceitos de CMM, SPICE e da Norma ISO 12207** - Dissertação de Mestrado – Curso de Pós Graduação em Ciência da Computação - Universidade Metodista de Piracicaba - UNIMEP - Piracicaba – SP, 2006.

O trabalho propõe um método para a melhoria da qualidade do Processo de Desenvolvimento de *Software* e sua validação experimental inicial em uma experiência piloto de demonstração, através de um trabalho de pesquisa-ação.

A elaboração do método baseia-se na aplicação de conceitos da Engenharia de Sistemas e de *Software*, nas abordagens consagradas de melhoria da qualidade de *software* e nas Normas da Qualidade de Software em estudo e em vigor, no Brasil e no exterior, especialmente o CMM, CMMI, SPICE e PMBOK. O objetivo primordial desta dissertação é permitir tornar mais eficiente o processo de criação de software e facilitar a obtenção de selos de certificação da qualidade, ou aprovação em auditorias da qualidade, por parte das *softwarehouses* brasileiras.

A validação experimental inicial foi possível através de um teste piloto em uma empresa brasileira de software de pequeno/médio porte constituindo um estudo de caso.

Termos de Indexação: *Software* – Engenharia de Sistemas – Engenharia de *Software* – Qualidade – Qualidade de *Software* - Método de Melhoria da Qualidade de *Software* - Método de Melhoria da Qualidade do Processo - Método de Melhoria da Qualidade do Processo de *Software* – Método de Melhoria da Qualidade do Processo de Desenvolvimento de *Software* - Programa de Melhoria da Qualidade de *Software* – Modelo de Melhoria da Qualidade de *Software* – CMM – CMMI – SPICE – NBR ISO/IEC 12207 – ISO / IEC 15504 – PMBOK – MC2Q-SW.

ABSTRACT

NAVAS, J.C, - **A Method for Software Process Improvement Applying CMM, SPICE and ISO 12207** – Master of Sciences Degree Dissertation – Computer Science Post Graduation Course - Universidade Metodista de Piracicaba - UNIMEP - Piracicaba – SP, Brazil - 2006.

This paper proposes a method for SPI (Software Process Improvement) and reports the experience of its first experimental evaluation through the action-research technique.

The knowledge bases for the method were Systems and Software Engineering, recognized frameworks for SPI and International Quality Standards, specially CMM, CMMI, SPICE and PMBOK. The main goal of this work is to support the Software Process in order to enhance its efficiency and to facilitate the submission of brazilian softwarehouses to Quality Certification or Quality Auditing Processes.

The first experimental validation was possible through a pilot-test in a small/medium-sized brazilian softwarehouse here documented as a case-study.

SUMÁRIO

RESUMO.....	ix
ABSTRACT.....	xi
DEFINIÇÕES.....	xxii
LISTA DE FIGURAS.....	xvi
LISTA DE TABELAS e QUADROS.....	xix
SUMÁRIO DOS ANEXOS.....	xx
SIGLAS.....	xxi
DEFINIÇÕES.....	xxii
1. INTRODUÇÃO.....	1
1.1. Objetivos	2
1.2. Justificativas da Pesquisa	3
2. REFERENCIAL TEÓRICO ..	10
2.1. Os Modelos dos Processos de Software.....	10
2.1.1. O modelo Cascata (Waterfall)	11
2.1.2. O Modelo de ciclo de vida com prototipação.....	13
2.1.3. O Modelo Espiral.....	14
2.1.4. Considerações finais a respeito de modelos.....	19
2.2. Qualidade aplicada aos Processos de Software.....	20
2.3. Modelos de Qualidade de Software—Uma Necessidade.....	25
2.4. A NBR ISO/IEC 12207	27
2.5. O CMM – Capability Maturity Model.....	29
2.5.1 O SEI (Software Engineering Institute)	31
2.5.2. Maturidade.....	32
2.5.3. A Estrutura do CMM	33
2.5.4. Os Níveis da Maturidade de Processo de Software	33
2.5.5. Áreas-chave de processo (Key Process Areas ou KPAs).....	36
2.5.6. Objetivos das áreas-chave de processo	36
2.5.7. As Características comuns e práticas-base	40
2.6. A Norma ISO / IEC 15.504 (SPICE).....	41
2.6.1. Histórico do SPICE.....	41

2.6.2.	Estrutura do SPICE.....	42
2.6.3.	Categorias de Processos.....	44
2.6.4.	Níveis de Capacitação e Atributos do Processo.....	47
2.6.5.	A Avaliação da Capacitação de Processos e o Perfil de Capacidade.....	50
2.6.6.	Considerações Finais sobre o SPICE.....	52
3.	METODOLOGIA DE PESQUISA E APLICAÇÃO.....	53
3.1	Procedimentos da Pesquisa-Ação.....	53
3.1.1.	Pesquisa Bibliográfica	55
3.1.2.	Determinação e Delimitação do Objeto de estudo	56
3.1.3.	Elaboração de Hipóteses para o Estudo de Caso	56
3.1.4.	Elaboração de um Modelo Teórico	57
3.1.5.	Validação das hipóteses restritas ao caso e o teste experimental inicial.....	58
4.	PROPOSTA DE UM MÉTODO.....	59
4.1.	Apresentação do Método.....	60
4.1.1.	Visão Geral do MC2Q-SW (Fases e Etapas).....	61
4.1.2.	Os Artefatos do MC2Q-SW.....	66
4.2.	Histórico do Desenvolvimento e a Evolução do Método.....	71
4.3.	Melhoria da Qualidade do Processo de Software como um Programa Contínuo.....	72
4.3.1.	A Síndrome da Fachada e o MC2Q-SW.....	74
4.4.	Diretrizes e Princípios ao Desenvolvimento do Método.....	78
4.5.	O Modelo Evolutivo dos três Estágios da Organização em direção a um Programa de Melhoria da Qualidade do Processo de Software.....	83
4.6.	O Diagnóstico Situacional – (Etapa 1.1) e a Proposição de Ações.....	88
4.6.1.	Uma Metodologia Prática de Diagnose Situacional.	
4.7.	As Ações de Capacitação – (Etapa 1.2).....	95
4.7.1.	A Espiral da Melhoria da Qualidade do Processo.	96
4.7.2.	Os Ciclos de Capacitação.....	97
4.7.2.1.	A organização Necessária a um Projeto de Melhoria Contínua da Qualidade do Processo de Software.	101
4.7.2.2.	A Determinação do Paradigma para a Base de Processos.....	103
4.8.	A Seleção dos Processos (Etapa 2.1).....	104
4.8.1.	A Tabela de Decisão para Determinação dos Processos (TDDP).	105
4.9.	A Identificação de Métricas (Etapa 2.2).....	106
4.9.1.	A Medida da Capacitação do Processo (MCP).....	108

4.9.2.	O Artefato de Medição da Melhoria da Capacitação do Processo.	115
4.9.3.	O Índice de Melhoria da Capacitação do Processo (IMCP).....	117
4.10.	A Avaliação da Capacitação do Processo (Etapa 2.3).....	118
4.11.	A Identificação de Melhorias (Etapa 2.4).....	121
4.12.	A Constituição do Dossiê do Processo (Etapa 2.5).....	122
4.13.	A Determinação / Concepção do Ciclo de Melhoria do Processo de Software – CMQ-SW (Etapa 3.1).....	123
4.13.1	A Proposta Formal de Ciclo (PFC), a Guia Mestre de Ciclo (GMC) e a Memória do Programa de Melhoria da Qualidade do Processo de Software (MPQM).....	124
4.14.	O Planejamento (Etapa 3.2).....	128
4.15.	O Desenvolvimento (Etapa 3.3).....	129
4.16.	A Implantação (Etapa 3.4).....	130
4.17.	A Institucionalização (Etapa 3.5).....	130
4.18.	A Avaliação (Etapa 3.6).....	131
5.	A APLICAÇÃO DO MÉTODO.....	133
5.1.	Requisitos de Seleção da Empresa para o Estudo de Caso.....	133
5.2.	Apresentação da Empresa.....	135
5.3.	A PRIMEIRA FASE (DIAGNÓSTICO E CAPACITAÇÃO).....	138
5.3.1.	PRÁTICA - A Compreensão do Cenário através do Diagnóstico Situacional – (Etapa 1.1).	139
5.3.2.	PRÁTICA - As Ações de Capacitação – (Etapa 1.2).....	141
5.3.2.1.	A Reorganização com vistas à Experiência-Piloto.....	145
5.3.2.2.	A Determinação do Paradigma para a Base de Processos.....	146
5.4.	A SEGUNDA FASE (PREPARAÇÃO).....	146
5.4.1.	PRÁTICA - A Seleção dos Processos (Etapa 2.1).....	147
5.4.2.	PRÁTICA - A Identificação de Métricas (Etapa 2.2).....	150
5.4.3.	PRÁTICA - A Avaliação da Capacitação do Processo (Etapa 2.3).....	152
5.4.4.	PRÁTICA - A Identificação de Melhorias (Etapa 2.4).....	154
5.4.5.	PRÁTICA - A Constituição do Dossiê do Processo (Etapa 2.5).....	156
5.5.	A TERCEIRA FASE (EXECUÇÃO).....	156
5.5.1.	PRÁTICA - A Determinação / Concepção do Ciclo de Melhoria da Qualidade do Proc. de Software – CMQ-SW (Etapa 3.1)..	158
5.5.1.1.	A Proposta Formal de Ciclo (PFC), a Guia Mestre de Ciclo (GMC) e a Memória do Programa de Melhoria da Qualidade do Processo de Software (MPQM).....	159
5.5.2.	PRÁTICA - O Planejamento do CQM (Etapa 3.2).....	160
5.5.3.	PRÁTICA - O Desenvolvimento das Melhorias do Processo (Etapa 3.3).....	161
5.5.4.	PRÁTICA - A Implantação das Melhorias do Processo	

	(Etapa 3.4).....	162
5.5.5.	PRÁTICA - A Institucionalização das Melhorias (Etapa 3.5).....	164
5.5.6.	PRÁTICA - A Avaliação das Melhorias (Etapa 3.6).....	166
5.6.	RESULTADOS.....	168
6.	CONCLUSÃO.....	179
7.	POSSIBILIDADES DE TRABALHOS FUTUROS.....	186
8.	REFERÊNCIAS.....	187
	ANEXOS.....	194
	GLOSSÁRIO.....	211

LISTA DE FIGURAS

Atenção: A numeração das figuras / quadros e tabelas é indexada pelo número do capítulo seguido pela ordem seqüencial de surgimento da figura no referido capítulo.

Figura 2.1.	O Modelo Cascata. (adaptada de Boehm, 1988).....	13
Figura 2.2.	Modelo de Ciclo de Vida com Prototipação. (Fonte: Pressman, 1995).....	14
Figura 2.3.	Risco Total vs Etapas de desenvolvimento. (Fonte: adaptada Kruchten, 2000)	16
Figura 2.4.	O Modelo Espiral. (Fonte: adaptado de Pressman, 1995)	17
Figura 2.5.	Modelo Espiral Detalhado. (Fonte: Boehm, 1988).....	19
Figura 2.6.	Tendência do Prazo/custo vs Controle de Qualidade. (Fonte: CMU/SEI-93-TR-24; tradução CPqD, 2001).....	21
Figura 2.7.	A Capacitação vs Nível de Maturidade. (Fonte: CMU/SEI-93-TR-24 ; tradução CPqD, 2001)	24
Figura 2.8.	Estrutura dos Processos do Ciclo de Vida do Software (Fonte: Rocha, 2001)	27
Figura 2.9.	Inter-relacionamento dos elementos básicos do CMM. (Fonte: Paulk, 1993).	33
Figura 2.10.	A evolução da maturidade via melhoria dos processos (Côrtes, 2001 / Fig. 5-2)	35
Figura 2.11.	Componentes do SPICE (adaptado do site SPICE)	43
Figura 2.12.	A dimensão de processos. (Fonte: Côrtes, 2001, p.113/F7-4)	44
Figura 2.13.	Caracterização dos Níveis de Processos conforme o atingimento dos graus de seus Atributos. (adaptada de Côrtes, 2001, p. 24)	51
Figura 2.14.	Exemplo de avaliação do perfil de capacidade para dois processos. (adaptada de Côrtes, 2001, p. 125)	51
Figura 3.1.	Diagrama de Interação dos Procedimentos de Pesquisa.....	55

Figura 4.1.	Diagrama overview do MC2Q-SW.....	62
Figura 4.2.	Etapas 1.1 a 2.5 do MC2Q-SW.	64
Figura 4.3.	Etapas 3.1 a 3.6 do MC2Q-SW.	65
Figura 4.4.	Qualidade do Produto e Qualidade do Processo.	72
Figura 4.5.	O MC2Q-SW e a Formalização da Melhoria do Processo nos 3 Estágios evolutivos da Organização.....	77
Figura 4.6.	Evolução na qualidade de um processo.....	81
Figura 4.7.	Esquema bidimensional das “n” possibilidades de caminhos de evolução de um processo.....	81
Figura 4.8.	Os 3 Estágios evolutivos da Organização em direção à formalização e aos Modelos Maduros de SPI.....	84
Figura 4.9.	O Diagnóstico Situacional - (Etapa 1.1) e a Proposição de Ações.....	91
Figura 4.10.	Atividades para o Diagnóstico Situacional e a Proposição de Ações.....	94
Figura 4.11.	As Ações de Capacitação - (Etapa 1.2).....	95
Figura 4.12.	Duas formas representativas de Ciclos.	98
Figura 4.13.	A possibilidade de múltiplos Ciclos de Capacitação combinados a Ciclos de Melhoria da Capacitação do Processo de SW.	99
Figura 4.14.	Visão esquemática do MC2Q-SW com o primeiro CMQ-SW completo e iniciando o PLANEJAMENTO do 2o. CMQ-SW após os CCs prévios.	100
Figura 4.15.	Estabelecimento de limites mínimos de capacitação a cada competência essencial.....	100
Figura 4.16.	Estrutura embrionária recomendada às organizações que demonstram preocupação mínima com a qualidade.....	102
Figura 4.17.	Exemplo de Estrutura Orgânica voltada à qualidade em empresas maduras (http://www.cit.com.br/aaem ago.2005).....	103
Figura 4.18.	A Seleção dos Processos (Etapa 2.1).	104
Figura 4.19.	A Identificação de Métricas (Etapa 2.2).	106
Figura 4.20.	24 possíveis combinações do array DIE3 no universo de MCP.....	113
Figura 4.21.	Esquema gráfico de 4 indicadores MCPs escolhidos dentre os 24 apresentados na Figura 4.20.....	113
Figura 4.22.	Esquema gráfico de 4 combinações do array DIE3 que não se caracterizam como MCPs por infringirem as regras básicas estabelecidas.....	114
Figura 4.23.	Os Níveis de Capacidade do Processo segundo o SPICE [SALVIANO, 2005].	115
Figura 4.24.	O artefato de Medição da Melhoria da Capacitação do Processo: TAMP (Tabela de Avaliação da Melhoria do Processo).....	117
Figura 4.25.	Demonstração do IMCP a partir das curvas Pp e Pa.....	118
Figura 4.26.	A Avaliação da Capacitação do Processo (Etapa 2.3).....	120
Figura 4.27.	A Identificação de Melhorias (Etapa 2.4).....	121
Figura 4.28.	A Constituição do Dossiê do Processo (Etapa 2.5).....	122
Figura 4.29.	A Determinação / Concepção do Ciclo de Melhoria do Processo de Software – CMQ-SW (Etapa 3.1).....	123
Figura 4.30.	Esquema estrutural dos Ciclos de Melhoria da Qualidade.....	124

Figura 4.31.	Exemplo de uma GMC para um novo Ciclo de Capacitação.....	127
Figura 4.32.	O Planejamento (Etapa 3.2).....	128
Figura 4.33.	O Desenvolvimento (Etapa 3.3).....	129
Figura 4.34.	A Implantação (Etapa 3.4).....	130
Figura 4.35.	A Institucionalização (Etapa 3.5).	131
Figura 4.36.	A Avaliação (Etapa 3.6).	132
Figura 5.1.	Diagrama escalar dos requisitos determinantes da escolha da empresa....	134
Figura 5.2.	Indicadores do baixo risco de descontinuidade da STORE Automação.....	135
Figura 5.3.	Relatório (parcial) RAS.01 de Avaliação da Situação resultante de reunião tipo REI para apuração de Problemas / Necessidades.....	141
Figura 5.4.	Esquema de Avaliação de Capacitação.....	144
Figura 5.5.	GMC para o Ciclo de Capacitação piloto.	145
Figura 5.6.	Relatório (parcial) de Avaliação da Situação utilizado para a referência cruzada.	149
Figura 5.7.	A primeira avaliação de processo levada a efeito, via o indicador MCP, para a ELICITAÇÃO DE REQUISITOS (ENG.1) na STORE.....	152
Figura 5.8.	A Proposição da TAMP ao Processo ENG.1 (Elicitação de Requisitos).....	151
Figura 5.9.	Vista parcial da TAP para a avaliação INICIAL da Capacitação do Processo ENG.1 (Elicitação de Requisitos).	154
Figura 5.10.	Vista parcial da GMCP para a Elicitação de Requisitos.....	155
Figura 5.11.	Filtragem aplicada à GMCP- ENG.1 para restringir o CMQ-SW piloto.....	159
Figura 5.12.	A GMP convertida da PFC para o CMQ-SW piloto.....	160
Figura 5.13.	O PAC-001 - Plano de Ação Circular do CMQ-SW piloto.....	161
Figura 5.14.	Aplicação de um Diagrama de Caso de Uso na Elicitação de Requisitos após o aprendizado do UML.....	163
Figura 5.15.	Utilização de Fluxo Funcional na Elicitação de Requisitos.	164
Figura 5.16.	Organograma Simplificado da STORE Automação em 2005.....	165
Figura 5.17.	Enquadramento de 4 possíveis situações da STORE face às melhorias praticadas na ELICITAÇÃO DE REQUISITOS.....	166
Figura 5.18.	TAMP.01 – A Avaliação das Melhorias após o 1º. CMQ-SW na STORE....	167
Figura 5.19.	Visualização gráfica do Índice de Melhoria da Capacitação do Processo ELICITAÇÃO de REQUISITOS na STORE após a implementação do CMQ experimental.....	167
Figura 5.20.	Avaliação das melhorias do processo Elicitação de Requisitos na Equipe On-Demand.....	172
Figura 5.21.	Gráfico do IMCP para a Elicitação de Requisitos -Equipe On-Demand.....	172
Figura 5.22a.	Lista de Avaliação de Melhorias (Início) ocorridas no prazo de 14 meses em 53 Práticas da Elicitação de Requisitos na Equipe On-Demand.....	174
Figura 5.22b.	Lista de Avaliação de Melhorias (Final) ocorridas no prazo de 14 meses em 53 Práticas da Elicitação de Requisitos na Equipe On-Demand.....	175
Figura 5.23.	Gráfico do Índice de Melhoria da Capacitação do Processo Elicitação de Requisitos na Equipe On-Demand.....	176
Figura 5.24.	A avaliação da Melhoria da Capacitação do Processo ELICITAÇÃO DE	

REQUISITOS focada na equipe “On-Demand” da STORE, após 14 meses de contato com o MC2Q-SW.....	177
---	-----

LISTA DE TABELAS e QUADROS

Tabela 1.1. Apropriação de custos da qualidade. Fonte: (MCT-SEPIN, 2002, Tabela 17 da p. 42)	4
Tabela 1.2. Padrões de qualidade exigidos na seleção de terceiros prestadores de serviços de desenvolvimento e manutenção de software. Fonte: (MCT-SEPIN, 2002) Tabela 19 da p, 43).....	5
Quadro 2.1. Processos Fundamentais.....	28
Quadro 2.2. Processos de Apoio.....	28
Quadro 2.3. Processos Organizacionais.....	29
Quadro 2.4. Comparativo da maturidade das organizações.....	32
Quadro 2.5. Descrição dos Níveis do CMM.....	34
Quadro 2.6. Os Focos dos 5 níveis e suas Áreas-chave de Processo.....	36
Quadro 2.7. Os objetivos de algumas das Áreas-chave de Processos.....	37
Quadro 2.8. As 5 características comuns e suas práticas-base.....	40
Quadro 2.9. Breve Descrição das Categorias de Processos.	45
Quadro 2.10. Estrutura das categorias, dos processos de cada categoria.....	46
Quadro 2.11. Descrição dos Níveis de Capacitação do SPICE.....	48
Quadro 2.12. Níveis de Capacitação vs Atributos de Processo (PAs)	49
Quadro 4.1. Denominação e Idealização dos Artefatos ou Práticas Citadas.....	69
Quadro 4.2. Quadro de Decisão para Determinação do Processo a Submeter à Melhoria da Capacitação.....	106
Quadro 5.1. Quadro de Decisão determinante à escolha da STORE como empresa escolhida ao estudo de caso.....	134
Quadro 5.2. Dados Referenciais Básicos da STORE Automação.	137
Quadro 5.3. Quadro Simplificado de Decisão para Determinação do Processo a Submeter ao 1o. CMQ-SW.....	147
Quadro 5.4. A Guia Mestre do Processo ELICITAÇÃO DE REQUISITOS (ENG.1).....	150

SUMÁRIO dos ANEXOS

5.01 - RAS.01 - Relatório de Avaliação de Situação - Apontamentos de Reunião de Eclosão de Idéias para Apuração de Problemas ou Necessidades da Equipe de Barra Bonita da STORE.....	194
5.02 - DSPA - Diagnóstico Situacional e Proposta de Ações para a STORE.....	196
5.03 - RAS.01b - Prévia à Tabela de Decisão de Determinação de Processo - Cruzamento de Problemas ou Necessidades com os PROCESSOS da BP (restringiu-se ao ENG.1 - Elicitação de Requisitos)	200
5.04 - GMP-ENG.1 - Guia Mestre do Processo ELICITAÇÃO DE REQUISITOS (ENG.1)	202
5.05 -TAP.01 - Tabela de Avaliação (INICIAL) do Processo ELICITAÇÃO DE REQUISITOS (ENG.1)	203
5.06 - GMCP-ENG.1 - Guia da Melhoria da Capacitação do Processo ELICITAÇÃO DE REQUISITOS (ENG.1)	206
5.07 – INFORMATIVO veiculado na STORE.	209

SIGLAS

Seguem abaixo as siglas criadas por força da necessidade de documentar o Método e deve ser complementada pelo Glossário anexado ao final desta dissertação, face as demais siglas de uso corrente.

SIGLA	Denominação (Para o caso de ARTEFATOS ou PRÁTICAS a 1a. REF é a indicação da ETAPA onde o artefato ou prática é referenciado pela 1a. vez no método ora proposto.)	1a. REF
ADDI	Atividades de Desenvolvimento, Diagnóstico e Integração	1.1
BP	Base dos Processos (congrega todos os DPs)	1.2
CC	Ciclo de Capacitação	1.2
CEO	Executivo principal de uma Organização, Presidente, “ <i>Chief Executive Officer</i> ”	
CMM	Capability Maturity Model	
CMMI	Capability Maturity Model Integration	
CMQ-SW	Ciclo de Melhoria da Qualidade da Capacitação do Processo de SW (ou, simplesmente, de Melhoria da Qualidade do Processo de SW)	3.1
DP	Dossiê de Processo (Engloba tudo o que pertine ao Processo, como a GMP, o QAP, a TAP, o RAP, a GMCP, e a TAMP.)	2.5
DSPA	Diagnóstico Situacional e Proposta de Ações	1.1
ERP	Sistema Integrado de Gestão, (<i>Enterprise Resources System</i>)	
GMC	Guia Mestre de Ciclo (Pode também indicar ser uma GMC-MQ)	3.1
GMC-C	Guia Mestre de Ciclo de Capacitação	1.2
GMC-MQ	Guia Mestre de Ciclo de Melhoria da Qualidade do Processo de SW (ou, simplesmente, de Melhoria do Processo de SW), usualmente referida como apenas GMC.	3.1
GMCP	Guia de Melhoria da Capacitação do Processo	2.4
GMP	Guia Mestre de Processos	1.2
MC2Q-SW	Método Circular para a Melhoria Contínua da Qualidade Aplicado ao Processo de Software	
MPMQ	Memória do Programa de Melhoria da Qualidade	3.1
PAC	Plano de Ação Circular	3.2
PADIMC	Acrônimo cunhado pelo pesquisador usado para se referir à prática gerencial em qualquer instância, especialmente em projetos, onde é fundamental: Planejar, Acompanhar, Documentar, Institucionalizar e Medir Continuadamente.	4.5
POC ₄	Acrônimo adaptado pelo pesquisador para sintetizar a função GERENCIAR como sendo o conjunto das funções: Planejar, Organizar, Comandar, Coordenar, Controlar e Comunicar.	4.9.1
QAP	Questionário de Avaliação do Processo (ou da Capacitação do Processo)	2.3
QAS	Questionário de Avaliação da Situação	1.1
RAC	Relatório de Avaliação do Ciclo (Pode também indicar ser um RAC-MQ)	3.6
RAC-MQ	Relatório de Avaliação do Ciclo de Melhoria da Qualidade (usualmente referido como apenas RAC)	3.6
RACP	Relatório de Avaliação da Capacitação do Processo	3.6
RAP	Relatório de Avaliação do Processo (ou da Capacitação do Processo)	2.3

RAS	Relatório de Avaliação da Situação	1.1
REIs	Reuniões de Eclosão de Idéias	2.1
RFIs	Reuniões Formais de Investigação	1.1
RI	Reunião de Inspeção	3.1
RP	Relatório de Participação Individual	1.1
SEI	Software Engineering Institute	
SPICE	Software Process Improvement and Capability dEtermination,	
TAMP	Tabela de Avaliação da Melhoria do Processo (ou da Capacitação do Processo)	2.2
TAP	Tabela de Avaliação do Processo (ou da Capacitação do Processo)	2.3
TDDP	Tabela de Decisão para Determinação dos Processos a Submeter a melhorias	2.1

DEFINIÇÕES

A terminologia básica aqui utilizada é entendida como a enunciada abaixo, resultante da compilação e adaptação a partir de definições pesquisadas na bibliografia **e deve ser complementada pelo Glossário anexado ao final desta dissertação**, face a profusão de termos correntes na língua portuguesa (alguns documentados em publicações especializadas mesmo sem a correspondente incorporação do verbete em dicionário) e inglesa utilizadas na área de qualidade de software.

A seguir opta-se pela ordem de apresentação dos termos segundo seu encadeamento de raciocínio para entendimento do objeto desta pesquisa, que pode ser expresso como: “Um Método a aplicar na Melhoria da Qualidade do Processo de Software, mais precisamente da Melhoria da Capacitação do Processo de Desenvolvimento de software, através das melhorias em suas práticas e das decisões pertinentes”.

MÉTODO *é um processo organizado, lógico e sistemático de trabalhar, ou o “caminho para chegar a um fim” [AURÉLIO, 1986].*

MODELO *é uma representação ou fonte de inspiração do que se deve executar, como forma ou padrão a ser imitado por sua perfeição [HOUAISS, 2006].*

QUALIDADE: *é a totalidade dos desempenhos em função das características de um produto ou serviço que se sustenta em sua possibilidade efetiva para atender às necessidades especificadas ou implícitas. [JURAN & GRZYNA, 1991]*

Neste trabalho preferiu-se adaptar esta definição para:

QUALIDADE *é a totalidade de características de uma entidade ou de uma decisão que lhe confere a capacidade de satisfazer suas necessidades ou requisitos explícitos*

e implícitos, entendendo-se por entidade, neste trabalho, uma “Organização” (uma instituição de direito público ou privado) ou um “Produto” (tais como: sistemas, programas, artefatos) ou um serviço (tais como: processos ou práticas);

PROCESSO: é um conjunto de atividades inter-relacionadas, que transforma entradas em saídas [NBR 12207, 1999].

Neste trabalho preferiu-se adaptar esta definição para:

PROCESSO é o *Conjunto de práticas necessárias a criar ou modificar artefatos;*

PROCESSO DE SOFTWARE é o *conjunto de práticas necessárias à construção e manutenção de software, segundo se compreende a partir dos trabalhos de Watts Humphrey nas décadas de 1980 e 1990 a partir de sua experiência com a IBM e para a criação do CMM através do SEI (Software Engineering Institute);*

PRÁTICA: é um conjunto de ações ou decisões com a finalidade de atender determinados objetivos;

ARTEFATO: *elemento real, concreto, que encerra conhecimento, informação ou dado (ex: relatório, programa, sistema, manual, documento, arquivo, gravação sonora);*

CAPACITAÇÃO DO PROCESSO: é a *capacidade intrínseca de um processo reproduzir seus resultados de forma consistente, durante ciclos múltiplos de operação [JURAN, 1992];*

Adicionalmente a estrutura do método se apoia em FASES subdivididas em ETAPAS, definidas a seguir:

FASE: é o *conjunto de Etapas que permite o atingimento de uma meta importante ao método (O MC2Q-SW se constitui de 3 fases);*

ETAPA: é o *conjunto de Processos que permite o atingimento de metas intermediárias à conclusão de uma Fase (O MC2Q-SW se constitui de 13 etapas);*

1. INTRODUÇÃO

“Por muitos anos, a garantia da qualidade do software esteve atrasada em relação à garantia da qualidade do hardware, em termos de modelos, métricas e resultados bem-sucedidos. Novas abordagens como ISO 9000, CMM, TQM e QFD estão começando a elevar os níveis de qualidade do software aos mesmos patamares alcançados pela indústria em geral”

Capers Jones, *em (Weber, 2000).*

A tendência em *software* é na direção de sistemas maiores e mais complexos e deseja-se isso mais rápido, portanto o tempo de liberação do produto no mercado é um fator importante. Mas isso é muito difícil, uma vez que a maioria dos especialistas continua desenvolvendo *software* da mesma forma que fazia há 25 anos atrás. A chave para a solução a esse problema está em uma gerência de processos bem definida e efetiva. Assim Jacobson et al (1999) inicia sua apresentação em seu artigo sobre o “*Unified Process*”, e sua afirmação nos ajuda a compreender a importância da melhoria da qualidade de processos de *software*.

Este trabalho trata da melhoria da qualidade do processo de desenvolvimento de *software* e contribui com uma experiência prática de implantação de um método com esse fim em uma *softwarehouse* brasileira, aproveitável a situações equivalentes ou semelhantes.

A estrutura do presente trabalho delinea o objeto de estudo, seus contornos, metodologia e objetivos de uma maneira coerente e encadeada.

A Introdução (capítulo 1) esclarece e limita o objetivo e apresenta a justificativa à proposição deste trabalho.

O capítulo 2 é composto pela revisão bibliográfica sobre o tema, que se constitui no referencial teórico que embasou a pesquisa e, por consequência, a elaboração do método aplicado na empresa objeto de estudo. Os conteúdos abordados nesse capítulo são: os modelos de processos de *software*; a qualidade aplicada aos processos de *software*; as iniciativas de melhoria da qualidade de *software*; a norma de qualidade NBR ISO/IEC 12207; o SPICE

que incorpora todas as outras normas elaboradas até o momento e o modelo CMM para melhoria da qualidade de *software*.

Conforme pode perceber no transcurso deste trabalho, embora pesquisa tenha sido feita ao CMMI, PMBOK e *Extreme Programming*, não há necessidade de incorporar teoria a respeito, pois os princípios contidos nesses ramos de conhecimento já estão incorporados nos modelos ou na Engenharia de Sistemas aqui documentados, sendo eles, essencialmente, o necessário ao trabalho ora em apresentação.

No capítulo 3 são apresentados a metodologia e os procedimentos adotados na pesquisa e na sua aplicação em um estudo de caso, onde se optou pela pesquisa-ação, como técnica de pesquisa qualitativa.

O capítulo 4 desenvolve a parte teórica do trabalho apresentando o Método, desde o esclarecimento das definições essenciais ao objeto estudado até o detalhamento das técnicas e instrumentos utilizados.

O Capítulo 5 complementa o anterior ilustrando a teoria ali desenvolvida, através de um exemplo completo de aplicação do método em um estudo de caso na empresa escolhida.

Os resultados são apresentados ao final do capítulo 5 seguido da conclusão (capítulo 6), possibilidades de futuros trabalhos (capítulo 7) e referências bibliográficas (capítulo 8).

Encerram o trabalho, anexos constituídos por alguns dos artefatos utilizados na experimentação e por um glossário parcial de termos correntes utilizados pela comunidade de pesquisa e atuação na área de melhoria da qualidade do processo.

Anexos e glossário são apresentados ao final.

1.1. Objetivos

O presente trabalho tem por objetivo desenvolver um método para a melhoria da qualidade do processo de desenvolvimento de *software*, com base nos métodos, modelos e normas utilizados internacionalmente para esse fim.

Propõe-se a revisar os conceitos e métodos, compilando o que for essencial e aproveitando as melhores práticas neles identificadas que propiciam não somente a melhoria da qualidade, como também o aumento da produtividade na criação de *software*.

O método foi idealizado originalmente com a intenção de solucionar um problema: o da pouca significância de ações para a melhoria da qualidade do processo de *software* nessa indústria.

Relativamente a custos, a intenção original também foi permitir propiciar a implementação de Programas de Melhoria da Qualidade do Processo de *Software* independentemente dos altos custos associados aos modelos existentes. No entanto, como se demonstrará no transcorrer deste trabalho e ressaltado na conclusão, o custo total da qualidade do processo de *software* é uma questão independente dos métodos e modelos, mas que decorre, sim, da maturidade da Organização e da intervenção cultural e gerencial (organizacional) necessárias para que as organizações se tornem eficientes nessa prática.

Além de estabelecer um método dirigido à indústria de *software*, propõe-se submeter o método a uma situação concreta de melhoria de um processo, permitindo uma primeira avaliação experimental, através de uma experiência piloto em uma empresa brasileira de *software* de pequeno porte.

1.2. Justificativas da Pesquisa

Desde seu início em 1990, o Programa Brasileiro da Qualidade e Produtividade (PBQP) e o Subcomitê Setorial da Qualidade e Produtividade em *Software* (PBQP/SSQP-SW), criado ao 1º. de junho de 1993, objetivam induzir ações de modernização industrial e tecnológica. Uma de suas estratégias é o desenvolvimento e a difusão de modelos modernos de gestão empresarial para a melhoria da qualidade de produtividade, estimulando assim, como um agente mobilizador, as iniciativas nesse sentido, mesmo quando originadas por

entidades ou empresas (Weber, 2000). Isto demonstra a sensibilidade, não só do governo, como também de diversas entidades nacionais e empresas vinculadas à produção e à utilização de *software* à relevância do tema qualidade e produtividade em *software*.

É, inclusive, um objetivo claro do PBQP/SSQP-SW buscar a obtenção de padrões internacionais de qualidade de produtividade nesse setor (Weber, 2000).

Os resultados da pesquisa bial de Secretaria de Política de Informática do Ministério da Ciência e Tecnologia (MCT-SEPIN, 2002) para acompanhamento da evolução da qualidade nas empresas de *software* atestam que, apesar do reconhecimento das empresas de que investimentos em qualidade resultam em ganhos, mais da metade (56,4%) das empresas não os apropriam. Adicionalmente, apenas 8,2% das empresas o fazem sistematicamente, conforme ilustra a tabela abaixo.

Categorias	Nº de organizações	%
Sistemática	36	8,2
Em projetos específicos	59	13,4
Em estudo ou implantação	97	22,0
Não apropriada	248	56,4
Base	440	100

Tabela 1.1. Apropriação de custos da qualidade. Fonte: (MCT-SEPIN, 2002, Tabela 17 da p. 42)

Do conjunto amostral de 446 organizações pesquisadas, a grande maioria desenvolvedoras de *software*, constata-se que 62% terceirizam, sistemática ou esporadicamente, serviços de desenvolvimento e manutenção de *software*, e, dentre estas 279 entidades, 72% não exigem padrões de qualidade na seleção de seus fornecedores desses serviços (veja tabela a seguir):

Categorias			Nº de organizações	%
ISO 9000			43	15,4
CMM			5	1,8
Avaliação próprias	segundo	normas	29	10,4
Outros			5	1,8
Não exige			201	72,0
Base			279	100

Tabela 1.2. Padrões de qualidade exigidos na seleção de terceiros prestadores de serviços de desenvolvimento e manutenção de software. Fonte: (MCT-SEPIN, 2002) Tabela 19 da p, 43)

Portanto, assumindo ser representativa essa amostra do universo real de empresas brasileiras de *software*, onde há reconhecimento dos benefícios da qualidade de *software* em contraposição ao significativo número de empresas que ainda não adotam, sistemática e globalmente, programas de qualidade em *software*, é plausível inferir que ocorrerá um crescente interesse por melhoria da qualidade nesse ramo de tecnologia. E, também um possível interesse em conhecer casos de aplicação de métodos e modelos de qualidade de *software* à realidade específica de determinados tipos de empresas nacionais.

Além disso, diversas abordagens de melhoria da qualidade em *software*, a exemplo de duas reconhecidamente abrangentes, o CMM e o SPICE, estabelecem orientações sobre como proceder para se atingir patamares ou níveis de qualidade em *software*, partindo de pressupostos ou constatações da real situação das empresas. Por exemplo, o CMM, originalmente concebido para o desenvolvimento de grandes projetos para o Departamento Americano de Defesa, se aplicado a projetos menores e em outras áreas exige um trabalho cuidadoso de interpretação e adequação à realidade da organização, como observado por Weber (2000) citando Humphrey (1989), Tingey (1997) e Fiorini (1998).

A já citada pesquisa do MCT-SEPIN (ver tabela anterior) indica ainda que: dos 28% das *softwarehouses* que contratam desenvolvimento, apenas 1,8% terceirizam o desenvolvimento com base no CMM; apenas 15,4% com

base na ISO 9000; e, também, apenas 10,4% o fazem segundo normas próprias ou outros critérios.

Ou seja, apesar da existência de modelos reconhecidamente apropriados à melhoria da qualidade de *software*, ainda é muito baixa a adesão a eles por parte das empresas geradoras de *software*.

Baskerville e Heje (1999) expandem essa constatação citando Yourdon (1998) que afirma que a indústria de *software* (mundial) estima que as organizações do setor classificadas no nível mais baixo de maturidade, definidas como nível 1 do CMM, o nível de desenvolvimento de *software* caótico ou *ad hoc*, equivale a cerca de 75% de todas as empresas desse segmento.

A determinação das causas, possivelmente diversas, dessa discrepância no mercado brasileiro, – baixa adesão ao CMM e à ISO 9000 (ao menos na terceirização) quando se constata o reconhecimento da importância dos investimentos em qualidade -, exigiria pesquisa específica, mas pode ser aventada a possibilidade de que aspectos culturais e de escassez de recursos componham o conjunto de causas.

Parece-nos inevitável que, decorrente desse cenário e da natural competição entre as empresas concorrendo por ampliar sua participação nos mercados em que já atuem ou na conquista de novos mercados, torne-se crescente o interesse e a busca por modelos e programas de melhoria da qualidade de *software* praticamente por uma questão de sobrevivência empresarial.

Segundo Watts S. Humphrey, mentor do CMM, em citação de Weber (2000 – p. v),

“para que a indústria de software contribua, de forma construtiva, para a sociedade, precisamos aprender a entregar produtos com qualidade, no prazo estabelecido e com custos planejados. Isto não é impossível. Outras indústrias, à medida que amadureceram, atingiram esse nível de desempenho. Não há razão para que isso não seja possível para software. As questões-chave são: (i) qual é a atual cultura de software?; (ii) quais são os problemas causados por esta cultura?; (iii) como deve ser a nova cultura?; e (iv) como podemos chegar lá?”

Assim, a avaliação da situação de cada empresa confrontada com as abordagens e normas de qualidade de *software* deve, provavelmente, induzi-las a escolher ou elaborar ou adaptar normas com base nas melhores práticas já identificadas gerando modelos que possibilitem a melhoria da qualidade de sua produção de *software*.

Mesmo empresas consideradas avançadas na aplicação de modelos de qualidade de *software* usualmente atuam dessa forma, a exemplo da Infosys Technologies Ltd,¹ (Jalote, 2000).

Pressupõe-se que as grandes empresas, por sua notória capacidade financeira e econômica, tenham melhores condições de adotar modelos, metodologias e ferramentas de apoio a uma gestão eficaz, bem como contratar consultorias e demais serviços de capacitação técnica e metodológica no mercado nacional e internacional. No entanto, muitas das pequenas e médias empresas brasileiras cujo fim é a produção de *software* certamente não contam com as facilidades das grandes empresas de avaliar o disponível e adotar tudo o necessário à sua melhoria de produtividade e qualidade. Observando-se que, muitas ainda em seu nascedouro e diversas outras em fase de maturação, têm de enfrentar um mercado globalizado hoje bastante competitivo pela crescente intervenção de competidores internacionais,

Além disso, as *softwarehouses* nacionais que ousarem exportar seus produtos podem encontrar barreiras de certificação da qualidade, exigidas no exterior, que as impelem a adotar programas de melhoria da qualidade capacitando-as à obtenção dos selos de qualidade exigidos internacionalmente.

Nesse contexto os modelos CMM e SPICE parecem ser excelentes paradigmas a serem utilizados por organizações envolvidas em planejar, gerenciar, monitorar, controlar e melhorar as atividades de aquisição, fornecimento, desenvolvimento, operação, evolução e suporte de *software*.

¹ A Infosys Technologies uma *softwarehouse* considerada de grande porte, estabelecida em Bangalore, Índia, e avaliada como CMM nível 4 por 2 representantes do SEI, partiu para a construção de um **modelo próprio** na busca de produzir software de alta qualidade e de aumentar sua produtividade (Jalote, 2000).

É oportuno, porém, ressaltar a crítica que alguns especialistas fazem ao CMM, especialmente a explícita por Baskerville e Heje (1999) de que *“induz a uma rígida burocracia e sufoca a criatividade e a inovação desmoralizando a equipe”* e, citando Bach, prossegue: *“Desenvolvedores notórios de software, competitivos e inovadores, exemplificados por Borland, Claris, Apple, Symantec, Microsoft e Lotus não se lançaram ao CMM junto com os pioneiros”*, reforçando a citação original:

“Now the SEI argues that innovation is outside of its scope and that the CMM merely establishes a framework within which innovation may more freely occur ... nothing could be further from the truth. Preoccupied with predictability the CMM is profoundly ignorant of the dynamics of innovation ... Because the CMM is distrustful of personal contributions, ignorant of the conditions needed to nurture nonlinear ideas, and content to bury them beneath a constraining superstructure achieving level 2 on the CMM scale may very well stamp out the only flame that lit the company to begin with.”
(Bach, 1994b, pp. 16-17)”

Além de tudo, por sua grande abrangência, os modelos citados têm de ser adaptados à realidade e particularidades das organizações que se interessem pela melhoria da qualidade de seus processos, a exemplo das pequenas e médias empresas brasileiras produtoras de *software*, foco de aplicação deste trabalho.

Nesse sentido, inspirando-se no próprio processo de criação do CMM, citado por Côrtes (2001), o esforço de compilação das práticas preconizadas nos modelos já citados e de sua adaptação para um adequado aproveitamento da empresa objeto de estudo desta dissertação pressupõe, além de observar os paradigmas de sucesso na engenharia de *software*:

- a) considerar a experiência prática dessa empresa;
- b) refletir o melhor estado da prática;
- c) atender as necessidades dos agentes que praticam a engenharia de *software*;
- d) ficar documentado e disponível à comunidade tanto acadêmica quanto empresarial.

As iniciativas brasileiras e a interação da comunidade de informática no País demonstram um vivo e crescente interesse pela produtividade e qualidade de *software* como demonstrado nos diversos quadros estatísticos e análises da

publicação do MCT-SEPIN, Qualidade e Produtividade no Setor de *Software* Brasileiro referente a 2002/2001.

Contribui a esta assertiva a declaração de Vanda Scartezini, Secretária de Política de Informática (<http://www.mct.gov.br.sepin>, dez.2002) que demonstra o funcionamento duradouro dos grupos de projetos e programas ligados à qualidade e produtividade de *software*, as inúmeras iniciativas em instituições de ensino técnico superior, temas e inscrições crescentes em congressos, simpósios e seminários, atendimento a chamadas de projetos, dentre outras iniciativas.

E tudo indica que esse vivo e crescente interesse é decorrente das oportunidades de negócio que impelem a comunidade nacional de informática à promoção de ações, tanto profissionais como acadêmicas, no sentido de impulsionar o conhecimento e a experimentação como moto propulsor dos avanços exigidos à qualidade e produtividade.

Também é de Scartezini declaração nesse sentido a respeito do crescente interesse por rodadas de negócios em congressos; do estabelecimento de linhas de financiamento específicas para empresas de *software* criadas junto ao FINEP e BNDES; das iniciativas de missões ao exterior; da evolução da promoção comercial do *software* brasileiro no exterior; da taxa média anual de crescimento da receita de 19% sobre os valores correntes; e da melhoria do desempenho do setor de *software* na década de 90, uma atividade que, em 2001, se projetava girar cerca de US\$8 bilhões no mercado nacional, onde US\$6,9 bilhões alimentariam a economia interna brasileira.

2. REFERENCIAL TEÓRICO

O desenvolvimento ou a adaptação de modelos de melhoria da qualidade de *software*, e especificamente o trabalho proposto ao processo de desenvolvimento, pressupõe um domínio bastante amplo dos modelos de ciclo de vida de sistemas e das melhores práticas da engenharia de *software*.

Acredita-se ser recomendável também o domínio de técnicas de gestão e desejável o conhecimento de modelos e normas direcionadas a esse objetivo, assim procurou-se efetuar um levantamento bibliográfico que desse respaldo a esse conhecimento que ora passa a ser exposto.

2.1. Os Modelos dos Processos de *Software*

São diversos os modelos aplicáveis ao processo de *software* (conjunto de práticas que levam à criação de *software*), cada qual enfatizando algum aspecto essencial ou mesmo orientando a utilização de algum modelo. Por exemplo, o Modelo Orientado a Objetos, aplicável quando se pretende utilizar a notação de objetos aos projetos, ou do Modelo de Processo de Sistema Embutido, direcionado à construção de *software* incorporado a mecanismos ou equipamentos, dentre vários outros, amplamente expostos por Peters e Pedrycz (2001).

Qualquer que seja, o modelo exige disciplina, dedicação, envolvimento, e conhecimentos técnicos específicos, por parte da equipe que dele se utiliza, para que se absorvam, de forma eficaz, os conhecimentos funcionais específicos da(s) rotina(s) ou sistema(s) objeto de construção do *software*, e se atinjam seus objetivos com qualidade, sendo que os atributos da liderança e da equipe designada ao projeto são os reais responsáveis pelo sucesso ou fracasso dos projetos .

Discorrer-se-á apenas sobre os modelos básicos e sua evolução, uma vez que todos esses modelos orientam as práticas que serão utilizadas no decorrer deste trabalho.

2.1.1. O modelo Cascata (*Waterfall*)

O modelo Cascata (*Waterfall*) ou modelo linear de desenvolvimento baseia-se num desenvolvimento em etapas sucessivas que incluem atividades de verificação, para averiguar se o processo está sendo feito conforme as boas práticas, e de validação, para propiciar maior segurança em relação à eficácia e de controle de qualidade.

As etapas são usualmente denominadas de:

- **Estudo de Viabilidade:** definição preliminar do escopo do sistema, restrições e conceitos alternativos;
- **Análise:** especificação funcional do sistema (Projeto Lógico ou Anteprojeto);
- **Projeto:** especificação completa da arquitetura de hardware e *software*, estruturas de controle, estruturas de dados do sistema, interfaces;
- **Implementação:** codificação e teste individual dos programas;
- **Teste:** teste integrado dos componentes do sistema;
- **Implantação:** disponibilização de todas as rotinas ou processos em funcionamento;
- **Operação e Manutenção:** utilização do sistema e modificações decorrentes de erros, mudança de necessidades, etc.

A documentação é uma atividade que se faz presente em todas as etapas e deve ser objeto de inspeção metódica por ocasião das avaliações ao final de cada etapa. Como em qualquer um dos modelos a adotar-se, requer forte acompanhamento da gerência para evitar que as desculpas usualmente

adotadas pelas equipes de desenvolvimento permitam interferir negativamente na qualidade do processo de *software* (Fayad, 1997).

A maior crítica a este modelo é relativa às etapas de teste e implantação, que, por se caracterizarem, originalmente, como etapas estanques após um período relativamente longo de desenvolvimento, costuma evidenciar conflitos decorrentes de especificação de requisitos insatisfatória, como o do alto custo de correção das especificações após os usuários tomarem contato com o projeto final.

Na tentativa de corrigir esse problema diversas melhorias foram sendo incorporadas ao modelo, que, no decorrer dos anos que se seguiram, travestiram-no em outros modelos, como por exemplo, o da prototipação, visto mais à frente.

Nessa linha Lott (1997) ilustra muito bem afirmando que *“many readers will already have recognized the waterfall life-cycle model hiding behind a new coat of silicon paint. But things have changed – the old waterfall model has accumulated a series of incremental improvements without anyone really noticing the changes.”*

Boehm (1988) ilustra o modelo, de forma mais completa, como mostra a figura 2.1.

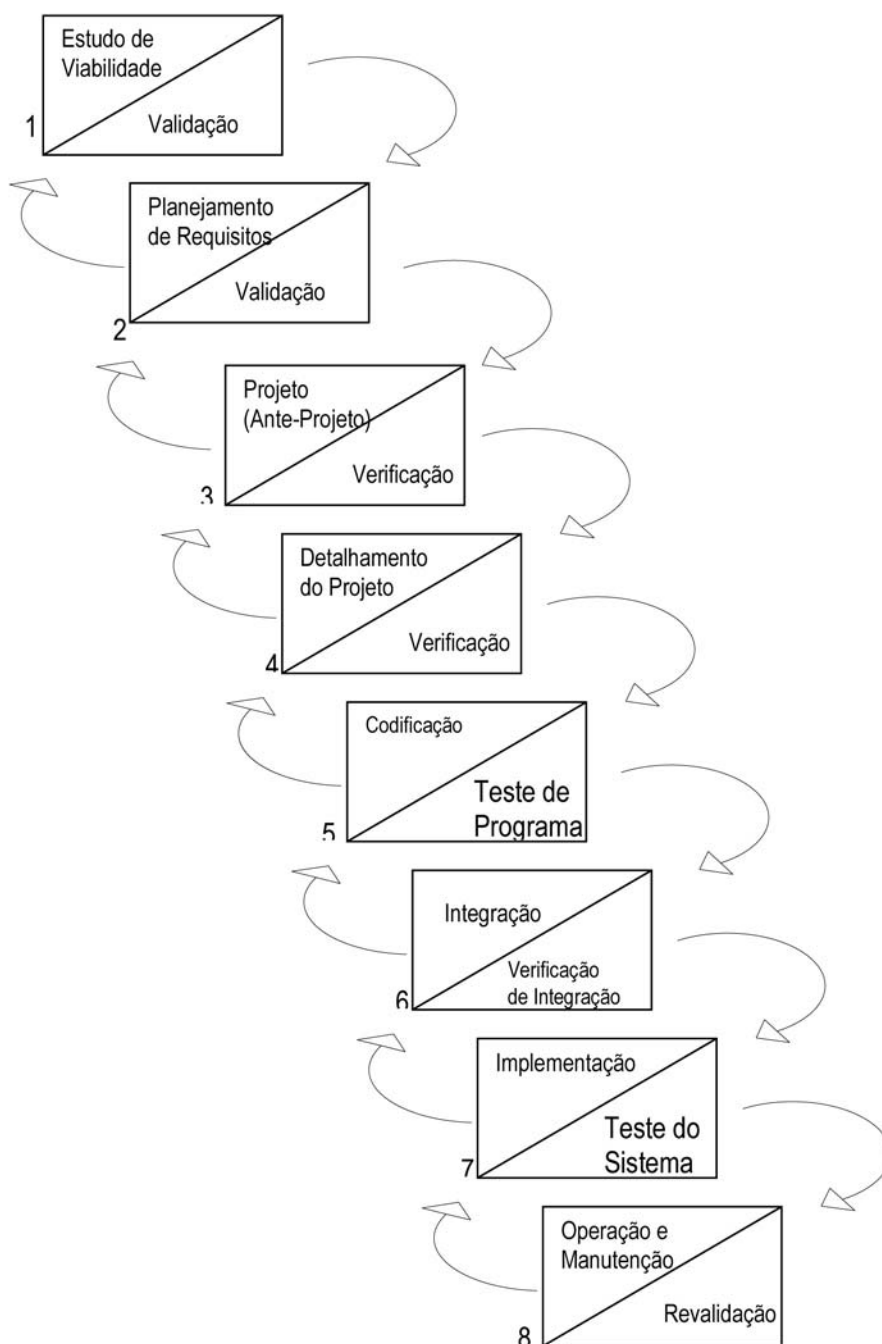


Figura 2.1. O Modelo Cascata. (adaptada de Boehm, 1988).

2.1.2. O Modelo de Ciclo de Vida com Prototipação

O Modelo de ciclo de vida com prototipação se baseia na utilização de um protótipo do sistema real, para auxiliar na determinação de requisitos, uma vez

que permite antecipar aos beneficiários do sistema uma visão aproximada de como funcionará o sistema ao findar seu desenvolvimento.

A geração de protótipos deve lançar mão de ferramentas que permitam essa funcionalidade a baixo custo e de forma rápida para que possa produzir o benefício gradativo de desenvolver parte(s) do sistema com o mínimo de investimento mas sem perder suas características básicas, e com a frequência necessária a possibilitar avaliações junto aos usuários do *software*, que permitirão correções e ajustes mais oportunos à medida que o projeto vai tomando corpo.

Com base nessa interação com os usuários as adaptações, correções ou mudanças durante o processo de desenvolvimento fluem com mais clareza do que se não houvesse a possibilidade de simulação que o protótipo permite. Tais procedimentos são executados de forma a aproximar o protótipo àquilo que o usuário realmente precisa evitando os já citados problemas de mudanças radicais após a conclusão do *software*.

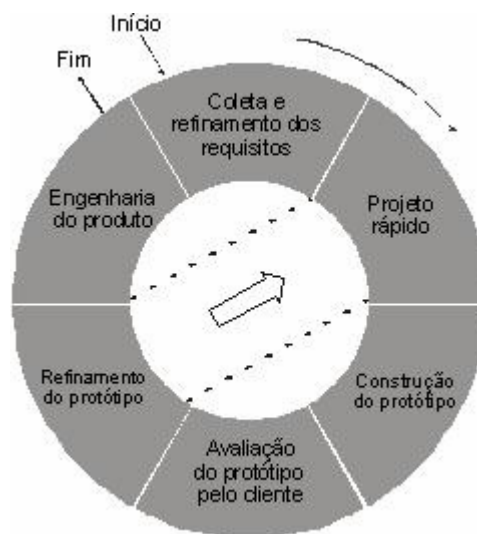


Figura 2.2. Modelo de Ciclo de Vida com Prototipação. (Fonte: Pressman, 1995)

2.1.3. O Modelo Espiral

O modelo espiral, proposto por Boehm em 1988, é representado basicamente por uma espiral de ciclos incrementais onde a cada ciclo precede uma análise de risco.

Procura abranger as melhores características de modelos anteriores, como a prototipação, acrescentando ao mesmo tempo, um novo elemento - a análise dos riscos - que falta nos modelos anteriores, apresentando-se assim como uma abordagem mais adequada ao desenvolvimento de *software* e sistemas em grande escala.

Baseia-se em um conceito de cadeia evolutiva, onde tanto os engenheiros de sistemas, como o cliente, seguem, gradativamente, um roteiro de atividades entendendo e reagindo aos riscos identificados a cada etapa evolutiva delineada na espiral, sendo que, ao final de cada ciclo pode-se gerar um produto que é uma versão de protótipo para avaliação.

Com base na experiência adquirida na criação da versão do protótipo resultante de cada ciclo da espiral, estabelecem-se novos requisitos para o sistema e uma nova versão é concebida e implementada. Assim, os protótipos gerados no ciclo de vida espiral cumprem o papel de favorecer a comunicação entre clientes e analistas/projetistas e de encurtar o ciclo "concepção-implementação-utilização-avaliação" do sistema; em uma constante avaliação dos processos de desenvolvimento e do sistema em si.

O modelo por si só, não garante a eliminação dos riscos, pois isso depende da habilidade dos envolvidos no processo, no entanto direciona a uma conduta que favoreça a detecção e avaliação, e, se for utilizado conjuntamente a técnicas específicas de avaliação de risco, como a de Kepner e Tregoe (1997), torna-se uma ferramenta bastante útil.

Os riscos são entendidos como circunstâncias adversas que podem surgir durante o desenvolvimento de *software* impedindo ou abortando o processo ou diminuindo a qualidade do produto, por exemplo, ferramentas incompatíveis, técnicas inadequadas a grandes volumes de processamento, perda de técnicos essenciais à equipe, falhas em equipamentos, mudança de objetivos ou metas, etc.

Nos modelos anteriores a característica de linearidade do processo de desenvolvimento de *software* implica em empurrar o risco para frente, e como bem ilustra Kruchten, (2000) na figura abaixo, de forma cumulativa.

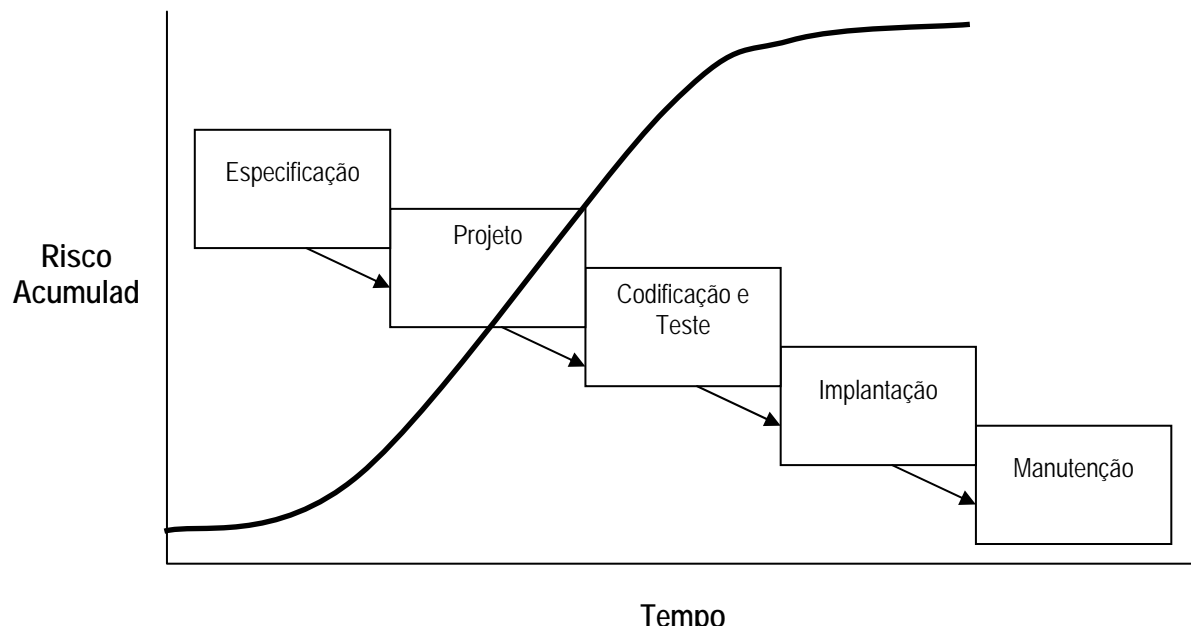


Figura 2.3. Risco Total vs Etapas de desenvolvimento. (Fonte: adaptada Kruchten, 2000)

Cada ciclo prevê 4 estágios, demonstrados no esboço a seguir, que vão se repetindo até que um produto seja obtido.

- Planejamento;
- Análise de Risco;
- Engenharia;
- Avaliação pelo Cliente.

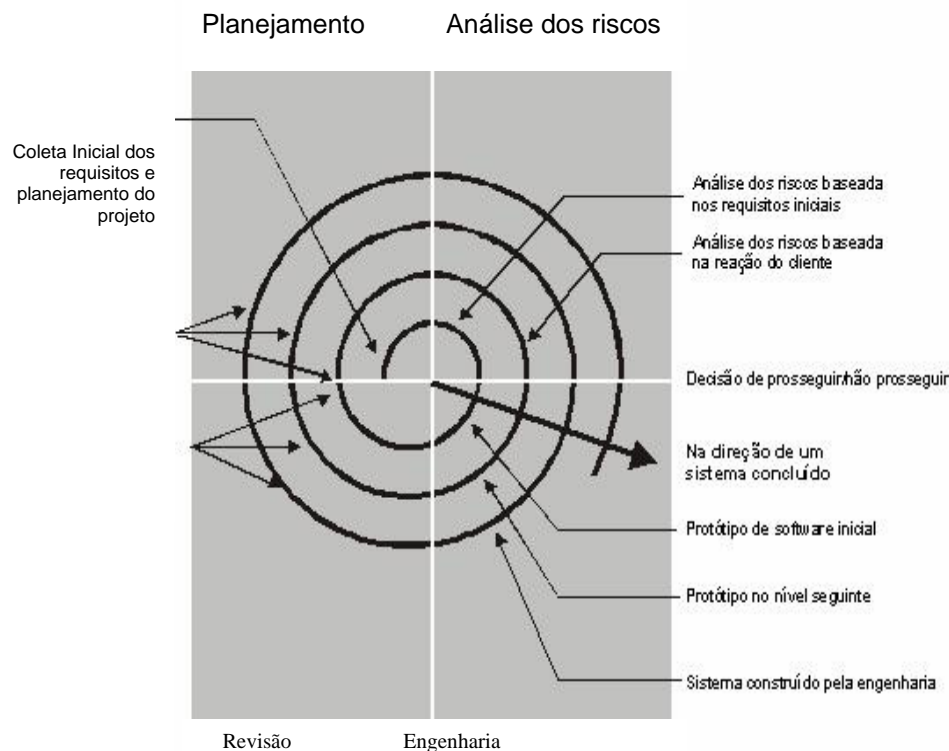


Figura 2.4. O Modelo Espiral. (Fonte: adaptado de Pressman, 1995)

Os estágios são descritos na seqüência, conforme seus processos principais:

- **Estágio 1 - Planejamento:** etapa em que se determinam os objetivos do sistema que será desenvolvido e se identificam as restrições impostas à aplicação, tais como: desempenho, funcionalidade, capacidade de acomodar mudanças, meios alternativos de implementação. Nesse estágio obtém-se o comprometimento dos envolvidos, suas atribuições e metas com o estabelecimento de planos para alcançá-las.
- **Estágio 2 - Análise de Risco:** etapa em que se analisam alternativas e se identificam e avaliam os riscos decorrentes das decisões do ciclo anterior, estabelecendo-se as medidas corretivas e de solução aos problemas potenciais passíveis de ocorrer. Nesse estágio podem ser construídos protótipos ou realizadas simulações do *software*. Após completo cada ciclo ao redor da espiral, e após o planejamento revisto, repete-se a análise de risco que resulta na decisão de

continuidade. Se os riscos forem muito altos, o projeto pode ser encerrado ou ter seu escopo reduzido ou alterado. Se a análise de risco indicar que existem dúvidas nos requisitos, a prototipação pode ser estendida ao quadrante de engenharia para assistir tanto o desenvolvedor como o cliente. Simulações e outros métodos podem ser usados depois para definir melhor o problema e refinar os requisitos.

- Estágio 3 - **Engenharia**: nesta etapa ocorre o desenvolvimento do produto. Uma vez que o modelo prevê retornos sucessivos a este quadrante pode-se considerar que, cada vez que se inicia nele um novo ciclo, pode-se aplicar o modelo cascata para o desenvolvimento. Ou seja, com base no protótipo e especificações gerados no estágio anterior produz-se uma versão mais próxima do sistema final, que agrega as funções pertinentes em um processo de contínua evolução. Como preconizado no modelo cascata, o estágio de Engenharia passa por fases de projeto, especificação detalhada, codificação e testes;
- O estágio 4 - **Revisão**: Nesta etapa o *software* liberado na etapa anterior de engenharia é avaliado e se prepara o início de um novo ciclo. O cliente faz sugestões para modificações que serão consideradas na próxima fase do planejamento e análise de risco.

O diagrama a seguir apresentado por Boehm (1988) e traduzido por Peters e Pedrycz (2001) incorpora mais detalhes ao modelo, dando uma idéia bastante abrangente do mesmo.

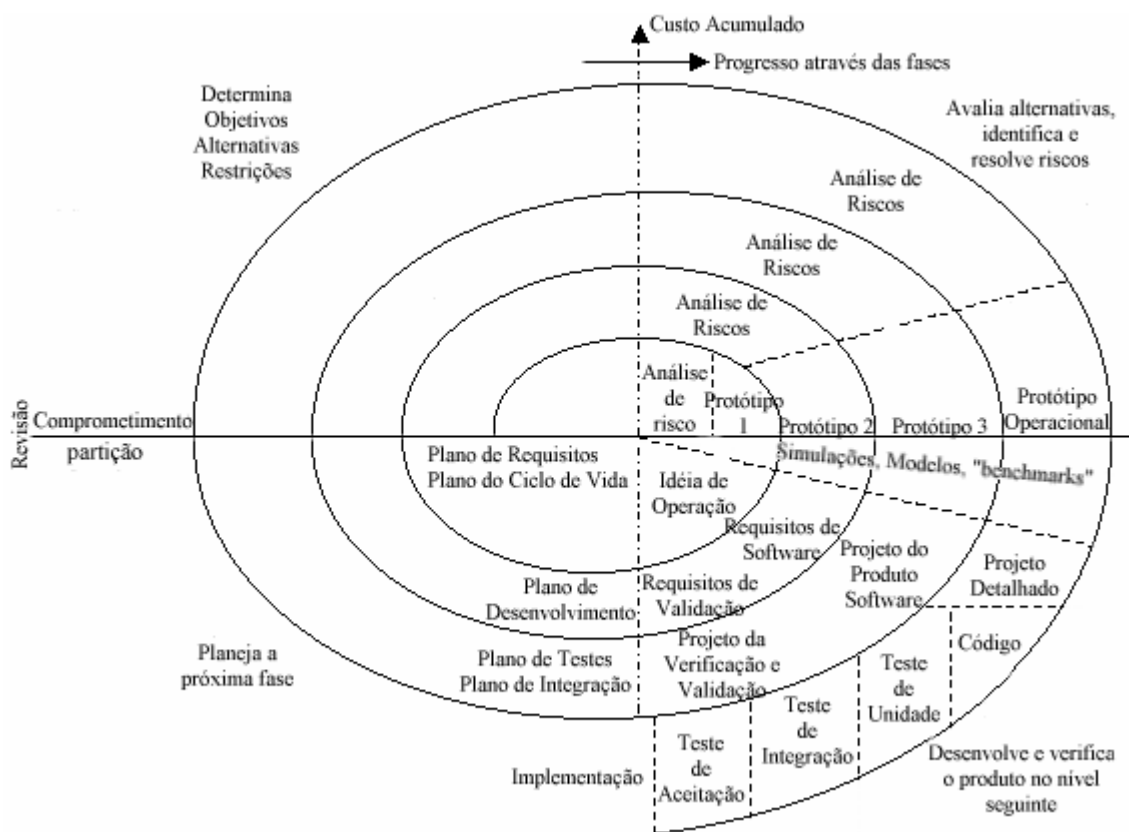


Figura 2.5. Modelo Espiral Detalhado. (Fonte: Boehm, 1988).

Pelo aqui demonstrado o modelo espiral para engenharia de *software* é um modelo bastante completo e realista, adequado para o desenvolvimento de sistemas complexos.

2.1.4. Considerações finais a respeito de modelos

Os modelos servem de guia para manter projetos de desenvolvimento de *software* sob um controle que permita assegurar qualidade e produtividade, neutralizando ou minimizando riscos, e mantendo-os em prazos e custos compatíveis com o previsto.

Assim, não devem ser entendidos como uma camisa de força, mas sim como um instrumento gerencial flexível e adaptável às circunstâncias de cada novo desenvolvimento, podendo agregar técnicas de qualquer um dos demais

modelos desde que elas somem eficazmente na direção da qualidade, factibilidade e produtividade.

Nesse sentido, em uma organização que tem por paradigma o modelo espiral para seu desenvolvimento de projetos, não seria impeditiva a aplicação de outras abordagens desde que se entendam as razões técnicas e se considerem os aspectos outros relevantes à situação específica, tais como capacidade gerencial, padronização, organização e demais detalhes funcionais.

Como um exemplo a essa consideração, tomando-se o modelo cascata, por ser considerado dos mais antigos e, por isso, corriqueiramente, tido como obsoleto, poder-se-ia perfeitamente aproveitá-lo em situações onde seja desejável a contratação de projetos por etapas, como defendido por Lott (1997), em seu trabalho de revisão do modelo cascata, que identifica diversos aspectos positivos no modelo como ainda sua integração a outras técnicas, como a prototipação.

Nesse trabalho, fica claro que as empresas Bellcore, Cambridge Technology Partners, Keane and Sapient creditam muito de seu sucesso na adoção de desenvolvimento de aplicativos *RAD (Rapid Application Development)* à utilização do modelo cascata combinado às novas tecnologias de apoio à prototipação.

2.2. Qualidade aplicada aos Processos de Software

O Processo Estatístico é a base para a melhoria da qualidade e, conforme demonstrado por Juran (1988) na figura a seguir, estabelece os fundamentos de qualquer programa nesse sentido, onde, a partir de métricas definidas, e da coleta sistemática de dados a respeito dos processos, pode-se melhorar sua performance, diminuindo-se o custo da baixa qualidade.

Conforme expõe o relatório CMU/SEI-93-TR-24 (tradução CPqD, 2001) Juran (1988) divide a gestão da qualidade em três processos gerenciais básicos que, aplicados à engenharia de *software*, propiciariam aos produtores

de *software* condições de desenvolver produtos de forma mais eficaz, ou seja, com qualidade e produtividade.

Juran parte do raciocínio que as empresas, ao desenvolverem seus produtos, incorrem em erros decorrentes de deficiências de qualidade, que geram retrabalho. O alto nível de custo associado ao retrabalho é crônico porque o processo é mal planejado, assim, o controle de qualidade deve ser realizado para identificar a freqüência de falhas apontando as possibilidades de melhoria.

Os picos esporádicos dentro do processo, como mostrados na Figura a seguir, representam as atividades conhecidas como “apagar incêndios”. O custo crônico indica a oportunidade de melhoria.

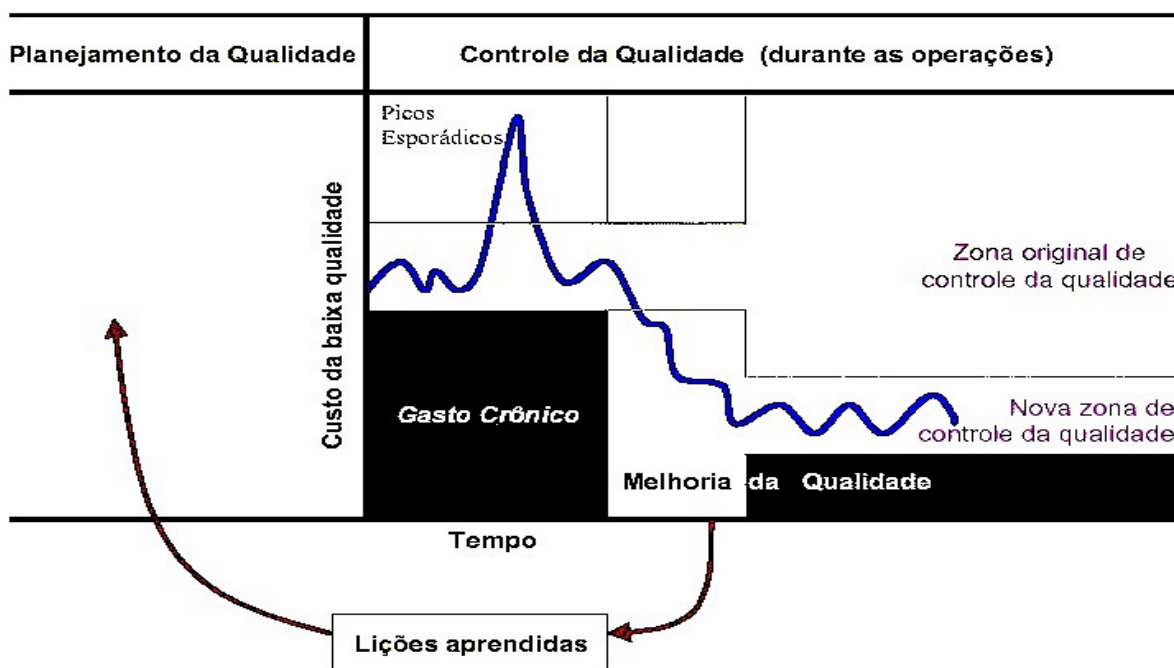


Figura 2.6. Tendência do Prazo/custo vs Controle de Qualidade. (Fonte: CMU/SEI-93-TR-24; tradução CPqD, 2001)

Assim, ainda sugerido pelo relatório CMU/SEI-93-TR-24 (tradução CPqD, 2001) uma abordagem de melhoria da qualidade deveria ter como primeira meta o controle de processo de *software*, que, passando a ser gerenciado, passaria a ser executado de forma estável dentro da zona de controle de qualidade. Pode haver algum custo crônico e pode haver picos nos

resultados medidos que precisam ser controlados, mas o sistema como um todo geralmente é estável. Esse é o ponto onde o conceito de controle de causas especiais de desvios se aplica. Uma vez que o processo é estável e medido, quando ocorre alguma circunstância excepcional, a “causa especial” do desvio pode ser identificada e tratada.

A segunda meta é a melhoria contínua do processo. O processo de *software* é alterado para melhorar a qualidade e, em conseqüência, a zona de controle de qualidade se move. Um novo padrão de referência para desempenho é estabelecido, reduzindo o custo crônico. Conhecimentos adquiridos com a melhoria de processos são aplicados no planejamento de processos futuros. Este é o ponto onde o conceito de tratamento de causas comuns de desvios se apresenta. Existe custo crônico em qualquer sistema, na forma de retrabalho, simplesmente devido aos desvios aleatórios. Custo adicional é inaceitável; os esforços organizados para eliminá-los resultam na alteração do sistema, isto é, na melhoria do processo através de alteração das “causas comuns” de ineficiência, objetivando prevenir a ocorrência de custos imprevistos.

Dessa forma, complementa o relatório CMU/SEI-93-TR-24 (tradução CPqD, 2001), seria previsível que as organizações mais avançadas em seu processo de controle de qualidade apresentassem um processo capaz de produzir *software* extremamente confiável dentro de limites de custo e de cronograma previsíveis.

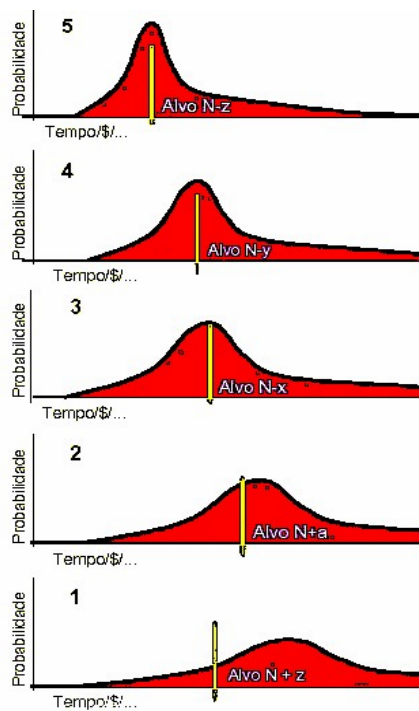
Mais à frente ao discorrer-se sobre o modelo (ou *framework*) CMM, caracterizar-se-á a maturidade das empresas em sua evolução na melhoria da qualidade de *software*, no entanto, é oportuno introduzir a idéia de maturidade das organizações em relação aos seus processos de *software* para que se visualize a tipificação do cumprimento de prazos e orçamentos.

O referido relatório CMU/SEI-93-TR-24 pressupõe que a maturidade do processo de *software* de uma organização ajuda a prever sua habilidade em atingir as suas metas de projetos de *software*. Para isso estabelece 5 níveis, desde o 1 estabelecido para a empresa imatura até o 5, em escala unitária, aplicado à empresa no auge da maturidade. Os projetos nas organizações de

Nível 1 apresentam grandes desvios em relação às metas de custo, cronograma, funcionalidade e qualidade. Como pode ser visto na Figura 2.7, três melhorias relacionadas ao alcance de metas são observadas à medida que o processo de *software* da organização vai se tornando mais maduro.

Primeiro, à medida que a maturidade cresce, a diferença entre os resultados esperados e os resultados reais tenderia a diminuir nos projetos. Por exemplo, se dez projetos de mesmo tamanho fossem programados para serem entregues na data “x”, então a data média de cada entrega estaria mais próxima de “x”, tanto quanto mais madura fosse a organização. As organizações de Nível 1 tenderiam a atrasar freqüentemente suas datas de entrega originalmente previstas, enquanto que as organizações de Nível 5 seriam capazes de prever essas datas com considerável precisão. Isso aconteceria porque as organizações de Nível 5 utilizariam um processo operacional de *software* cuidadosamente elaborado dentro de parâmetros conhecidos, sendo a escolha da data de entrega baseada no grande volume de dados que possuem sobre seus processos e em suas habilidades de tirar proveito do mesmo. (Isso é ilustrado na Figura 2.7 através da parte da área sob a curva que se encontra à direita da data de entrega.)

Segundo, à medida que a maturidade cresce, a variação dos resultados reais em torno dos resultados esperados tenderia a diminuir. Por exemplo, as datas de entrega nas organizações de Nível 1, em projetos de tamanhos similares, seriam imprevisíveis e muito variáveis. Projetos similares em organizações de Nível 5, entretanto, seriam entregues dentro de um intervalo de tempo muito menor. Isso ocorreria nos níveis de maturidade mais elevados porque, na prática, todos os projetos seriam executados dentro de parâmetros controlados, aproximando-se da Capacitação do processo para custo, cronograma, funcionalidade e qualidade. (Isso é ilustrado na Figura 7 através da parte da área sob a curva concentrada próximo à linha alvo).



O desempenho tenderia a melhorar continuamente nas organizações de Nível 5.

Com base no entendimento quantitativo do processo e do produto, o desempenho continuaria a melhorar nas organizações de Nível 4.

Com os processos bem definidos, o desempenho melhoraria nas organizações de Nível 3.

Os planos baseados em desempenhos anteriores tenderiam a ser mais realistas nas organizações de Nível 2.

As previsões de prazo e orçamento seriam tipicamente inatingíveis pelas organizações de Nível 1.

Figura 2.7. A Capacitação vs Nível de Maturidade. (Fonte: CMU/SEI-93-TR-24 ; tradução CPqD, 2001).

Terceiro, os resultados esperados tenderiam a melhorar à medida que a maturidade da organização aumentasse. Isto é, à medida que a organização de *software* amadurecesse, os custos diminuiriam, o tempo de desenvolvimento se tornaria menor, a produtividade e a qualidade aumentariam. Numa organização de Nível 1, o tempo de desenvolvimento poderia ser muito longo devido ao volume de retrabalho necessário para correção de erros. Contrastando com isso, as organizações de Nível 5 utilizariam melhoria contínua de processo e técnicas de prevenção de defeitos para aumentar a eficiência do processo e eliminar retrabalho oneroso, permitindo que o tempo de desenvolvimento fosse reduzido. (Isso é ilustrado na Figura 7 através do deslocamento horizontal da linha alvo em relação à origem).

Os aprimoramentos na previsão dos resultados do projeto representados na Figura 2.7 assumem que os resultados do projeto de *software* se tornam mais previsíveis à medida que a interferência, na forma de retrabalho, é eliminada. Sistemas sem precedentes complicam o quadro, uma vez que as

novas tecnologias e as novas aplicações reduzem a Capacitação do processo através do aumento da variabilidade.

O relatório prossegue afirmando que mesmo no caso de sistemas sem precedentes, as características das práticas de gestão e de engenharia de muitas organizações maduras ajudam a identificar e encaminhar os problemas no ciclo de desenvolvimento antes mesmo de serem detectados em organizações menos maduras. A detecção antecipada de defeitos contribui para a estabilidade e desempenho do projeto, eliminando o retrabalho durante as fases posteriores. A gestão de riscos é uma parte integrante da gestão de projeto num processo maduro. Em alguns casos, um processo maduro pode significar que projetos “fadados ao fracasso” sejam identificados mais cedo no ciclo de vida de *software* e o investimento em causa perdida seja minimizado.

2.3. Modelos de Qualidade de *Software* – Uma Necessidade

A motivação em aplicar modelos ou estruturas ou, ainda, *frameworks* de qualidade aos processos de criação de *software* decorre basicamente da necessidade de se obter o máximo de satisfação do cliente com o mínimo custo possível.

Para o caso de empresas produtoras de *software*, além dessa necessidade básica advêm outras, tais como: maximização do lucro da *softwarehouse*, garantia da perenidade do negócio, crescimento da *softwarehouse*.

Para o caso de clientes internos dos Departamentos de Informática (ou SPUs – *Software Producing Units*) nas organizações a motivação básica continua sendo obter o máximo de satisfação dos usuários com o menor custo, pois é isso que garante a sobrevivência dos *software* e das equipas de engenheiros de *software* que os criam e os mantêm.

Assim, entendemos que modelos de qualidade de *software* são vitais às entidades que os criam, os comercializam e os mantêm operacionais, de forma

análoga à conclusão do DoD ("*USA Department of Defense*"), citado em Côrtes (2001), que compeliu esse Departamento de Defesa a patrocinar a Instituição que criaria o modelo de qualidade de *software* CMM (*Capability Maturity Model*). A mesma conclusão estende-se aos métodos, a exemplo do ora desenvolvido por esta pesquisa.

É graças a eles que se procurará garantir cumprimento de prazos e custos dimensionados aos projetos de *software* conforme as possibilidades de cada negócio, assumir riscos calculados, minimizar conseqüências desastrosas, enfim, dar condições a uma gestão mais eficaz das atividades de engenharia de *software* inerentes a todo o ciclo de vida dos projetos de sistemas.

Serão destacados 2 dos modelos mais difundidos internacionalmente, e com algumas iniciativas de utilização no mercado brasileiro por grandes empresas, o CMM e o SPICE.

O CMM (*Capability Maturity Model*) é o modelo de qualidade de *software* para avaliação e melhoria de processos de *software* inicialmente encomendado pelo DoD em 1984 e proposto por Watts S.Humphrey, que tem sido aperfeiçoado pelo SEI (*Software Engineering Institute*) da Universidade Carnegie Mellon. Propõe-se a disciplinar o controle e a visibilidade dos processos bem como os resultados técnicos e atividades de gestão de forma que, em uma escala crescente, permita que a organização que venha a aplicá-lo evolua em uma escala de 5 níveis de maturidade, numerados de 1 a 5. Baseia-se portanto, em um conceito de maturidade de processos e foi inspirado nas técnicas de TQM (*Total Quality Management – Gestão da Qualidade Total*).

O SPICE (*Software Process Improvement and Capability dEtermination - Futura Norma ISO/IEC 15504*) é um projeto fruto do esforço da comunidade internacional de engenheiros de *software* que têm se dedicado, nos últimos anos, a compilar e organizar os processos de *software* baseados na experiência e nas melhores características de vários métodos e modelos de avaliação de processos existentes, principalmente o CMM.

2.4. A NBR ISO / IEC 12207

A norma brasileira ISO/IEC 12207 é a norma com base na equivalente internacional que descreve em detalhes os processos, atividades e tarefas que envolvem o fornecimento, desenvolvimento, operação e manutenção de produtos de *software*. Esta norma começou a ser elaborada em junho de 1989 e foi aprovada em agosto de 1995.

Sua principal finalidade é servir de referência para os demais padrões que venham a surgir, assim a norma ISO 15504 (SPICE), organiza seu trabalho segundo o que está descrito na 12207.

Ela descreve uma arquitetura do ciclo de vida do *software* em termos gerais, como apresentado na figura 2.8.



Figura 2.8. Estrutura dos Processos do Ciclo de Vida do Software (Fonte: Rocha, 2001)

Esquemáticamente a norma agrupa os processos do ciclo de vida do *software* em três classes: processos fundamentais; processos de apoio e cada processo é definido com base em suas atividades, que por sua vez são definidas com base em suas tarefas.

A seguir está descrita a lista completa dos processos, agrupadas classe a classe:

Quadro 2.1. Processos Fundamentais

Processos Fundamentais	Atendem o início e execução do desenvolvimento, operação ou manutenção do <i>software</i> durante o seu ciclo de vida.
Aquisição	Atividades de quem adquire um <i>software</i> . Inclui a definição da necessidade de adquirir um <i>software</i> (produto ou serviço), pedido de proposta, seleção de fornecedor, gerência da aquisição e aceitação do <i>software</i> .
Fornecimento	Atividades do fornecedor de <i>software</i> . Inclui preparar uma proposta, assinatura de contrato, determinação recursos necessários, planos de projeto e entrega do <i>software</i> .
Desenvolvimento	Atividades do desenvolvedor de <i>software</i> . Inclui a análise de requisitos, projeto, codificação, integração, testes, instalação e aceitação do <i>software</i> .
Operação	Atividades do operador do <i>software</i> . Inclui a operação do <i>software</i> e suporte operacional aos usuários.
Manutenção	Atividades de quem faz a manutenção do <i>software</i> .

Quadro 2.2. Processos de Apoio

Processos de Apoio	Auxiliam um outro processo.
Documentação	Registro de informações produzidas por um processo ou atividade. Inclui planejamento, projeto, desenvolvimento, produção, edição, distribuição e manutenção dos documentos necessários a gerentes, engenheiros e usuários do <i>software</i> .
Gerência de Configuração	Identificação e controle dos itens do <i>software</i> . Inclui o controle de armazenamento, liberações, manipulação, distribuição e modificação de cada um dos itens que compõem o <i>software</i> .
Garantia da Qualidade	Garante que os processos e produtos de <i>software</i> estejam em conformidade com os requisitos e os planos estabelecidos.
Verificação	Determina se os produtos de <i>software</i> de uma atividade atendem completamente aos requisitos ou condições impostas a eles.

Validação	Determina se os requisitos e o produto final atendem ao uso específico proposto.
Revisão Conjunta	Define as atividades para avaliar a situação e produtos de uma atividade de um projeto, se apropriado.
Auditoria	Determina adequação aos requisitos, planos e contrato, quando apropriado.
Resolução de Problemas	Análise e resolução dos problemas de qualquer natureza ou fonte, descobertos durante a execução do desenvolvimento, operação, manutenção ou outros processos.

Quadro 2.3. Processos Organizacionais

Processos Organizacionais	Implementam uma estrutura constituída de processos de ciclo de vida e pessoal associados, melhorando continuamente a estrutura e os processos.
Gerência	Gerenciamento de processos.
Infra-estrutura	Fornecimento de recursos para outros processos. Inclui hardware, <i>software</i> , ferramentas, técnicas, padrões de desenvolvimento, operação ou manutenção.
Melhoria	Atividades para estabelecer, avaliar, medir, controlar e melhorar um processo de ciclo de vida de <i>software</i> .
Treinamento	Atividades para prover e manter pessoal treinado.

2.5. O CMM - Capability Maturity Model

O SW-CMM (*Capability Maturity Model for Software*), modelo de Maturidade da Capacidade, ou, simplesmente CMM, segundo definição do SEI (*Software Engineering Institute*) da Universidade Carnegie Mellon, entidade que o desenvolveu, é um modelo estrutural que descreve princípios e práticas relacionadas ao processo de desenvolvimento de *software*. O modelo visa ajudar organizações envolvidas com o desenvolvimento de *software* a melhorar a capacidade de seus processos, por meio de um caminho evolucionário que considera desde processos com resultados imprevisíveis e até mesmo caóticos

para processos disciplinados e definidos, com resultados previsíveis e com possibilidade de melhoria contínua.

Propõe-se a atingir esse objetivo disciplinando o controle e a visibilidade dos processos bem como os resultados técnicos e atividades de gestão de forma que, em uma escala crescente, permita que a organização que venha a aplicá-lo evolua em uma escala de 5 níveis de maturidade, numerados de 1 a 5. Baseia-se portanto, em um conceito de maturidade de processos e foi inspirado nas técnicas de TQM (*Total Quality Management* – Gestão da Qualidade Total).

O CMM descreve os elementos chave de um processo de desenvolvimento realmente efetivo. É um caminho evolutivo para a melhoria de processos, partindo de um estágio desestruturado (nível inicial) até um estágio onde o gerenciamento quantitativo e qualitativo do desenvolvimento de *software* se torne uma prática natural na empresa (nível otimizado).

Conforme explica o relatório CMU/SEI-93-TR-24, a estrutura em estágios do CMM é baseada em princípios de qualidade de produto dos últimos sessenta anos. Walter Shewhart definiu os princípios de controle estatístico da qualidade na década de 30. Esses princípios foram desenvolvidos e demonstrados com sucesso no trabalho de W. Edwards Deming e Joseph Juran. Esses princípios foram adaptados pelo SEI dentro da estrutura de maturidade que estabelece a gestão de projeto e os fundamentos de engenharia para o controle quantitativo do processo de *software*, que é a base para a contínua melhoria do processo.

O relatório CMU/SEI-93-TR-24 prossegue narrando que a estrutura de maturidade da qual esses princípios de qualidade foram adaptados foi primeiramente inspirada por Philip Crosby, que elaborou um quadro de maturidade de gestão da qualidade descrevendo as cinco etapas para a adoção das práticas de qualidade. Essa estrutura de maturidade foi adaptada para o processo de *software* por Ron Radice e seus colegas, sob a direção de Watts Humphrey da IBM. Humphrey trouxe essa estrutura de maturidade para

o Instituto de Engenharia de *Software* em 1986, acrescentou o conceito de níveis de maturidade e desenvolveu o fundamento para o seu uso atual na indústria de *software*.

O presente estudo tomou por base o CMM, versão 1.1, que apresenta os fundamentos necessários ao trabalho, e as proposições contidas nas normas e propostas de normas posteriores (ISO 9000-3, ISO 12207 e ISO 15504).

2.5.1. O SEI (*Software Engineering Institute*)

O CMM surgiu em 1984 quando o governo americano procedeu a uma revisão dos maiores contratos do Departamento de Defesa (DoD) para o desenvolvimento de *software* por terceiros, da ordem de milhões de dólares, que concluiu por apontar a existência de sérios problemas de não cumprimento de prazos, alguns com anos de atraso, todos atrasados.

A partir desse cenário o governo americano, sentindo a importância da necessidade de investir na criação de um padrão de qualidade nessa área, patrocinou a criação do SEI (*Software Engineering Institute*) da Universidade Carnegie Mellon, em Pittsburgh, no estado da Pensilvânia, para, a partir do modelo proposto por Watts S.Humphrey, desenvolver o CMM.

O SEI se destina à Pesquisa e Desenvolvimento na área de engenharia de *software*, com a missão de liderar iniciativas no sentido de melhorar radicalmente a qualidade de produção de sistemas que dependam de *software*, conforme se depreende de sua página na Internet:

The SEI provides the technical leadership to advance the practice of software engineering so the DoD can acquire and sustain its software-intensive systems with predictable and improved cost, schedule, and quality. The SEI mission includes four objectives:

- 1. accelerate the introduction and widespread use of high-payoff software engineering practices and technology by identifying, evaluating, and maturing promising or underused technology and practices;*
- 2. maintain a long-term competency in software engineering and technology transition;*

3. *enable industry and government organizations to make measured improvements in their software engineering practices by working with them directly;*
4. *foster the adoption and sustained use of standards of excellence for software engineering practice.*

(<http://www.sei.cmu.edu>)

Embora o CMM não seja uma norma emitida por uma instituição internacional (como a ISO ou o IEEE), tem tido uma grande aceitação mundial, até mesmo fora do mercado americano.

Praticamente os resultados de seu trabalho começaram a ser aplicados em 1987, com a aplicação de um questionário preliminar, desenvolvido pelo SEI, a partir de necessidades do Departamento de Defesa Americano (DoD), para avaliar a maturidade do processo de desenvolvimento de *software* das empresas interessadas em fornecer sistemas à Força Aérea Americana.

Em agosto de 1991, foi entregue à comunidade a versão 1.0 do SW-CMM, que, desde então vem sofrendo evolução continuada.

2.5.2. Maturidade

O modelo exige uma avaliação prévia da maturidade da organização que a ele se submete em relação ao processo de *software*. A definição do que é "Maturidade" pode ser mais bem compreendida através do quadro a seguir:

Quadro 2.4. Comparativo da maturidade das organizações

Organizações maduras	Organizações imaturas
Papéis e responsabilidades bem definidos e documentados	Processo improvisado
Existe base histórica dos processos e experiências bem como de métricas para garantir a objetividade das avaliações	Não existe base histórica
É possível julgar a qualidade do produto em função de padrões de qualidade pré-determinados.	Não há maneira objetiva de julgar a qualidade do produto
A qualidade dos produtos e processos é monitorada	Qualidade e funcionalidade do produto sacrificadas
O processo pode ser atualizado sistematicamente	Não há rigor no processo a ser seguido
Existe comunicação sistemática entre o gerente e seu grupo	Resolução de crises imediatas

2.5.3. A Estrutura do CMM

O CMM apresenta-se estruturado nos seguintes elementos básicos:

- Níveis de maturidade;
- Áreas-chave de processo;
- Características comuns; e
- Práticas-base;

que se interligam conforme o diagrama a seguir:

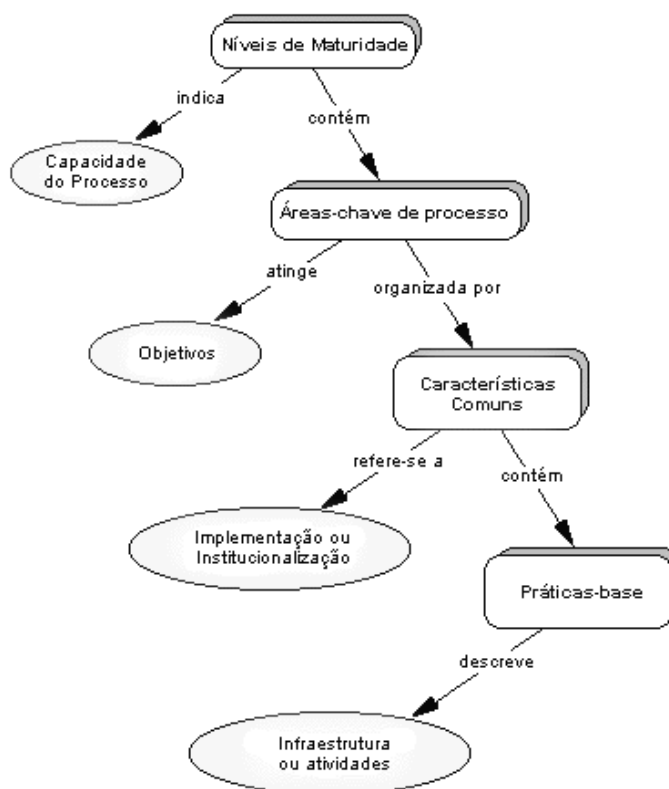


Figura 2.9. Inter-relacionamento dos elementos básicos do CMM. (Fonte: Paulk, 1993).

2.5.4. Os Níveis da Maturidade de Processo de *Software*

O CMM não apresenta nenhum processo revolucionário, mas é, sim, um processo sistematicamente organizado que permite uma melhoria contínua de processo. Assim, parte de uma classificação das organizações em cinco níveis

distintos de maturidade, cada um com características próprias, que estabelecem fundamentos sucessivos para a contínua melhoria do processo permitindo às organizações priorizar seus esforços de melhorias.

Os níveis de maturidade são, portanto, um estágio evolutivo bem definido em busca de um processo maduro de obtenção e manutenção de *software*.

Cada nível compreende um conjunto de objetivos de processos que, quando satisfeitos, estabilizam um componente importante do processo de *software*. Alcançando cada nível da estrutura de maturidade, estabelecem-se diferentes componentes no processo de *software*, resultando em uma evolução na escala de capacitação de processo da organização (Veja o quadro 2.5) .

No nível 1, das organizações mais imaturas, não há nenhuma metodologia implementada e tudo ocorre de forma desorganizada ou desestruturada. Este nível é também utilizado como base de comparação para as melhorias de processos alcançadas nos níveis mais elevados de maturidade.

Os níveis de maturidade de 2 a 5 podem ser caracterizados pelas atividades executadas pela organização para estabelecer ou melhorar o processo de *software*, pelas atividades realizadas em cada projeto e pela Capacitação de processo resultante nos projetos.

No nível 5, o das organizações mais maduras, cada detalhe do processo de desenvolvimento está definido, quantificado e acompanhado e a organização consegue até absorver mudanças no processo sem prejudicar o desenvolvimento.

Quadro 2.5. Descrição dos Níveis do CMM

Nível CMM	Descrição
1) Inicial	O processo de desenvolvimento é desorganizado, desestruturado ou confuso. Poucos processos são definidos e o sucesso depende do talento individual dos envolvidos. Por definição, todas as empresas não avaliadas estão no nível 1.
1) Inicial	O processo de desenvolvimento é desorganizado, desestruturado ou confuso. Poucos processos são definidos e o sucesso depende do talento individual dos envolvidos. Por definição, todas as empresas não avaliadas estão no nível 1.
2) Repetível	Os processos básicos de gerenciamento de projetos estão estabelecidos e

	permitem acompanhar custo, cronograma e funcionalidade. É possível repetir o sucesso de um processo utilizado anteriormente em outros projetos similares, graças ao disciplinamento dos processos garantido por normatização e gerenciamento efetivos.
3) Definido	Tanto as atividades de gerenciamento quanto as de engenharia do processo de desenvolvimento de <i>software</i> estão documentadas, padronizadas e integradas em um padrão de desenvolvimento da organização. Todos os projetos usam uma versão customizada aprovada do processo de <i>software</i> padrão da organização para desenvolvimento e manutenção de <i>software</i> . O sucesso passa a depender de bons processos.
4) Gerenciado	A organização se utiliza de métricas. Medidas detalhadas da qualidade do produto e processo de desenvolvimento de <i>software</i> passam a ser obtidas e gerenciadas. O processo de <i>software</i> e os produtos são quantitativamente entendidos e controlados.
5) Otimizado	O melhoramento contínuo do processo é obtido através do retorno quantitativo e sistemático dos processos e pelo uso pioneiro de idéias e tecnologias inovadoras.

Uma empresa no nível 1 caracteriza-se pela impossibilidade de garantir prazos, custos ou funcionalidade. No nível 2, a empresa já consegue produzir bom *software*, no prazo e a um custo previsível. O nível 3 garante um excelente nível de qualidade, tanto no produto quanto no processo de desenvolvimento como um todo. Já os níveis 4 e 5 exigem um aperfeiçoamento a partir do uso de métricas e de inovações contínuas em engenharia de *software*, sendo que poucas empresas no mundo atingiram esses estágios, nenhuma no Brasil.

Organizando o CMM nos cinco níveis mostrados na figura anterior, priorizam-se ações de melhoria para o crescimento da maturidade do processo de *software*. As setas rotuladas na figura seguinte indicam o tipo de Capacitação de processo que está sendo institucionalizado pela organização a medida que ela evolui na escala de maturidade.

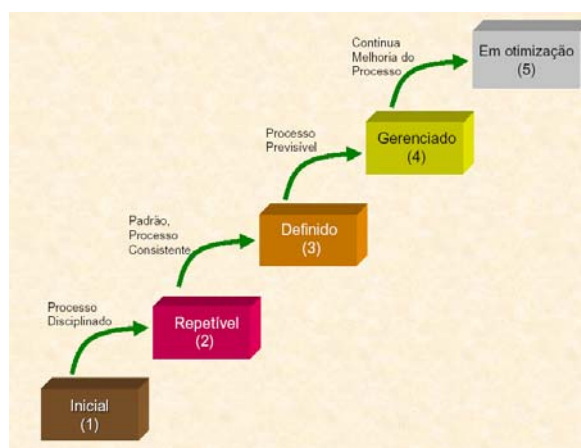


Figura 2.10. A evolução da maturidade via melhoria dos processos (Côrtes, 2001 / Fig. 5-2)

2.5.5. Áreas-chave de processo (*Key Process Areas* ou *KPAs*)

Exceto no nível 1, todos os níveis são detalhados em áreas-chave de processo ou *KPAs* (*Key Process Areas*). Estas áreas são exatamente aquilo no que a organização deve focar para melhorar o seu processo de desenvolvimento de *software*. Para que uma empresa possa se qualificar em um determinado nível de maturidade CMM, deve estar realizando os processos relacionados às áreas-chave daquele determinado nível. Todas as áreas-chave estão citadas na tabela abaixo:

Quadro 2.6. Os Focos dos 5 níveis e suas Áreas-chave de Processo

Nível CMM	Foco	Áreas-chave de processo
1) Inicial	Os processos dependem do talento individual dos agentes	
2) Repetível	Processos de gerenciamento de projetos	Visão geral e acompanhamento do projeto Gerenciamento de Requisitos Planejamento do Projeto de <i>Software</i> Acompanhamento e Rastreamento do Projeto Gerenciamento de Configuração de <i>Software</i> Gerenciamento de Subcontratação Gerenciamento de configuração Garantia de Qualidade em <i>Software</i>
3) Definido	Processos de engenharia e apoio	Foco no Processo Organizacional Definição do Processo Organizacional Gerenciamento Integrado de <i>Software</i> Engenharia de Produto de <i>Software</i> Coordenação entre Grupos Revisões Conjuntas entre Pares Programa de Treinamento
4) Gerenciado	Qualidade do produto e do processo	Gerenciamento Quantitativo dos Processos Gerenciamento da Qualidade do <i>Software</i>
5) Otimizado	Melhoramento contínuo do processo	Prevenção de Defeitos Gerenciamento de Mudanças Tecnológicas Gerenciamento de Mudanças em Processos

2.5.6. Objetivos das áreas-chave de processo

O modelo CMM define um conjunto de dois a quatro objetivos para cada área-chave. Esses objetivos definem aquilo que deve ser alcançado no caso de os

processos desta área-chave serem efetivamente realizados conforme a tabela a seguir:

Quadro 2.7. Os objetivos de algumas das Áreas-chave de Processos

Nível CMM	Áreas-chave de processo	Objetivos
1) Inicial		
2) Repetível	Gerenciamento de requisitos	<p>Os requisitos do sistema definidos para o <i>software</i> são controlados de forma a estabelecer um perfil mínimo a ser utilizado pela engenharia de <i>software</i> e pela administração</p> <p>Os planos, produtos e atividades do <i>software</i> são sempre consistentes com os requisitos de sistema definidos para o <i>software</i>.</p>
	Planejamento do projeto	<p>Estimativas relativas ao <i>software</i> são documentadas para uso no planejamento e acompanhamento do projeto do <i>software</i>.</p> <p>As atividades de projeto de <i>software</i> e compromissos assumidos são planejados e documentados.</p> <p>Grupos e pessoas afetadas concordam com seus compromissos relacionados ao projeto do <i>software</i>.</p>
	Visão geral e acompanhamento do projeto	<p>Resultados reais são acompanhados de acordo com o planejamento do <i>software</i>.</p> <p>Quando os resultados apresentam um significativo desvio do planejamento do <i>software</i>, são tomadas ações corretivas que são acompanhadas até o final do projeto.</p> <p>Mudanças nos compromissos assumidos são feitas em comum acordo com os grupos e indivíduos afetados.</p>
	Gerenciamento de subcontratados	<p>O contratante seleciona subcontratos qualificados</p> <p>O contratante e os subcontratados estão de acordo no que diz respeito aos compromissos assumidos um com o outro.</p> <p>O contratante e os subcontratados mantêm uma comunicação constante</p> <p>O contratante acompanha os resultados reais do subcontratado de acordo com os compromissos assumidos</p>

	Garantia da qualidade do <i>software</i>	As atividades de garantia de qualidade de <i>software</i> são planejadas A conformidade dos produtos de <i>software</i> e atividades com os padrões, procedimentos e requisitos é verificada objetivamente. Os grupos e indivíduos afetados são informados das atividades de garantia de qualidade de <i>software</i> e de seus resultados. Questões relacionadas à não conformidade que não são resolvidas dentro do projeto de <i>software</i> são encaminhadas à gerência geral
	Gerenciamento de configuração	As atividades de gerenciamento de configuração são planejadas. Os produtos de trabalho de <i>software</i> são identificados, controlados e estão disponíveis. Mudanças nos produtos de trabalho identificados são controladas. Os grupos e pessoas afetadas são informados da situação atual e projetada dos produtos de trabalho de <i>software</i> .
3) Definido	Foco do processo organizacional	São coordenadas atividades de desenvolvimento e melhoramento do processo de <i>software</i> em toda a organização Os pontos fortes e fracos do processo de desenvolvimento de <i>software</i> utilizado são identificados, de acordo com um padrão de processo. São planejadas atividades de desenvolvimento e melhoramento do processo, a nível de organização.
	Definição do processo organizacional	O processo padrão de desenvolvimento de <i>software</i> da organização é desenvolvido e mantido. A informação relacionada ao uso do processo padrão de desenvolvimento de <i>software</i> é coletada, revisada e disponibilizada.
	Programa de treinamento	As atividades de treinamento são planejadas É fornecido treinamento para o desenvolvimento de habilidades e conhecimentos necessários para realizar o gerenciamento do <i>software</i> e as funções técnicas. As pessoas no grupo de engenharia de <i>software</i> e outros grupos relacionados a <i>software</i> recebem o treinamento necessário para realizar as suas funções
	Gerenciamento de <i>software</i> integrado	O processo de <i>software</i> definido para o projeto é uma versão adaptada do processo padrão de desenvolvimento de <i>software</i> da organização O projeto é planejado e gerenciado de acordo com o processo de desenvolvimento de <i>software</i> definido para o projeto

	Engenharia de produto de <i>software</i>	As atividades de engenharia de <i>software</i> são definidas, integradas e consistentemente realizadas para produzir o <i>software</i> . Os produtos de trabalho do <i>software</i> são mantidos consistentes entre si.
	Coordenação intergrupos	Todos os grupos de trabalho afetados concordam com os requisitos dos cliente. Todos os grupos de trabalho afetados concordam com os acordos entre os grupos de engenharia Os grupos de engenharia identificam, acompanham e resolvem todas as questões intergrupos.
	Revisão conjunta	Atividades de revisão conjunta são planejadas Defeitos nos produtos de trabalho são identificados e removidos.
4) Gerenciado	Gerenciamento quantitativo dos processos	As atividades de gerenciamento quantitativo dos processos são planejadas A performance do processo de desenvolvimento de <i>software</i> definido para o projeto é controlada quantitativamente A capacidade do processo desenvolvimento de <i>software</i> padrão da organização é conhecida em termos quantitativos.
	Gerenciamento da qualidade de <i>software</i>	As atividades de gerenciamento da qualidade de <i>software</i> do projeto são planejadas. Objetivos mensuráveis da qualidade do produto de <i>software</i> e suas prioridades são definidos. O progresso real em direção à realização dos objetivos de qualidade para os produtos de <i>software</i> é quantificado e gerenciado.
5) Otimizado	Prevenção de defeitos	As atividades de prevenção de defeitos são planejadas As causas comuns de defeitos são procuradas e identificadas As causas comuns de defeitos são priorizadas e sistematicamente eliminadas.
	Gerenciamento de mudanças tecnológicas	A incorporação de mudanças tecnológicas é planejada Novas tecnologias são avaliadas para determinar seu efeito na qualidade e na produtividade Novas tecnologias adequadas são incorporadas na prática normal de toda a organização.
	Gerenciamento de mudanças no processo	O melhoramento contínuo do processo é planejado Toda a organização participa das atividades de melhoramento do processo de <i>software</i> O padrão de processo de <i>software</i> da organização e os processos de <i>software</i> de cada projeto definido são melhorados continuamente.

2.5.7. As Características comuns e práticas-base

As características comuns são itens a serem observados para que se possa verificar a implementação e institucionalização de cada área-chave de processo. Elas podem indicar se a área-chave de processo é eficiente, repetível e duradoura. São cinco as características comuns no modelo CMM e cada uma possui suas práticas-base a serem realizadas.

Quadro 2.8. As 5 características comuns e suas práticas-base

Característica Comum	Descrição	Práticas-base
Compromisso de realizar	Atitudes a serem tomadas pela organização para garantir que o processo se estabeleça e seja duradouro.	Estabelecimento de políticas e apadrinhamento de um gerente experiente.
Capacidade de realizar	Pre-requisitos que devem existir no projeto ou na organização para implementar o processo de forma competente.	Alocação de recursos, definição da estrutura organizacional e de treinamento.
Atividades realizadas	Papéis e os procedimentos necessários para implementar uma área-chave de processo.	Estabelecimento de planos e procedimentos, realização do trabalho, acompanhamento do trabalho e tomada de ações corretivas, se necessário.
Medições e análise	Necessidade de medir o processo e analisar as medições.	Realização de medições para determinar o estado e a efetividade das atividades realizadas.
Implementação com Verificação	Passos para garantir que as atividades são realizadas de acordo com o processo estabelecido.	Revisão, auditoria e garantia de qualidade.

As práticas-chave descrevem as atividades que contribuem para atingir os objetivos de cada área-chave do processo. Em geral são descritas com frases simples, seguidas de descrições detalhadas (chamadas de subpráticas) que podem até incluir exemplos. As práticas-base devem descrever "o que" deve ser feito e não "como" os objetivos devem ser atingidos. O modelo CMM

inclui um extenso documento em separado, chamado "Práticas-base para o CMM", que lista todas as práticas-chave e subpráticas para cada uma das áreas-chave de processo.

2.6. A Norma ISO / IEC 15.504 (SPICE).

O SPICE, acrônimo para *Software Process Improvement and Capability dEtermination*, é uma norma em elaboração pela ISO / IEC objetivando tornar-se um padrão para a avaliação do processo de *software* e para determinar a capacitação de uma organização. A norma pretende orientar a organização para uma melhoria contínua do processo cobrindo todos os aspectos da Qualidade do Processo de *Software*.

2.6.1. Histórico do SPICE

Em junho de 1991 a ISO / IEC aprovou um período de estudo, por sua resolução ISO/IEC JTC1/SC7 n. 144, para a investigação das necessidades e requisitos exigidos à padronização da avaliação dos processos relacionados à criação e uso de *software*. Em Janeiro de 1993 foi aprovado pela ISO/IEC JTC1 o programa de trabalho para o desenvolvimento de um padrão internacional. Nesse sentido, em junho de 1993 constituiu-se a Organização do Projeto SPICE (*SPICE Project Organisation*) que liberou os trabalhos preliminares, na forma de esboço, em junho de 1995 (*Concepts and Introductory Guide* do SPICE v1.00).

Os trabalhos que se seguiram à elaboração da norma, na forma que se encontra hoje, têm sido desenvolvidos graças a um esforço conjunto de cinco centros técnicos espalhados pelo mundo (EUA, Canadá/América Latina, Europa, Pacífico Norte e Pacífico Sul). O Brasil participa nesse projeto através de um grupo de estudos da ABNT que atua no processo de desenvolvimento, e na tradução das versões preliminares da norma para o português.

Somente em 2003 a norma se encontra acabada, porém disponível apenas como versão preliminar em formato de relatório técnico (TR – *Technical Report*) da ISO/IEC com previsão de ser lançada como Norma. Este foi o trabalho utilizado nesta pesquisa, referenciado como ISO/IEC TR 15504, que, por simplificação, é referido apenas por TR 15504 ou, simplesmente, 15504, ou mesmo SPICE.

Apenas no primeiro trimestre de 2006 foi liberada, na versão de norma, e não mais Relatório Técnico, sua parte 5 que descreve o Modelo de Avaliação do Processo (ISO/IEC 15504-5).

Esta norma, em decorrência do amplo trabalho de pesquisa na década de 90, onde vários outros modelos e normas se consolidaram, integra muito do que compõe os modelos CMM, TickIT (derivado da ISO 9000-3), o Trillium, o Bootstrap, dentre outros (Côrtes, 2001), que podem ser, cada qual, combinados à norma em um programa de melhoria da qualidade conforme o interesse da organização que os adote.

Ainda, segundo Côrtes (2001), por ter causado fortes reações dos detentores dos modelos existentes, o que significava alto risco de continuidade do projeto SPICE, a ISO “*modificou a abordagem e retirou do modelo detalhes de implementação com o objetivo de torná-lo, de certa maneira, compatível com os existentes (upward compatible)*”. Assim, o SPICE tornou-se um modelo mais genérico permitindo a utilização complementar dos demais modelos.

2.6.2. Estrutura do SPICE

Para aplicar o SPICE, como qualquer outro modelo, a organização interessada na melhoria contínua do processo, ou na determinação de sua capacidade, deve proceder, previamente, a uma avaliação. Dentro do contexto de melhoria de processos a avaliação significa a caracterização das práticas correntes da organização, unidade organizacional ou projeto em termos da capacidade dos processos selecionados. A análise dos resultados é feita em relação às

necessidades de negócio da organização, identificando os aspectos positivos e negativos, e os riscos associados aos processos. Permite identificar como os processos atingem seus objetivos e a identificar causas de baixa qualidade, alto custo ou tempo excessivos, como priorizar a melhoria dos processos.

A 15504 define um modelo de referência com um conjunto universal de processos fundamentais para engenharia de *software* e um roteiro racional para avaliação e melhoria de cada processo (capacidade de processos) e por isso pode ser combinada a qualquer outro modelo de melhoria da qualidade.

Portanto, pode-se considerar que o SPICE é um modelo ou estrutura composto de um conjunto padronizado de processos fundamentais baseado nas melhores práticas da engenharia de *software*.

É dividido em 9 partes conforme a figura a seguir:

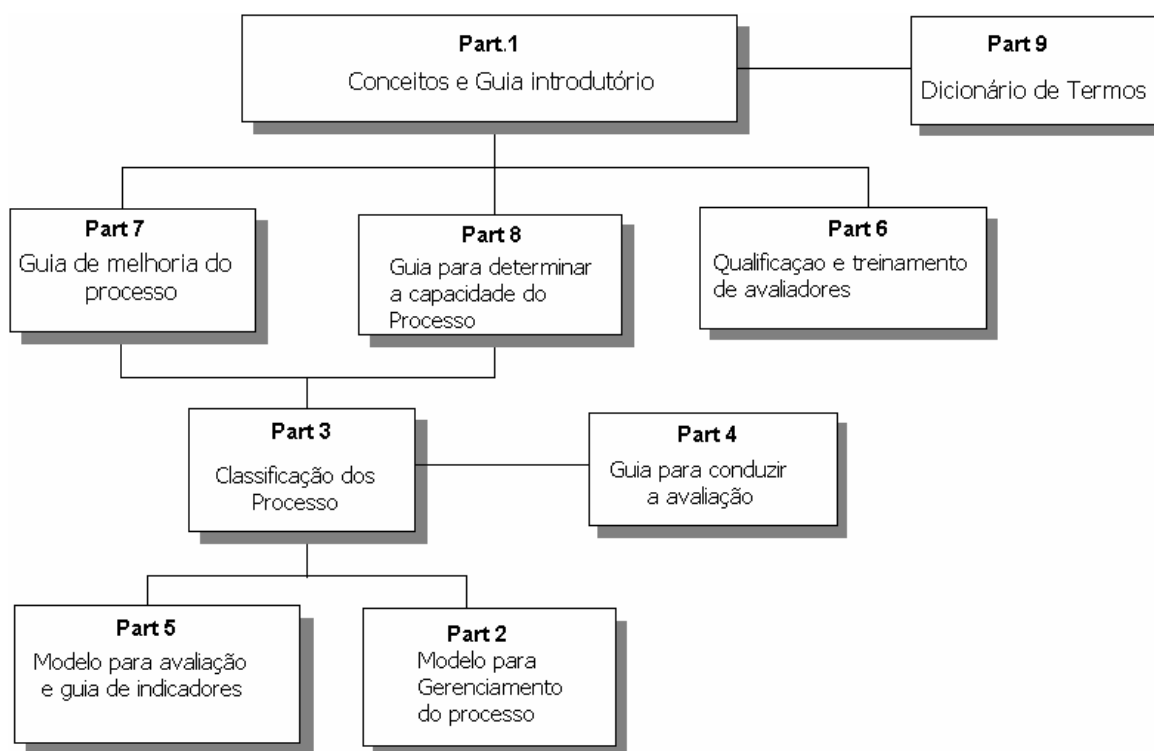


Figura 2.11. Componentes do SPICE (adaptado do site SPICE)

2.6.3. Categorias de Processos

Os processos do SPICE são agrupados como na figura seguinte:

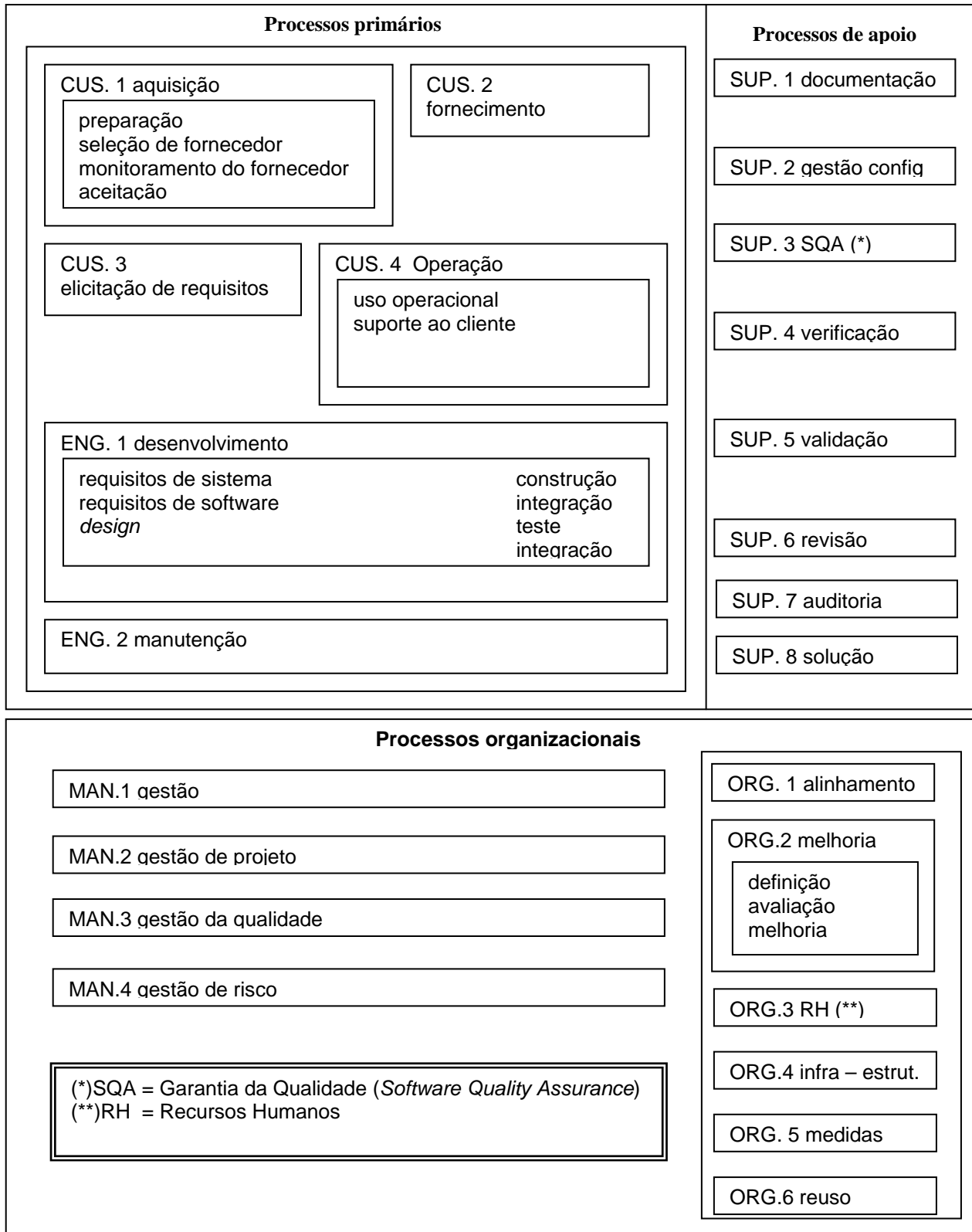


Figura 2.12. A dimensão de processos. (Fonte: Côrtes, 2001, p.113/F7-4)

O SPICE é derivado da norma ISO/IEC 12207 que detalha bem os processos inerentes à criação e utilização de *software*, tendo sido esquematizado no diagrama mostrado à página anterior, semelhantemente àquela norma.

Assim, a dimensão processos do SPICE apresenta-se dividida em cinco grandes categorias de processo esquematizadas a seguir: Cliente-Fornecedor (CUS), Engenharia (ENG), Suporte (SUP), Gerência (MAN) e Organização (ORG). Cada uma destas categorias é detalhada em processos mais específicos. Tudo isso é descrito em detalhes pela norma e ilustrado sinteticamente no quadro a seguir, compatível com a figura anterior de Côrtes (2001)²:

Quadro 2.9. Breve Descrição das Categorias de Processos

CUS - Cliente-Fornecedor

Processos que impactam diretamente os produtos e serviços de *software* na fornecedor para o cliente.

ENG – Engenharia

Processos que especificam, implementam ou mantêm um sistema ou produto de *software* e sua documentação

SUP – Suporte

Processos que podem ser empregados no apoio a qualquer um dos outros processos

MAN – Gerência

Processos que contém práticas de natureza genérica destinadas ao gerenciamento ou coordenação dos elementos e recursos dos projetos ou processos

ORG – Organização

Processos que estabelecem os objetivos de negócios da organização e desenvolve processos, produtos ou recursos destinados a ajudar a organização a atingir seus objetivos

Nos documentos da norma esses processos são descritos pelos seguintes atributos: identificador (ex: CUS 1.1); nome (Adquirir *Software*); objetivo, resultados e demais informações relevantes ao processo.

² Posteriormente, ao instante de aplicação do Método (capítulo 4) lançou-se mão de nova versão ainda experimental do SPICE, que reestrutura a categoria de processos ENG em 12 processos ao invés dos 7 descritos por Côrtes, e ilustrado à figura 4.23.

Todos os processos envolvidos no desenvolvimento de *software*, agrupados em suas categorias, são apresentados a seguir:

Quadro 2.10. Estrutura das categorias, dos processos de cada categoria

Processo	Descrição
CUS - Cliente-Fornecedor	
Processos que impactam diretamente os produtos e serviços de <i>software</i> na fornecedor para o cliente.	
CUS.1	Adquirir <i>Software</i>
CUS.2	Gerenciar necessidades do Cliente
CUS.3	Fornecer <i>Software</i>
CUS.4	Operar <i>Software</i>
CUS.5	Prover Serviço ao Cliente
ENG - Engenharia	
Processos que especificam, implementam ou mantêm um sistema ou produto de <i>software</i> e sua documentação	
ENG.1	Desenvolver requisitos e o projeto do sistema
ENG.2	Desenvolver requisitos de <i>software</i>
ENG.3	Desenvolver o projeto do <i>software</i>
ENG.4	Implementar o projeto do <i>software</i>
ENG.5	Integrar e testar o <i>software</i>
ENG.6	Integrar e testar o sistema
ENG.7	Manter o sistema e o <i>software</i>
SUP - Suporte	
Processos que podem ser empregados por qualquer um dos outros processos	
SUP.1	Desenvolver a documentação
SUP.2	Desempenhar a gerência de configuração
SUP.3	Executar a garantia da qualidade
SUP.4	Executar a verificação dos produtos de trabalho
SUP.5	Executar a validação dos produtos de trabalho
SUP.6	Executar revisões conjuntas
SUP.7	Executar auditorias
SUP.8	Executar resolução de problemas

MAN - Gerência

Processos que contêm práticas de natureza genérica que podem ser usadas por quem gerencia projetos ou processos dentro de um ciclo de vida de *software*

MAN.1	Gerenciar o projeto
MAN.2	Gerenciar a qualidade
MAN.3	Gerenciar riscos
MAN.4	Gerenciar subcontratantes

ORG - Organização

Processos que estabelecem os objetivos de negócios da organização

ORG.1	Construir o negócio
ORG.2	Definir o processo
ORG.3	Melhorar o processo
ORG.4	Prover recursos de treinamento
ORG.5	Prover infra-estrutura organizacional

A norma define detalhes de cada um dos processos mencionados acima com uma lista dos resultados da sua implementação bem sucedida e uma descrição detalhada de cada uma das práticas básicas.

2.6.4. Níveis de Capacitação e Atributos do Processo

Como um dos objetivos do SPICE é avaliar a capacitação da organização, ele define, além dos processos, seis níveis de capacitação para a classificação de cada processo, portanto, comparado ao CMM, apresenta 1 nível a mais, o incompleto (nível 0). Os níveis são: incompleto (0); realizado / executado (1); gerenciado (2); estabelecido (3); previsível (4) e otimizado (5), descritos a seguir.

A medida da capacitação é baseada em um conjunto de atributos de processos (*Process Attributes – PA*) definidos como necessários de existir para garantir que o processo tenha atingido o nível desejado, sendo que a descrição dos PAs será desenvolvida após o quadro a seguir.

Quadro 2.11. Descrição dos Níveis de Capacitação do SPICE

Nível	Nome	Descrição
0	Incompleto	Há uma falha geral em realizar o objetivo do processo. Não existem produtos de trabalho nem saídas do processo facilmente identificáveis. Não há atributos de processo para este nível.
1	Executado	O objetivo do processo em geral é atingido, embora não necessariamente de forma planejada e controlada. Há um consenso na organização de que as ações devem ser realizadas e quando são necessárias. Existem produtos de trabalho para o processo e eles são utilizados para atestar o atendimento dos objetivos. Há um atributo de processo para este nível, o PA 1.1.
2	Gerenciado	O processo produz os produtos de trabalho com qualidade aceitável e dentro do prazo. Isto é feito de forma planejada e controlada. Os produtos de trabalho estão de acordo com padrões e requisitos, e , para avaliar este processo há 2 PAs, o PA 2.1 e o PA 2.2.
3	Estabelecido	O processo é realizado e gerenciado usando um processo definido, baseado em princípios de Engenharia de <i>Software</i> . As pessoas que implementam o processo usam processos aprovados, que são versões adaptadas do processo padrão documentado, e , para avaliar este processo há 2 PAs, o PA 3.1 e o PA 3.2.
4	Previsível	O processo é realizado de forma consistente, dentro dos limites de controle, para atingir os objetivos. Medidas da realização do processo são coletadas e analisadas. Isto leva a um entendimento quantitativo da capacitação do processo a uma habilidade de prever a realização, e , para avaliar este processo há 2 PAs, o PA 4.1 e o PA 4.2.
5	Otimizado	A realização do processo é otimizada para atender às necessidades atuais e futuras do negócio. O processo atinge seus objetivos de negócio e consegue ser repetido. São estabelecidos objetivos quantitativos de eficácia e eficiência para o processo, segundo os objetivos da organização. A monitoração constante do processo segundo estes objetivos é conseguida obtendo feedback quantitativo e o melhoramento é conseguido pela análise dos resultados. A otimização do processo envolve o uso piloto de idéias e tecnologias inovadoras, além da mudança de processos ineficientes para atingir os objetivos definidos, e , para avaliar este processo há 2 PAs, o PA 5.1 e o PA 5.2.

Observa-se que ao nível 0 (Processo Incompleto) não se atribui nenhum PA, ao nível 1 (Processo Executado) atribui-se apenas um PA, e aos demais níveis, sempre há 2 PAs para um deles, listados a seguir:

Quadro 2.12. Níveis de Capacitação vs Atributos de Processo (PAs)

Nível	Nome	Atributos de Processos
0	Incompleto	Não há atributos de processo para este nível.
1	Executado	PA 1.1. Atributo de execução do processo
2	Gerenciado	PA 2.1. Atributo de gestão de execução do processo PA 2.2. Atributo de gestão dos produtos de trabalho
3	Estabelecido	PA 3.1. Atributo de definição do processo PA 3.2. Atributo de recursos do processo
4	Previsível	PA 4.1. Atributo de medidas do processo/produtos PA 4.2. Atributo de controle do processo
5	Otimizado	PA 5.1. Atributo de mudança do processo PA 5.2. Atributo de de melhoria contínua do processo

Cada um desses atributos é detalhadamente descrito na norma, acreditando-se ser desnecessário transcrevê-los a este trabalho uma vez que o Relatório Técnico está disponível na Internet, e ao trazê-lo a esta qualificação só a tornaria excessivamente volumosa, no entanto observa-se que, de acordo com o já apresentado a respeito do controle de qualidade aplicado aos processos de *software*, e, semelhantemente ao já descrito no CMM, os atributos vão caminhando linearmente, na escala de 0 a 5, no sentido do aperfeiçoamento contínuo dos mesmos, sendo que por exemplo:

- o PA 4.1. (Atributo de medidas do processo/produtos) se destina a medir até onde se utilizam métricas (associadas aos objetivos do processo/produtos) para assegurar que o processo é executado efetivamente (ou eficazmente) contribuindo a atingir os objetivos empresariais;
- o PA 5.2. (Atributo de melhoria contínua) se destina a medir até que ponto as mudanças de processo contribuem para a eficaz melhoria contínua, o que significa que tal atributo prescreve que as mudanças dos processos devem ser estabelecidas segundo sua contribuição aos objetivos de negócio da organização, e que os riscos associados à mudança devem ser avaliados, bem como deve haver planos de ação a problemas potenciais que decorram

da mudança, como também deve haver uma lista de possíveis oportunidades de melhoria do processo permanentemente atualizada e gerenciada, e, ainda, um programa de melhoria contínua do processo, e assim por diante.

2.6.5. Avaliação da Capacitação de Processos e o Perfil de Capacidade

A avaliação da capacidade de um processo é feita com base em uma escala de quatro valores ordenados do menor ao maior, denominados de (1^o) **N** (*not achieved*); (2^o) **P** (*partially achieved*); (3^o) **L** (*Largely achieved*) e (4^o) **F** (*Fully achieved*), conforme descreve-se a seguir:

- (1^o) **N** (*not achieved*); ou não entendido: 0% a 15% - há pouca ou nenhuma evidência de que o atributo foi satisfeito.
- (2^o) **P** (*partially achieved*); ou parcialmente atendido: de 16% a 50% - há evidências de uma prática sistemática no sentido da satisfação do atributo. Entretanto alguns aspectos do atendimento podem ser imprevisíveis.
- (3^o) **L** (*Largely achieved*); ou largamente atendido: de 51% a 85% - há evidências de uma prática sistemática no sentido da satisfação do atributo. Entretanto a execução do processo pode variar em algumas unidades ou áreas de atividade.
- (4^o) **F** (*Fully achieved*). Ou totalmente atendido: de 86% a 100% - há evidências de uma prática sistemática no sentido da satisfação do atributo. Não há deficiências significativas em todas as áreas da organização.

Assim, o que caracterizará um processo ter atingido um determinado nível será o grau (L) ou (F) aos seus atributos, pois atingimento parcial (P), não lhe dá a significância necessária de caracterização.

E ainda, para caracterizar perfeitamente o atingimento de determinado nível, além de o processo ter atingido pelo menos o grau (L) em todos os atributos desse nível, é também condição indispensável que todos os níveis anteriores já tenham atingido o grau (F).

A figura a seguir demonstra como a atribuição dos graus aos diversos níveis se comporta para que um processo evolua do nível 0 ao nível 5.

Atributos	0	1	2	3	4	5	Legenda:
5.2						F-L	Símbolo Grau <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; background-color: white;"></div> N ou P </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; background-color: yellow;"></div> F-L </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; width: 20px; height: 20px; background-color: cyan;"></div> F </div>
5.1						F-L	
4.2					F-L	F	
4.1					F-L	F	
3.2			F-L	F	F	F	
3.1			F-L	F	F	F	
2.2		F-L	F	F	F	F	
2.1		F-L	F	F	F	F	
1.1	F-L	F	F	F	F	F	

Figura 2.13: Caracterização dos Níveis de Processos conforme o atingimento dos graus de seus Atributos. (adaptada de Côrtes, 2001, p. 124)

E a figura a seguir ilustra que o processo ENG.2 tem pontuação no nível 2, pois satisfaz os requisitos desse nível ((F) no atributo 1.1 e (F / L) nos atributos 2.1 e 2.2) e não satisfaz os requisitos para o nível 3, por ter um (P) no atributo 3.2.

De forma análoga o processo ORG3 encontra-se no nível 3.

	PA 1.1	PA 2.1	PA 2.2	PA 3.1	PA 3.2	PA 4.1	PA 4.2	PA 5.1	PA 5.2
ENG 2	F	F	L	F		L	L		
ORG 3	F	F	F	L	L				

Legenda:

Símbolo	F	L		
Grau	F	L	P	N

Figura 2.14: Exemplo de avaliação do perfil de capacidade para dois processos. (adaptada de Côrtes, 2001, p. 125)

2.6.6. Considerações Finais sobre o SPICE

Presentemente a norma que dispõe sobre o SPICE se apresenta como um excelente guia de capacitação em processos.

Salviano (2004) esclarece que a 15504, por ser um modelo contínuo, permite que a seleção de processos e a determinação do nível de Capacitação alvo atendam o contexto, as restrições e os objetivos do negócio de forma superior aos modelos por estágios.

Relativamente a modelos de avaliação apresenta o PRO2PI (*Process capability Profile for Processo Improvement*) explicando que tal técnica compõe características das arquiteturas contínuas e por estágios, destacando-se sobre os modelos de PCP (*Process Capability Profile*) e SPI (*Software Process Improvement*) documentados até então, todos não formais.

3. METODOLOGIA DE PESQUISA E APLICAÇÃO

O trabalho de pesquisa ora apresentado caracterizou-se por um estudo de natureza qualitativa na medida em que privilegiou o aspecto da qualidade e intensidade dos dados trabalhados em contraposição à quantidade ou extensão desses resultados. Demo (2001) relaciona a idéia de intensidade a dimensões do fenômeno, objeto de estudo, marcadas pela profundidade, pelo envolvimento e pela participação.

Caracteriza-se como proposição de uma ferramenta, o Método, e sua validação experimental inicial através de um estudo de caso (Ludke e André, 1986) com um enfoque de pesquisa-ação, conforme definido por Thiollent (1997): *“A pesquisa-ação consiste essencialmente em acoplar pesquisa e ação em um processo no qual os atores implicados participam, junto com os pesquisadores, para chegarem interativamente a elucidar a realidade em que estão inseridos, identificando problemas coletivos, buscando e experimentando soluções em situação real. Simultaneamente, há produção e uso de conhecimento.”* (pág. 14).

3.1. Procedimentos da Pesquisa-Ação

Sinteticamente os procedimentos de pesquisa foram os seguintes:

1. Pesquisa bibliográfica;
2. Determinação e delimitação do objeto de estudo;
3. Elaboração de hipóteses aplicáveis ao caso;
4. Elaboração de um método, a partir da literatura e da realidade do objeto de estudo;
5. Validação das hipóteses restritas ao caso e a um teste experimental inicial.

É importante ressaltar que tais procedimentos estão propostos como uma estrutura dinâmica de interação, numa relação dialética³, que pressupõe um constante diálogo entre seus atores, na medida em que cada procedimento retro alimentará o outro, conforme forem sendo desenvolvidos.

A interação desses procedimentos metodológicos pode ser esquematizada conforme o diagrama da figura 3.1. elaborado pelo pesquisador a partir de referenciais já estabelecidos nos estudos sobre metodologia de pesquisa com a finalidade orientar esquematicamente o trabalho.

Tal diagrama apóia-se nos trabalhos de Marconi e Lakatos (1982) e Demo (2001), onde, a partir da determinação e delimitação do objeto de estudo, se destaca a importância de estar o objeto intimamente relacionado à temática da pesquisa, com uma bibliografia sólida como referência, bem como ser viável sua execução, tanto em relação a prazos como a meios humanos e físicos disponíveis. Uma vez determinado e delimitado o objeto de estudo, ocorre o estabelecimento de hipóteses iniciais que orientam provisoriamente a pesquisa e contribuem para que a realidade concreta encontrada e a pesquisa bibliográfica efetuada sejam permanentemente confrontadas no decurso da experiência, propiciando ao desenvolvimento da pesquisa-ação uma orientação preliminar para que seja confluyente e não dispersante (Demo, 2001). As etapas seguintes de elaboração do modelo teórico e a validação de hipóteses por meio da aplicação piloto indicam a avaliação constante que ocorre entre as hipóteses iniciais, a pesquisa bibliográfica e a pesquisa-ação.

A dinâmica da realidade concreta, dada pela implementação da pesquisa-ação, requer uma constante análise baseada no contato direto com o objeto de estudo, nas hipóteses iniciais e seu confronto com esta última, como também com a bibliografia já sistematizada sobre o tema em estudo.

³ O conceito de dialética tem como referência o usado por Demo (2001) em que afirma: “a dialética privilegia o fenômeno da transição histórica, que significa a superação de uma fase por outra”. Tal conceito, tem como pressuposto uma constante reconstrução da realidade, construção esta que, no presente caso, seria fruto das interações entre o pesquisador e os atores envolvidos no processo e na realidade em estudo.

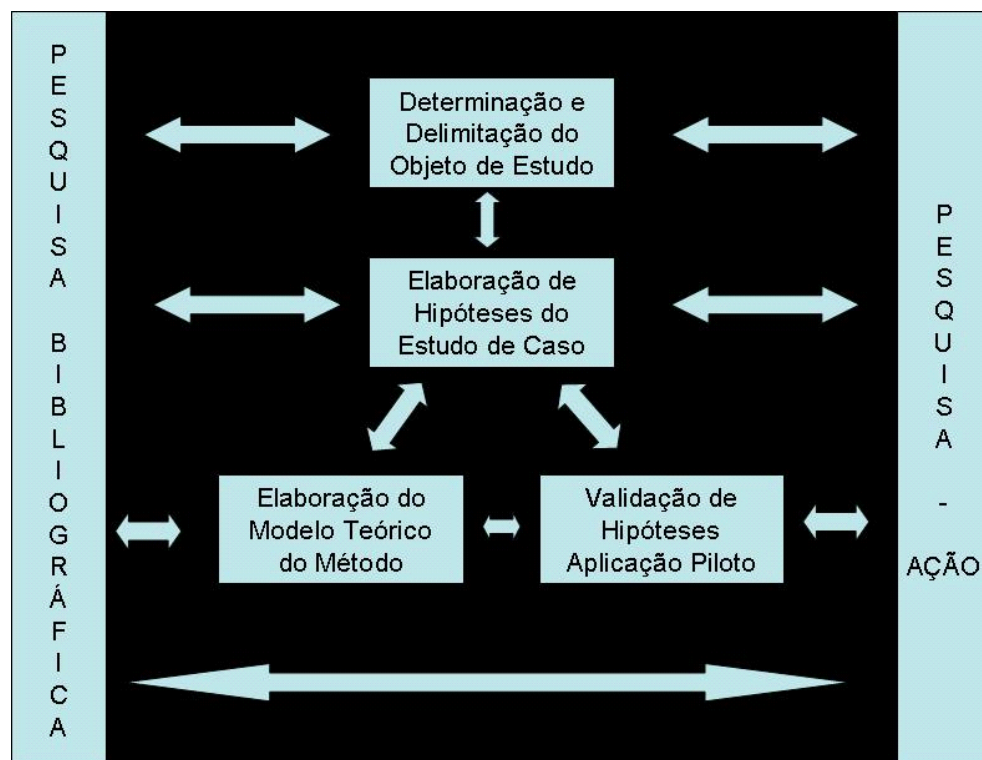


Figura 3.1. Diagrama de Interação dos Procedimentos de Pesquisa

3.1.1. Pesquisa bibliográfica

A pesquisa bibliográfica teve a finalidade de fornecer o referencial teórico para a elaboração de um método de intervenção e por uma pesquisa-ação para aplicação do referido método.

Justificando a importância da pesquisa bibliográfica para a elaboração do quadro conceitual, Thiollent (1997) afirma que esta tem o objetivo de, além de subsidiar a ação a ser implementada na pesquisa, contribuir que o grupo de participantes possa desvendar aspectos da realidade concreta por meio de um referencial mais consistente do que a simples prática do cotidiano. É a forma oportuna de colocar os participantes em dia com a temática escolhida para o estudo, propiciando os meios para definir, encaminhar soluções e resolver problemas conhecidos, além de abrir caminho para novas áreas e novas questões a serem exploradas (Marconi e Lakatos, 1982).

3.1.2. Determinação e Delimitação do Objeto de Estudo

A delimitação do objeto de estudo teve como função definir o caso a ser estudado e seus contornos, pois, como afirma Ludke & André (1986), nunca será possível explorar todos os ângulos do fenômeno a ser estudado, em um espaço de tempo razoavelmente limitado.

A ênfase, na definição do objeto de pesquisa, direcionou-se às pequenas e médias empresas produtoras de *software*, partindo de alguns pressupostos tais como: esse segmento é, possivelmente, (a) o mais carente na aplicação de *frameworks* de qualidade, em função da escassez de recursos, (b) o que demonstrará maior interesse pelos benefícios que essa aplicação possa trazer.

Para identificar uma *softwarehouse* brasileira que estivesse disposta a compartilhar da experiência de experimentação do método contataram-se diversas empresas, algumas produtoras de *software*, outras que apenas se utilizavam de equipes de desenvolvimento de aplicativos próprios, tendo a escolha recaída pela STORE Automação, cuja descrição se encontra no item 5.2, através de critérios de escolha também explicados nesse capítulo.

3.1.3. Elaboração de hipóteses aplicáveis ao caso

O estabelecimento de hipóteses iniciais constituiu-se em um conjunto de idéias que delinearão o tema em estudo, orientando a análise e contribuindo para que a pesquisa seja confluyente e não dispersante (Demo, 2001).

No caso da pesquisa em questão, as hipóteses iniciais foram fruto da observação da realidade e de experiências vivenciadas no setor, que não necessariamente encontram-se sistematizadas na literatura. Dessa maneira, o estabelecimento das hipóteses do presente trabalho desenvolve um diálogo entre diferentes experiências e relatos (mídia, eventos, experiências de vida em empresas do ramo, sites da web, etc.) e a literatura sistematizada sobre a temática em questão.

As hipóteses preliminares que podem ser apontadas são as que seguem:

1. Há cada vez mais um reconhecimento por empresas nacionais da importância da melhoria da qualidade no ramo da tecnologia de *software* e conseqüentemente um interesse cada vez mais no desenvolvimento de *frameworks* ou modelos da qualidade de *software* voltados especificamente para a realidade de empresas nacionais;
2. Embora o interesse por modelos de qualidade esteja avançando, ainda é baixa a adesão a eles por parte das empresas geradoras de *software*;
3. As grandes empresas, por sua capacidade financeira, têm condições de implementar os modelos de qualidade à frente das pequenas e médias empresas brasileiras de produção de *software*, antecipando vantagens competitivas;
4. Embora diferentes abordagens de melhoria da qualidade em *software*, com seus modelos ou normas, tais como o CMM e SPICE estabeleçam orientações sobre como proceder para se atingir patamares ou níveis de qualidade em *software*, observa-se a necessidade de adaptações às diferentes realidades onde serão implementadas, o que induz da necessidade de um método que oriente esse trabalho.

Tais hipóteses, exceto a 3^a, que especifica, podem ser lançadas ao universo de empresas independentemente do porte, cuja base amostral de 446 empresas produtoras de *software* de portes variados foi o foco da pesquisa bienal MCT-SEPIN (2002) (item 1.2),. Esse relatório associado foi a fonte de inspiração das referidas hipóteses.

3.1.4. Elaboração de um Modelo Teórico

Dentre as questões fundamentais que permeiam as pesquisas em processos de *software*, encontram-se os modelos e tecnologias de avaliação, apoio e melhoria das atividades de desenvolvimento de *software*. Nesse sentido, Rocha et al. (2001, pág. 2), citando Fuggetta, afirmam que “a definição e o uso

de processos de software envolve a complexa inter-relação de fatores organizacionais, culturais, tecnológicos e econômicos”.

Tendo em vista esses aspectos, os referidos autores afirmam que um único processo não pode ser ideal e adequado para qualquer tipo de empresa e projeto, necessitando adequações a características próprias das diferentes realidades. As características relacionadas à cultura organizacional, recursos disponíveis, tecnologias de desenvolvimento, experiência da equipe, porte da empresa e outros fatores similares são importantes de serem levadas em conta nesses processos de avaliação e melhoria de desenvolvimento de *software*.

Assim, a elaboração de um “**modelo teórico**”, de orientação das práticas a aplicar ao processo de desenvolvimento de *software*, é em realidade a elaboração de um **MÉTODO** (cuja definição é, segundo AURÉLIO (1986), “caminho para chegar a um fim”).

Para tanto, a partir da literatura pesquisada, mais especificamente das melhores práticas gerenciais, dos modelos e normas de qualidade, especialmente o CMM e o SPICE, foi desenvolvido um MÉTODO que permitiu a empresa de *software* implantar um Programa de Melhoria da Qualidade do Processo de *Software* de forma razoavelmente independente dos *frameworks* disponíveis no mercado.

3.1.5. Validação das hipóteses restritas ao caso e o teste experimental inicial

Por fim procedeu-se a uma experimentação do método proposto em uma situação específica de estudo de caso que permitiu validar algumas das hipóteses circunstanciadamente ao caso em questão, conforme se expõe na conclusão (item 6).

4. PROPOSTA DE UM MÉTODO

"O automóvel é só uma novidade. O cavalo está aqui para ficar."

*De um banqueiro ao advogado de Henry Ford,
desencorajando-o a investir na
Ford Motor Company, em 1900.*

http://pt.wikiquote.org/wiki/Henry_Ford

O **Método Circular para a Melhoria Contínua da Qualidade Aplicado ao Processo de Software (MC2Q-SW⁴)** surgiu a partir de idéias, conceitos e técnicas das áreas de Engenharia de Sistemas, Administração, Gestão de Projetos e Qualidade consubstanciados em modelos, normas técnicas e métodos desenvolvidos nos últimos 20 anos conforme breve descrição em 4.1.

A conjugação dos conceitos e técnicas citadas permitiu uma ordenação conceitual e metódica que se consubstanciou no trabalho ora apresentado resultando no MC2Q-SW que, tendo sido submetido a uma experiência de pesquisa-ação, foi amadurecido e complementado pela experiência piloto de aplicação em uma empresa, que se dispôs a servir de estudo de caso.

Este método, considerando a natureza e características específicas da organização onde venha a ser implantado, visa permitir a melhoria da qualidade de processos (usualmente referenciada como melhoria da “**capacitação**” dos processos) de Engenharia de Sistemas, através da conjugação de modelos, métodos e demais técnicas ou ferramentas que a organização entenda adequados à sua utilização.

⁴ MC2Q-SW (**Método Circular para a Melhoria Contínua da Qualidade Aplicado ao Processo de Software**) onde o 2 indica duplicidade das iniciais MC (de **Método Circular** e de **Melhoria Contínua**).

4.1. Apresentação do MC2Q-SW

O **MC2Q-SW** apoiou-se essencialmente em modelos, normas técnicas e métodos como o IDEAL, a ISO 12207, o CMM/CMMI, o SPICE (ISO 15504) referenciados em Côrtes (2001), no PMBOK (2003), e nos consagrados modelos de Desenvolvimento de Sistemas, especialmente o Espiral (Boehm, 1988), e na teoria geral da administração a exemplo do que segue:

- do CMM / CMMI, aproveitaram-se as idéias de escalabilidade da maturidade, tanto da Organização como da Capacitação do Processo; de avaliação de maturidade da Organização e da Capacitação; da possibilidade de Integração de várias técnicas e modelos; da necessidade de adaptação dos modelos às necessidades específicas das Organizações; e toda a terminologia em vigor. Particularmente, aproveitou-se a idéia dos 5 estágios da maturidade para, criticamente, estabelecerem-se 5 atributos aos processos que possam ser medidos no sentido de avaliar sua capacidade, ou “capacitação”;
- da norma ISO12207, aproveitaram-se a estruturação em que se classificam os processos e toda a terminologia em vigor;
- da norma ISO 15504 (SPICE), aproveitaram-se a estruturação evoluída da 12207, a abordagem “contínua” para a melhoria da capacitação do processo, em oposição à melhoria “estagiada” original do CMM, e toda a terminologia em vigor. Particularmente, em relação ao capítulo 5 (ISO/IEC 15504-5), aproveita-se a idéia dos 6 estágios de capacidade, para, criticamente, estabelecer 6 notas pontuais, de zero a cinco, para a avaliação dos 5 atributos determinados para avaliação do processo. A referida norma serviu também de paradigma para a formação de uma base de

conhecimento dos processos, na aplicação prática levada a cabo através de pesquisa-ação na empresa estudo de caso;

- do IDEAL, aproveitou-se a idéia do ciclo básico de melhoria, já utilizada em modelos diversos de gestão;
- do PMBOK, aproveitaram-se conceitos, especialmente de “projeto”; a racionalização da gestão de projetos e a terminologia associada;
- da Engenharia de Sistemas, aproveitaram-se os modelos de Desenvolvimento, especialmente o Espiral desenvolvido por Boehm (1988) e toda a terminologia associada; e, particularmente da metodologia *Extreme Programming* (XP, 2006) a idéia de redução do escopo em favor da qualidade (em contraposição ao paradigma de fixação de escopo em sacrifício da qualidade) que serviu de base ao estabelecimento dos princípios de racionalização que nortearam este trabalho;
- da Administração, aproveitaram-se os conceitos da prática da gestão e o método de Kepner e Tregoe (1997) para a tomada de decisões e resolução de problemas.

4.1.1. Visão Geral do MC2Q-SW (Fases e Etapas)

Esquemáticamente, apresenta-se a seguir uma visualização geral do método seguida de sucinta descrição de suas etapas principais, sendo oportuno introduzir, agora, as definições básicas de seus elementos estruturais:

- a) **FASE:** Conjunto de Etapas que permite a concretização de uma meta importante ao método (O MC2Q-SW se constitui de 3 fases);
- b) **ETAPA:** Conjunto de Processos que permite a concretização de metas intermediárias à conclusão da Fase (O MC2Q-SW se constitui de 13 etapas);

O MC2Q-SW se estrutura, segundo o foco de sua atenção, basicamente em três FASES (Diagnóstico e Capacitação; Preparação; Execução). A **PRIMEIRA FASE** foca a Organização, através de um **DIAGNÓSTICO SITUACIONAL (1.1)** e de uma **Proposição de Ações** que objetiva (a) avaliar a prontidão da Organização a um processo contínuo de melhoria da qualidade dos processos de *software* e identificar obstáculos a esses propósito; e (b) propor ações que capacitem a Organização a melhorar, continuamente, a capacitação dos processos de *software*; seguido por **AÇÕES DE CAPACITAÇÃO (1.2)**, através de Ciclos de Capacitação, que visam deixá-la apta a promover um Programa de Melhoria da Qualidade desses processos.

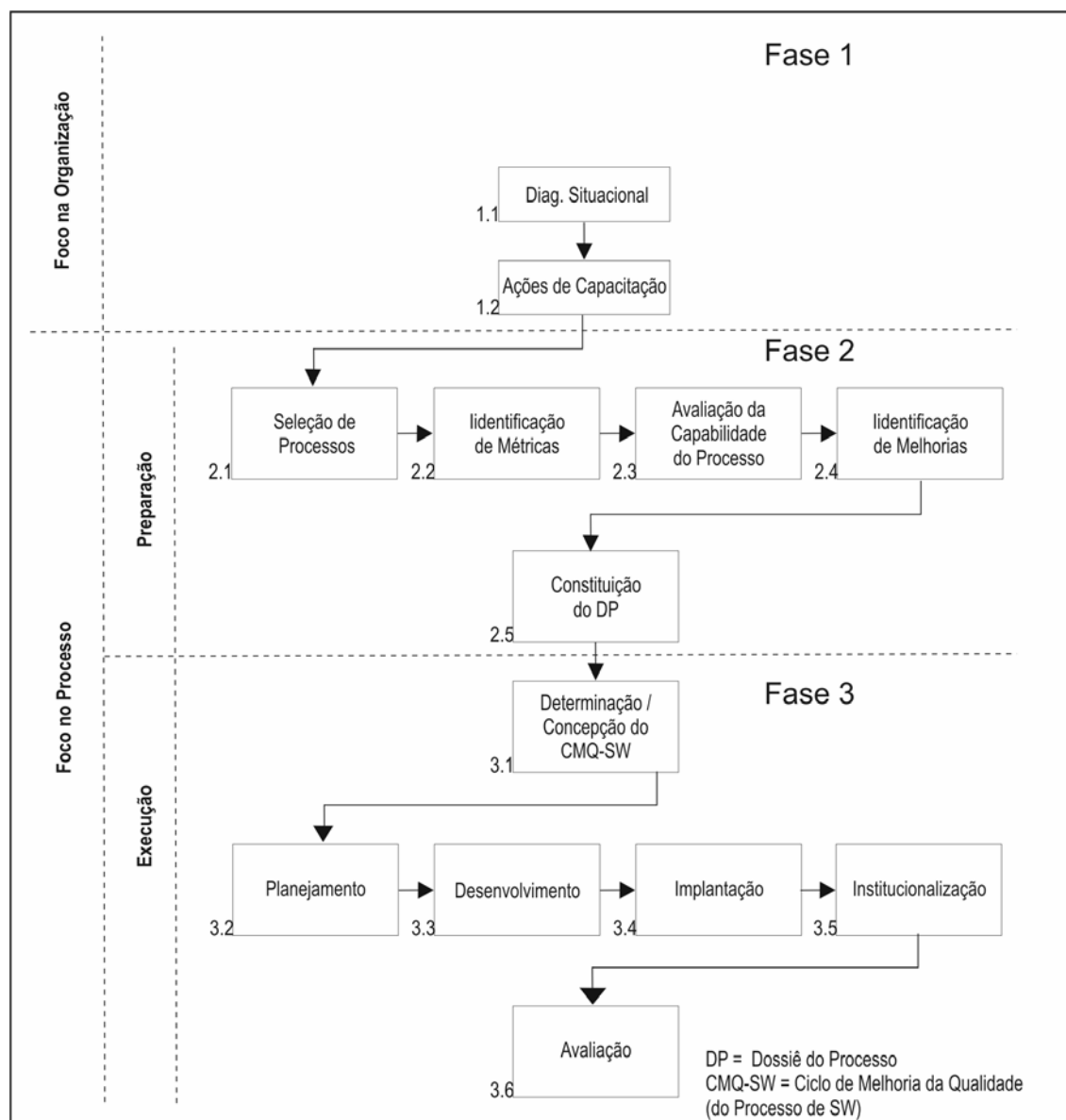


Figura 4.1. Diagrama de Visão Geral do MC2Q-SW

Uma vez encerrada essa fase, o método passa a focar o **PROCESSO**, (ou processos) objeto de melhoria, prosseguindo por um trabalho de preparação - **SEGUNDA FASE** - caracterizado pela determinação dos processos prioritários a serem melhorados (**SELEÇÃO (2.1)**); seguido pela **IDENTIFICAÇÃO DAS MÉTRICAS (2.2)** que permitirão a avaliação das melhorias intrínsecas aos processos; pela **AValiação DA CAPACITAÇÃO (2.3)** desses processos; e, ainda, pela **IDENTIFICAÇÃO DAS MELHORIAS (2.4)** - medidas (ações, práticas ou decisões) - que permitam o aperfeiçoamento das práticas.

O trabalho de preparação se encerra resultando na elaboração, para cada processo selecionado, de um **Dossiê de Processo (DP)**⁵, na etapa **CONSTITUIÇÃO DOS DPs (2.5)** que comportará tudo o necessário a orientar a melhoria da capacitação do processo escolhido, e que será complementado permanentemente, sempre que alterações ocorram nesse processo.

Por fim, o MC2Q-SW estipula regras de gestão da melhoria da capacitação dos processos que será efetivamente implementada, através de **Ciclos de Melhoria da Qualidade do Processo de Software (CMQ-PSW**, ou, simplesmente, **CMQ-SW**, eventualmente apenas **CMQ**), definidos e implementados na **TERCEIRA FASE** (de execução).

Tais ciclos são definidos em **DETERMINAÇÃO / CONCEPÇÃO DOS CMQS (3.1)** e implementados através das etapas de **PLANEJAMENTO (3.2)**; **DESENVOLVIMENTO (3.3)**; **IMPLANTAÇÃO (3.4)** e **INSTITUCIONALIZAÇÃO (3.5)**, devendo ser avaliados ao final (tanto o CMQ-SW como as melhorias da capacitação do processo objeto em si) pela etapa final **AValiação (3.6)**.

O fluxo diagramado às duas páginas seguintes é explicado ao longo deste trabalho. Ele se presta aqui a expandir a visão geral da estrutura do MC2Q-SW e serve de guia a referências futuras mostrando as vinculações dos

⁵ O **DP** (Dossiê de Processo), melhor descrito mais à frente, por agregar tudo o que pertence ao Processo, compõe-se de 5 artefatos básicos: a **GMP** (Guia Mestre de Processos); o **QAP** (Questionário de Avaliação do Processo (ou da Capacitação do Processo)); a **TAP** (Tabela de Avaliação do Processo (ou da Capacitação do Processo)); o **RAP** (Relatório de Avaliação do Processo (ou da Capacitação do Processo)); a **GMCP** (Guia de Melhoria da Capacitação do Processo); e a **TAMP** (Tabela de Avaliação da Melhoria do Processo (ou da Capacitação do Processo))

principais artefatos de Entrada e Saída às Etapas já descritas resumidamente. Tais artefatos são apresentados sucintamente em 4.1.2.

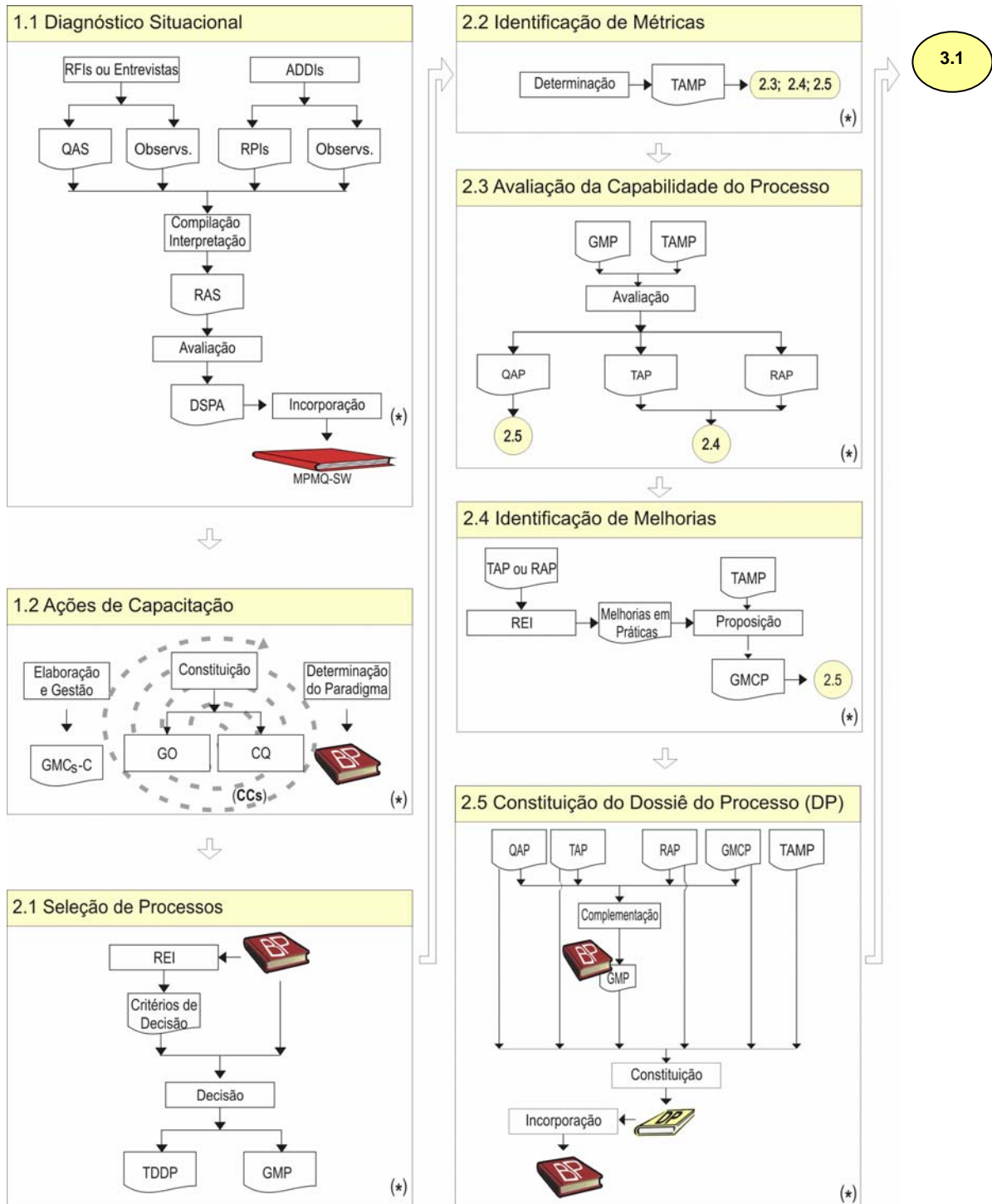


Figura 4.2. Etapas 1.1 a 2.5 do MC2Q-SW

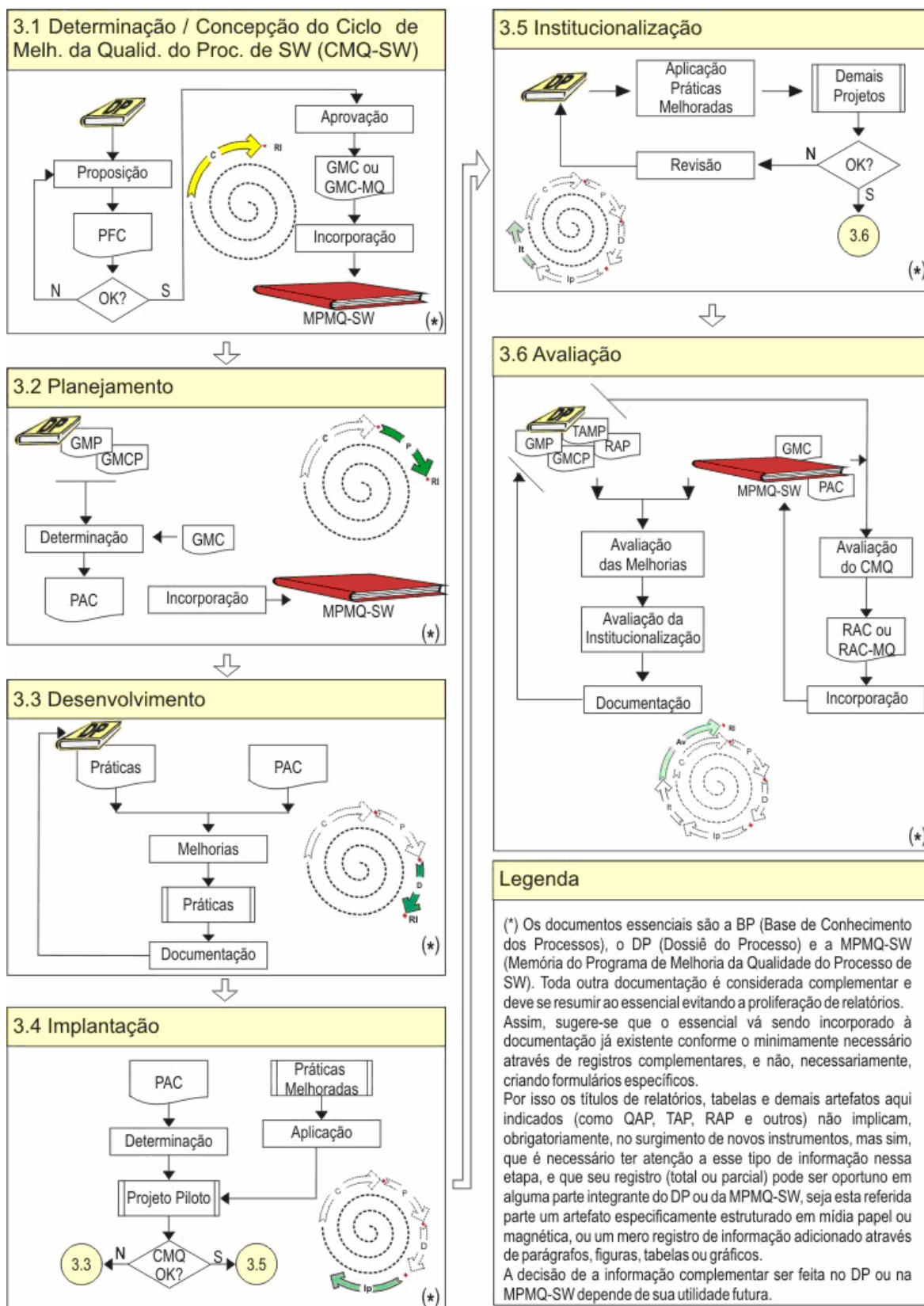


Figura 4.3. Etapas 3.1 a 3.6 do MC2Q-SW

4.1.2. Os Artefatos do MC2Q-SW

O método, conforme princípios gerais de racionalização, descritos no item 4.4 deste trabalho, evita a proliferação de artefatos, a maioria documentos de gestão em mídia papel ou magnética, e, quando sua necessidade é recomendável, ilustra-os apenas de forma orientadora ou sugestiva, cabendo a cada organização os utilizar ou não, ou, ainda, adaptá-los conforme sua conveniência ou circunstâncias. A existência mínima de artefatos decorre da necessidade de algum controle formal que favoreça o sucesso de **Programas de Melhoria da Qualidade do Processo de Software (PMQ-PSW, ou, simplesmente, PMQ-SW)**. Isso visa possibilitar que a informalidade e as iniciativas pessoais não excedam os limites a partir dos quais se tornem impróprios à organização, sem, necessariamente, desestimular os colaboradores ou inibir sua criatividade.

A razão de se ilustrar e identificar artefatos, a cada passo considerado notório, durante um processo de Melhoria da Qualidade de Processos de SW é a de que eles encerram a evolução de uma idéia, ou de uma decisão, enfim, encerram informações que são necessárias à continuidade do processo de melhoria.

Desta forma, as ETAPAS combinadas aos ARTEFATOS propiciam um “*check-list*” que visa assegurar que não se deixem de lado as considerações essenciais à melhoria do processo de *software*.

Em atenção aos princípios da ESSENCIALIDADE, SIMPLICIDADE e ECONOMICIDADE, conforme as características e situação de cada ciclo de melhoria da qualidade combinadamente ao estágio evolutivo da Organização e circunstâncias particulares a cada caso, os artefatos podem até serem suprimidos, desde que haja consciência de que isso não venha implicar no aumento da “Informalidade indesejável”.

Uma forma de se minimizar o risco da “Informalidade Indesejável” é estabelecer mínimos controles, ou mínimas obrigações. Assim o MCQ2-SW estabelece que os artefatos a serem obrigatoriamente mantidos são a **BP**

(Base de Conhecimento dos Processos), o DP (Dossiê dos Processos) e a MPMQ-SW ⁶(Memória do Programa de Melhoria da Qualidade do Processo de Software). Toda outra documentação é considerada complementar e deve se resumir ao essencial evitando a proliferação de relatórios.

Tudo isto ora exposto fica claro, por exemplo, na pesquisa-ação conduzida junto à empresa estudo de caso descrita no item 6 deste trabalho.

O MC2Q-SW sugere que o essencial vá sendo incorporado à documentação já existente, conforme o minimamente necessário através de registros complementares, e não, necessariamente, criando formulários específicos, respeitando as circunstâncias de cada Organização em seu aprendizado em relação à Melhoria da Qualidade do Processo de SW.

Isto posto, embora se mencionem diversos artefatos por títulos (como QAP, TAP, RAP⁷ indicados na Etapa 2.3), não se deve imaginar necessária uma documentação exaustiva a respeito, porém, recomenda-se, ao menos, uma síntese que registre toda informação essencial à melhoria do processo no Dossiê do Processo. Por exemplo, a organização pode eleger um artefato guia a cada processo que titularemos por **GMP (Guia Mestre do Processo) como sendo o único instrumento a compor o Dossiê do Processo**, em adição às descrições ISO ou CMM relativas ao processo e suas práticas que se tornaram parte integrantes do DP.

Por isso, os títulos de relatórios, tabelas e demais artefatos aqui indicados (como QAP, TAP, RAP (etapa 2.3) e outros) não implicam, obrigatoriamente, no surgimento de novos instrumentos, mas sim, que é necessário ter atenção a esse tipo de informação nessa etapa, e que seu registro (total ou parcial) pode ser oportuno em alguma parte integrante do DP ou da MPMQ-SW, seja esta referida parte um artefato especificamente

⁶ A MPMQ-SW pode ser eventualmente referenciada por apenas MPMQ.

⁷ O DP descrito em nota de rodapé anterior é, em síntese, a descrição do PROCESSO adicionada das melhorias às práticas incorporadas através do tempo, sendo que os artefatos que o compõem são simples referências a conteúdos que devem ser considerados a compor um instrumento de descrição do PROCESSO e de suas PRÁTICAS. Assim, no exemplo comentado, QAP, TAP, RAP podem ser apenas inserções de parágrafos, tabelas, diagramas ao texto original tomado emprestado da Base de Processos (ISO ou CMM, por exemplo) que foi eleita como paradigma por ocasião das Ações de Capacitação.

estruturado em mídia papel ou magnética, ou um mero registro de informação adicionado através de parágrafos, figuras, tabelas ou gráficos.

A decisão de a informação complementar ser feita no DP ou na MPMQ-SW depende de sua utilidade futura.

Adicionalmente alguns artefatos têm a característica de serem incorporados a outros à medida que se evolui nas etapas de melhoria da qualidade do processo de *software*, exemplo mais notório é o da PFC (Proposta Formal de Ciclo), que se transforma completamente na GMC (Guia Mestre de Ciclo) tão logo aquela tenha sido aprovada, passando ou não por sucessivas adaptações exigidas pelo CQ (Comitê de Qualidade).

A facilidade de se trabalhar com informações (texto e imagem) em meio magnético permite que a evolução dos artefatos ocorra sem a proliferação desnecessária de instrumentos a abarrotar pastas ou arquivos (magnéticos ou não). Este é o caso dos RAS (Relatórios de Avaliação da Situação), que, residentes em *winchester*, acabam por desaparecer tão logo tenham se tornado parte do DSPA (Diagnóstico Situacional e Proposta de Ações).

Desta forma, a indicação de um artefato em um determinado ponto do MC2Q-SW é uma forma de alerta de que, em determinado instante no roteiro de melhoramento do processo, uma prática deve ser exercitada no sentido de cumprir uma ou mais ações idealizadas⁸ àquele artefato que facilitarão as práticas de melhoria que surgirão a seguir.

Portanto, a idéia de lançar mão de artefatos é uma forma de esclarecer melhor o roteiro indicado pelo MC2Q-SW, através da qual engenheiros de sistemas possam entender mais facilmente como promover melhorias em processos de forma organizada e produtiva. No entanto, qualquer melhoria em processo pode ser feita sem gerar, sequer, um relatório ou documentação! Prova disto é que engenheiros de sistemas experimentados praticam, vez ou outra, melhorias em processos dessa forma!

⁸ A **IDEALIZAÇÃO** é entendida como a idéia que norteia o artefato ao fim que ele se prestaria, pois, em determinadas situações, basta uma reunião ou uma iniciativa, que, uma vez verbalmente exposta, aprovam ou documentam o “passo” avançado em direção à melhoria da qualidade do processo. Tal idéia reforça a não obrigatoriedade de se criar ou preencher um determinado artefato que não seja um dos três já identificados como minimamente obrigatórios.

Na seqüência apresenta-se a denominação de cada artefato e de algumas práticas recomendadas, segundo a ordem de surgimento no método com uma explicação sucinta a seu respeito (sua idealização).

1a. REF	Quadro 4.1. Denominação e Idealização dos Artefatos ou Práticas Citadas (1a REF é a indicação da ETAPA onde o artefato ou prática é referenciado pela 1a. vez.)
1.1	RFIs - Reuniões Formais de Investigação Reuniões planejadas e conduzidas por alguém com experiência e habilidades na condução de reuniões com o objetivo de eliciar aspectos importantes à implantação de um PMQ-SW. Documentada através de Ata.
1.1	QAS - Questionário de Avaliação da Situação Questionário que direciona o foco da atenção a determinada apuração de informação que se considere fundamental ao entendimento do clima organizacional e à motivação a implantar um PMQ-SW bem sucedido.
1.1	ADDI - Atividades de Desenvolvimento, Diagnóstico e Integração Conjunto de atividades diversas (usualmente informais para estimular maior espontaneidade) de caráter motivacional, de integração, desenvolvimento pessoal e interpessoal, e de diagnose situacional, preferencialmente, apoiadas em apresentação e discussão de filmes apropriados, jogos, dramatizações que simulem situações do dia-a-dia, construção de murais, dentre outras, que permitam revelar indícios inibidos à apresentação a indivíduos estranhos ao grupo.
1.1	RPI - Relatório de Participação Individual Qualquer tipo de registro da participação individual dos que desenvolveram alguma das atividades desenvolvidas no processo de Diagnóstico Situacional.
1.1	RAS - Relatório de Avaliação da Situação Relatório intermediário, ou simples registro de observações colhidas durante as atividades ou reuniões conduzidas na Organização com lembretes ou conclusões que sirvam de esboço ou suporte à elaboração do Diagnóstico Situacional e Proposta de Ação (DSPA).
1.1	Anexo 02 - DSPA - Diagnóstico Situacional e Proposta de Ações Relatório conclusivo da análise situacional, contendo proposição de ações que permitam implantar um PMQ-SW, que deve ser aprovado pela direção da organização.
1.2	GMC-C - Guia Mestre de Ciclo de Capacitação Uma GMC exclusiva de Ciclo e Capacitação.
1.2	CC - Ciclo de Capacitação Ciclo destinado ao aprendizado do conhecimento teórico e prático necessários a implantar um PMQ-SW ou ao provimento de recursos físicos, estruturais e organizacionais fundamentais ao sucesso do PMQ-SW.
1.2	BP - Base dos Processos (congrega todos os DPs) Conjunto do conhecimento atualizado dos Processos de <i>Software</i> , desde sua especificação e características até a descrição das práticas que o constituem.
1.2	GMP - Guia Mestre de Processos Documento integrante do Dossiê de Processo, que contém a descrição, práticas e demais vinculações do Processo em foco. Surge, originariamente, copiado da descrição do Processo constante na Base de Processos (BP) adotada conforme o paradigma escolhido (ISO, ou CMM, ou outro qualquer)
2.1	REIs - Reuniões de Eclosão de Idéias Reuniões "brain-storm" conduzidas sistematicamente com o fim de elencar causas ou itens de atenção a problemas ou situações e fazer surgir alternativas de solução ou de melhorias.
2.1	TDDP - Tab. de Decisão para Determinação dos Processos a Submeter a melhorias Uma tabela que separa os critérios Imprescindíveis dos Critérios Ponderáveis utilizados na Decisão de escolha de um processo
2.2	TAMP - Tabela de Avaliação da Melhoria do Processo (ou da Capacitação do Processo)

	Tabela que demonstra o “grau de utilização apropriada” de cada PRÁTICA no processo, tanto ANTES como APÓS as MELHORIAS terem sido implementadas, em confronto com a utilização ideal dessa prática, estabelecida como sendo 100%.
2.3	QAP - Questionário de Avaliação do Processo (ou da Capacitação do Processo) Questionário complementar aos RAS, caso se avalie sua necessidade, que objetiva avaliar as práticas e identificar as possíveis melhorias em práticas (eliminação, substituição ou criação) pertinentes ao processo.
2.3	TAP - Tabela de Avaliação do Processo (ou da Capacitação do Processo) Qualquer tabela que se julgue necessária a complementar a TAMP para permitir avaliação da Capacitação do Processo
2.3	RAP - Relatório de Avaliação do Processo (ou da Capacitação do Processo) Qualquer relatório adicional aos RAS e TAMP que permita, caso se julgue necessário, melhor avaliação da capacitação do processo.
2.4	GMCP - Guia de Melhoria da Capacitação do Processo Artefato elaborado um a cada processo, designado a compilar as medidas de melhoria em práticas pertinentes ao processo em foco.
2.5	DP - Dossiê de Processo Conjunto de informações que engloba tudo o que diz respeito ao Processo e às melhorias a que se submete, como a GMP, o QAP, a TAP, o RAP, a GMCP, e a TAMP.).
3.1	CMQ-SW - Ciclo de Melhoria da Qualidade da Capacitação do Processo de SW (ou, simplesmente, de Melhoria da Qualidade do Processo de SW) Ciclo destinado ao aperfeiçoamento das práticas do processo que se deseja melhorar.
3.1	PFC - Proposta Formal de Ciclo Artefato resultante da etapa de CONCEPÇÃO “considerado o gatilho” (C) de um CC ou CMQ-SW, que identifica a amplitude, a complexidade e o risco do ciclo, através das ações propostas de melhoria a um processo.
3.1	GMC - Guia Mestre de Ciclo (Pode também indicar ser uma GMC-MQ) Artefato de orientação ao planejamento que formaliza as intenções de melhoria de um processo resultantes de uma Proposta Formal de Ciclo já revisada e aprovada.
3.1	GMC-MQ - Guia Mestre de Ciclo de Melhoria da Qualidade do Processo de SW (ou, simplesmente, de Melhoria do Processo de SW), usualmente referida como GMC. Uma GMC exclusiva de Ciclo de Melhoria da Qualidade do Processo de SW.
3.1	MPMQ - Memória do Programa de Melhoria da Qualidade Conjunto de informações que mantém tudo o que documenta a evolução da melhoria do processo
3.1	RI - Reunião de Inspeção Reunião de avaliação do andamento de alguma atividade
3.2	PAC - Plano de Ação Circular Artefato que consolida, de forma sintética, dados essenciais de planejamento de um Ciclo de Capacitação ou de Melhoria da Qualidade, como: metas, prazos, responsabilidades e força de trabalho.
3.6	RACP - Relatório de Avaliação da Capacitação do Processo Simples relatório ou tabela e anexos que documentem a avaliação da capacitação de um processo.
3.6	RAC - Relatório de Avaliação do Ciclo (Pode também indicar ser um RAC-MQ) Simples relatório que documenta como foi o desenvolvimento do Ciclo.
3.6	RAC-MQ - Relatório de Avaliação do Ciclo de Melhoria da Qualidade (usualmente referido como apenas RAC) Simples relatório que documenta como foi o desenvolvimento do Ciclo de Melhoria da Qualidade

4.2. Histórico do Desenvolvimento e a Evolução do Método

A preocupação em contribuir ao tema Melhoria da Qualidade de *Software*, especialmente Melhoria do Processo de *software*, ou *SPI* (*Software Process Improvement*), deveu-se basicamente a três razões principais:

- a) Os modelos existentes, a exemplo do CMMI e SPICE, são bastante abrangentes e completos, assim, salvo alguma inovação de paradigma, qualquer tentativa de um novo modelo possivelmente pouco viria a acrescentar;
- b) Qualquer novo modelo exigiria um estudo exaustivo pelo menos do CMMI e SPICE como também um longo prazo de desenvolvimento, haja vista que os modelos CMM / CMMI vêm sendo aperfeiçoados há mais de quinze anos;
- c) Por outro lado, pouco se tem desenvolvido em relação a um **método** **(não um framework)** dirigido a empresas de qualquer porte, focando inicialmente, porém, as de pequeno e médio porte, onde é maior a dificuldade de adoção dos modelos consagrados.

Disto se concluiu que bons modelos não faltam, mas sim métodos que, conjugadamente aos modelos, permitissem deles tirar maior proveito. Isto posto, e tendo sido o método delineado, era importante submetê-lo a, pelo menos, uma experimentação.

Com isto em mente, foram estipulados requisitos de seleção (item 5.1) que nortearam a escolha de uma empresa para o Estudo de Caso, que permitiram apontar a **STORE Automação** (item 5.2) como a melhor alternativa a se aplicar o MC2Q-SW, ajustando-o à medida que evoluíamos em sua utilização.

Como primeiro passo de desenvolvimento do método, estabeleceram-se algumas diretrizes que pudessem nortear este trabalho, descritas no item 4.4 a seguir (**Diretrizes e Princípios ao Desenvolvimento do Método**). Tais diretrizes foram discutidas junto à empresa escolhida, e foram fundamentais à aceitação da proposta de pesquisa-ação que resultou no MC2Q-SW, uma vez

que a empresa contava com pouca disponibilidade de recursos a alocar a essa experiência.

4.3. Melhoria da Qualidade do Processo de Software como um Programa Contínuo

A implantação de Programas de Melhoria da Qualidade, inclusive do Processo de Software, passa por estágios de conscientização e desejo até sua efetiva implementação. No entanto, qualquer programa de Melhoria da Qualidade, corre o risco de parar após sua implantação, ou após algum tempo de vigência. Isso ocorre naturalmente nas organizações que não assumem a melhoria da qualidade como uma de suas missões permanentes, ou como parte essencial de sua missão institucional,

Este trabalho de pesquisa distingue que um Programa de Melhoria da Qualidade é CONTÍNUO quando, além de normatizado e implantado, incorpora ou reforça valores imprescindíveis de disciplina e organização na cultura da entidade, de forma que esta passe, permanentemente, a considerá-los como parte de sua existência.



Figura 4.4. Qualidade do Produto e Qualidade do Processo.

A melhoria da qualidade do produto está intrinsecamente associada à melhoria da qualidade do processo, uma vez que produtos são resultantes de

processos. Esquemáticamente, podemos ilustrar o foco da qualidade do produto e do processo através da figura 4.4.

A referida figura induz ao entendimento de que a Qualidade do Produto *Software* está intimamente associada à Qualidade do Processo de *Software*, e em razão direta, ou seja, melhores processos induzem a melhor *software*.

É essa incorporação de valores voltados à melhoria da qualidade que permitirá que a organização se prepare, capacitando-se, e promova melhorias de forma continuada em seus processos de desenvolvimento de *software*. As melhorias devem, ainda, ser institucionalizadas, ou seja, compartilhadas com todos os que se envolvem com o processo melhorado através de uma especificação documentada objetiva que assegure a sobrevivência do processo melhorado e seu contínuo aperfeiçoamento.

Quando os Programas de Melhoria da Qualidade em geral, e em particular os de melhoria da qualidade do processo de *software*, falham na continuidade, podemos chamá-los de temporários, esporádicos, aleatórios, eventuais ou, simplesmente, conclusos, encerrados. Qualquer termo que identifica que o PMQ-SW não seja contínuo, ele denuncia que a melhoria da qualidade não é um valor prioritário na organização, ou, em outras palavras, a organização não se preocupa com a qualidade, pois não atingiu o grau de amadurecimento que lhe permita assumir a melhoria da qualidade como essencial à sua evolução.

Assim, despreocupadamente, organizações podem sobreviver no mundo dos negócios, como qualquer ser vivo sobrevive e evolui, sem se preocupar conscientemente em melhorar a qualidade de seus processos vitais ou de relacionamentos. Pode, inclusive, melhorar a qualidade de seus processos de forma eventual, de forma assistemática, não metódica, ou INFORMAL, no entanto, presume-se que ficará muito mais à mercê das forças do mercado, do que as que já se encontram maduras e encaram a melhoria da qualidade como parte intrínseca à sua evolução, gerenciando-a e não apenas deixando-a às iniciativas informais dos subordinados.

No caminho ao amadurecimento e à incorporação da melhoria da qualidade, faz-se necessário, antes de tudo, um método de trabalho. É o

método que permite a idealização de cenários, a identificação de objetivos, a avaliação da situação, a escolha de caminhos, ferramentas, melhores passos e o momento certo de fazer as coisas que devem ser feitas para se atingir e se manter em cenários idealizados. O método impõe a disciplina e a alocação de recursos adequados, cada qual em seu tempo preciso, favorecendo a tomada de decisão e viabilizando o bom cumprimento da missão.

Assim, doravante, quando este trabalho se referir a PMQ-SW sem nenhuma outra qualificação, entenda-se de que se trata de um **PMQ-SW contínuo**.

4.3.1. A Síndrome da Fachada e O MC2Q-SW

A constatação de que ainda é baixa a adesão de *softwarehouses* brasileiras a **Programas de Melhoria da Qualidade do Processo de Software (PMQs-SW)** apesar de o estado da arte em Qualidade de *Software* estar bastante desenvolvido (NAVAS, 2004, com base em MCT-SEPIN, 2002), com modelos consolidados e abrangentes como o CMMI, SPICE e o recente MPS-Br, pode denunciar que tais empresas ainda não percebiam retornos atrativos⁹ em incorporar tais programas.

Pode-se deduzir que o volume de conhecimento específico a respeito, a profusão de tecnologias e ferramentas, junto com os custos de alocação de recursos próprios e de consultores especializados, associado a prazos longos de desenvolvimento quando se trata de obter certificações, a exemplo do CMMI, traz mais dúvidas aos que têm de tomar decisões sobre qual o melhor caminho a seguir do que se o estado da arte fosse insipiente, com poucas alternativas à disposição.

Afinal, projetos de altos custos e prazos longos envolvem muita discussão e exposição na alta direção até a aprovação e muita cobrança

⁹ O confronto com a realidade do estudo de caso permitiu constatar que a atratividade de um retorno é relativa às dos demais projetos. Assim, um Projeto de Melhoria da Qualidade pode, frequentemente, “perder” em atratividade aos demais projetos (de sistemas, a atender o mercado) da empresa de *software* devido a percepção gerencial a respeito.

depois, e pressupõe-se que ninguém esteja disposto a tais desgastes se não for muito seguro de si e dos resultados.

Talvez o receio de se expor a retornos duvidosos por empreender um PMQ-SW, quando comparado à profusão de ferramentas de pequeno custo relativo que permitem inúmeras e pequenas melhorias pontuais, desestimule executivos a tomar decisão em favor de programas que levem a frameworks como o CMMI ou o SPICE, ainda mais quando há inúmeras empresas bem sucedidas que preferem se valer do talento de seus colaboradores com o mínimo de formalização na melhoria do processo.

O cenário descrito pode levar à constatação de que algumas empresas pratiquem a melhoria do processo de *software* de maneira informal, com pouca documentação, sem muita transparência, através de iniciativas pessoais e isoladas que penalizam a Organização pelo retrabalho; redundância de esforços, não compartilhamento do conhecimento, e com o alto risco de perda, pelo “*turn-over*” ou o “*burn-out*” dos envolvidos, desse conhecimento adquirido.

Esta foi a situação caracterizada neste estudo de caso, à qual se denominou de “**Síndrome da Fachada**” por ilustrar uma situação crítica onde dirigentes e técnicos, ao serem levados a decidir por implantar um PMQ-SW, mascaram suas dúvidas, limitações, receios, ou quaisquer outras intenções.

Também pode ter sido esta uma possível explicação do desinteresse manifestado por algumas outras empresas alternativamente consideradas para se submeter a esta experiência piloto.

Tal situação se evidencia pela não adesão aos modelos e métodos formais, especialmente os reconhecidamente maduros, nem nenhuma outra ação concreta em benefício da melhoria da qualidade, embora com os executivos demonstrando seu engajamento com medidas de prontidão virtual, de incentivo a grupos de estudo ou discussão e eterno reconhecimento da importância da melhoria contínua da qualidade do processo de *software*.

É inconsistente como os discursos de grande parte dos políticos em campanha eleitoral, por mostrarem-se corretos nas idéias, mas persistindo em desenvolver sistemas como se tem feito há mais de trinta anos: apenas

baseando-se nos valores individuais e experiência pessoal de especialistas em desenvolver *software*, semelhante ao já citado na Introdução (item 1) por Jacobson et al (1999) em seu artigo sobre o “*Unified Process*”.

O MC2Q-SW veio contribuir com a empresa estudo de caso em seu enfrentamento ao dilema de o que fazer em relação à melhoria da qualidade do processo diante do cenário descrito, enquanto não se decidia pela melhor alternativa de modelo. Possibilitou que a empresa experimentasse praticar a melhoria da qualidade do processo sem perder os investimentos que seriam aplicados nesse trajeto inicial de melhoria e, ainda, a um baixo custo e com aplicação imediata.

Em outras palavras o MC2Q-SW serviu de antídoto à Síndrome da Fachada no período em que se praticou o ciclo experimental de melhoria da qualidade do processo de *software*, por oferecer medidas de combate aos problemas, principalmente ao da informalidade, que induzem empresários, gerentes e técnicos a persistirem na atuação improdutiva de desenvolvimento de *software* pela precariedade de seus processos.

Conforme a figura 4.5 a seguir o MC2Q-SW é mostrado nesse contexto, onde sua aplicação sugere três estágios, explicados detalhadamente no item 4.5 (O Modelo Evolutivo dos três Estágios da Organização em direção a um Programa de Melhoria da Qualidade do Processo de *Software*), pelos quais possivelmente algumas Organizações passam quando se propõem a implantar um PMQ-SW.

Resumidamente esses estágios foram denominados como:

- **Estágio I** – Inicial ou Intuitivo;
- **Estágio II** – Orientado;
- **Estágio III** – Orientado e Direcionado.

O **Estágio I** caracteriza-se pela inexistência de um método com a Organização insegura em relação às ações que deve adotar e às tecnologias e metodologias que deve contratar. O **Estágio II** é o da Organização se orienta por um método que lhe permita conduzir PMQs-SW independentemente de *frameworks* ou grupos de tecnologia. O **Estágio III** é o da Organização que

prática metodicamente a melhoria da qualidade do processo de *software* apoiado em algum *framework* consagrado.

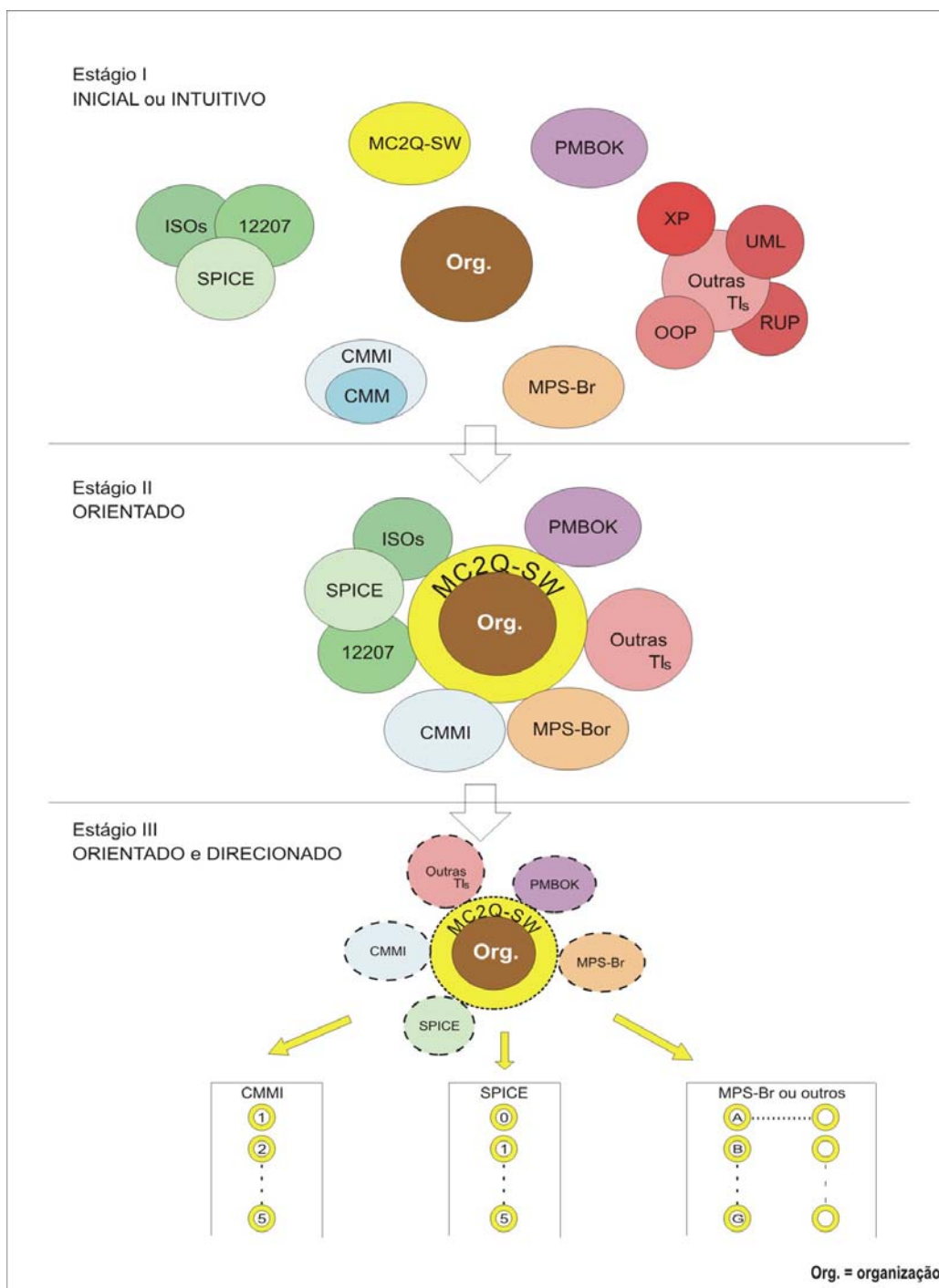


Figura 4.5. O MC2Q-SW e a Formalização da Melhoria do Processo nos 3 Estágios da Organização

A referida figura ilustra que, enquanto no **estágio I** os métodos são apenas mais uma ferramenta ou possibilidade de solução, no **estágio II** o método é a ferramenta essencial. Ele é utilizado de forma a encasular a

Organização imunizando-a contra os efeitos deletérios da informalidade e da dúvida de forma a permitir que prossiga melhorando a capacitação dos processos, com, ao menos, um mínimo de formalismo, organização e gerenciamento que torne essa prática eficaz e eficiente.

Assume-se, portanto, que a determinação de um método de trabalho é o primeiro passo concreto a ser dado em direção à Melhoria da Qualidade do Processo de *Software*, como em qualquer processo de mudança nas Organizações.

4.4. Diretrizes e Princípios ao Desenvolvimento do Método

O MCQ2-SW foi concebido a partir de 6 princípios básicos de racionalização (**princípios racionais** de **ESSENCIALIDADE, SIMPLICIDADE, ECONOMICIDADE, EVOLUCIONARIEDADE, ITERATIVIDADE e GERENCIABILIDADE**) que orientaram seu desenvolvimento e demais princípios compatíveis. Os princípios são auto-explicativos, por isso não se fazem necessárias maiores explicações além da compilação de sua definição e correlações lingüísticas, com breve justificativa. No entanto, são esclarecidos sucintamente os **seis primeiros** enunciados¹⁰, que, de certa forma, regem ou induzem a todos os demais e servem como demonstração essencial, simples e econômica do que pretendemos demonstrar. Esses seis princípios são a essência diretora da elaboração do MC2Q-SW, entendidos como suas diretrizes.

- **Essencialidade** - caráter do que é essencial;
Essencial - que constitui o mais básico ou o mais importante em algo; fundamental; que é necessário, indispensável; a coisa principal; o indispensável;
Sinônimos - ver sinonímia de necessário e principal;
Antônimos - acessório, acidental, adventício, contingente, episódico, secundário; ver tb. antonímia de necessário.

¹⁰ Compilações e adaptações efetuadas com base no dicionário Houaiss <http://houaiss.uol.com.br> em 12.jan.2006

O princípio da **ESSENCIALIDADE** nos leva a considerar apenas o que é essencial, evitando o supérfluo, redundante, secundário e tudo o mais que dispersa esforços e gera custos desnecessários.

- **Simplicidade** - ausência de complicação; ausência de luxo, de pompa, de sofisticação;

Antônimos - afetação, complexidade, ênfase, fasto, pretensão.

O princípio da **SIMPLICIDADE** nos obriga a fazer que tudo o que seja essencial de forma simples, sem excessos, formalismos ou burocracias.

- **Economicidade** – qualidade daquilo que é econômico; relação entre custo e benefício a ser observada;

Econômico - *respeitante a economia; caracterizado pelo uso cauteloso, eficiente e ponderado dos recursos materiais; que controla gastos, que evita desperdícios; que gera economia, que reduz gastos; que custa pouco; barato; que obtém resultados com o mínimo de perdas, erros, dispêndios, tempo;*

Economia - *aproveitamento racional e eficiente de recursos materiais.*

O princípio da **ECONOMICIDADE** nos orienta a fazer o essencial e simples apenas quando for necessário, evitando o desperdício de recursos ou a antecipação de desembolso financeiro ou dispêndio de energia.

Assim, deve-se procura aplicar os princípios da **ESSENCIALIDADE**, **SIMPLICIDADE** e **ECONOMICIDADE** nas palavras, nos textos; na criação de símbolos, formulários e artefatos; na geração de novos procedimentos, modelos, métodos; na tomada de decisões, tanto ao se desenvolver um método como ao aplicá-lo, evitando perda de tempo em explicar o óbvio, com tudo o que não inova gerando ganhos e com tudo o que atrapalha mais do que, realmente, agrega valor.

Tais princípios induzem a aproveitar ao máximo tudo o que já foi criado, também de essencial, simples e econômico, e a afastar tudo o que não se enquadra nesses princípios. A exceção desta orientação fica por conta dos que desejarem ou perceberem que algo sofisticado ou complexo possa ser útil à melhoria da qualidade.

Uma característica natural que se pode observar no desenvolvimento de organismos vivos e também nas organizações, sistemas e artefatos criados pelo homem é o **quarto princípio** que foi adotado como diretriz ao desenvolvimento do método: a **EVOLUCIONARIEDADE**.¹¹ Pois a melhoria dos processos, para ocorrer naturalmente, acontece de forma gradativa. Melhorias, uma ou algumas de cada vez, usualmente (ou pelo menos, inicialmente) de pequeno impacto, de forma a não causar tumulto, ônus exagerado, estresse negativo, resistências, arrependimentos ou prejuízos. É exatamente o contrário da revolução, onde grandes avanços podem ser feitos, comumente acarretando altos custos ou conseqüências indesejáveis. Não que a evolucionariedade não possa levar a uma maior rapidez e amplitude de melhorias, que assim podem ocorrer caso a organização precise promovê-las por algum motivo imperioso ou acredite haver evoluído o suficiente a ponto de neutralizar ou compensar os transtornos disso decorrentes.

- **Evolucionariedade** – qualidade daquilo que é evolucionário, que evolui no decorrer do tempo;

Evolucionismo - qualquer teoria que explique a evolução ao longo do tempo;

Evolucionário - concernente à evolução ou evoluções;

Evolução – ato, processo ou efeito de evoluir / série de movimentos desenvolvidos contínua e regularmente, geralmente completando um ciclo harmonioso / todo processo de desenvolvimento e aperfeiçoamento de um saber, de uma ciência / processo gradativo, progressivo de transformação, de mudança de estado ou condição; progresso / processo através do qual algo se modifica ao longo do tempo.

¹¹ O princípio da *evolucionariedade*, combinado com os demais princípios em discussão, permitirá que, em próximas edições, um *software*, ou um livro, torne-se melhor. Por exemplo, sua combinação com o princípio da *praticidade*, determinará o quanto mais se deve investir em melhorar algo que já foi produzido, e, com o da *oportunidade*, se devem ou não ser desenvolvidas novas funções ou descritos mais detalhes de algo. Assim este conjunto de princípios norteia, naturalmente, o caminho à eficácia e à eficiência da elaboração de um trabalho. O pesquisador aproveita ainda para mostrar que, na prática, estamos repletos de exemplos de produtos eficazes liberados com muita crítica mas que serviram a aprimorar cada vez mais a indústria de *software* e a fortalecer seu mercado, como as notórias versões de sistema operacionais de microcomputadores desde a década de 90 até o presente.

Assim, o princípio da **EVOLUCIONARIEDADE** ampara a implantação de versões de processos na medida em que novas melhorias evidenciem-se realmente oportunas ou necessárias, sem a necessidade de abordagens radicais que impliquem em altos custos ou tumulto na organização pela antecipação desnecessária ou exagerada de trabalho.

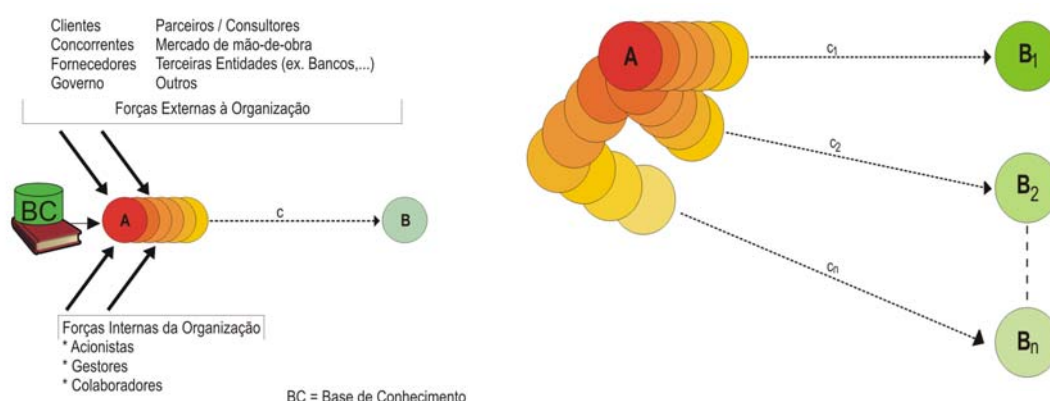


Figura 4.6. Evolução na qualidade de um processo

Figura 4.7. Esquema bidimensional das “n” possibilidades de caminhos de evolução de um processo

A Figura 4.6 ilustra forças que influem nas melhorias dos processos e a Figura 4.7 ilustra que são possíveis inúmeros diferentes caminhos a se chegar a processos eficazes, sugerindo 2 observações: (1) dentre os n processos eficazes, um pode ser mais eficiente que os demais (onde se assume que B1 seja mais eficiente que B2, que, por sua vez é mais eficiente que todos os n outros processo imaginados); e (2) dentre os n caminhos a se atingir um processo eficaz, um pode ser mais eficiente que os demais por evitar o retrabalho, a redundância de esforços, o ensaio-erro, e tantas outras ineficiências decorrentes do desconhecimento, da falta de experiência ou do mau gerenciamento (onde se assume que c1 seja mais eficiente que c2, que, por sua vez é mais eficiente que todos os n outros caminhos evolutivos de melhoria imaginados).

O quinto princípio que rege o MC2Q-SW é o da **ITERATIVIDADE**, pois o caráter de revisões periódicas favorece o aprendizado, o amadurecimento, o planejamento e permite calibrar a velocidade e a amplitude das melhorias.

- **Iteratividade** – qualidade ou condição do que é iterativo;

Iterativo - relativo a iteração / repetido, reiterado, feito mais de uma vez; freqüente.

O sexto princípio é o da **GERENCIABILIDADE**:

- **Gerenciabilidade** – qualidade ou condição do que é gerenciável, sujeito à gestão, gerência, administração;

Gerenciar - dirigir (empresa, negócio, serviço) na condição de gerente; administrar, gerir, exercer gerência sobre; administrar, dirigir, encarregar-se de, cuidar de um negócio;

Sinonímia: confrontar com **conduzir**.

Este princípio é também muito importante, pois visa, principalmente, garantir 3 objetivos:

- **Evolucionariedade não aleatória**, planejada, de forma a tornar mais eficiente o caminho evolutivo de melhorias em direção ao processo eficaz que possa ser o mais eficiente, coibindo a evolução errática, ou fora de direção, ou improdutiva, ou redundante, demonstrada pelo caminho ineficiente **cn**, e pouco eficiente do caminho **c2** (Figura 4.7).
- **Avaliação**, pois não se gerencia bem o que não se pode medir;
- **Aprendizado**, pois uma vez que se gerencia um processo de melhorias, é possível aprender e aperfeiçoar-se, melhorando continuamente os caminhos evolutivos, afastando cada vez mais o desperdício da evolução aleatória.

Estes princípios induzem a outros, que por sua clareza de sentido, acredita-se desnecessário discorrer a respeito. Tais princípios, listados a seguir, também nortearam a implantação do método e sugere-se que estejam sempre em mente durante qualquer PMQ-SW: Princípio da **Oportunidade**; da **Racionalidade**; da **Necessidade**; da **Praticidade**; da **Realidade**; da **Notoriedade**; da **Informalidade**; da **Facilidade**; da **Obviedade**; da **Flexibilidade**; da **Adaptabilidade**; da **Aplicabilidade**; da **Factibilidade**; da **Prioridade**; da **Espontaneidade**; da **Criatividade**; da **Instantaneidade**; da

Atratividade; da **Reusabilidade;** da **Escalabilidade;** dentre outros que encerrem as idéias de trabalho (a) eficaz e (b) bastante eficiente.

4.5. O Modelo Evolutivo dos três Estágios da Organização em direção a um Programa de Melhoria da Qualidade do Processo de *Software*

A pesquisa-ação junto à equipe da empresa foco do estudo de caso permitiu perceber uma situação que pode ocorrer com muita freqüência nas organizações conforme um modelo evolutivo de melhoria de *software* esquematizado na Figura 4.8.

Este modelo evolutivo repousa na hipótese de que as organizações passam por três estágios básicos quando se avalia sua evolução em direção à melhoria contínua do processo de *software*. Baseia-se em uma análise não qualitativa, advinda da percepção (do pesquisador e de outros participantes da experiência) a respeito do comportamento dos envolvidos na implantação do MC2Q-SW como um estudo de caso experimental na STORE.

A motivação, ao desenvolver tal modelo, foi buscar uma explicação a respeito da Influência da Informalidade e da Formalidade na Melhoria da Qualidade do Processo de *Software*. Ela surgiu a partir da percepção do pesquisador e de alguns especialistas em desenvolvimento de *software* de que a absoluta Informalidade é, em tese, prejudicial à melhoria da qualidade desse processo, assim como o excesso de formalidade pode, também, sê-lo, como, por exemplo, em situações onde o excesso de formalismo tolha a criatividade.

Para auxiliar na solução desse dilema, formulou-se um princípio de que deve haver um Limite mínimo de Formalidade (**LmF**) e um Limite Máximo de Informalidade (**LMI**) a serem respeitados para que uma formalidade mínima seja obrigatória e para que uma informalidade máxima seja tolerada na Organização.

Assim, estabeleceu-se um modelo de duas curvas em um plano cartesiano que representam as curvas da Formalidade e da Informalidade,

como resultantes, respectivamente, das forças das ações formais e das ações informais relacionadas à melhoria da qualidade dos processos.

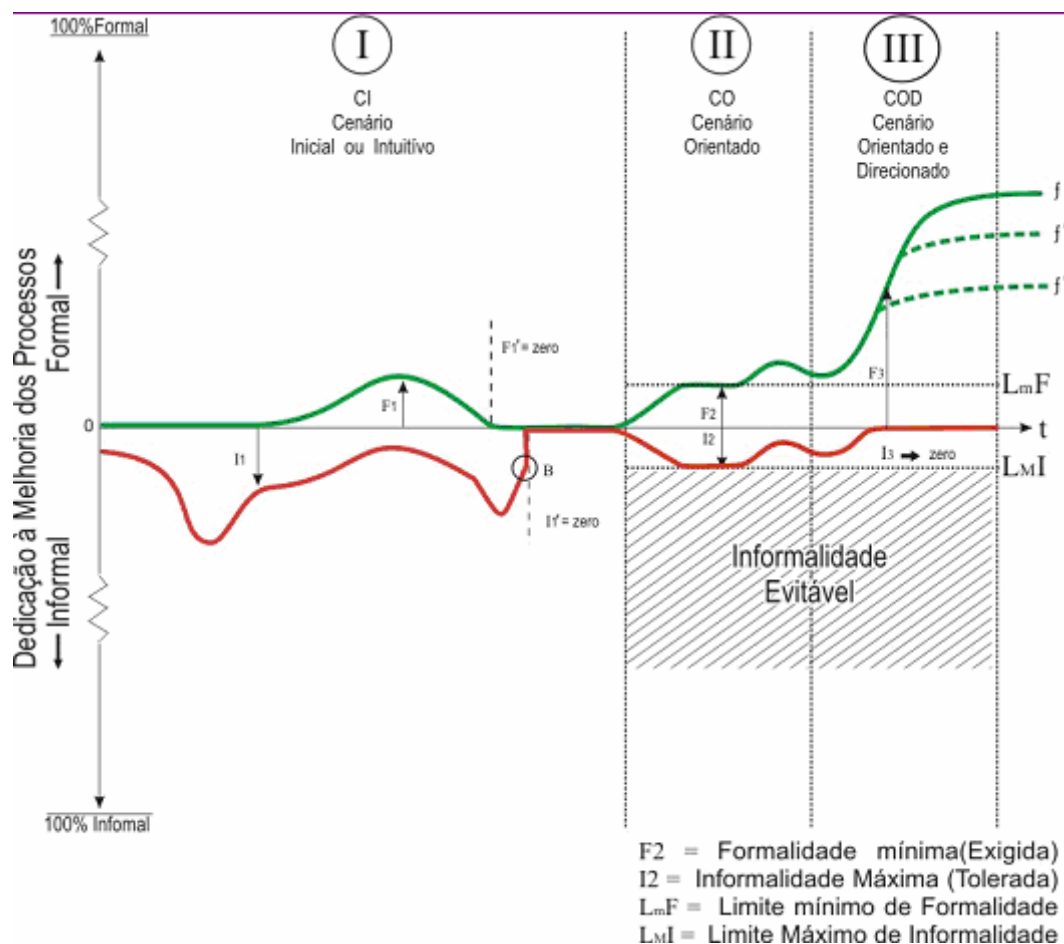


Figura 4.8. O Modelo Evolutivo dos 3 Estágios da Organização em direção à formalização e aos Modelos Maduros de SPI

A curva desenhada na parte **superior** (verde) do gráfico representa a evolução das melhorias implementadas às práticas associadas aos processos através de ações **formais**, gerenciadas, e a curva **inferior** (vermelha) representa a ocorrência de melhorias caracterizadas como **informais**, isto é, decorrentes do interesse e empenho pessoais de gerentes ou técnicos, sem método nem padronização, sem gerenciamento nem divulgação eficaz e eficiente.

As forças que constroem a curva superior foram ilustradas como “**F**” (forças de Ações **F**ormais que induzem a melhorias) e as que resultam na

curva inferior, como “I” (forças de Ações Informais que induzem a melhorias) mostrando qual sua prevalência em cada estágio. Tais forças representam a “intensidade de trabalho” ou a “força de trabalho” alocada a cada instante em que ocorre a melhoria nas práticas dos processos, seja formal ou informalmente, ilustrando diversas combinações.

O gradiente de cada força depende de circunstâncias múltiplas nas Organizações, desnecessárias de serem aqui e agora exploradas e que resultam em pontos máximos e mínimos meramente ilustrativos em ambas as curvas.

Esses estágios já apresentados como: **Estágio I** – Inicial ou Intuitivo; **Estágio II** – Orientado e **Estágio III** – Orientado e Direcionado, servem para explicar como as influências da Informalidade e da Formalidade na melhoria da qualidade dos processos afetam as organizações.

O **Estágio I** caracteriza-se quando as organizações, não tendo nenhum método, se vêem à mercê das múltiplas tecnologias e metodologias com seus modelos, técnicas e ferramentas diversas sendo oferecidas por consultores ou representantes dos fornecedores através de propostas comerciais de variadas formas.

Alguns desses produtos implicam em treinamentos de baixo custo ou custo zero, caso de algumas ferramentas simples de produtividade onde, em muitos casos, se lança mão do auto-aprendizado, outros, no entanto, exigem grande investimento da empresa contratante, seja pela dedicação de seus colaboradores ou pela contratação de consultores especializados por longos períodos de atuação.

As organizações que contam com métodos de trabalho cujos propósitos não sejam direcionados à melhoria da qualidade de *software* e que também se vêem assediadas pelos representantes comerciais dessas variadas soluções sem conseguir decidir com firmeza que caminho adotar também são enquadradas nesse estágio inicial.

O estágio é caracterizado, tendo em vista a melhoria do processo de *software*, pela intuição, personalismo, imediatismo, oportunidade, adoção de

soluções pontuais, ferramentas de pequena abrangência, mistura de conceitos, estudos superficiais, ações descoordenadas de melhoria, baixa integração de ferramentas ou colaboradores ou departamentos, informalidade na execução e evolução dos processos.

Sofre alto risco de retrabalho, redundância de esforços, não compartilhamento do conhecimento, e de perda desse conhecimento adquirido. Podem ocorrer situações onde prevaleça o formalismo (**F**), mas não forçosamente anulando a falta de formalidades (**I**). Qualquer combinação de **F** e **I** pode ocorrer. O ponto (**B**) no gráfico (onde **I** cai abruptamente a zero) simula possivelmente o pior resultado, com risco relativamente alto de ocorrer, para esse cenário: o da interrupção total de ações de melhoria de qualidade do processo de *software* pela demissão de um engenheiro de sistemas que se dedicava, por iniciativa própria, a aperfeiçoar processos, quando a Organização não se preocupava com nenhum formalismo (pois **F** já era zero) nem contava com outros engenheiros envolvidos em melhorias. Neste caso, além da perda do conhecimento adquirido, ainda há o risco de transferi-lo à concorrência pela possibilidade de ela vir a admitir esse ex-funcionário que leva sua experiência adquirida por sua exclusiva iniciativa.

Sinteticamente, este estágio se caracteriza pela falta uma linha-mestra de conduta que organize, capacite, planeje, coordene, integre, avalie, divulgue, uniformize e institucionalize, permanentemente, as melhorias nos processos de *software*.

O **Estágio II** indica a Organização que conta com um método que permita conduzir PMQs-SW independentemente de se concentrar em determinado *framework* ou grupo de tecnologia. Denominou-se por ORIENTADO em função de o método orientar o caminho da melhoria dos processos, independentemente da situação da Organização. No caso do MC2Q-SW, o método se propõe a envolver a Organização através de duas formas: (a) protegendo-a da procrastinação e da precipitação; (b) aproximando-a dos modelos e demais tecnologias, enquanto ela se capacita e atua em melhorias graduais de processos de *software* até decidir que direção assumir em relação a modelos maduros. As melhorias em seus processos passam a

ocorrer de forma organizada e formal, desestimulando a informalidade e o personalismo, pois o MC2Q-SW orienta ações de melhoria de forma **gerenciada**, ou “**PADIMC**” (Planejada; Acompanhada; Documentada; Institucionalizada; Medida e Continuada).

É neste estágio que se impõem limites de Formalidade e de Informalidade. O primeiro, **Limite mínimo de Formalidade (LmF)**, visa garantir que mínimos procedimentos gerenciais sejam praticados, tal como uma reunião periódica com todos os envolvidos no desenvolvimento de sistemas para avaliar como estão sendo executados os processos e que melhorias estão sendo necessárias, se houve alguma implementação em melhoria e se há necessidade ou intenção de planejar-se ou efetivar-se alguma nova melhoria. O segundo, **Limite Máximo de Informalidade (LMI)**, estabelece o limite a partir do qual seja desestimulante qualquer um praticar melhorias sem o conhecimento ou autorização do GO e CQ.

Políticas de estímulo e desestímulo tais como premiações, promoções e punições, sempre com bom senso, devem ser estudadas de forma a tornar o ambiente mais propício fortalecer a formalidade e eliminar a informalidade.

O **Estágio III** é a evolução natural do **Estágio II**, onde, já tendo optado por um *framework* consagrado (a exemplo do CMMI ou SPICE ou MPS-Br), o método MC2Q-SW ou equivalente, continua orientando, de forma “**PADIMC**”, a melhoria dos processos de *software* da Organização - através de seus Ciclos de Melhoria do Processo de SW - porém, fortemente baseado nas estruturas e práticas adotadas pelo *framework* escolhido.

É no **Estágio III** que se enquadram as Organizações que, metodicamente, praticam a Melhoria da Qualidade com o objetivo de obter certificação de alguma entidade Certificadora da Qualidade do Processo de *Software*. O prolongamento da curva superior (verde) conforme as tendências f , f' e f'' ilustra que podem ser diversas as possibilidades de alternativas de comportamento da curva da FORMALIDADE segundo a intensidade do formalismo exigido às ações de melhoria e que a tendência da curva da INFORMALIDADE deveria ser tornar-se praticamente nula.

A Síndrome da Fachada associada ao O Modelo Evolutivo dos 3 Estágios da Organização enfatiza a importância da função gerencial e as repercussões que Programas de Melhoria da Qualidade do Processo causam no corpo gerencial de uma Organização, sugerindo que a falha em atingir metas decorre de, e denuncia, fraco gerenciamento.

4.6. O Diagnóstico Situacional (Etapa 1.1) e a Proposição de Ações

Um dos pressupostos do MC2Q-SW é que o que determinará o comportamento de uma organização em relação à qualidade é a sua maturidade ***combinada*** à sua predisposição a esse fim. Assim é importante avaliar previamente essa maturidade (que engloba o conhecimento específico, o domínio de técnicas e o preparo da organização) junto com o clima organizacional e motivacional a esse objetivo, pelo menos para que não se frustrem expectativas quanto aos resultados esperados em relação à efetiva melhoria da qualidade dos processos.

O MCQ2-SW assume que, havendo indícios suficientes de que a empresa se enquadre em determinada situação, não é necessário despende muito esforço para validar essa posição através de análises exaustivas, principalmente quando ocorre o consenso ou a evidência notória de que a empresa se encontra nesse estágio. Postura ainda mais sensata se o referido nível for equivalente ao INICIAL preconizado pelo CMM/CMMI, aliás, exatamente a situação de nosso caso estudado.

Referências à origem do CMMI/CMM, como constata Paulk et al. (1995), já indicam que a idéia é essa, de fato, quando afirma, por exemplo, que “um método de avaliação baseado no CMM: a) fornece a estrutura básica para a investigação de processos; e b) ***permite um rápido e consistente desenvolvimento das constatações que apontam os pontos fortes e fracos da organização*** (grifo nosso)”.

O rigor acadêmico exige que se determine precisamente o que significariam "indícios suficientes", "consenso" e "evidência notória", com métricas adequadas a avaliar essa assertiva. Porém, é importante realçar que o valor econômico da informação e seu benefício confrontado ao custo exigido para sua obtenção tornam a aplicação exaustiva de métodos formais de avaliação da maturidade da empresa restritos às situações onde, de fato, seja importante esse rigor como, por exemplo, previamente à submissão da empresa a alguma certificação.

Para o caso de empresas que desejam confirmar seu enquadramento além do nível 1, há diversas possibilidades de avaliação, previstas no universo de ferramentas do CMM/CMMI.

Assim, o foco da avaliação da Organização pelo MC2Q-SW é da sua situação diante do desafio de implantar um PMQ-SW, principalmente porque se partiu do seguinte princípio: como boa parte das pequenas e médias empresas enquadra-se no Nível Elementar de Maturidade, e se as que não se enquadram nesse nível estão preocupadas em melhorar a qualidade, então a avaliação de maturidade deve se concentrar na identificação das deficiências, dos temores e dos obstáculos que possam ameaçar o sucesso do PMQ-SW. Isto leva a que se tomem medidas de capacitação e outras decisões que neutralizem essas ameaças, deficiências e temores.

Portanto este método sugere que se promova uma avaliação do clima organizacional, das deficiências, da predisposição e do estado de prontidão a implantar um PMQ-SW, uma vez que programas desta ordem implicam em mudança atitudinal, organizacional e em sobrecarga de trabalho, circunstâncias estas que usualmente provocam temores e tendem a produzir resistências.

Foi pressuposto também que o processo de diagnóstico sugerido (descrito no item seguinte) estimula o compartilhamento do conhecimento a respeito dos *frameworks* disponíveis no mercado; a uniformização de conceitos; a conscientização da importância da preocupação com a qualidade e o estabelecimento da prontidão adequada aos gestores e demais colaboradores da organização em direção ao PMQ-SW. Tudo isso propiciado

pela intensa interação dos colaboradores da organização (dirigentes ou não) provocada pela indução das atividades conduzidas de diagnóstico.

4.6.1. Uma Metodologia Prática de Diagnose Situacional

O MC2Q-SW, adota uma metodologia bastante simples de diagnóstico da situação baseada na consolidação de evidências e na interpretação de indícios após a confrontação da percepção de gerentes da Organização em **Reuniões Formais de Investigação (RFIs)** ou entrevistas (com ou sem questionários), com a percepção dos colaboradores lotados na Área de Desenvolvimento de Sistemas, engenheiros de sistemas ou não, em **Atividades de Desenvolvimento, Diagnóstico e Integração (ADDIs)**, usualmente informais, em resposta à necessidade de estimular maior espontaneidade.

Sugere-se que para resultar um bom diagnóstico, ele seja conduzido por alguém com experiência em condução de reuniões e atividades em grupo, com habilidade de comunicação e bons conhecimentos de Engenharia de Sistemas e nas técnicas de aplicação de brain-storm e disposto a praticar a Análise de Conteúdo conforme indica BARDIN (2000). Nesse papel, o condutor deve ainda atuar como um Facilitador que conduza Reuniões de Eclosão de Idéias (**REIs**) com o fim de estimular o afloramento, a coleta e a comparação e consistência desses indícios.

Nesta etapa é recomendável a aplicação individual de questionários que explorem os aspectos que possam identificar riscos ou problemas que causem descontinuidade ao PMQ-SW e documente a avaliação de cada participante.

Complementarmente, as ADDIs são um conjunto de atividades diversas de caráter motivacional, de integração, desenvolvimento pessoal e interpessoal, e de diagnose situacional, preferencialmente aplicadas a intervalos semanais, em um período sugerido de um mês, para o estreitamento e fortalecimento de das relações de convívio entre os participantes. Visam também quebrar resistências interpessoais, inclusive a consultores externos, apoiadas em apresentação e discussão de filmes apropriados, jogos, dramatizações que simulem situações do dia-a-dia, construção de murais, dentre outras, que

permitam revelar indícios gerais inibidos à apresentação a indivíduos estranhos ao grupo.

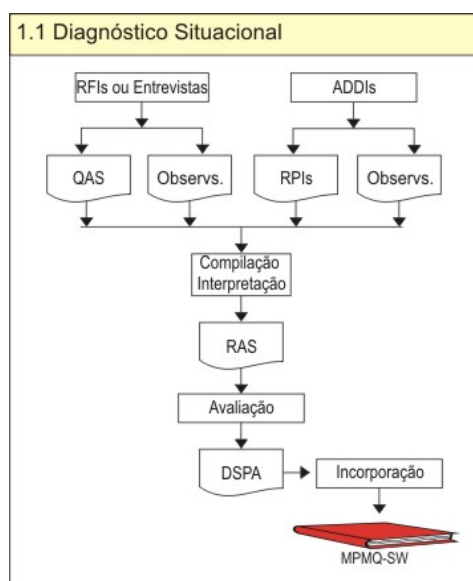


Figura 4.9. O Diagnóstico Situacional - (Etapa 1.1) e a Proposição de Ações

A cultura de um grupo de *software* revela-se pelo comportamento dos indivíduos, grupos e gerentes, além de características gerais da organização (WIEGERS, 1996), assim a maior atenção deve ser dada a essas características obtidas a partir dos indícios coletados, pois a conduta a ser estabelecida no intuito de tornar bem sucedida a implantação do PMQ-SW dependerá dessa avaliação.

ROCHA et al. (2001, pág. 2) citando Fuggetta, afirmam ainda que “a definição e o uso de processos de software envolve a complexa inter-relação de fatores organizacionais, culturais, tecnológicos e econômicos” e SOMMERVILLE (2003, p. 115) enquadra essa técnica de observação cotidiana na etnografia, que diferencia o trabalho suposto do trabalho real que é desenvolvido por uma organização, evidenciando os fatores sociais e organizacionais que influem na situação.

Complementando, Wilson, citado por LUDKE e ANDRÉ (1986), afirma que, ao procurar descobrir as estruturas de significado dos participantes nas diversas formas em que são expressas, os tipos de dados relevantes em uma pesquisa etnográfica são, dentre outros: forma e conteúdo da interação verbal dos participantes; comportamento não verbal e padrões de ação e não ação.

Assim, revelam-se aspectos mais fundamentais e profundos da realidade a ser estudada, contribuindo para que um maior número de variáveis que afetam o caso possa ser considerado.

A análise de conteúdo é um conjunto de técnicas de análise das comunicações visando obter, por procedimentos sistemáticos e objetivos de

descrição do conteúdo das mensagens, indicadores (quantitativos ou qualitativos) que permitam a inferência de conhecimentos relativos às condições de produção/recepção dessas mensagens (BARDIN, 2000, pág. 42).

Por tratar-se de um exercício de interpretação, recomenda-se que as atividades relacionadas ao diagnóstico, contem ainda com a participação de um observador, trabalhando em dupla com o condutor, com os mesmos atributos já sugeridos. Isto permite que, além de apoiar as atividades, o observador registre as ocorrências pertinentes, e auxilie na elaboração de Relatórios de Avaliação da Situação (**RAS**) da organização.

Um importante elemento da análise de conteúdo é descobrir os núcleos de sentido, que compõem a comunicação e cuja presença, ou frequência de aparição, pode ter um significado para o objetivo a ser atingido explicitando sentimentos e impressões que nem sempre se manifestam claramente (BARDIN, 2000, p.105). Os núcleos de sentido são referenciais que se relacionam diretamente aos elementos que se quer explicitar, como categorias de análise elaboradas a partir dos objetivos da pesquisa. No caso em estudo os elementos organizacionais que permitiram conhecer o contexto de implementação do modelo de qualidade formaram a base dos núcleos de sentido. Como exemplo pode-se citar: o conceito compartilhado de qualidade; os padrões de comunicação e de relacionamento diversos (como chefia-subordinados; suporte-cliente; analistas (de negócios) - analistas de desenvolvimento; analistas de desenvolvimento - programadores; analistas (diversos) – cliente; colaboradores em geral (entre si)); a concepção de autonomia no trabalho, a concepção de criatividade no trabalho, dentre outros.

Um relatório conclusivo a respeito desta avaliação, O **Diagnóstico Situacional e Proposta de Ação (DSPA)**, deve ser gerado e aprovado pela organização, cuja elaboração resulta das atividades esquematizadas na Figura 4.10 a seguir.

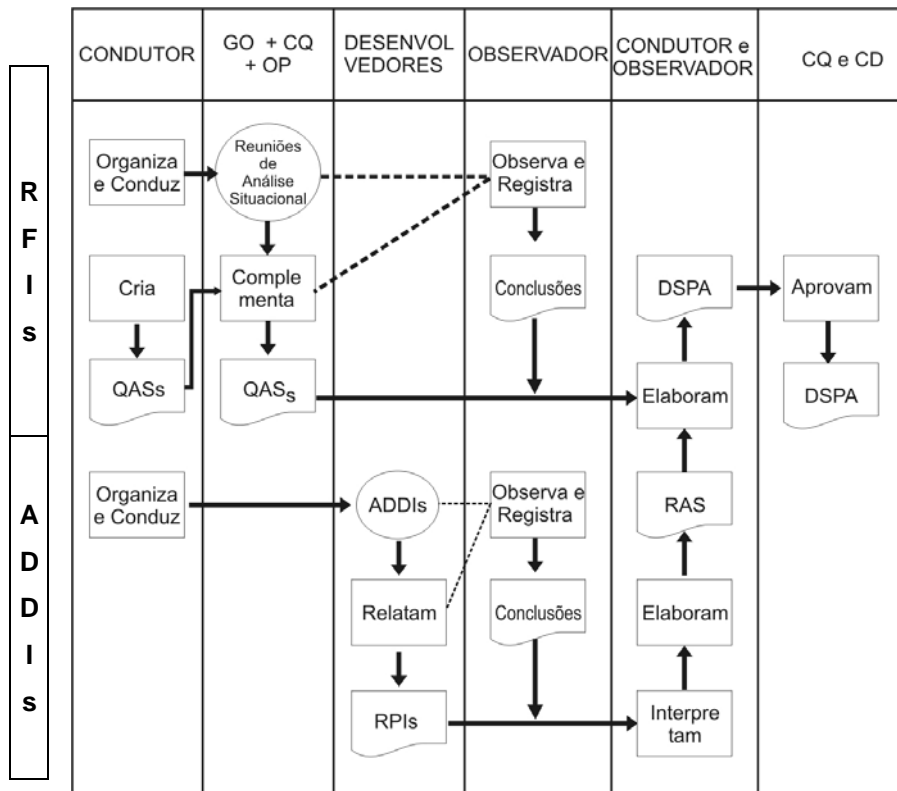
Através do relacionamento individual e em grupo, formal e informal, com o objetivo de **elicit**¹², apoiado em questionamento oral ou impresso e conduzido em ocasiões distintas, que permitam refinar, filtrar e interpretar as informações coletadas, é possível obter informações significativas a respeito da maturidade, cultura e clima organizacional da empresa.

O risco de que este tipo de elicitação possa conduzir a opiniões pessoais subjetivas sem fundamento e deixar de apontar algo muito importante é praticamente nulo, pois, além de colocar em evidência visões às vezes antagônicas que ajudam a compreender o clima organizacional dos diversos grupos ou “células” da Organização (exemplo: “corpo de dirigentes vs corpo de funcionários”; “acionista(s) vs executivos”; “grupo técnico vs equipe de vendas”; “equipe de desenvolvimento vs equipe de manutenção”, dentre outros), o MC2Q-SW recomenda que se observem, ao menos, as regras que seguem:

- as atividades devem ser conduzidas e acompanhadas, no mínimo, por dois observadores experientes (ao menos um conduzindo e outro observando);
- os relatórios de avaliação de situação / diagnóstico / proposta de ação devem ser elaborados conjuntamente pelo condutor e observador(es);
- os relatórios devem ser criticados por, pelo menos, 2 colaboradores (um representante dos dirigentes e outro representante dos funcionários);
- o relatório DSPA deve ser submetido a um referendo pelos dirigentes da empresa.

É oportuno enfatizar que a Etapa 1.1 de Diagnóstico da Situação propicia a integração de todos os que devem se envolver com a melhoria da qualidade do processo de *software*, constituindo-se, por sua característica sócio-integradora, como a atividade básica fundamental de lançamento do PMQ-SW. Recomenda-se ainda que seja aproveitada como motivação e oportuno passo essencial ao aumento da maturidade da organização em relação à Qualidade.

¹² **elicit**: termo adaptado do Inglês *elicit* que significa compreender, deduzir, concluir, adquirir, entender, trazer à tona (a verdade), fazer sair, expulsar, eliciar, extrair, revelar (algo que está latente ou é potencial) ou seja, **muito mais do que simplesmente coletar**.



- RFI = Reunião Formal de Investigação OP = Outros Participantes
- ADDI = Atividade de Desenvolvimento, QAS = Questionário de Avaliação da Situação
- Diagnóstico e Integração
- CD = Comitê Diretor RPI = Relatório de Participação Individual (*)
- CQ = Comitê da Qualidade RAS = Relatório de Avaliação da Situação
- GO = Grupo Organizador DSPA = Relatório de Diagnóstico Situacional e Proposta de Ação (**)

(*) Relatórios ou formulários não necessitam, forçosamente, de leiaute específico, mas apenas, a denominação (TÍTULO) e o conteúdo que documente apropriadamente sua especificidade.

(**) O DSPA é o primeiro relatório formal a constituir a Memória do Programa de Melhoria da Qualidade do Processo de Software (MPMQ-SW).

Figura 4.10. Atividades para o Diagnóstico Situacional e a Proposição de Ações

Esta é uma excelente ocasião para lançar oficialmente o PMQ-SW e iniciar a divulgação periódica do andamento desses trabalhos através de veículos internos, Intranet, Internet e demais mídia para a comunidade de clientes e “prospects”, seja como elemento de marketing interno e externo, seja como de auto-estímulo, incentivo e de consolidação de valores.

Por fim, as atividades desenvolvidas nesta etapa revestem-se da maior importância, pois é graças à qualidade com que se desenvolvem que se pode

obter boa noção da cultura e do cenário situacional da Organização com informações a respeito de obstáculos, fraquezas e temores face ao desafio da mudança inerente à implantação de um PMQ-SW.

Ao ter-se concluído e aprovado o DSPA, este é incorporado à Memória do Programa de Melhoria da Qualidade do Processo de SW (MPMQ-SW) da Organização, como primeiro registro de intervenção organizacional formal que caracterizará o enquadramento da Organização no **Estágio II (ORIENTADO)**.

4.7. As Ações de Capacitação. (Etapa 1.2)

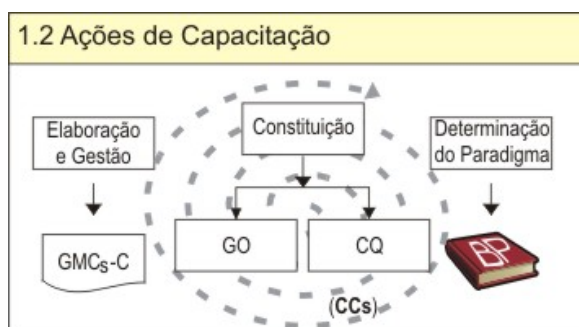


Figura 4.11. As Ações de Capacitação - (Etapa 1.2)

A primeira providência ao se pensar em capacitação, como em qualquer atividade, é organizar-se para o bom atingimento do que se propõe, e é com esse fim que o MC2Q-SW sugere o estabelecimento de uma **GUIA MESTRA DE CICLO DE CAPACITAÇÃO (GMC-C)** (fig.4.31) explicada em 4.13.1.

Dentre as diversas possíveis ações de capacitação destacam-se duas fundamentais, sem as quais torna-se impraticável implantar o MC2Q-SW ou qualquer PMQ-SW. A primeira, de caráter organizacional, é a constituição da estrutura mínima necessária a colocar em andamento o PMQ-SW, e a segunda é a determinação do paradigma a ser utilizado para nortear as práticas em processos.

A operacionalização das ações de capacitação e das posteriores ações de melhoria da qualidade do processo ocorrem em ciclos, conforme se explica a seguir.

4.7.1. A Espiral da Melhoria da Qualidade do Processo

Ao se classificar o MC2Q-SW segundo a natureza dos agrupamentos de atividades (as 13 etapas), percebem-se 2 grandes grupos: um de preparação (Etapas: 1.1; e 2.1 a 2.5) e outro de execução (Etapas: 1.2; e 3.1 a 3.6) (veja figura 4.1).

As etapas de execução, sejam com a finalidade de capacitação, sejam com a finalidade de melhoria direta do processo, desenvolvem-se segundo uma quantidade indeterminada de ciclos com relativa padronização genérica, encadeados seqüencialmente em uma disposição espiral, destinados a promover a capacitação gradual dos envolvidos e a melhoria permanente da qualidade dos processos.

A idéia de ciclos espiralados partiu da observação que abordagens circulares ou iterativas são uma forma natural de se desenvolver e implantar programas de ação (em nosso caso, um Programa de Melhoria da Qualidade do Processo de *Software*), uma vez que revisões constantes ocorrem desencadeando adaptações e decisões que mantêm o programa de ação atualizado e em andamento.

Adicionalmente, cada ciclo deve ser encarado como um projeto, pois os ciclos nada mais são que empreendimentos temporários, cada qual com um objetivo determinado, conforme o caracterizado no *Project Management Body of Knowledge* (PMBOK, 2003). Assim, cada ciclo pode ser também referenciado como um Projeto.

Até mesmo a implantação de um PMQ-SW apoiado pelo MC2Q-SW deve ser encarada como mais um projeto na Organização, cujo custo e benefícios têm de ser demonstrados aos dirigentes, pois estes somente patrocinarão a implantação do PMQ-SW se tiverem consciência de que os benefícios advindos proporcionarão retorno compensador.

Esta é uma questão usualmente delicada, típica da Síndrome da Fachada, onde o discurso dos dirigentes não coincide com a prática, mas se

esta for a situação vigente em determinado caso, isso é facilmente identificado através da elicitação via RFIs e ADDIs cuidadosamente conduzidas na Etapa de Diagnóstico.

Neste caso identifica-se um grave problema potencial à implantação do PMQ-SW, cuja solução somente pode ocorrer se, através de ações de capacitação junto aos dirigentes, conseguir-se obter seu comprometimento ao grande projeto de melhoria permanente da qualidade do processo de *software*.

Cabe ao executivo de Tecnologia assumir o desafio de conduzir a Organização nesse sentido, desde provar seu benefício à alta direção, obtendo seu comprometimento e patrocínio, até a adaptação cultural da entidade, que permita o engajamento com disciplina de todos. Essa condução tem de ser gerenciada, com estabelecimento de metas e avaliação de resultados, como ocorre com qualquer projeto.

O MC2Q-SW vem ao encontro desse desafio, oferecendo um caminho prático e gradual a avançar na melhoria do processo, mesmo antes de se atingir domínio pleno dos *frameworks* disponíveis, através de ciclos espiralados gerenciáveis de capacitação e de melhoria direta da qualidade do processo de *software*. Pode (e deve) ser utilizado pelas Organizações no Estágio I como uma ferramenta que favorecerá, ao menos, o início de uma evolução cultural em direção à disciplina, sem inibir a criatividade.

Uma recomendação bastante pertinente é que os ciclos iniciais sejam projetos-piloto, de escopos reduzidos e facilmente controláveis, com a finalidade de aprendizado e de incorporar o método mais facilmente na organização.

4.7.2. Os Ciclos de Capacitação

Em princípio o PMQ-SW se inicia por Ciclos de Capacitação (CCs), destinados ao menos à mínima capacitação dos envolvidos no conhecimento teórico e prático necessários a implantar um PMQ-SW.

É oportuno deixar claro que a capacitação não se resume à aquisição do conhecimento, mas também incorpora a organização e a estruturação necessárias à viabilização do PMQ-SW em toda a sua magnitude. Assim, um importante Ciclo de Capacitação é o de prover os recursos necessários ao projeto que concretizará o PMQ-SW. É nesse CC que os dirigentes não só determinam papéis e responsabilidades mas também se comprometem a prover os recursos necessários à missão, tais como: treinamentos, equipamentos, instalações, instrumentos, alocação de talentos e especialidades, disponibilização de tempo operacional ou gerencial às tarefas, enfim, tudo o que se fizer necessário à manutenção do PMQ-SW.



Figura 4.12. Duas formas representativas de Ciclos

Os referidos ciclos de capacitação englobam, ainda, todo o trabalho descrito até aqui de integração, desenvolvimento, diagnóstico e avaliação, pois consideramos tais atividades também como capacitação, que, neste caso, é específica de conhecimento da organização e de integração, motivação e potencialização de seus talentos humanos.

Um mínimo controle gerencial deve ser mantido a cada Ciclo de Capacitação. Assim sugere-se a criação de uma **GUIA MESTRA DE CICLO DE CAPACITAÇÃO (GMC-C)** (ver ilustração à figura 4.31) que nada mais é que uma formalização metódica de idéias que oriente o ciclo, e que pode incorporar controles gerenciais conforme as necessidades ou exigências circunstanciais. (Este mesmo instrumento será utilizado nos futuros **Ciclos de Melhoria da Qualidade do Processo de SW (CMQs-SW)**, quando serão melhor detalhados a estrutura dos ciclos e os passos essenciais ao seu desenvolvimento).

Estes ciclos, portanto, antecedem os CMQ-SW propriamente ditos e podem (e devem!) continuar existindo e constituir uma espiral própria ou

paralela de capacitação, ou, ainda, amalgamada à espiral de CMQs-SW, pois a capacitação é também contínua e evolutiva, em função da permanente evolução do conhecimento e de suas aplicações práticas, propiciando que o grau mínimo de capacitação inicial evolua à plena capacitação e possibilite, até, o grau máximo de criação de conhecimento.

A Figura 4.13 mostra, esquematicamente, a possibilidade de múltiplos Ciclos de Capacitação combinados a Ciclos de Melhoria da Qualidade do Processo de SW.

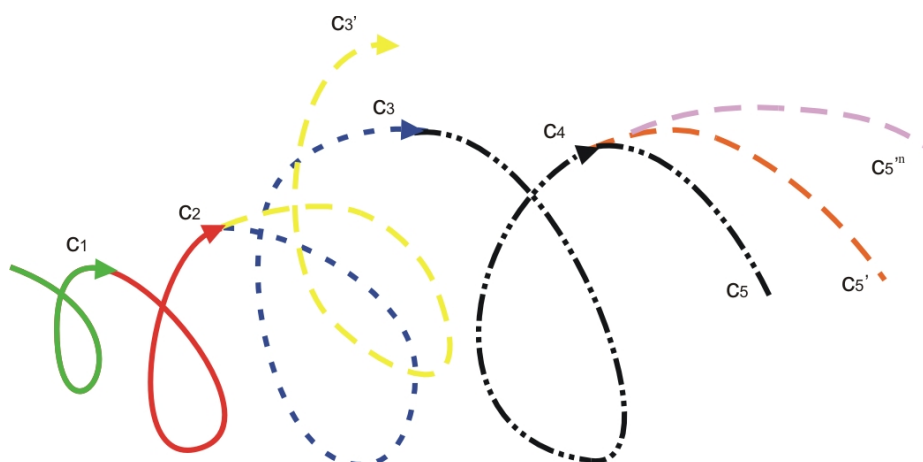


Figura 4.13. A possibilidade de múltiplos Ciclos de Capacitação combinados a Ciclos de Melhoria da Qualidade do Processo de SW

O que fundamenta nossa argumentação de que ciclos diversos (não só de capacitação) podem ser disparados e mantidos é paralelo, é que eles nada mais são que projetos. Portanto, o fator limitante da quantidade de ciclos a administrar é a capacidade gerencial e operacional da organização.

Adicionalmente, é importante ressaltar que um baixo nível de capacitação não deve ser fator inibidor de utilização do método, pois se se estabelecer que é necessário um perfeito domínio dos modelos e das melhores práticas da Engenharia de Sistemas, possivelmente será necessário um volume incômodo e indesejável de investimentos sem o amparo de planos de ação coerentes aos objetivos da organização, além de retardar, mais do que o desejável, o início do PMQ-SW.

O MC2Q-SW apresenta a vantagem de ser um instrumento orientador e disciplinador também da capacitação que ocorre paralelamente ao desenvolvimento dos CMQs-SW.

A partir do instante que se considerar que a Organização atingiu um nível mínimo de capacitação, podem ser iniciados os CMQs-SW, conforme o esquema sugerido na Figura 4.14. Pressupõe-se que quanto maior for o grau de conhecimento dos envolvidos, como também do comprometimento, da maturidade (associada à cultura - de disciplina, de gerenciamento, de resultado, de valorização profissional, de reconhecimento do mérito) da empresa, e da capacidade gerencial dos líderes, mais ousados ou ambiciosos poderão ser estes ciclos, e resultados melhores e mais rápidos deverão ser obtidos.

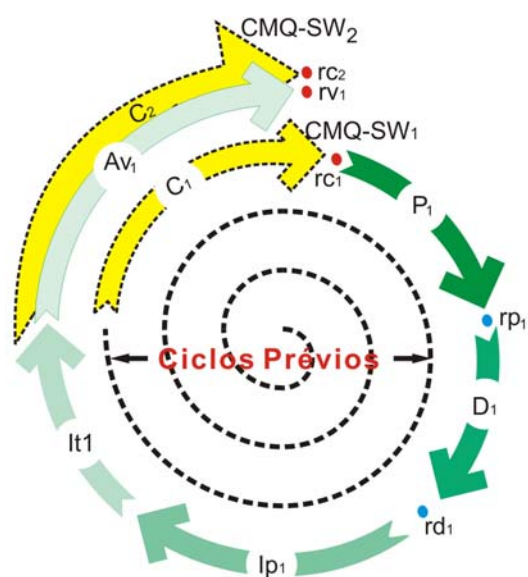


Figura 4.14. Visão esquemática do MC2Q-SW com o primeiro CMQ-SW completo e iniciando o PLANEJAMENTO do 2o. CMQ-SW após os CCs prévios. (ver Legenda na fig.4.30)

Recomenda-se a utilização de algum instrumento de avaliação da capacitação, quando for necessário avaliar a competência de grandes grupos, ou quando a precisão da avaliação seja mais rigorosa. Por exemplo, quando a empresa esteja entrando em fase de INSTITUCIONALIZAÇÃO de medidas de melhoria a processos, ou, quando for o caso da adoção de modelos consagrados como o SPICE ou o CMMI, onde o conhecimento sobre os mesmos pode fazer grande diferença na performance da equipe.

	Nenhum conhecimento
2	Participou de treinamento
4	Domínio básico
6	Experiência prática
8	Muito Boa Experiência
10	M.B.Experiência e Domínio Teórico

Figura 4.15. Estabelecimento de limites mínimos de capacitação a cada competência essencial

O ponto de partida de um instrumento desse tipo pode ser uma tabela de enquadramento dos colaboradores nas competências

desejadas, que evolui a quadros como o desenvolvido para a STORE Automação (fig 5.4) onde a avaliação das médias e desvios-padrão podem ser confrontadas com limites mínimos de capacitação pré-determinados a cada competência essencial, como o exemplificado à Figura ao 4.15.

Por exemplo, *“a organização somente deve iniciar o PMQ-SW quando pelo menos 50 por cento dos colaboradores da área técnica tiverem participado de treinamento nas competências predeterminadas”*; ou *“se a organização optar pelo CMMI, ela somente prescindirá de consultoria especializada quando a média da competência em CMMI dos engenheiros de sistemas do Controle de Qualidade for superior a 7 e variância máxima = 1”* .

4.7.2.1. A Organização Necessária a um Projeto de Melhoria Contínua da Qualidade do Processo de Software

Partiu-se de um princípio de que a melhoria de processos não ocorre dissociada da melhoria ou da reorientação da Cultura Organizacional, ou, ainda, da melhoria da maturidade da organização (se assim se preferir referenciar). Ainda, levou-se em consideração que somente a melhoria dos processos não basta, tem de ser acompanhada de um movimento de melhoria da maturidade ou da cultura organizacional da organização, segundo Curtis (1998). Assim, o MC2Q-SW recomenda a constituição de um **Grupo Organizador (GO)** e de um **Comitê da Qualidade (CQ)** como resultado de uma das decisões iniciais de capacitação na direção de um PMQ-SW.

Observado o princípio de EVOLUCIONARIEDADE, a criação do **GO** tem a finalidade de funcionar como a célula mater, o embrião, a partir do qual a organização inicia um processo evolutivo de estruturação e organização que permita seu aprendizado, a adaptação de modelos e métodos, e a elaboração e execução de planos de ação para melhorar a qualidade de seus processos.

Pode ser constituído, inicialmente, por uma equipe de um ou mais especialistas, com tempo integral ou parcial, cujo perfil englobe: interesse no tema QUALIDADE; alto comprometimento; capacidade de realização; conhecimento; liderança e um grande grau de autonomia, direcionada ao

objetivo prioritário de desenvolver a nova Estrutura de Qualidade da organização e determinar tudo o que deve ser feito para iniciar e manter ativo o PMQ-SW.

O CQ é o núcleo inicial de homologação das ações propostas pelo GO, devendo ser composto (idealmente) por colaboradores da organização que tenham legítima autoridade para tomar as decisões necessárias a garantir a perenidade do PMQ-SW. Alertou-se “idealmente”, pois a experiência demonstra que dirigentes não habituados à Tecnologia da Informação tendem a não incorporar tal papel, assim “alternativamente” os componentes desse núcleo pode assumir essa função por delegação, o que deve garantir a legitimidade de poder de decisão. Funcionará como órgão para decisões colegiadas visando garantir que as ações do PMQ-SW atendam, de fato, os interesses da organização, e em conformidade com os princípios e valores cultuados por ela. Portanto, o GO e o CQ devem trabalhar em perfeita sintonia para esse fim.

A figura 4.16 ilustra uma configuração básica de GO e CQ onde este representa a alta administração por delegação de competência, com poderes para homologar as ações de melhoria da qualidade.

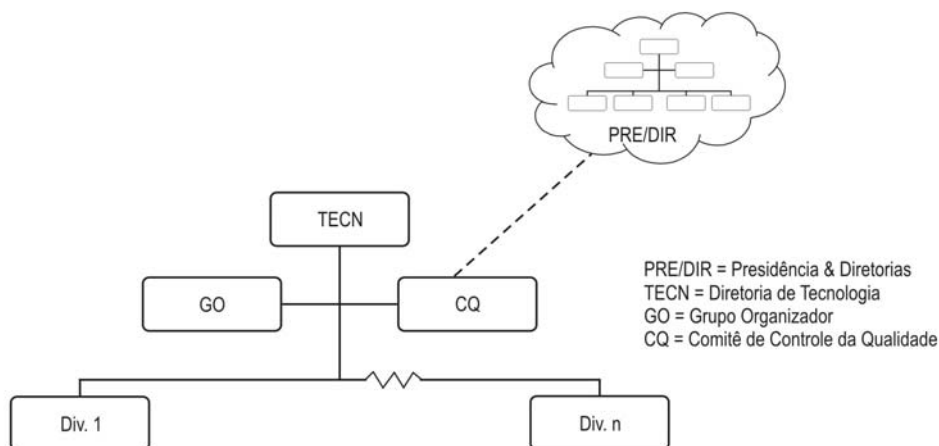


Figura 4.16 Estrutura embrionária recomendada às organizações que demonstram preocupação mínima com a qualidade

A criação desses dois núcleos iniciais permite o caminho às configurações mais sofisticadas da Estrutura da Qualidade para organizações mais maduras, como Grupos de: Teste; Garantia da Qualidade (*Software Quality Assurance* - SQA); Engenharia de Processos (*Software Engineering Process Group* - SEPG); dentre outros, a exemplo da estrutura divulgada pela empresa Ci&T para garantir a melhoria da qualidade de seus produtos e processos, ilustrada na figura 4.17.

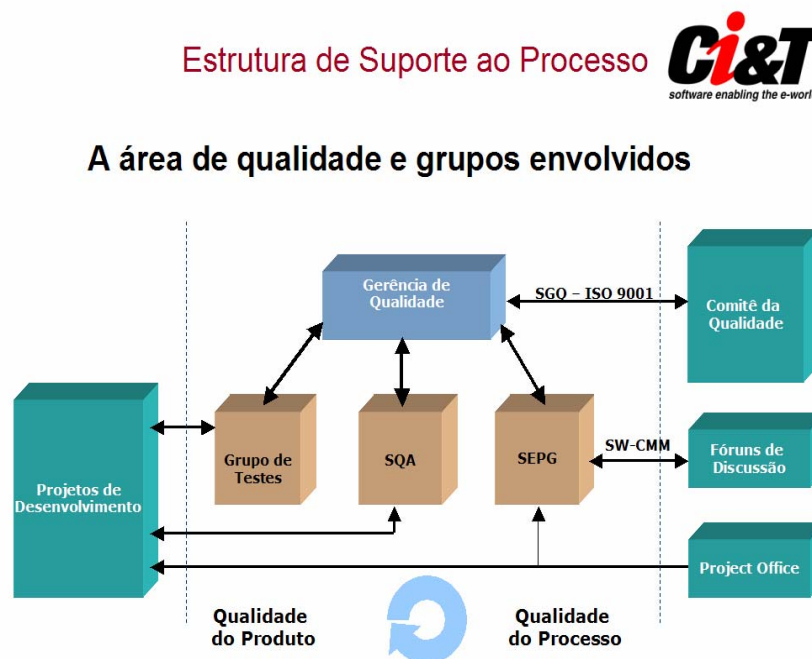


Figura 4.17. Exemplo de Estrutura Orgânica voltada à qualidade em empresas maduras (<http://www.cit.com.br/aa> em ago.2005)

4.7.2.2. A Determinação do Paradigma para a Base de Processos

O CM2Q-SW sugere que se adotem como paradigma as descrições de processos de *software* e suas práticas do CMM ou do SPICE por representarem o que há de mais completo a respeito dos processos de SW.

O importante é que a adoção de uma Base consagrada de Conhecimento dos Processos de *Software* irá poupar um enorme trabalho de pesquisa.

Uma vez adotada a **Base de Processos (BP)**, cada processo de interesse passará a ter suas especificações ali descritas confrontadas com as práticas correntes na Organização. Desse confronto podem surgir situações de substituição, supressão ou melhoria de práticas.

O CM2Q-SW recomenda que a avaliação de cada processo seja feita de forma organizada e orientada a partir de uma seleção objetiva, que resulte em um ou mais ciclos de melhoria do processo somente após a formal aprovação pelo CQ.

Portanto, segundo tal recomendação, somente se deve empreender esforços no detalhamento do processo e de suas práticas, à medida que a Organização decidir que vale a pena investir nessa tarefa.

4.8. A Seleção dos Processos (Etapa 2.1)

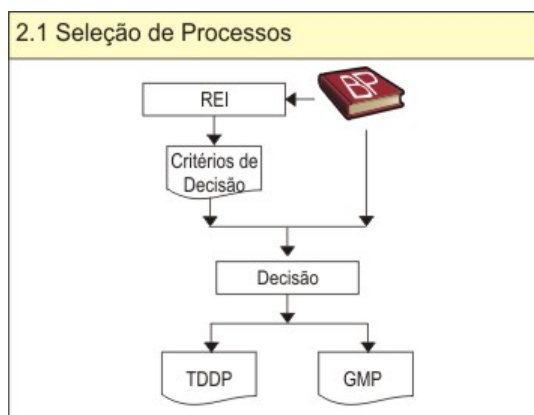


Figura 4.18. A Seleção dos Processos (Etapa 2.1)

A decisão de quais processos devem ser submetidos ao MC2Q-SW depende de múltiplos fatores: alinhamento dos processos à missão e negócios da organização; interesse; oportunidade; necessidade; que variam conforme peculiaridades da organização ao longo do tempo.

Assim, a decisão de escolha decorre da importância que se dê aos diversos fatores considerados essenciais à determinação dos processos, como exemplifica a experiência de Salviano (1999) junto à *softwarehouse* Senior Sistemas, de Blumenau.

Qualquer método que determine com clareza critérios de seleção e classificação de processos através de graus atribuídos segundo ponderações e notas, incluindo critérios de desempate quando se atingir graus idênticos, pode ser utilizado.

O MCQ2-SW sugere, como alternativa simplificadora, a aplicação de uma ou mais reuniões de eclosão de idéias (**REIs**) encerrada por uma atividade de tomada de decisão que deve resultar em uma documentação clara de porque determinado processo, ou processos, foi o escolhido. A referida documentação pode ser apenas uma **Tabela de Decisão para Determinação dos Processos (TDDP)** a submeter a melhorias, ou um simples relato em ata de reunião dos critérios utilizados para essa determinação.

Tendo ficado claro o processo escolhido a submeter às melhorias, o MC2Q-SW sugere que nesse instante se destaque sua descrição, práticas e demais vinculações da Base de Processos (BP) adotada, constituindo a primeira **GUIA MESTRA DO PROCESSO (GMP)**, que passará a ser atualizada a cada melhoria ou adaptação de práticas que se considere relevante documentar.

Assim, resumidamente, a GMP surge como cópia do texto completo a respeito do processo e suas práticas, constante na Base de Processo eleita como paradigma.

4.8.1. A Tabela de Decisão para Determinação dos Processos (TDDP)

A forma mais prática encontrada a determinar processos a submeter a melhorias, é a simples adaptação do método Kepner & Tregoe (Kepner, 1997).

O método consiste, sumariamente, em, primeiro, delinear **o mais precisamente** o objetivo da decisão, e, com base nesse objetivo, identificar critérios que permitam eliminar alternativas indesejáveis rapidamente, e, a seguir, classificar as desejáveis.

Para isso basta separar 2 grupos de critérios: o primeiro denominado como grupo dos critérios IMPRESCINDÍVEIS, que funciona como uma peneira, ou funil, tipo “passa-não-passa”, que elimina de pronto os processos que não se tem interesse.

O segundo, denominado de grupo dos critérios PONDERÁVEIS, é o conjunto de quaisquer critérios (que pode conter também os imprescindíveis, agora em ponderação) a que se atribuem pesos e que serão avaliados com notas de 0 a 10 para cada alternativa.

Como exemplo de decisão podemos ter diversas situações a configurar cada instante, ilustrada a seguir

Quadro 4.2. Quadro de Decisão para Determinação do Processo a Submeter à Melhoria da Capacitação		
OBJETIVO: selecionar um ou mais processos a promover a melhoria da capacitação no ano de 2006.		
Instante	Critérios Imprescindíveis	Critérios Ponderáveis
A	1. Processo mais crítico. 2. Processo que seja praticado pelo maior número de engenheiros de sistemas.	Nenhum além dos imprescindíveis.
B	1. Processo de grande interação com o cliente.	1. Maior grau de interação com o cliente 2. Maior ganho em performance decorrente das melhorias já identificadas por fazer
C	1. Processo de maior absorção de recursos.	1. Maior grau percebido de redução da carga de trabalho. 2. Maior possibilidade de aplicação de ferramentas de produtividade.

O resultado da “filtragem” pelos critérios imprescindíveis pode ser vários processos, ou apenas um único processo. Caso resulte mais de um processo, seu desempate entre eles decorre da classificação que se obtém pela avaliação obtida dos Critérios Ponderáveis, passível de uma Análise Complementar de Conveniência ou Risco cuja finalidade é ajustar eventuais falhas de atribuição de pesos na ponderação.

4.9. A Identificação de Métricas (Etapa 2.2)

O propósito da ETAPA 2.2 é identificar métricas, gerando uma **Tabela de Avaliação das Melhorias do Processo (TAMP)**, que, a este instante, somente

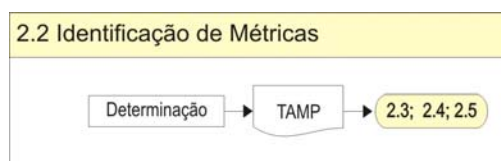


Figura 4.19. A Identificação de Métricas (Etapa 2.2)

pode avaliar a capacitação do processo segundo este se apresenta em função de suas práticas **ANTES** de sua submissão às melhorias pretendidas ainda em deliberação.

O maior desafio em todo o desenvolvimento do MC2Q-SW foi identificar qual seria (ou quais seriam), obedecendo os critérios de racionalização, a(s) métrica(s) práticas eficazes e eficientes a demonstrar o grau de sucesso *a posteriori* da(s) melhoria(s) aplicada(s) à capacitação do processo.

Entendeu-se que a teorização dos modelos CMMI (dos já apresentados 5 níveis de capacitação) e do SPICE (dos já apresentados 6 níveis de capacidade) é boa, na medida em que sensibiliza os envolvidos no grau de precariedade ou excelência do *status-quo* dos processos em uso. Porém, considerou-se que a simples identificação do processo na escala de 0 a 5, por exemplo, estabelecida pela ISO 15504 (ISO/IEC 15504-5, 2006) ainda é insuficiente para uma boa gestão de melhorias em processos de *software*.

O MC2Q-SW preferiu adotar uma sistemática de medição baseada em um conceito de que um processo tem de ter alguns atributos avaliados em relação às suas práticas significativas conforme as circunstâncias ditadas pelo negócio da Organização.

Assim, é importante saber o quanto de práticas significativas a determinado processo, em um determinado contexto vivenciado pela Organização, está sendo praticado em um instante específico.

Esse exercício de saber “o quê e como está sendo feito em relação ao quê deve ser feito” para obter o melhor rendimento parece ser o que concentra, consciente ou inconscientemente, os esforços gerenciais dos executivos de Tecnologia de Informação que lidam com o desenvolvimento de *software* liderando suas equipes nessa atividade.

Os práticos, que não perdem tempo em avaliações acuradas, mas confiam em sua percepção da realidade através de avaliações subjetivas aproximadas, parecendo focar sua atenção, alocar recursos e estabelecer prioridades de ação conforme critérios implícitos quando executam ou gerenciam os processos de criação de *software*.

O MC2Q-SW tenta aqui explicitar, para que fiquem mais claros, quais os atributos que devem ser avaliados, e como poderiam sê-lo, permitindo, assim, identificar se a execução de um processo está mais próxima da precariedade ou mais próxima da excelência.

Assim, o MC2Q-SW propõe dois tipos de avaliação, a do primeiro tipo, focando genericamente o processo (mas sem perder de vista suas práticas), denominada **Medida da Capacitação do Processo (MCP** ou **PMCP** conforme se explica no item a seguir). A do segundo tipo focando a evolução das melhorias em suas práticas, através da “Curva de Melhoria”. Ambas só têm sentido prático quando se deseja empreender melhorias, de fato, para avaliar a evolução das práticas, e não apenas “medir” estaticamente a situação do processo.

Desta forma as medições de MCP e de Curva de Melhoria implicam em uma avaliação ANTES e outra avaliação DEPOIS de terem sido empreendidas as ações de melhoria ao processo.

4.9.1. A Medida da Capacitação do Processo (MCP)

Relativamente a medições, o MC2Q-SW, inspirado no modelo SPICE, descrito em 2.6.4 e 2.6.5, estipulou que o processo deve considerar 5 atributos básicos (melhor detalhados logo à frente) denominados de **DIE₃**, e que têm de ser avaliados na ordem apresentada a seguir:

- **DEFINIÇÃO (D);**
- **INCORPORAÇÃO (I);**
- **EXECUÇÃO (E₁);**
- **EVOLUÇÃO (E₂)**
- **EXCELÊNCIA (E₃).**

Tais atributos devem ser avaliados em uma escala pontual de 6 pontos, de 0 a 5, conforme a tabela a seguir:

- 0 – o atributo não é satisfeito;
- 1 – o atributo é precariamente satisfeito;
- 2 – o atributo é satisfeito, porém, de forma ainda intuitiva ou informal¹³, sem formalização;
- 3 – o atributo é satisfeito com um mínimo de formalização;
- 4 – o atributo é satisfeito com boa formalização;
- 5 – o atributo é perfeitamente satisfeito, segundo padrões normatizados que garantam sua qualidade.

A seguir, maiores esclarecimentos a respeito dos atributos **DIE₃**:

- **DEFINIÇÃO** – diz respeito à clareza, abrangência, contextualização, documentação e gerenciamento da definição do processo. Ou seja o processo precisa estar perfeitamente definido, sem margens a dúvidas, enquadrado em uma estrutura de grupos de processos, com suas práticas identificadas e com o inter-relacionamento com demais processos (ou práticas) também explicados.

Se tal definição não estiver documentada já é o bastante para que tal atributo tenha zero de avaliação, caso em que, pelas regras 1 e 2 expostas a seguir, todos os demais atributos receberão também avaliação zero.

Caso a definição exista, porém seja incompleta, ou não perfeitamente contextualizada, a avaliação do atributo poderá ser de 1, 2, 3, ou, no máximo, 4. Apenas para o caso em que a definição seja bastante abrangente, perfeitamente clara, contextualizada, documentada, e gerenciada é que esse atributo receberá a nota 5.

¹³ A formalização refere-se a ocorrência de normas e padrões de conduta explicitamente documentados na Organização sobre as quais há gerenciamento, manutenção, revisões periódicas, aperfeiçoamentos sempre que necessário e treinamento dos envolvidos. O inverso, quando não há normas nem gerenciamento sobre as mesmas, caracteriza a informalidade, ou informalização. Entre cada extremo, desde o totalmente INFORMAL (intuitivo, precário, elementar, desorganizado, não gerenciado) e o FORMALIZADO GERENCIADO (perfeitamente gerenciado, atualizado, medido), estabeleceram-se os 6 graus na intenção de realçar a importância do gerenciamento e da formalização de procedimentos na garantia da qualidade.

No entanto, o fato de receber 5, não significa que a definição é a melhor que se possa ter em definir perfeitamente o processo. Apenas significa que, para a execução desse processo, ele está perfeitamente definido e documentado em relação às necessidades da Organização e que há um controle gerencial perene sobre sua definição. Assim, tão logo a Organização mude seu foco de ação, ou descubra um paradigma que permita que o processo se demonstre mais abrangente, melhor estruturado e contextualizado, pode simplesmente promover a substituição dessa definição por outra, que nesse momento será reavaliada segundo os critérios de clareza, abrangência, contextualização e documentação, já citados.

Um princípio fundamental para que cada atributo obtenha a nota 5 é que ele seja formalmente gerenciado. Ou seja, por melhor que seja a avaliação de um atributo, é impossível ele obter a nota 5 se não for gerenciado¹⁴. No caso da **DEFINIÇÃO** de um processo, esse atributo somente receberá nota 5 se, obrigatoriamente, alguém periódica, sistemática e metodologicamente, cuidar de verificar se a definição expressa a realidade de forma eficaz e eficiente e necessita de melhoramentos;

- **INCORPORAÇÃO** – diz respeito ao quanto a definição é perfeitamente conhecida pelos atores que a utilizam na Organização. Só poderá obter avaliação 5 se, além de todos os envolvidos no desenvolvimento de sistemas, for gerenciada, pois tal situação demonstra o cuidado em garantir permanentemente a efetiva incorporação da definição institucionalizada do processo na Organização.

Por exemplo, é graças ao gerenciamento que se garante treinar os novos colaboradores que passem a trabalhar com o desenvolvimento de *software*, independentemente de serem funcionários antigos da

¹⁴ Gerenciado, na concepção da Teoria Geral da Administração, onde Gerenciar engloba as funções Planejar, Organizar, Comandar, Coordenar e Controlar, adicionado de Comunicar, no que o pesquisador alcunha de POC₄.

Organização que atuavam em outras áreas, ou doutores recém-contratados oriundos dos melhores cursos de Engenharia de Sistemas;

- **EXECUÇÃO** – diz respeito ao uso eficaz e eficiente das práticas definidas como tal pelos atores no processo de desenvolvimento de sistemas. Só pode obter 5 se for gerenciada pelos pares ou chefia através de reuniões de inspeção ou demais práticas gerenciais;
- **EVOLUÇÃO** – diz respeito ao conhecimento das melhorias necessárias às práticas de um processo que permitam que ele atinja a excelência e ao esforço em, continuamente, melhorar essas práticas segundo os **princípios racionais** que orientaram o MC2Q-SW. Para se obter o grau 5 neste atributo, não é necessário que todas essas práticas estejam sendo praticadas, mas sim que elas sejam de domínio e que haja gerenciamento, liderança sobre o quê, quando e como devem ser melhoradas, com uma revisão sistematizada que confronte o já conhecido com o estado da arte.
- **EXCELÊNCIA** – é o atributo que indica se o estado da arte de práticas do processo em questão é aplicado na Organização segundo¹⁵ os princípios de racionalidade, garantindo confiabilidade, eficácia e máxima eficiência do processo. A designação do grau 5 a este atributo significa que todos os atributos anteriores obrigatoriamente foram 5, conforme as regras 1 e 2 explicadas a seguir.

São duas as regras básicas para avaliação de processos segundo tais atributos:

Regra 1 – Os atributos têm de ser avaliados na ordem exposta **DIE₃**, ou seja: o atributo **DEFINIÇÃO** tem de ser avaliado **antes** do atributo

¹⁵ Importante considerar os princípios de racionalidade, pois práticas inócuas e o desperdício da força de trabalho em atividades improdutivas, que não agreguem valor ao processo devem ser evitadas. Ilustrando como exemplo extremo, se determinada prática, imprescindível a um sistema “*critical-mission*”, não for sequer considerada em determinado processo de desenvolvimento de um *software* “*ad-hoc*” de missão não crítica, não há porque não aceitar ser o processo excelente.

INCORPORAÇÃO, que deve ser avaliado *antes* da **EXECUÇÃO**, que deve ser avaliado *antes* da **EVOLUÇÃO**, que deve ser avaliado *antes* da **EXCELÊNCIA**.

Esta regra foi estabelecida pelo pressuposto de que tal disciplinamento de ordem induz à maior eficiência, uma vez que quanto melhor entendido e praticado cada atributo antecedente, melhor deve resultar o atributo seguinte.

Regra 2 – Qualquer atributo nunca pode obter grau maior que o atribuído ao imediatamente antecedente, assim é impossível atribuir 5 à **EXECUÇÃO** de um processo se parte dos engenheiros de sistemas que trabalham com o desenvolvimento de *software* não conhecem a definição institucional do processo, adotada pela Organização, caso em que a **INCORPORAÇÃO** apresenta um grau menor que 5.

Esta regra foi estabelecida complementarmente ao pressuposto da regra anterior, de uma ordem natural favorecendo a melhoria da qualidade do processo, onde melhor se torna o processo cujos atributos venham melhorando de grau disciplinadamente, com o atributo anterior antecedendo sua melhoria à melhoria do seguinte.

A melhor forma de comunicar a avaliação de um processo pelo MC2Q-SW é lançar mão de um indicador denominado de **Medida de Capacidade do Processo (MCP)**, a medida criada pelo pesquisador e utilizada experimentalmente no estudo de caso. Constitui-se de um “*array*” dos 5 atributos, um seguido do outro, da esquerda para a direita, com as possíveis combinações abaixo ilustradas, melhor visualizadas de forma gráfica através de um Pentágono, batizado de **Pentágono MCP (ou PMCP – Pentágono da Medida de Capacitação do Processo)** um gráfico tipo “radar” auto-explicado através das figuras a seguir.

Convencionou-se que o “UNIVERSO de MCP” é o conjunto de todos os valores do *array* **DIE₃** que, exclusivamente, atendam às regras estipuladas, assim um **DIE₃** de valor 54324, não se considera um MCP, mas sim é apenas uma combinação de valores de **DIE₃** que não obedece às regras estipuladas, mesmo que tal valor tenha sentido de explicar determinada situação de avaliação.

	D	I	E ₁	E ₂	E ₃
1	0	0	0	0	0
2	1	0	0	0	0
3	1	1	0	0	0
4	1	1	1	0	0
5	1	1	1	1	0
6	1	1	1	1	1
7	2	0	0	0	0
8	2	1	0	0	0

	D	I	E ₁	E ₂	E ₃
9				
10	2	2	1	0	0
11				
12	3	3	3	2	0
13				
14	4	2	2	1	0
15				
16	5	0	0	0	0

	D	I	E ₁	E ₂	E ₃
17				
18	5	4	4	4	2
19				
20	5	5	3	3	3
21				
22	5	5	5	5	3
23	5	5	5	5	4
24	5	5	5	5	5

Figura 4.20. 24 possíveis combinações do *array* **DIE₃** no universo de **MCP**

Observa-se que a demonstração gráfica (Figura 4.21) permite perceber mais rápida e claramente que o indicador MCP54442 refere-se a um estado de capacitação de processo muito mais maduro que o MCP11100.

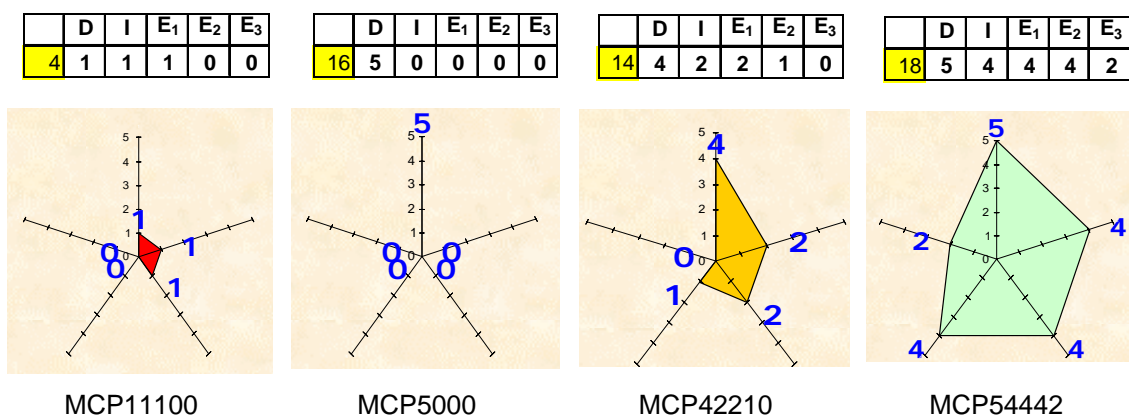


Figura 4.21. Esquema gráfico de 4 indicadores **MCPs** escolhidos dentre os 24 apresentados na Figura 4.20

Complementarmente, acrescentaram-se 4 exemplos de avaliações errôneas por infringirem as regras estabelecidas (Figura 4.22), que, portanto, não se enquadram no universo de **MCP**.

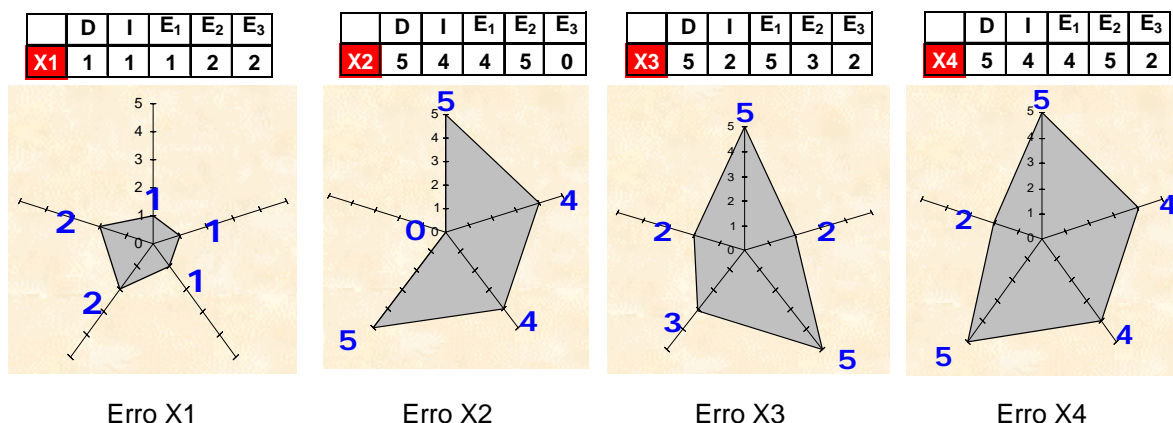


Figura 4.22. Esquema gráfico de 4 combinações do array **DIE₃** que **não** se caracterizam como **MCPs** por infringirem as regras básicas estabelecidas

Propositadamente o MC2Q-SW exagera no rigor de avaliação quando determina a segunda regra, pois, desta forma, força que o processo seja continuamente revisto em relação à sua aplicação, de fato, na Organização.

Justifica-se tal rigor na intenção que o mau gerenciamento seja punido, pois, presume-se, praticamente tudo se resume a resultado de gerenciamento. Bons resultados parecem ser grandemente favorecidos por bom gerenciamento e maus resultados parecem ser grandemente influenciados por gerenciamento de má qualidade.

Distorções comuns nas Organizações, como por exemplo: a não promoção de treinamentos periódicos, ao menos de reciclagem, dos colaboradores; ou o não treinamento de novos integrantes da equipe de desenvolvimento; ou as falhas de comunicação quando se relaxa na formalização de documentação e na baixa interação pessoal ou organizacional, são, sem dúvida, resultantes de muitas variáveis, mas que incorporam o descuido ou o relaxamento e isto tem a ver com gerenciamento, conforme se constata em qualquer manual básico de Administração.

Assim, é perfeitamente natural que, um processo outrora considerado MCP5542 possa, hipoteticamente, cair para MCP5332¹⁶ um ano após, em

¹⁶ O processo que outrora ganhara **5** na INCORPORAÇÃO (I) – o 2º dígito, da esquerda para a direita no MCP5542 - passou a ser avaliado como **3** pela quantidade significativa de pessoas que estão trabalhando com ele sem treinamento formal. Isto é um erro grave, que deve penalizar toda a medida MCP, assim E1 e E2 (EXECUÇÃO e EVOLUÇÃO) – respectivamente os 3º e 4º dígitos no MCP5542, mesmo não tendo regredido, não podem, agora, obter grau acima de **3**, embora já fosse sabido que estavam em nível **5** e **4**, respectivamente, outrora. A penalidade de rebaixamento de grau a atributos seguintes que excedam o

função da contratação de um grupo de *trainees* que, trabalhando com o processo em questão, não tenha sido treinado, de alguma forma concreta, nesse processo.

4.9.2. O Artefato de Medição da Melhoria da Capacitação do Processo

Como avaliar um processo implica na avaliação de suas práticas, muito mais racional é avaliar o “estágio de uso” de cada uma das práticas associadas em relação ao seu possível “uso pleno”, através de uma escala percentual, onde 100%, obviamente, representem esse “uso pleno” (a prática sendo utilizada em toda sua potencialidade). Isto pareceu ser melhor que, simplesmente, rotular o nível de capacidade do processo entre 0 e 5, a exemplo do SPICE (figura a seguir), discutido em 2.6.4 e 2.6.5, como também, ao invés de simplesmente utilizar o indicador MCP, comparando sua avaliação ANTES e DEPOIS da aplicação das melhorias aos processos.



Figura 4.23. Os Níveis de Capacidade do Processo segundo o SPICE (SALVIANO, 2005)

novo grau de um atributo antecedente foi instituída para realçar a gravidade do rebaixamento do atributo antecedente, fazendo saltar aos olhos que o processo perdeu em qualidade.

Este é o propósito da **TAMP**, o artefato resultante da ETAPA 2.2 ora em apresentação: o de avaliar mais de perto as melhorias nas práticas, tanto numérica como graficamente, através da comparação entre as Curvas de Capacitação do Processo, que resultará no **Índice de Melhoria da Capacitação do Processo (IMCP)** explicado mais à frente.

Assim a TAMP pode ser definida como o artefato que demonstra o “**grau de utilização real**” ou, ainda, o “**estágio de uso**” de cada PRÁTICA no processo, tanto ANTES como APÓS as MELHORIAS terem sido implementadas, em confronto com a utilização ideal dessa prática, estabelecida como sendo 100%.

Isto posto, o “estágio de uso” foi definido como uma medida de avaliação relativa entre o quê, de fato, **era feito** como prática, em relação ao que **poderia ser feito** completamente se essa prática fosse executada em toda sua potencialidade de forma bastante eficiente.

Para melhor ilustrar essa medida, imaginou-se um processo X hipotético, composto por apenas 4 práticas onde seus estágios de uso caracterizaram, conforme a figura 4.24, as situações ANTES e DEPOIS das melhorias. Neste exemplo, a prática 01 não sofreu nenhuma melhora; a prática 02, cujo estágio de uso era de 25%, melhorou para a posição de 50%; a prática 03, inexistente antes da aplicação de melhorias passou a ser perfeitamente praticada, em toda sua potencialidade, após a melhoria; e a prática 04 melhorou seu estágio de uso de 25% para 75%.

Desta maneira, as métricas relacionadas à medição da melhoria da Capacitação do Processo ficaram associadas às práticas conforme apresentado à figura anterior (cujo preenchimento se explica em sua legenda) que mostra o estabelecimento dessas métricas, e como é possível avaliar a melhoria, mediante duas medidas sucessivas, uma antes e outra posterior à execução das melhorias implementadas.

TAMP - Tabela de Avaliação da Melhoria do Processo X													
PRÁTICAS inerentes ao Processo X	A PRÁTICA já ocorria, no proc X, em até x% do ideal (ANTES das MELHORIAS):					A prática ocorre no processo X em até y% do ideal (DEPOIS das MELHORIAS):					AVANÇO de Melhoria		
	zero %	até 25%	até 50%	até 75%	até 100 %	ATÉ	zero %	até 25%	até 50%	até 75%		até 100 %	ATÉ
prática 01		25				25		25				25	0
prática 02		25				25		50				50	25
prática 03	0					0					100	100	100
prática 04		25				25			75			75	50
Indicar qtos % a prática era utilizada no processo em foco ANTES das Melhorias													
Preencha com 0, 25, 50, 75 ou 100 na célula apropriada conforme seja o caso.													
Exemplo:													
Se a prática 03 era nula (não praticada), então preencha 0 (zero) na célula D8													
Se a prática 04 ocorria em 25% do ideal, então preencha 25 na célula E9													
Indicar qtos % do uso ideal a prática passou a ser utilizada nesse processo APÓS a implementação das melhorias.													
Preencha com 0, 25, 50, 75 ou 100 na célula apropriada conforme seja o caso.													
Exemplo:													
Se a prática 03 passar a ser praticada no que se considerar 100% do que poderia ser aproveitada nesse processo, então preencha 100 na célula N8. Analogamente, para o caso de a prática 04 passar a ser praticada no que se considerar 75% do que poderia ser aproveitada nesse processo, então preencha 75 na célula M9.													
Obs: neste exemplo a aplicação da prática 03 subiu de 0% para 100% do uso ideal no processo X (AVANÇO de Melhoria = 100%) e a aplicação da prática 04 subiu de 25% para 75% do uso ideal no processo X (AVANÇO de Melhoria = 50%)													

Figura 4.24. O artefato de Medição da Melhoria da Capacitação do Processo: TAMP (Tabela de Avaliação da Melhoria do Processo)

4.9.3. O Índice de Melhoria da Capacitação do Processo (IMCP)

Outra avaliação de melhoria criada, considerada importante pelos envolvidos no estudo de caso, foi o **Índice de Melhoria da Capacitação do Processo (IMCP)**, uma demonstração gráfica comparativa do “estágio de uso” das práticas exigidas a um processo.

O gráfico resulta da listas de valores extraídas do modelo do artefato TAMP ilustrado na figura 4.24, que geram duas curvas, P_a e P_p , demonstradas na figura 4.24a, onde P_a é definida como sendo a curva de uso das práticas **antes** da aplicação das melhorias e P_p como sendo a curva uso das práticas **após** a aplicação das melhorias.

A curva P_a obtém-se dos valores da **coluna I**, que indicam o “estágio de uso” de cada PRÁTICA no processo, **ANTES** das MELHORIAS terem sido implementadas.

A curva P_p obtém-se dos valores da **coluna O**, que indicam o “estágio de uso” de cada PRÁTICA no processo, **APÓS** as MELHORIAS terem sido implementadas.

Ao se avaliar a diferença entre as áreas sob as curvas P_p e P_a (que são, respectivamente, as curvas que demonstram os estágios de uso de **todas** as práticas exigidas ao bom desempenho de um determinado processo, posterior e anteriormente à implantação de melhorias) pode-se visualizar a demonstração gráfica do **Índice de Melhoria da Capacitação do Processo (IMCP)**.

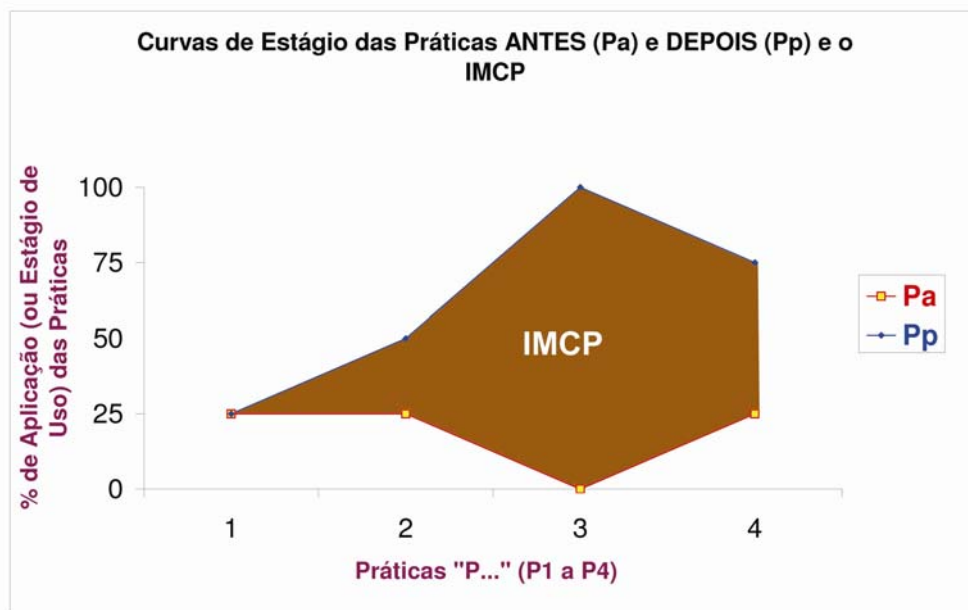


Figura 4.25. Demonstração gráfica do IMCP a partir das curvas P_p e P_a

4.10. A Avaliação da Capacitação do Processo (Etapa 2.3)

Os métodos vinculados às abordagens de melhoria da qualidade são bastante completos e pode-se adotar qualquer um deles segundo sua compatibilização aos fins desejados, a exemplo dos CMM *Appraisal Framework Version 1.0* – CAF V1.0 (Masters&Bothwell, 1995) e do CMM *Based Appraisal for Internal Process Improvement Version 1.2* segundo Donna e Masters (2001), e o *SEI Appraisal Program* (2004).

Relativamente ao modelo de avaliação contínua SPICE tem-se as propostas e experiências de uso demonstradas por Salviano (1999, 2001, 2004 e 2005).

Também é oportuno lembrar os trabalhos de aplicação prática de Arruda, Villas Boas et al. (1998), Villas Boas (2001), e Kohan (2003) que exploram bem a condução de trabalhos de avaliação ou melhoria do processo.

O MC2Q-SW, em observância aos princípios de racionalização, especialmente os da OPORTUNIDADE e da ECONOMICIDADE, recomenda, no entanto, que se procedam as avaliações dos processos praticamente a partir do instante em que se dispararão os Ciclos de Melhoria da Qualidade (CMQs) descritos mais à frente pelas seguintes vantagens:

- Restringe-se a quantidade de processos a serem avaliados, uma vez que os ciclos visam delimitar a amplitude e a complexidade das atuações de melhoria;
- Os custos inerentes à tarefa de avaliação da capacitação de um processo são apropriados somente à véspera de atuação para sua melhoria;
- O esforço e a concentração da atenção dos envolvidos na avaliação da capacitação do processo vincula-se, de imediato, ao planejamento e à execução de sua melhoria, como também à sua efetiva INSTITUCIONALIZAÇÃO.

Tal racionalização minimiza o tumulto e o impacto do desembolso financeiro provocados pela atividade de avaliação se, desnecessariamente, se passasse a avaliar inúmeros processos, sem proceder ao trabalho de melhoria, um a um, logo a seguir, vinculando, de imediato, a referida avaliação dos poucos processos escolhidos ao planejamento e à execução de suas melhorias, como também à sua efetiva INSTITUCIONALIZAÇÃO.

O MC2Q-SW, portanto, recomenda que a Avaliação da Capacitação de cada processo ocorra vinculadamente ao Ciclo de Melhoria da Qualidade do Processo de SW (CMQ-SW) (descrito mais à frente) que vier tratar de sua melhoria, antecedendo sua correspondente elaboração da Proposta Formal de Ciclo (PFC) (explicada mais adiante), pois é em função dos objetivos de cada ciclo que se restringe a quantidade de processos e práticas a serem

melhorados, além de somente se apropriarem custos a essa tarefa a esse instante.

E, ainda de acordo com os princípios de racionalização, oferece o MCP, a TAMP e o Índice de Melhoria da Capacitação do Processo (IMCP) como os instrumentos de medição a serem utilizados na avaliação da Capacitação do Processo.

No entanto, para a identificação das melhorias necessárias às práticas, podem ser necessários passos adicionais caso não tenha sido possível, durante o Diagnóstico Situacional, apurar as necessidades de melhoria.

Esses passos adicionais podem ser executados nesta etapa 2.3 através de questionários, ou tabelas ou relatórios complementares aos RAS e DSPA, resultantes de atividades tipo REI ou de qualquer outra forma de elicitação.

Assim surgiram as sugestões de uso do QAP, TAP e RAP¹⁷, conforme ilustra a figura seguinte, para o caso de alguém sentir necessidade de expandir a avaliação que já se originou na etapa de Diagnóstico Situacional, caso em que a TAMP deverá ser revista e complementada em função das possíveis novas melhorias apuradas através destes novos instrumentos, transformando-se na **Guia de Melhoria da Capacitação do Processo (GMCP)** na etapa 2.4 a seguir.

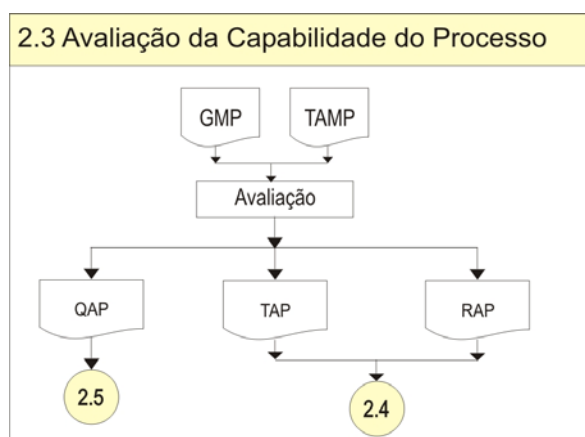


Figura 4.26_ A Avaliação da Capacitação do Processo (Etapa 2.3)

4.11. A Identificação de Melhorias (Etapa 2.4)

¹⁷ QAP, TAP e RAP têm suas idealizações descritas no Quadro 4.1, à referência 2.3

Da avaliação dos RAS, DSPA, complementada pelos eventuais QAP, TAP e RAP, e tendo em vista as melhorias já previstas na TAMP devidamente atualizada e preparada para a avaliação das melhorias, surge a **Guia de Melhoria da Capacitação do Processo (GMCP)**, o artefato designado, um a cada processo, a compilar as medidas de melhoria em práticas pertinentes ao processo em foco.

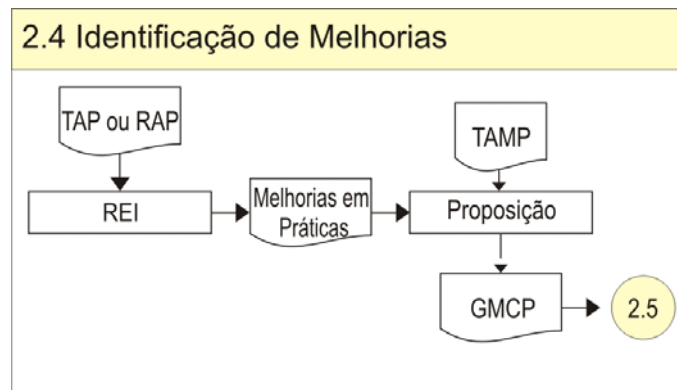


Figura 4.27. A Identificação de Melhorias (Etapa 2.4)

Por razões de simplificação a GMCP é apenas uma versão complementada da TAMP, mas, dependendo do grau de detalhamento de especificação de melhorias, pode ser uma evolução da TAMP que anexe relatório com esse fim.

Como evoluções à TAMP, ela ainda pode ser utilizada para agrupar e consolidar inúmeras melhorias a práticas comuns, racionalizar melhorias e priorizar as inúmeras tarefas de melhorias.

A GMCP tem, portanto, a finalidade de funcionar como um “*check-list*” de todas as melhorias que se pretende sejam implementadas ao longo de um ou mais CMQs-SW para determinado processo, até que as práticas desse processo possam ser consideradas as melhores práticas que o tornem reconhecidamente maduro ou com alta capacitação.

4.12. A Constituição do Dossiê do Processo (Etapa 2.5)

Todo o trabalho da **FASE 2 (PREPARAÇÃO)** deve ser concluído com a montagem de um **DOSSIÊ DO PROCESSO (DP)**.

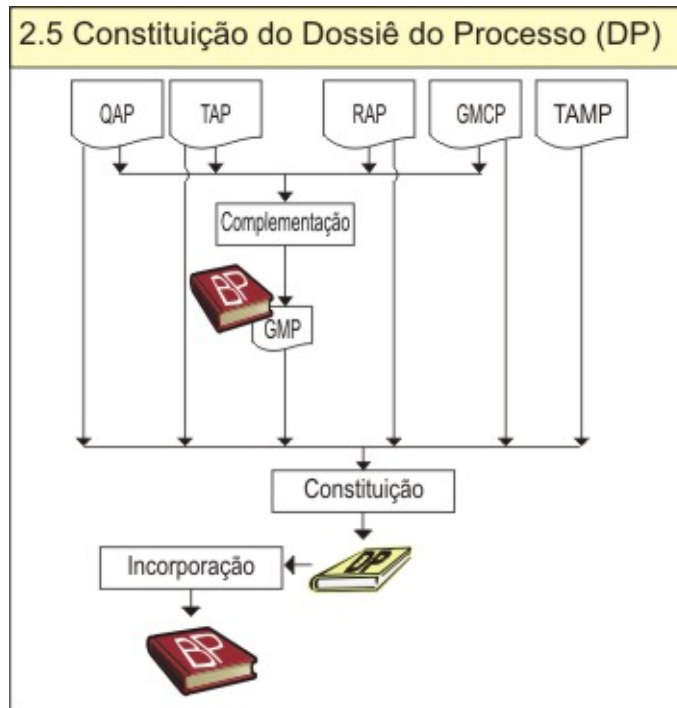


Figura 4.28. A Constituição do Dossiê do Processo (Etapa 2.5)

A partir de sua elaboração, o dossiê de cada processo incorpora-se à BP eleita, como o artefato central orientador e esclarecedor da natureza, constituição e demais características e peculiaridades do processo. Contem ainda a evolução prevista das melhorias a incorporar em suas práticas, na forma que se prevê seja ele utilizado na Organização com o objetivo de garantir sua mais eficiente aplicação na tarefa de se desenvolver *software*.

4.13. A Determinação / Concepção do Ciclo de Melhoria do Processo de *Software* – CMQ-SW (Etapa 3.1)

O MC2Q-SW estipula que, antes de se iniciar cada ciclo, este seja definido e detalhado com o fim de deixar claro a todos os envolvidos **para onde** e **como** se pretende evoluir.

Assim, **antes** do estabelecimento de planos de trabalho, recomenda-se a criação de um artefato que seja o marco-zero, ou o gatilho, do CC ou do CMQ-SW. Esse instrumento é a **Proposta Formal de Ciclo (PFC)**, que pode passar por sucessivas modificações até sua aprovação.

Uma vez aprovada, a PFC converte-se em uma **Guia Mestre de Ciclo (GMC)**, que passa a compor a **Memória do Programa de Melhoria da Qualidade do Processo de *Software* (MPQM)** da Organização.

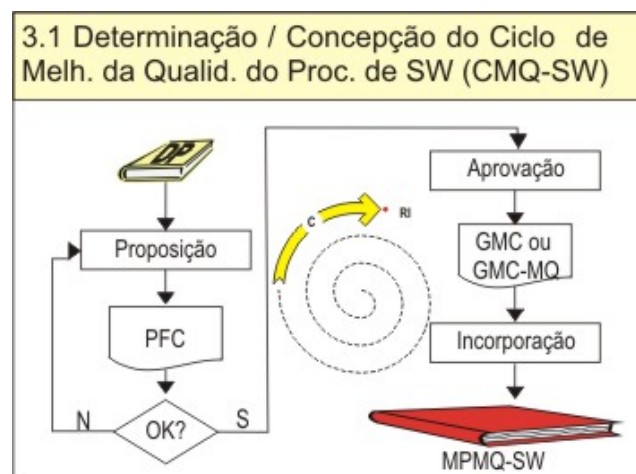


Figura 4.29. A Determinação / Concepção do Ciclo de Melhoria do Processo de *Software* – CMQ-SW (Etapa 3.1)

Os ciclos são estruturados pelas etapas de: Concepção (C); Planejamento (P); Desenvolvimento (D); Implantação (Ip); Institucionalização (It) e Avaliação (Av), conforme ilustra a Figura a seguir.

Reuniões de Apreciação ou Revisão ou Aprovação, denominadas **Reuniões de Inspeção – (RIs)**, devem ocorrer junto ao CQ, pelo menos ao

final das etapas de Concepção, Planejamento, Desenvolvimento e Avaliação (quando deve ser encerrado o Ciclo) sendo que, com o objetivo de imprimir maior velocidade ao Programa de Melhoria de Qualidade, é recomendável que, em paralelo à etapa de Avaliação de um determinado ciclo, se inicie a PFC do novo ciclo para, ao ocorrer a reunião do Comitê da Qualidade para encerramento daquele ciclo, aproveite-se a ocasião para aprovar o planejamento do novo ciclo.

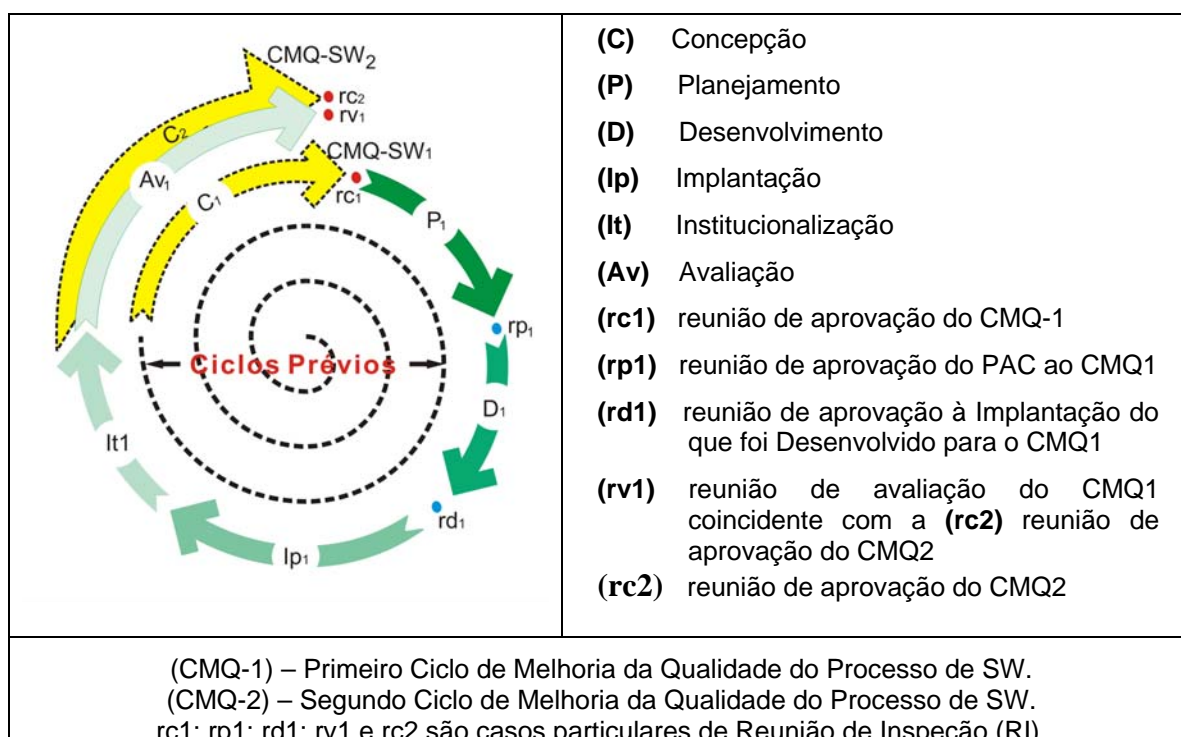


Figura 4.30. Esquema estrutural dos Ciclos de Melhoria da Qualidade

Assim, o exemplo da Figura anterior mostra que ao encerramento do 1º. CMQ-SW sua Reunião de Inspeção estaria ocorrendo com pauta conjunta à proposta do 2º. CMQ-SW.

4.13.1. A Proposta Formal de Ciclo (PFC), a Guia Mestre de Ciclo (GMC) e a Memória do Programa de Melhoria da Qualidade do Processo de Software (MPQM)

A PFC é um componente fundamental e relevante, pois é em função do realismo das metas e missões escolhidas a cada ciclo, e da avaliação dos

resultados obtidos, que se pode verificar o grau de sucesso da implantação do MC2Q-SW ou de um PMQ-SW.

A PFC deve servir de instrumento a avaliar a amplitude, a complexidade e o risco do ciclo, tendo-se sempre em mente que quanto mais abrangente e complexo for o ciclo, maior será a carga de trabalho, mais ações terão de ser desenvolvidas, ou maior será o tempo de duração do ciclo, ou mais especialidades terão de ser aplicadas.

Por isso é que ela tanto pode transformar-se de imediato em uma GMC, quando, por exemplo, o ciclo é de muito baixa amplitude ou complexidade¹⁸, como pode ser objeto de muita deliberação e sofrer sucessivas modificações até poder ser satisfatoriamente convertida na GMC (caso que pode denunciar um ciclo de maior complexidade ou amplitude).

Sua importância consiste no fato de fazer com que idéias propostas de melhoria que impliquem em ultrapassar um **“limite máximo de esforço significativo”**¹⁹ tenham de passar por uma aprovação de um colegiado, no caso o Comitê da Qualidade, que, ou a aprova, ou a reorienta, por exemplo, restringindo ciclos ambiciosos ou fragmentando-os a conjuntos de ciclos mais gerenciáveis. Outra atenção do CQ sobre cada PFC que lhe chegue à aprovação, diz respeito a sintonizar o proposto com o que seja o consenso de mais oportuno à Organização.

A PFC, além de delimitar a ação do ciclo, oferece a oportunidade ao CQ de rejeitar, complementar ou referendar a elaboração de planos de trabalho. Adicionalmente é a melhor forma de comprometer os dirigentes e o CQ com a

¹⁸ No caso dos Ciclos de Capacitação para gestão das ações de capacitação (Etapa 1.2) não se comentou da necessidade de elaboração de uma PFC simplesmente por razões de simplificação da apresentação do MC2Q-SW, uma vez que a GMC-C (Guia Mestra do Ciclo de Capacitação) é a PFC discutida e aprovada, conforme está sendo explicado neste item.

¹⁹ O **“limite máximo de esforço significativo”** é um conceito relativo, que muda de acordo com a Organização. Por exemplo, em uma pequena empresa pode ser algumas horas.pessoa, em outra, considerada grande, pode ser alguns dias.pessoa. O que se deseja ressaltar é que não se deve incentivar que as melhorias em processo permaneçam na informalidade, porém, sem tolher as iniciativas e a criatividade positivas. Assim, o importante é que haja um Limite Máximo de Informalidade tolerável a partir do qual seja impossível persistir na informalidade, e, adicionalmente, haja um controle gerencial periódico que resgate as melhorias ocorridas nessa zona de informalidade, incorporando-as na formalidade através de atualizações na BP e da divulgação e institucionalização dessas melhorias na Organização. Estas idéias são melhor descritas no item 5.5 (Os três Estágios Evolutivos da Organização em direção a um Programa de Melhoria da Qualidade do Processo de *Software*).

carga de trabalho, custos e mudanças a ocorrer na organização, e mantê-los permanentemente atualizados em relação ao PMQ-SW.

Assim, uma vez elaborada uma PFC, esta pode entrar em um “looping” curto ou longo de discussão-modificação-submissão-revisão que retarde sua aprovação, conforme o CQ perceba alguma impropriedade no ciclo proposto.

Daí, supõe-se que, com o correr do tempo e em função da curva de aprendizado, naturalmente a Organização encontrará um padrão de ciclo, especialmente de melhoria da qualidade do processo (CMQ-SW), que produzirá PFCs de rápida aprovação, que gerem ciclos de melhoria eficazes e bastante eficientes.

Uma vez aprovada, incorporando ou não alterações face à discussão no CQ, a PFC passa a ser automaticamente (e sem nenhuma outra modificação) considerada uma **Guia Mestre de Ciclo (GMC)**, e assim identificada, seja como GMC-C, para os casos de Ciclos de Capacitação, seja como GMC-SW, para os casos de Ciclos de Melhoria da Qualidade do Processo de SW.

Assim, a GMC, praticamente um espelho da versão bem sucedida da PFC, se resume na formalização metódica de intenções resultantes de uma Proposta Formal de Ciclo já revisada e aprovada, que implicará, nas etapas posteriores do ciclo; em detalhamento de Planos de Ação, Métricas e tudo o mais essencial e pertinente ao gerenciamento e operacionalização do ciclo em questão, como qualquer projeto.

O exercício de elaboração de cada PFC / GMC não é uma tarefa difícil, pois à medida que a espiral evolui, os novos ciclos resumem-se a réplicas dos ciclos anteriores adicionados ou alterados em detalhes conforme os objetivos estabelecidos ao ciclo.

Se o escopo da GMC for de curta duração, por exemplo, semanal, quinzenal, ou, no máximo mensal, fica mais fácil sua avaliação, pois se for de longa duração, como ocorreu no estudo de caso, os resultados demoram a ocorrer, propiciando descontinuidades. Em qualquer caso, uma avaliação periódica, ao menos quinzenal, deve ser feita.

Como em qualquer processo produtivo, um aspecto fundamental é o **controle** (a função gerencial que permite avaliar se o planejado está ocorrendo conforme se espera). Assim, espera-se que gerentes assumam sua responsabilidade de, periodicamente, acompanharem a melhoria da qualidade do processo através de reuniões rápidas de avaliação.

A seguir é mostrado um exemplo de GMC ora em proposição na empresa piloto para a continuidade do PMQ-SW nessa companhia, uma vez que se considerou encerrada a experiência piloto com a melhoria do processo.

PROJETO: Programa de Melhoria Contínua da Qualidade do Processo de SW na **STORE**

SUBPROJETO: Implantação do MC2Q-SW

ETAPA: Capacitação complementar no MC2Q-SW do restante da equipe.

TAREFA: Implantar o MC2Q-SW **data:** 20/jun/2006

REF: GMC-C_002

Objetivo Imediato: Apresentar os resultados da experiência piloto com o MC2Q-SW à equipe, sua situação na **STORE**, sua vinculação com a Avaliação Anual de Desempenho e estabelecer os Compromissos que resultarão nas GMCs de:

- Diretrizes para Definição e Implantação da Estrutura Básica Inicial do Controle de Qualidade na **STORE**;
- Disparar o Programa de Capacitação em Fundamentos de Controle da Qualidade; PMBOK; PSP e Frameworks de interesse: CMM; CMMI; SPICE; e MPS.BR;
- Disparar o Programa de Capacitação no MC2Q-SW.

Prazo (sugerido 1 semana) de: (a definir) **até:** (a definir)

CRONOGRAMA: (a desenvolver)

OBS: Prazo e cronograma são desnecessários - só interessa na PFC e na GMC

Figura 4.31. Exemplo de uma GMC para um novo Ciclo de Capacitação

Graças ao aprendizado do estudo de caso, que prolongou demasiadamente o primeiro CMQ-SW, o escopo desta GMC foi reduzido. No entanto, percebe-se que, deste Ciclo de Capacitação, vários outros ciclos terão de ser desenvolvidos para dar continuidade a essa capacitação.

4.14. O Planejamento (Etapa 3.2)

Tanto os Ciclos de Capacitação anteriores aos Ciclos de Melhoria da Qualidade do Processo de SW (CMQs-SW) como estes próprios pressupõem o estabelecimento de um Plano de Ação para que possam ser minimamente controlados e propiciar organização e gerenciamento adequados à sua evolução.

O Plano de ação surge a partir do confronto do processo em seu *status-quo* (espelhado na GMP - Guia Mestra do Processo) com as melhorias esperadas, identificadas na GMCP (Guia de Melhoria da Capacitação do Processo). Melhorias que já foram restringidas em amplitude, complexidade e risco na GMC (Guia Mestra do Ciclo) para tornar viável e bem sucedido o próximo “projeto” de Ciclo de Melhoria.

Assim, a etapa **PLANEJAMENTO (P)**, visa garantir que as atividades inerentes a essa importante função gerencial sejam consolidadas em um artefato denominado de **Plano de Ação Circular (PAC)**, que minimamente, pode ser um gráfico de Gantt, onde fiquem claras as metas, prazos, responsabilidades e a força de trabalho pertinentes ao Ciclo de Melhoria da Qualidade que se desenvolverá.

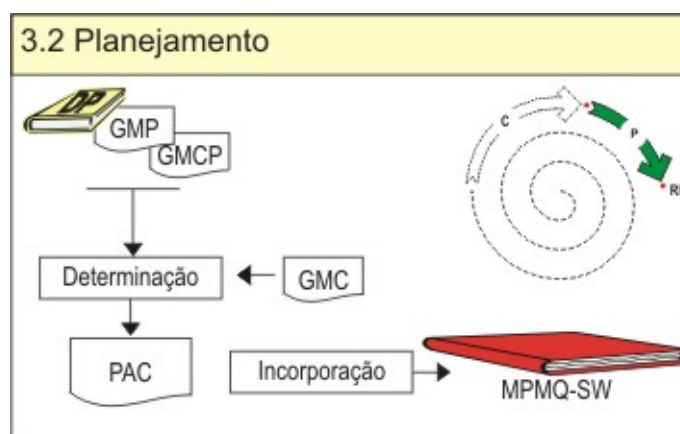


Figura 4.32. O Planejamento (Etapa 3.2)

Assim, o PAC se torna o instrumento gerencial a auxiliar para que:

- a) produzam-se resultados formais de melhoria das práticas inerentes aos processos de *software*, permitindo transformá-los, desde elementares processos “*ad hoc*” para processos maduros e altamente disciplinados;
- b) produza-se informação que propicie a melhoria do desempenho dos futuros CMQs.

4.15. O Desenvolvimento (Etapa 3.3)

A etapa **DESENVOLVIMENTO (D)** visa:

- a) Detalhar, a cada melhoria identificada nas práticas delimitadas, o que deve ser feito;
- b) Especificar artefatos a serem utilizados nessas práticas;
- c) Descrever os artefatos, ao menos através de exemplos ou modelos;
- d) Identificar referências conceituais ou de procedimentos porventura oriundas de outras Bases de Conhecimento das quais se lance mão, além da já constante na BP adotada como paradigma;
- e) Manter o Dossiê do Processo atualizado com tudo o que se desenvolver de melhoria em suas práticas.

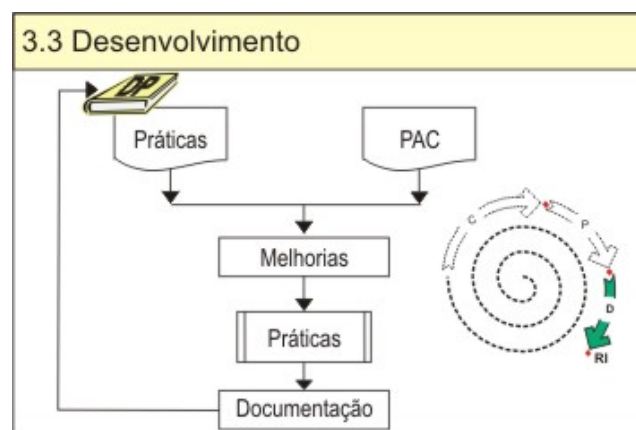


Figura 4.33. O Desenvolvimento (Etapa 3.3)

4.16. A Implantação (Etapa 3.4)

A **IMPLANTAÇÃO (Ip)** é a etapa onde se inicia a institucionalização dos procedimentos de melhoria determinados pelo ciclo. Ao iniciar-se a implantação das melhorias ao processo através de ações concretas em pelo menos um projeto da empresa (considerado como piloto à validação das melhorias propostas) inicia-se também sua divulgação, seja pela de documentação ou interação entre os envolvidos, seja por sessões de treinamento.

As melhorias incorporadas às práticas passam a ser executadas nesse projeto escolhido à medida que o processo, agora remodelado, tiver de ser utilizado.

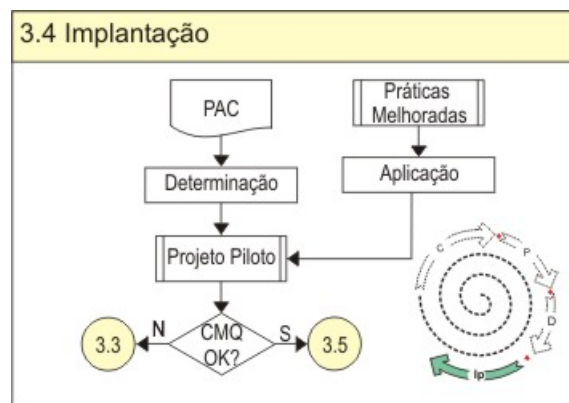


Figura 4.34. A Implantação (Etapa 3.4)

4.17. A Institucionalização (Etapa 3.5)

A **INSTITUCIONALIZAÇÃO (It)** é a etapa que visa garantir que:

- os demais projetos em andamento, de fato, incorporem as mudanças dos processos conforme se planejou ou,
- eventuais adaptações gerais ou específicas a esses processos, decorrentes da aplicação das novas melhorias descritas nos demais projetos da Organização, sejam incorporadas às práticas constantes no DP.

É oportuno observar que a etapa **INSTITUCIONALIZAÇÃO** pode se estender por um período muito longo. Se tal situação ocorrer, é recomendável estabelecer um procedimento de controle em paralelo que garanta a institucionalização do processo, de forma que o prolongamento desta etapa não retarde o disparo de novos ciclos de melhorias de processos que estejam em fila de espera.

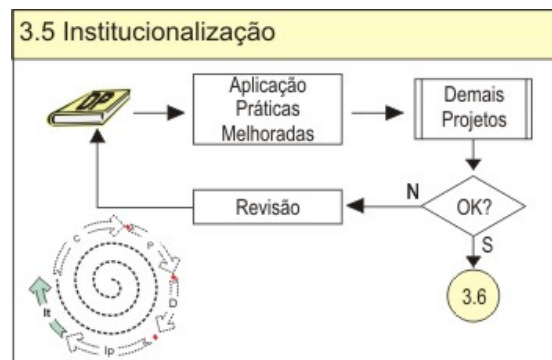


Figura 4.35. A Institucionalização (Etapa 3.5)

A pesquisa-ação serviu a mais este aprendizado, pois o idealizado nem sempre se realiza como se imagina, seja por especificidades não previstas às generalizações iniciais, seja por falhas gerenciais na condução de todo o processo de melhoria. Além disso, é possível surgirem oportunidades de melhoria no processo após sua prática pelos “usuários” (neste caso os engenheiros de sistemas que desenvolvem os diversos projetos da Organização). Assim, quando esse for o caso, recomenda-se que a nova melhoria seja objeto de inovação ou mudança a incorporar em nova versão do processo, gerenciada em um novo CMQ-SW.

É durante esta etapa que o Dossiê do Processo se completa com os resultados advindos das melhorias incorporadas pelo Ciclo de Melhoria da Qualidade a que a Organização tenha se submetido.

4.18. A Avaliação (Etapa 3.6).

A **AValiação (Av)** objetiva apurar os ganhos reais decorrentes dos esforços de melhoria do processo e apontar os incidentes e alterações ocorridas no

decorrer do ciclo para permitir o amadurecimento da utilização deste método e sua continuidade.

É nesta fase que se tem a oportunidade de saber a quantas anda o desempenho do CMQ-SW, se produtivo, prático, eficaz, eficiente, útil.

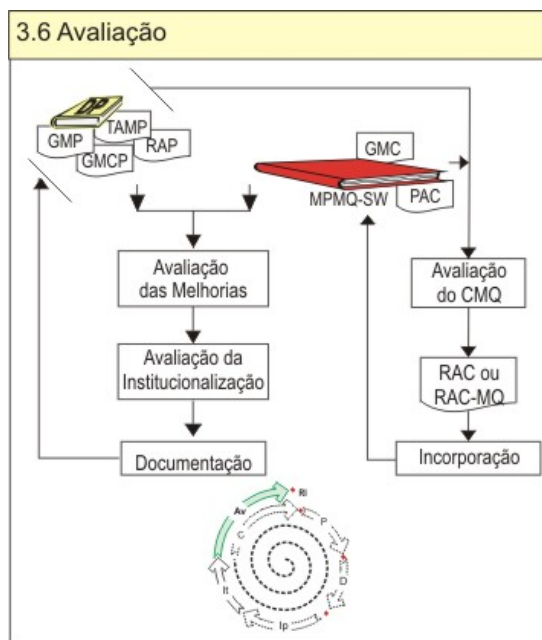


Figura 4.36. A Avaliação (Etapa 3.6)

O MC2Q-SW lança mão das comparações dos estágios de uso das práticas associadas aos processos a ele submetidas, já anteriormente descritas, para avaliar, principalmente, a variação no indicador MCP (Medida da Capacitação do Processo) e o IMCP (Índice de Melhoria da Capacitação do Processo) conforme também já explicados.

Caso as melhorias tenham permitido melhorar a performance de execução do processo, tornando-o mais eficiente, a **Medida de Capacitação do Processo (MCP)** será reavaliada, resultando em um grau superior ao de estágio anterior ao CMQ-SW implementado.

Um relatório final descritivo da evolução da melhoria da capacitação do processo pode ser gerado para documentar o ocorrido no CMQ-SW e tudo o que for considerado pertinente e válido para armazenamento, arquivado no Dossiê do Processo (**DP**) ou na Memória do Programa de Melhoria da Qualidade do Processo (**MPMQ-SW**).

5. A APLICAÇÃO DO MÉTODO

Para o teste de aplicação do MC2Q-SW planejou-se um projeto piloto que permitisse reduzir ao mínimo o envolvimento de pessoas, em situação mais controlada e melhor gerenciada, caracterizando uma experiência que implicasse em custos reduzidos e com o menor impacto aos envolvidos nos demais projetos em curso na Organização escolhida como parceira.

Assim, é sob esta ótica e diretrizes que a experiência de pesquisa-ação se desenvolveu.

As ocorrências de cada Fase e Etapa seguem registradas com miniatura da figura ilustrativa da Fase ou Etapa (com a sugestão de artefatos pertinentes) ao canto esquerdo ou direito do texto para facilitar a referência do leitor.

As figuras ilustrativas dos artefatos utilizados, em sua maioria tabelas apresentadas parcialmente, assim foram apresentadas para facilitar o manuseio da dissertação sem ter de se remeter aos anexos referenciados, onde estão documentadas de forma completa.

5.1. Requisitos de Seleção da Empresa para o Estudo de Caso

Por ocasião do exame de qualificação explicou-se, conforme o slide a seguir, que se buscava uma Organização que atendesse aos seguintes Requisitos Imprescindíveis: INTERESSE; CONDIÇÕES²⁰; ADERÊNCIA e ENGAJAMENTO. Como havia oito alternativas²¹, e duas empresas (PiX-DF e STORE) atenderam todos os requisitos imprescindíveis, lançou-se Mão de ponderação dos seguintes quesitos: INTERESSE; MATURIDADE EMPRESARIAL; DISPONIBILIDADE; ACESSIBILIDADE e PUBLICIDADE; PROXIMIDADE GEOGRÁFICA e CONFLITO DE INTERESSES.

²⁰ As condições referem-se à acessibilidade; publicidade; abrangência; maturidade; proximidade e de não conflito de interesses, conforme ilustrou o slide apresentado.

²¹ duas das quais (P e Y) preferiu-se omitir a identificação por não terem se enquadrado no item “maturidade empresarial”, considerado por demais subjetivo.



Figura 5.1. Diagrama escalar dos requisitos determinantes da escolha da empresa.

Quadro 5.1. Quadro de Decisão determinante à escolha da STORE como empresa escolhida ao estudo de caso

		ROMI	DEDINI	COSAN	SRF	PIX - DF	STORE	P	Y
Critérios Imprescindíveis									
Interesse / Aderência		NC	NC	NC	NOK	OK	OK	OK	NC
Maturidade Empresarial		OK	OK	OK	OK	OK	OK	NOK	NOK
Disponibilidade		NC	NC	NC	NOK - B	OK	OK	NOK	NC
Acessibilidade e Publicidade		NOK	OK	OK	OK	OK	OK	NOK	NC
Critérios Ponderáveis									
	Peso	G	GP	G	GP	G	GP	G	GP
	Interesse	10						10	100
	Maturidade Empresarial	5						10	50
	Disponibilidade	6						6	36
	Acessibilidade e Publicidade	8						10	80
	Proximidade Geográfica	5							5
	Não Conflito de interesses	10						10	100
									366
									410
Análise de Risco									
		NA	P	P	NA	ALTO	baixo	NA	NA

Legenda
 NC = Não Consultada
 OK = OK
 NOK = Não OK
 B = Burocracia / Lentidão
 G = Grau
 GP = Grau Ponderado
 NA = Não aplicável
 P = Prejudicado

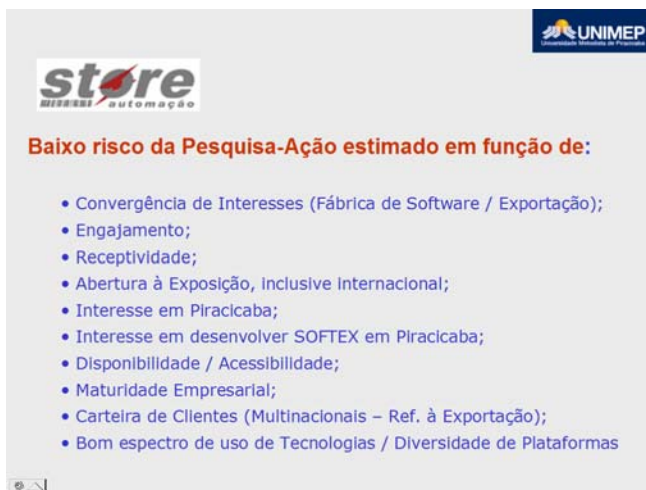


Figura 5.2. Indicadores do baixo risco de descontinuidade da STORE Automação

Adicionalmente procedeu-se a uma análise do risco de descontinuidade a que a pesquisa-ação poderia se sujeitar, conforme exposto à ocasião da Qualificação desta dissertação.

Tais critérios permitiram desempatar as duas alternativas finalistas, deixando claramente demonstrado que a opção pela

STORE Automação, além de obter maior quantidade de pontos, também se mostrava como a de menor risco à descontinuidade do trabalho.

A tabela da Figura 5.2 demonstra a praticidade de uso de critérios Imprescindíveis e Ponderáveis quando, por multiplicidade de alternativas, a decisão de melhor escolha não fica evidente. Por isso é que o MC2Q-SW, baseando-se no método de Tomada de Decisão de Kepner e Tregoe (1997), sugere sejam eles utilizados em qualquer decisão de escolha, como, por exemplo, no caso de seleção de processos na Etapa 2.1,

5.2. Apresentação da Empresa

A STORE Automação é uma *softwarehouse* brasileira considerada de pequeno porte, com um faturamento anual da ordem de US\$1300mil (em 2005), que encontrou seu nicho de mercado no ramo de *software* para operadores logísticos e da automação comercial, industrial e de distribuição.

Atua desde 1991 na prestação de serviços de Tecnologia da Informação e de Automação focando sua atenção na automação da gestão da cadeia de

suprimentos objetivando oferecer soluções de *software* para a logística integrada e atividades afins.

Desenvolve projetos, presta serviços de consultoria e de gerenciamento e implantação de sistemas informatizados e automatizados baseados em tecnologias atualizadas exigidas em seu mercado de atuação (banco de dados relacional, código de barras, aplicações com coletores de dados, transmissão por rádio frequência e intercâmbio eletrônico de dados).

Possui soluções específicas para armazenagem multidepósito e multiciente, envolvendo toda a cadeia de suprimento, ou seja, a logística de Coleta, Armazenagem, Fabricação e Distribuição de Produtos, abrangendo funções como controle de lotes, gerenciamento de ordens de *picking* e *put away*, montagem de *kits*, etc.

Alguns de seus aplicativos de gestão são listados a seguir:

- Sistema de Gerenciamento da Armazenagem e Gestão Operacional de Almojarifados, Depósitos e Centros de Distribuição;
- Sistema de Gerenciamento de Chão de Fábrica e Produção;
- Sistema de Gerenciamento de Transportes, que otimiza de forma ágil, segura e integrada, a gestão da coleta e distribuição de mercadorias e o controle de fretes;
- Sistema de gestão de EADI (porto seco), REDEX e TRA (Terminal Retroportuário Alfandegado) visando otimizar o gerenciamento das atividades de importação, exportação, armazenagem, transportes e integração com a Receita Federal;
- Módulos complementares para o controle de custos, receitas (faturamento de serviços) e documentos fiscais.

Conta com um escritório Central e Administrativo em São Paulo – SP, e com uma fábrica de *Software* em Barra Bonita – interior de SP, onde atuam cerca de 30 colaboradores regularmente contratados, além de contar com consultores em parceria a cada empreendimento que exige conhecimentos

específicos. Mantem equipes de manutenção (especificamente para novas demandas complementares) aos seus sistemas dentro de grandes clientes, alguns multinacionais.

Em Barra Bonita funciona também uma equipe denominada “ON DEMAND” que visa atender sua base instalada de clientes nas solicitações de pequenos serviços de TI complementares aos sistemas já implantados.

Os dados identificadores da STORE Automação foram agrupados na tabela a seguir como base parcial de referência a contextualizar a experiência piloto.

Quadro 5.2. Dados Referenciais Básicos da STORE Automação.		
Capital	:	nacional
Estrutura acionária	:	100% um único empresário
Tipo de Negócio	:	Desenvolvimento de <i>Software</i> Aplicativo
Nicho de Atuação	:	Automação Industrial, Comercial, de Distribuição e Logística
Tempo de atuação no mercado	:	15 anos
Corpo de Dirigentes	:	Profissional, sem vínculo de parentesco
Total de Funcionários regularmente contratados + CEO	:	36
Média anual mdo terceirizada (em 2005)	:	5
Qtd. de Executivos em nível de direção	:	4 (inclue CEO)
Qtd. de Funcionários - Área Comercial	:	3 (inclui diretoria/gerência)
Qtd. de Funcionários - Área Administrativa	:	2 (inclui diretoria/gerência)
Qtd. de Funcionários - Produção/Operações	:	30
Qtd de Escritórios próprios e Instalações de Desenvolvimento de Sistemas	:	2
Cidades/UF das instalações	:	São Paulo (SP) e Barra Bonita (SP)
Qtd. de Empresas Parceiras e de Contratos de Representação	:	4
Estados da Federação onde se localizam Escritórios ou Representantes comerciais	:	SP; RJ; PR; SC
Faturamento nos últimos 6 anos	:	(2000: US\$550); (2001: US\$1100); (2002: US\$1000); (2003: US\$850); (2004: US\$980); (2005: US\$1300);
Qtd. de Clientes	:	15
Qtd. de Clientes Multinacionais	:	3
Estados da Federação onde tem seus sistemas operando em clientes	:	SP RJ PE SC AM DF PA
QtDs de Instalações Físicas onde operam seus sistemas	:	21

Qtd total de terminais em instalações de clientes onde os sistemas ficam disponível para operação	:	250
Plataformas	:	Windows, Windows-CE, Unix-like
SGBD utilizados	:	Oracle, SQL-Server
Linguagens de Programação e ferramentas utilizadas	:	Delphi 5 e 7, Java, Dot-net; Visual Basic 6, Visual Studio 2003, Dreamweaver; J-builder; PL-SQL Developer; Star-Team; Cristal Report; Java Server (Apache Tomcat); Source-Safe, ER-Win (modelagem de BD)
Sistemas de Terceiros que Interfaceiam com seus Sistemas	:	SAP; Datasul; Microsiga; Senior (SAPIENS) e apto a interfacear com demais ERPs do mercado

Em 2003 a STORE se conscientizou ser estratégico implantar um Programa de Melhoria da Qualidade e aceitou a proposta de um trabalho cooperado com o pesquisador através do proposto nesta dissertação. A proposta consistiu em complementar, testar e ajustar um Método de Melhoria Contínua da Qualidade em *Software*, priorizando os processos de Desenvolvimento e Manutenção de *Software*, com base no universo de conhecimento da Engenharia de *Software* e da Gerência de Projetos, na metodologia CMM e normas ISO / IEC 15.504 (SPICE) e ISO/IEC 12207.

5.3. A PRIMEIRA FASE (DIAGNÓSTICO E CAPACITAÇÃO)



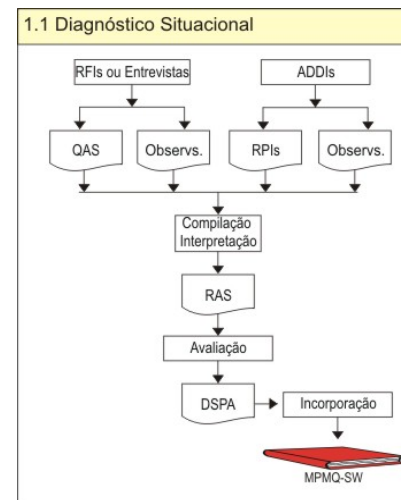
Após os primeiros contatos formais de apresentação à diretoria foram mantidas conversações a respeito da atuação da empresa no mercado, seus clientes e

concorrentes, sua estrutura organizacional, perfil de colaboradores e a visão da alta administração em relação ao futuro e em relação à qualidade.

Desenvolveram-se reuniões formais (RFIs) e Atividades de Desenvolvimento, Diagnóstico, e Integração (ADDIs) conforme o detalhamento a seguir, e promoveram-se algumas ações de capacitação consideradas oportunas à experiência-piloto.

5.3.1. PRÁTICA - A Compreensão do Cenário através do Diagnóstico Situacional - (Etapa 1.1)

Ocorreram duas reuniões formais (RFIs), uma com os dirigentes e outra com os demais colaboradores da área técnica (em Barra Bonita). Como desde o primeiro contato com o empresário e dirigentes da STORE apurou-se não haver dúvidas de que a empresa se enquadrava no estágio inicial de maturidade, não havia necessidade de maiores avaliações, pois não havia indícios a compilar, mas sim, evidências de uma situação perfeitamente conhecida e confirmada nas



demais reuniões ocorridas com praticamente todos os demais colaboradores da STORE em um período de aproximadamente um mês.

No entanto, por ser fundamental conhecer a situação e o clima organizacional vigentes na empresa, além de conseguir a integração necessária entre o pesquisador e os colaboradores para levar a cabo o trabalho, promoveram-se três sessões, em dias e semanas diferentes, de Atividades de Desenvolvimento, Diagnóstico, e Integração (ADDIs).

Os questionários elaborados e as atividades desenvolvidas tiveram como preocupação central, além da ampliação do conhecimento da Organização pelo pesquisador e compreensão do cenário vigente, o desenvolvimento de um clima propício às ações de melhoria da qualidade do processo e a identificação de temores e obstáculos que pudessem causar prejuízo a esse objetivo.

Pode-se considerar que as RFIs e ADDIs se constituíram na primeira interação caracterizadora da pesquisa-ação pela implicação da tomada de consciência dos princípios que conduziriam o trabalho, tornando claro o que seria feito e o porquê seria feito, como aponta McNiff (2002) citado em (TRIPP 2005, pág 449), como um reconhecimento (TRIPP 2005, pág. 453) prévio ao que seria feito em uma modalidade de cooperação (TRIPP 2005, pág. 454), e,

principalmente, de forma deliberada, “turbinando” o modo habitual de aprender com a experiência (TRIPP 2005, pág. 462).

A consolidação das informações, inicialmente transmitida verbalmente à diretoria, além de apontar a capacitação específica necessária, focou basicamente os problemas (de pequena repercussão, porém dignos de preocupação) de percepções diferenciadas das equipes lotadas em São Paulo e em Barra Bonita, como também do corpo de colaboradores vs os dirigentes. Tais percepções, resumidamente expostas a seguir, implicavam no entendimento de que a equipe técnica resistiria a se empenhar em um Programa Formal de Melhoria da Qualidade, pois seria mais uma demanda sem a contrapartida de renegociação dos prazos (já apertados) dos compromissos com projetos em andamento:

- 1) falta de compreensão de uns a respeito do trabalho “dos outros”, mais ainda das dificuldades desse trabalho, subestimando uns o trabalho dos outros (onde uns, e outros aqui se referem como equipes lotadas em São Paulo e em Barra Bonita e vice e versa);
- 2) sentimento generalizado, especialmente na equipe “Fábrica”, de excesso de trabalho, “sufoco”, “pressão constante por resultados”;
- 3) Desejo, especialmente em Barra Bonita, de mais capacitação, seja por treinamento, seja por ferramentas de projeto seja por infra-estrutura de tecnologia e conforto ambiental.

Adicionalmente, uma das ADDIs constituiu-se de uma REI, onde participaram cerca de vinte colaboradores STORE, através da qual apuraram-se 78 Problemas ou Necessidades relacionados a práticas ou ações ou decisões que o corpo técnico da STORE acreditava ser importante resolver.

Essas “necessidades” foram listadas no artefato RAS.01 (ANEXO 5.01) parcialmente apresentado a seguir:

RAS.01 - Relatório de Avaliação de Situação - Apontamentos de Reunião de Eclosão de Idéias para Apuração de Problemas ou Necessidades da Equipe de Barra Bonita da STORE	
Problemas / Sugestões / ATIVIDADE / Processo / Artefato / etc...	
0	
0	
1	elaborar mapa de competências
2	elaborar relatório de aderência
3	inserir atividades no MIS
4	definir fluxos de processos
5	levantamentos informais junto ao cliente através de interações informais sociais
6	trazer o cliente à STORE na fase de elicitação e posteriores
7	montar uma estratégia de conquistar a simpatia do cliente com base no dossiê
8	elaborar e manter permanentemente atualizado um dossiê do cliente
9	promover sempre segunda reunião para rever o entendimento da reunião anterior
10	ao início de cada reunião, sempre rever a reunião anterior
11	elaborar um planejamento de cada reunião
12	identificar no planejamento da reunião as características da empresa, contexto e stakeholders/participante
13	idem ao que está por traz da necessidade do cliente (real causa)
14	elaborar protótipo
15	idem levar sempre alternativas, proposições que conquistem o cliente ajudando a resolver seus problemas
16	elaborar uma ata da reunião de elicitação não deixando dúvidas sobre o negócio/problema do cliente (não se inibir de demonstrar o desconhecimento ao cliente com cautela...)
17	não deixar escapar termos essenciais ao negócio do cliente

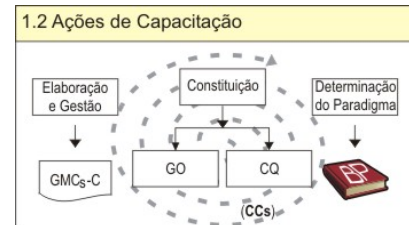
Figura 5.3. Relatório (parcial) RAS.01 de Avaliação da Situação resultante de reunião tipo REI para apuração de Problemas / Necessidades (ver ANEXO.5.01)

O relatório DSPA (Diagnóstico Situacional e Proposta de Ações), conclusivo, fruto da compilação de tudo o observado nas atividades desta etapa, encontra-se no ANEXO 5.02, tendo sido discutido, e considerado aprovado, sem formalidades, com dois executivos, um deles o presidente e um engenheiro de sistemas *senior*.

A Memória do Programa de Melhoria da Qualidade de SW (MPMQ-SW) foi organizada com toda a documentação das atividades e relatórios produzidos nesta etapa, e não apenas os RAS e DSPA.

5.3.2. PRÁTICA - As Ações de Capacitação - (Etapa 1.2)

As Ações de Capacitação decorrentes do Diagnóstico são descritas a seguir, tendo sido desnecessária a aplicação prévia de um artefato de avaliação da capacitação da equipe, embora



houvesse sido desenvolvido (modelo na figura 5.4), por que a experiência se restringiu ao apenas pequeno grupo designado de cinco integrantes na experiência do projeto-piloto.

As seguintes ações de capacitação foram desenvolvidas:

1. relativamente à REORGANIZAÇÃO (GO e CQ) (conforme item 5.3.2.1):
 - Determinação de um responsável (nas instalações de Barra Bonita) que seria o ponto focal das ações de qualidade. O indicado foi o então Gerente da Fábrica de *Software*;
 - Determinação de um Grupo de Trabalho para estudar os modelos; métodos e tecnologias/ferramentas relacionados ao tema; referendar e desenvolver as ações relativas à melhoria da qualidade do processo.
2. relativamente à DETERMINAÇÃO do PARADIGMA (conforme item 5.3.2.2):
 - Eleição do SPICE como a Base de Processos devido à facilidade de o pesquisador interagir com o representante do SPICE no Brasil, sem nenhum ônus de consultoria;
3. relativamente à expansão do Conhecimento e aperfeiçoamento do MC2Q-SW:
 - Aprendizado através de leituras indicadas, especialmente o PMBOK (2003), o CMM (CMU/SEI-93-TR-17 (1993), CMU/SEI -

93-TR-24 (2002) e CMU/SEI - 93-TR-25 (2002))²², e o SPICE (15504 TR (1998) e 15504-5.5 CD (2004));

- Participação de alguns dos membros em treinamentos nessas metodologias/ferramentas e em demais eventos correlacionados à qualidade;
- Amadurecimento e detalhamento do MC2Q-SW por parte do pesquisador;
- Sessões de treinamento, ministradas pelo pesquisador, no MC2Q-SW a toda a equipe.

Entendeu-se que as deficiências de conhecimento do grupo de trabalho, associadas ao grau de incerteza inicial, considerado não-temerário, não deveriam impedir o avanço às etapas seguintes do PMQ, especialmente por tratar-se de uma experiência-piloto.

Um instrumento possível de Avaliação da Capacitação é o ilustrado na figura a seguir, resultante da evolução de um instrumento desenvolvido pelo Gerente da Fábrica de *Software* para a avaliação das competências de seus subordinados, em mais uma demonstração caracterizadora da pesquisa-ação-participativa.

Tal avaliação é apenas ilustrativa, pois foi considerada desnecessária em função do tamanho restrito do grupo envolvido na experimentação, onde todos já conheciam suas limitações e cujas experiências complementares, incluindo a do pesquisador, foram sendo intercambiadas à medida que a experimentação do MC2Q-SW se desenrolou.

A avaliação parte de um elenco de habilitações consideradas necessárias, dispostas nas colunas, a que se atribuem pesos segundo o foco de interesse. Daí, atribuem-se notas aos envolvidos, dispostos nas linhas, segundo um critério a exemplo do sugerido no quadro menor superior ao canto esquerdo intitulado “Perfil de Qualificação”. O resultado ponderado das notas pode ser confrontado com uma grade pré-determinada de resultados que

²² O CMMI à época ainda estava embrionário no Brasil.

indique o risco ou as ações que devem ser tomadas de capacitação do grupo para a implantação de um PMQ-SW.

AValiação de Capacitação

Perfil de Qualificação		STORE (hipotético)									
	Nenhum conhecimento	Revisado em jul.2005 Ciclo de Capacitação ao PMQ - STORE Responsável: Navas									
2	Participou de treinamento										
4	Domínio básico										
6	Experiência prática										
8	Muito Boa Experiência										
10	M.B.Experiência e Domínio Teórico										

		Habilitação									
		Domínio do MC2Q-SW	PMBOK	CMM ou CMMI	SPICE	ISO 12207	X-P e Agile Develop. Method	Técnicas de Eng. de Sistemas	Programação OO	UML	
Colaborador (e A.P.) \ PESO		10	2	2	2	2	5	5		10	
Aldo	Avaliação Ponderada		2				2	6			
Bia	Avaliação Ponderada		4				10	30			
Cido	Avaliação Ponderada		2					6			
Douglas	Avaliação Ponderada		4	4	4	4	4	6			
Eduardo	Avaliação Ponderada		40	8	8	8	8	20	30		
X	Avaliação Ponderada		4	2			2		8		
	Avaliação Ponderada		40	4			4		40		
	Avaliação Ponderada		6	4	4	4	4	4	8		
	Avaliação Ponderada		60	8	8	8	8	20	40		
	Avaliação Ponderada		100	4	4	4	4	6	8	2	
	Qtd. de Colaboradores		6								
	SOMATORIO PONDERADO		260	32	24	24	28	80	210	20	
	Média Ponderada		43,3	5,33	4	4	4,67	13,3	35	3,33	
	Desvio Padrão		34,4	3,27	4,38	4,38	3,93	12,1	5,48	8,16	
	Variância		1187	10,7	19,2	19,2	15,5	147	30	66,7	

(*) baseado em MAPA DE COMPETÊNCIAS INDIVIDUAIS - Roger (Gte. Da Fábrica)

Figura 5.4. Esquema de Avaliação de Capacitação

A Guia Mestre do Ciclo de Capacitação (GMC-C) resultante para o projeto piloto não foi bem elaborada, em função de seu escopo muito abrangente, o que serviu de lição a incorporar ao MC2Q-SW quanto à limitação de objetivos aos Ciclos, sejam de Capacitação, sejam de Melhoria da Qualidade. Sua ilustração é apresentada na figura a seguir, conforme originalmente concebida:

PROJETO: Programa de Melhoria Contínua da Qualidade do Processo de SW na **STORE**
SUBPROJETO: Implantação do MC2Q-SW
ETAPA: Capacitação INICIAL no MC2Q-SW.
TAREFA: Capacitar-se como Laboratório à Experiência Piloto. **data:** 20/jun/2004
REF: GMC-C_001
Objetivos: Organizar Grupo de Trabalho; Determinar Responsabilidades; Treinar-se no ESSENCIAL e Determinar a Base de Processos.

- CMM; CMMI; SPICE; e MPS.BR;
- Disparar MC2Q-SW.

Prazo: (sugerido 4 meses)
CRONOGRAMA: (a desenvolver)

Figura 5.5. GMC para o Ciclo de Capacitação piloto

5.3.2.1. A Reorganização com vistas à Experiência-Piloto

Infelizmente não se deu a ênfase à constituição de um Comitê da Qualidade (CQ) designando-se o então Gerente da Fábrica de *Software*, que era um entusiasta do Programa de Melhoria da Qualidade, como responsável (nas instalações de Barra Bonita) para a experiência-piloto de melhoria da qualidade do processo de *software*.

Também se formou um Grupo de Trabalho para estudar os modelos; métodos e tecnologias/ferramentas relacionados ao tema e referendar e desenvolver as ações relativas à melhoria da qualidade do processo. Essa equipe foi composta por 5 representantes da STORE, mais o pesquisador:

- ✓ um Diretor, o de Tecnologia (Ditec);
- ✓ um Gerente, o da Fábrica de *Software* (GteFSW);
- ✓ um Gerente de Projetos (GteProj);
- ✓ dois analistas de sistemas.

Finda a experiência piloto do MC2Q-SW e tendo o pesquisador encerrado seu trabalho, ficou à STORE a recomendação de implantar o GO e o CQ, como passos seguintes à continuidade do PMQ-SW.

5.3.2.2. A Determinação do Paradigma para a Base de Processos

Não se decidiu, de imediato, qual seria a Base de Processos a eleger, pois se acreditava ser necessário um aprofundamento do conhecimento dos modelos CMMI e SPICE, antes dessa decisão. Paralelamente surgia uma iniciativa no mercado, de um novo modelo, o MPS-Br, que achou-se melhor avaliar.

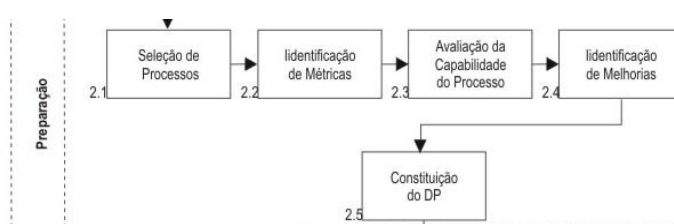
Isto retardou a escolha da Base de Processos (BP), por cerca de dois meses, até o instante da decisão sobre qual processo deveria ser o escolhido como o primeiro a submeter ao MC2Q-SW.

Na evidente indecisão que procrastinava a escolha da BP a um horizonte imprevisível, ficou decidido que se utilizaria como paradigma o SPICE (15504-5.5 CD, 2004), ainda em versão preliminar de revisão (em inglês).

A referida Base de Processos prevê 48 processos (baseados na ISO 12.207) e 6 níveis de capacitação, agrupados em 9 Principais Grupos de Processos, conforme o esquema já ilustrado à figura 4.23.

5.4. A SEGUNDA FASE (PREPARAÇÃO)

Foco no Processo



A fase de PREPARAÇÃO demonstrou-se como a mais complicada, pois a questão de determinação de métricas a

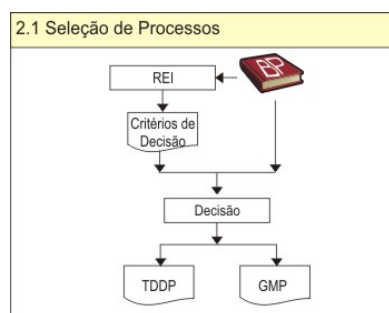
respeito da capacitação dos Processos como apresentadas pelos modelos SPICE e CMM geravam resistência velada ao grupo de trabalho. A alternativa foi encontrar um caminho de avaliação da Capacitação do Processo que não exigisse muita atividade nem estudo ou pesquisa da profusão de métodos disponíveis indicados pelo pesquisador.

Optou-se, portanto, pelo desenvolvimento e aplicação de um método prático que gerou a teoria desenvolvida em 4.9 (Identificação de Métricas).

5.4.1. PRÁTICA - A Seleção dos Processos (Etapa 2.1)

Utilizou-se a técnica de Reunião de Ecloração de Idéias (REI) para a determinação do processo ideal para iniciar a aplicação do MC2Q-SW com base nos critérios mostrados no quadro abaixo.

Foram 3 as alternativas de Critérios de Decisão aventadas à ocasião em que venceu a alternativa “Aa”



para o objetivo estabelecido de **“selecionar um ou mais processos para promover a melhoria da capacitação através da aplicação do MC2Q-SW”**.

Quadro 5.3. Quadro Simplificado de Decisão para Determinação do Processo a Submeter ao 1º. CMQ-SW		
Instante	Critérios Imprescindíveis	Critérios Ponderáveis
Aa	3. Processo mais crítico. 4. Processo que seja praticado pelo maior número de engenheiros de sistemas.	Nenhum além dos imprescindíveis.
Ab	2. Processo de grande interação com o cliente.	3. Maior grau de interação com o cliente 4. Maior ganho em performance decorrente das melhorias já identificadas por fazer
Ac	2. Processo de maior absorção de recursos.	3. Maior grau percebido de redução de mão de obra alocada 4. Maior possibilidade de aplicação de ferramentas de produtividade.

Os critérios que orientaram a escolha do processo foram, portanto, apenas dois, um de impacto/criticidade e outro de atratividade, conforme a alternativa “Aa”, que, de certa forma, incorpora, ainda que parcialmente, os critérios da alternativa “Ab”.

O processo ideal deveria:

- a) ser o foco ou causa da maioria dos problemas que a STORE enfrentava no desenvolvimento de sistemas; e
- b) cujo conjunto de melhorias em suas práticas se mostrasse o mais atrativo para servir de ilustração ao aprendizado dos demais colaboradores que não tivessem tido oportunidade de participar da experiência-laboratório;

Em atenção ao critério (a) aproveitou-se o levantamento das 78 “necessidades” apurado na Etapa 1.1 (Diagnóstico Situacional) procurando-se identificar, através do cruzamento de tais necessidades de melhoria contra os 12 processos estruturados na Base de Processos do SPICE relativos ao **Grupo de Processos de Engenharia**.

O critério (b) mostrou-se de suma importância por permitir reduzir a amplitude e o escopo do primeiro Ciclo de Melhoria do Processo a servir de experiência-piloto, conforme se demonstrará no item 5.5.1. (Determinação/Concepção do CMQ-SW).

Ao efetuar tal cruzamento, das “necessidades” em RAS.01a, com o processo ENG.1 (Elicitação de Requisitos), saltou aos olhos que esse, senão o mais importante, era um processo vital para a STORE, conforme se documentou no relatório RAS.01b (ANEXO 5.03), parcialmente mostrado na figura que segue.

A1	B	C	D	E
	RAS.01b - Prévía à Tabela de Decisão de Determinação de Processo - Cruzamento de Problemas ou Necessidades com os PROCESSOS da BP (restringiu-se ao ENG.1 - Elicitação de Requisitos)			
2	0			
3	0	Problemas / Sugestões / ATIVIDADE / Processo / Artefato / etc...	F ou D	ENG.1
4	1	elaborar mapa de competências		1
5	2	elaborar relatório de aderência		1
6	3	inserir atividades no MIS		
7	4	definir fluxos de processos		1
8	5	levantamentos informais junto ao cliente através de interações informais sociais	D	1
9	6	trazer o cliente à STORE na fase de elicitação e posteriores	D	1
10	7	montar uma estratégia de conquistar a simpatia do cliente com base no dossiê	D	1
11	8	elaborar e manter permanentemente atualizado um dossiê do cliente	D	1
12	9	promover sempre segunda reunião para rever o entendimento da reunião anterior		1
13	10	ao início de cada reunião, sempre rever a reunião anterior	D	1
14	11	elaborar um planejamento de cada reunião	D	1
15	12	identificar no planejamento da reunião as características da empresa, contexto e stakeholders/participante	D	1
82	501		F ou D	ENG.1
83	503	F = Futuramente ; D = Desejavelmente		
84	504	ENG.1 - Requirements Elicitation		

Figura 5.6. Relatório parcial de Avaliação da Situação utilizado para a referência cruzada

Uma vez percebido que as “necessidades” se concentravam em práticas relacionadas a esse processo decidiu-se ser desnecessário prosseguir em discussões a respeito de demais processos e sequer desenhar a TDDP correspondente, por sua simplicidade. Caso fossem necessários critérios variados, onde pudessem confundir a análise para a tomada de decisão, então, nesse caso, sim, teria sido recomendável sua classificação em Critérios Imprescindíveis e Ponderáveis aplicados em uma tabela similar à utilizada na decisão de escolha da empresa-caso a esta pesquisa-ação conforme se demonstrou em 5.1 (Requisitos de Seleção da Empresa).

A complementação do cruzamento dessas necessidades com os demais processos foi deixada para futuros CMQs-SW que pudessem considerar ser essa avaliação oportuna.

O RAS.01b permitiu observar que a não padronização do processo Elicitação de Requisitos associada à pressão por resultados, tanto da chefia como dos clientes, usualmente conduzia a desentendimentos, problemas de

comunicação, entrega de serviços incompletos, ou um tanto diferentes do esperado, segundo a ótica dos clientes.

Isso causava muito desgaste de relacionamento, estresse nos técnicos, insatisfação tanto na chefia como nos analistas por conflitos de avaliação, perda de tempo em explicações e retrabalhos, atrasos, não só na entrega do serviço em questão como em outros serviços, pelo desvio de dedicação da mão de obra.

A documentação era precária, com pouca fluxogramação, com poucos cuidados nas validações e aceite, sem quase nenhuma prototipação. O interesse em aplicar UML e ferramentas CASE era grande, mas a Diretoria de Tecnologia não obtinha sucesso em determinar e implantar essas tecnologias com a velocidade que gostaria.

A seguir (Quadro 5.4) encontra-se fragmento da primeira GMP (GMP-ENG.1) (ANEXO 5.04) que, tomando emprestada da Base de Processos adotada (SPICE), a descrição do processo escolhido, ENG.1 (Elicitação de Requisitos), constituiu a primeira peça do Dossiê do Processo, a surgir, formalmente, na etapa 2.5.

Quadro 5. 4. A Guia Mestra do Processo ELICITAÇÃO DE REQUISITOS (ENG.1)

GMP-ENG.1
Guia Mestra do Processo ELICITAÇÃO DE REQUISITOS (ENG.1)

Process ID	ENG.1	Updated on	30/01/2004
Process Name	Requirements elicitation		
Process Purpose	The purpose of the Requirements elicitation process is to gather, process, and track evolving customer needs and requirements throughout the life of the product and/or service so as to establish a requirements baseline that serves as the basis for defining the needed work products. Requirements elicitation may be performed by the acquirer or the developer of the system.		
Process Outcomes	As a result of successful implementation of Requirements elicitation process: 1) continuing communication with the customer is established; ... 6) enhancements arising from changing technology and customer needs are identified and their impact managed.		

Base Practices	<p>ENG.1.BP1: Obtain customer requirements and requests. Obtain and define customer requirements and requests through direct solicitation of customer and user input.</p> <p>...</p> <p>ENG.1.BP6: Establish customer query mechanism. Provide a means by which the customer can be aware of the status and disposition of their requirements changes. [Outcome: 5]</p> <p>NOTE 3: This may include joint meetings with the customer or formal communication to review the status for their requirements and requests; Refer to the process QUA.4 Joint Review.</p>
-----------------------	--

5.4.2. PRÁTICA - A Identificação de Métricas (Etapa 2.2)

A avaliação inicial do processo, quando se iniciou a experiência do MC2Q-SW através da Medida da Capacitação do Processo

2.2 Identificação de Métricas



(MCP) resultou no Pentágono MCP da figura abaixo, cuja interpretação é clara: pouco entendimento formal existia sobre a ELICITAÇÃO DE REQUISITOS, nenhuma documentação a seu respeito, confusão entre a contextualização desse processo na estrutura de processos utilizados na construção de *software*, e nenhuma formalização a respeito de suas práticas.

Determinou-se um MCP11111 ao ENG.1 para não se penalizar excessivamente a situação, visto que a empresa praticava a Elicitação de Requisitos, embora de forma não padronizada e com base teórica elementar; tinha noção das melhorias necessárias, mas não se articulava coordenada nem metodicamente em praticá-las; e não tinha domínio de ferramentas gráficas, com baixo uso de fluxogramação.

	D	I	E ₁	E ₂	E ₃
A	1	1	1	1	1

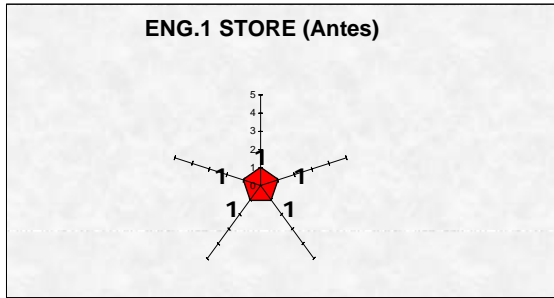


Figura 5.7. A primeira avaliação de processo levada a efeito, via o indicador MCP, para a ELICITAÇÃO DE REQUISITOS (ENG.1) na STORE

Ou seja, era notória a situação de um processo que exigiria muito esforço em DEFINIÇÃO, INCORPORAÇÃO, padronização de EXECUÇÃO, compreensão e gerenciamento da EVOLUÇÃO de suas práticas até atingir um alto grau de capacitação aproximando-se da EXCELÊNCIA.

Posta esta situação, e entendido que bastaria elencar e avaliar as práticas associadas ao ENG.1, compôs-se a Tabela de Avaliação da Melhoria do Processo (TAMP.01) com a lista oriunda do RAS.01b (ANEXO 5.03), conforme a figura a seguir.

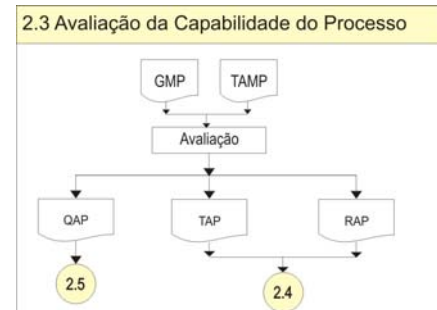
A1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
2	1a	Avaliação das MELHORIAS em PRÁTICAS após 14 meses																
3	0	TAMP.01 - Tabela de Avaliação da Melhoria do Processo				A PRÁTICA já ocorria, no ENG.1, em out.04 em até x% do ideal:					Hoje (jan.2006) a prática ocorre no ENG.1 em até y% do ideal:					AVANÇO de Melhoria OCORRIDO via Práticas no ENG.1 entre		
4	0	ELICITAÇÃO DE REQUISITOS (ENG.01)																
5	0	PRÁTICAS (ATIVIDADE/ Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)		C => U; D; F	ENG. 01	zero %	até 25%	até 50%	até 75%	até 100%	ATÉ	zero %	até 25%	até 50%	até 75%	até 100%	ATÉ	
9	4	definir fluxos de processos		U ₀	1						0						0	A calcular
13	8	elaborar e manter permanentemente atualizado um dossiê do cliente		U ₀	1						0						0	A calcular
19	14	elaborar protótipo		U ₀	1						0						0	A calcular
41	36	não adianta apenas documentar, mas explicar item a item o que foi levantado apresentando ao cliente		U ₀	1						0						0	A calcular
57	52	Mudar comportamento => Desenhar mais Fluxogramas		U ₀	1						0						0	A calcular
58	53	Prover Recursos => Ferramenta simples de fluxogramação		U ₀	1						0						0	A calcular
60	55	Utilização de fluxogramação com notação UML;		U ₀	1						0						0	A calcular
77	72	elaborar um dossiê de cliente (DOSSIE)		U ₀	1						0						0	A calcular
84	90	C (considerando) => U (Urgente); D (Desejável); F (Quem sabe Futuramente)																
85	90	Preencher com 1 quando a PRÁTICA está relacionada ao Processo em questão																
86	90	Indicar em quantos % a prática já era utilizada no processo em foco (ELICITAÇÃO) em out.04.																
87	90	Preencha com 0, 25, 50, 75 ou 100 na célula apropriada conforme seja o caso.																
88	90	Exemplo:																
89	90	Se a utilização do UML na ELICITAÇÃO (ENG.1) era nula, então preencha 0 na célula F60																
90	90	mesmo que o uso do UML fosse total em demais PROCESSOS como, hipoteticamente, no ENG.5 (SW Design)																
91	90																	
92	90	Indicar em quantos % do uso ideal da prática nesse processo, ela está sendo aplicada, hoje, no processo em foco (ENG.01).																
93	90	Preencha com 0, 25, 50, 75 ou 100 na célula apropriada conforme seja o caso.																
94	90	Exemplo:																
95	90	Se a utilização do UML na ELICITAÇÃO (ENG.1) atende, hoje, 75% do que poderia ser aproveitada nesse processo,																
96	90	então preencha 75 na célula O60, mesmo que o uso do UML seja incipiente em demais PROCESSOS como,																
97	90	hipoteticamente, no ENG.5 (SW Design).																
98	90	Observe que se os exemplos ocorrerem, então o uso do UML subiu de 25% para 75% do uso ideal na ELICITAÇÃO.																

Figura 5.8. A Proposição da TAMP ao Processo ENG.1 (Elicitação de Requisitos)

Observe-se que a TAMP já foi estruturada a permitir, que ao final do MCQ-SW, se possa obter o **Índice de Melhoria da Capacitação do Processo (IMCP)** conforme previsto em 4.9.3.

5.4.3. PRÁTICA - A Avaliação da Capacitação do Processo (Etapa 2.3)

Ainda guiado pela GMP-ENG.1, avaliou-se, quando aplicável, cada uma das práticas apontadas em seu “estágio de uso”, diante da possibilidade de uso pleno, através da escala percentual (conforme explicado em 4.9.3) onde 100%, obviamente, representava a prática em toda sua potencialidade.



Não se utilizou nenhum questionário (QAP - Questionário de Avaliação da Capacitação do Processo), pois a determinação do critério de avaliação pelo “estágio de uso” das práticas demonstrou que, neste caso de uma experiência-piloto, seria muito mais proveitoso trabalhar diretamente nas melhorias das práticas identificadas, assim a referência ao artefato QAP fica como um “*check-point*” a futuros CMQs que entendam ser necessário utilizar-se de questionários, como os sugeridos nos trabalhos de outros pesquisadores ou previstos nos modelos consagrados.

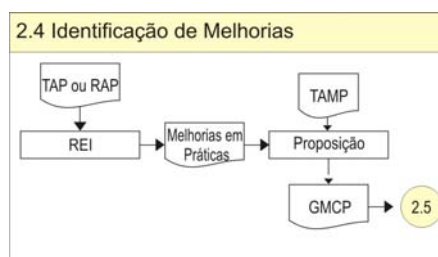
O TAP.01 (ANEXO 5.05), parcialmente mostrado na figura 5.9 é um relatório constituído de simples tabela auto-explicativa, tendo sido considerado desnecessário, neste caso particular da experiência piloto, criar um relatório adicional RAP, pois nada mais essencial, oportuno ou importante se considerou necessário acrescentar para o entendimento da Capacitação do Processo de ELICITAÇÃO.

A1	B	C	E	F	G	H	I	J	K
		Avaliação dos 78 PROBLEMAS / NECESSIDADES ou PRÁTICAS antes do 1o. CMQ							
2	0								
3	0	TAP.01 - Tabela de Avaliação (INICIAL) do Processo			A PRÁTICA já ocorria, no ENG.1, em out.04 em até x% do ideal:				
4	0	ELICITAÇÃO DE REQUISITOS (ENG.01)							
5	0	PRÁTICAS (ATIVIDADE / Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)	ENG.01	zero %	até 25%	até 50%	até 75%	até 100%	ATÉ
6	1	elaborar mapa de competências	0						0
7	2	elaborar relatório de aderência	1	0					0
8	3	inserir atividades no MIS						100	100
9	4	definir fluxos de processos	1		25				25
10	5	levantamentos informais junto ao cliente através de interações informais sociais	1		25				25
11	6	trazer o cliente à STORE na fase de elicitação e posteriores	1	0					0
12	7	montar uma estratégia de conquistar a simpatia do cliente com base no dossiê	1	0					0
13	8	elaborar e manter permanentemente atualizado um dossiê do cliente	1	0					0
14	9	promover sempre segunda reunião para rever o entendimento da reunião anterior	1	0					0
15	10	ao inicio de cada reunião, sempre rever a reunião anterior	1	0					0
16	11	elaborar um planejamento de cada reunião	1	0					0
17	12	identificar no planejamento da reunião as características da empresa, contexto e stakeholders/participante	1		25				25

Figura 5.9. Vista parcial da TAP para a avaliação INICIAL da Capacitação do Processo ENG.1 (Elicitação de Requisitos). (ver ANEXO 5.05)

5.4.4. PRÁTICA - A Identificação de Melhorias (Etapa 2.4)

As 78 necessidades de melhoria, agora agrupadas no TAP.01, foram submetidas a uma classificação de prioridades, face às limitações de tempo e recursos, fazendo surgir a GMCP-ENG.1 (ANEXO 5.06), a primeira Guia de Melhoria da Capacitação do Processo ENG.1.



A GMCP-ENG.1 tornou-se, desta forma, o “*check-list*” de todas as melhorias, agrupadas conforme prioridades, que deveriam ser perseguidas até que as práticas do ENG.1 pudessem ser consideradas as melhores práticas a tornar a ELICITAÇÃO DE REQUISITOS um processo de alta capacitação.

Ficou claro que muitos CMQs- SW teriam de ser percorridos, e ainda mais alguns de Capacitação, para que se atingisse a excelência da capacitação do processo ENG.1, como já era, de fato, esperado face ao estágio de desenvolvimento em que se encontrava a empresa estudo de caso e ao caráter evolutivo das melhorias no decorrer do tempo.

A figura a seguir mostra fragmentos da GMCP-ENG.1 que se apresenta completa no ANEXO 5.06.

Classificação de PROBLEMAS / NECESSIDADES ou PRÁTICAS antes do 1o. CMQ									
GMCP-ENG.01 - Guia da Melhoria da Capacitação do Processo ELICITAÇÃO DE REQUISITOS (ENG.01)			A PRÁTICA já ocorria, no ENG.1, em out.04 em até x% do ideal:						
0	PRÁTICAS (ATIVIDADE / Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)	C => U; D; F	ENG. 01	zero %	até 25%	até 50%	até 75%	até 100 %	ATÉ
4	definir fluxos de processos	U ₀	1		25				25
8	elaborar e manter permanentemente atualizado um dossiê do cliente	U ₀	1	0					0
14	elaborar protótipo	U ₀	1		25				25
36	não adianta apenas documentar, mas explicar item a item o que foi levantado apresentando ao cliente	U ₀	1	0					0
31	elaborar curvas de Pareto (C ₃ U ₂₃₄)	U _m	1	0					0
32	montar uma base de conhecimento das demandas do cliente (C ₃ U ₂₃₄)	U _n	1	0					0
33	montar uma base de conhecimento de atas (C ₃ U ₂₃₄)	U _q	1	0					0
62	ENTREVISTAS (ENTREV)	U _r	1	0					0
63	elabora questionário (ENTREV)	U _r	1	0					0

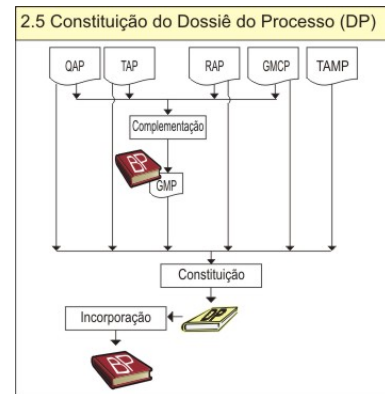
Figura 5.10. Vista parcial da GMCP para a Elicitação de Requisitos. (Ver ANEXO 5.06)

5.4.5. PRÁTICA - A Constituição do Dossiê do Processo (Etapa 2.5)

Todo o material desenvolvido até então passou a constituir o Dossiê do Processo, complementando a GMP-ENG.1.

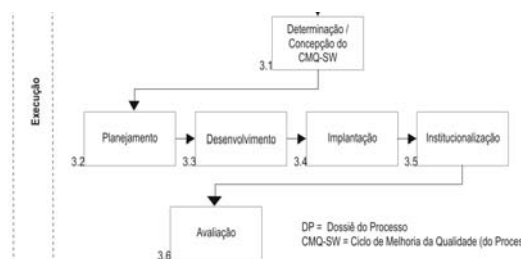
O Dossiê do Processo, similarmente à Memória do Programa de Melhoria da Qualidade de SW (MPMQ-SW) foi organizado com toda a documentação das atividades e relatórios produzidos

nesta etapa.



5.5. A TERCEIRA FASE (Execução)

Foco no Processo



Graças a todo o trabalho antecedente e à ênfase em organização e planejamento, esta foi a fase de mais fácil implementação, pois as medidas previstas foram implementadas sem desvios significativos do planejado.

No entanto, um incidente prejudicou a EXECUÇÃO: a demissão, a pedido, do gerente da Fábrica, coincidentemente o gerente do projeto de melhoria da qualidade, ao qual o grupo se subordinava.

Como a execução já mostrava seus resultados, considerados satisfatórios relativamente ao teste do MC2Q-SW, e como a pressão por novos projetos a disponibilizar ao mercado era crescente, a STORE Automação protelou o prosseguimento da implantação do MC2Q-SW à contratação de um novo gerente da Fábrica. Deu-se, então, por encerrado o CMQ-SW piloto apesar de a etapa de INSTITUCIONALIZAÇÃO não ter ocorrido em sua plenitude, conforme explicado mais à frente.

A decorrência da demissão e vacância do cargo de gerência, assumida provisoriamente por outros colaboradores da STORE Automação²³ foi a de que o grupo se dispersou e, assim o prazo de conclusão do CMQ-SW piloto dilatou-se exageradamente. Também diversas das melhorias previstas que deveriam ser implementadas através de novos CMQs-SW foram sendo praticadas em meio à informalidade, da mesma maneira que ocorriam anteriormente, algumas em paralelo à implementação do CMQ-SW piloto, sem nenhuma formalização.

Mais uma vez a característica de pesquisa-ação mostrou-se proveitosa, pois foi graças à aplicação prática que se elaborou o Modelo Evolutivo dos três Estágios da Organização em direção a um Programa de Melhoria da Qualidade do Processo de *Software* (descrito em 4.5) e da Síndrome da Fachada (descrita em 4.3.1). Tanto o referido modelo como a situação de Síndrome da Fachada resultaram das constatações propiciadas pela pesquisa-ação, especialmente as decorrentes da percepção gerencial dos participantes a respeito da atratividade de retornos a projetos (ver nota de rodapé do item 4.3.1) e o “comodismo” gerencial relegando a melhoria da qualidade à iniciativa individual, foram fundamentais

²³ Notar que, a fábrica localiza-se em Barra Bonita, distante do escritório de São Paulo cerca de 4 horas por rodovia, único meio atual de transporte viável à empresa, e que os gerentes concentram-se nesse escritório em São Paulo. Também cabe observar que a oferta de mão de obra gerencial para Tecnologias de Informação concentra-se na Grande São Paulo. Assim, a dificuldade da STORE em preencher um cargo gerencial vago, é sempre um grande problema que tem de ser contornado com medidas paliativas até a solução definitiva de contratação de um candidato adequado que se disponha a residir em Barra Bonita.

5.5.1. PRÁTICA - A Determinação / Concepção do Ciclo de Melhoria da Qualidade do Processo de Software – CMQ-SW (Etapa 3.1)

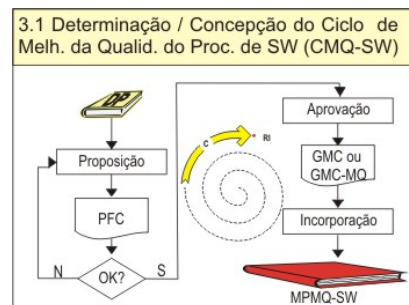
Tendo sido identificado que o processo a submeter à melhoria seria o de ELICITAÇÃO DE REQUISITOS, focou-se a amplitude do ciclo.

Uma vez constatada a grande quantidade de melhorias a implementar, a viabilização do ciclo só

foi possível graças à observação do critério (b) indicado na ETAPA.2.1²⁴ por ocasião da elaboração da Proposta Formal de Ciclo PFC.01, que reduziu o escopo de melhorias a um mínimo de práticas inter-relacionadas ao que se identificou como adequadamente gerenciável.

Tendo sido verificado um interesse muito grande de praticamente a totalidade dos colaboradores da STORE na ferramenta UML, e como havia uma consciência geral de se utilizar mais fluxogramação, e aperfeiçoá-la, objetivando documentação mais eficiente, nada mais natural que promover uma melhoria no processo de ELICITAÇÃO relacionada à fluxogramação apoiada em UML.

A figura a seguir mostra a filtragem que se procedeu à lista das 78 melhorias em práticas originalmente identificadas, agora formatadas na GMCP-ENG.1a. com o objetivo de redução do escopo para viabilizar a avaliação de um ciclo experimental de melhoria da qualidade do processo de SW.



²⁴ “um conjunto de melhorias que se mostrasse o mais atrativo a servir de ilustração ao aprendizado dos demais colaboradores que não tivessem tido a oportunidade de participar da experiência-laboratório”

Filtragem de PROBLEMAS / NECESSIDADES ou PRÁTICAS para o 1o. CMQ.
GMCP-ENG.1a - Guia de Melhoria da Capacitação do Processo ELICITAÇÃO DE REQUISITOS (ENG.1)
PRÁTICAS
1 - Privilegiar fluxogramação ao invés de textos
2 - Estabelecer e aplicar padrão de conduta a reuniões e aplicar
3 - Elaborar protótipo
4 - Promover Reuniões de Inspeção com o Cliente
5 - Manter um Especialista Store como Gerente do Cliente
6 - Prover Recursos => Ferramenta simples de fluxogramação
7 - Utilização de fluxogramação com notação UML
8 - Elaborar Dossiê de cliente

Figura 5.11. Filtragem aplicada à GMCP-ENG.1 para restringir o CMQ-SW piloto

5.5.1.1. A Proposta Formal de Ciclo (PFC), a Guia Mestre de Ciclo (GMC) e a Memória do Programa de Melhoria da Qualidade do Processo de Software (MPQM)

Assim, a PFC do 1º. Ciclo de Melhoria da Qualidade do Processo restringiu-se ao foco de melhorias às práticas que haviam sido consideradas de primeira prioridade (prioridades U_0) após sucessivos encontros de deliberações sobre as diversas possibilidades de alcance do 1º. CMQ que seria o projeto-piloto de teste do MC2Q-SW.

Sucessivos esboços de PFC resultaram na PFC.01, que, uma vez tendo sido aprovada, tornou-se a 1a. Guia Mestre de Ciclo, demonstrada na figura seguinte.

PROJETO: Programa de Melhoria Contínua da Qualidade do Processo de SW na **STORE**
SUBPROJETO: Avaliação do MC2Q-SW
ETAPA: Experiência Piloto - MC2Q-SW.
TAREFA: MQP (*) Elicitação de Requisitos data: 10/ago/2004
REF: GMC-SW.001
Objetivo Imediato: Promover melhoria da documentação dos requisitos elicitados através de fluxogramação UML e testar o MC2Q-SW.
Prazo: (sugerido 3 meses).
Objetivos Mediatos: Melhorar a qualidade de comunicação com o cliente e novos colaboradores; facilitar a elaboração de protótipos; facilitar revisões e aprovações de propostas de trabalho ou Ordens de Serviço; dinamizar a constituição dos futuros dossiês.

CRONOGRAMA: (a desenvolver)

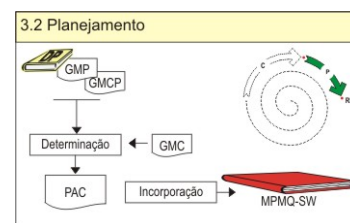
Prazo e cronograma são desnecessários só interessa na PFC e na GMC (mas optou-se por 3,5 meses pela natureza de ser laboratório e envolver estudo / aprendizado....)

(*) MQP = Melhoria da Qualidade do Processo

Figura 5.12. A GMP convertida da PFC para o CMQ-SW piloto

5.5.2. PRÁTICA - O Planejamento (Etapa 3.2)

A GMC-SW.001 produziu um Plano de Ação Circular (PAC), cujo gráfico de Gantt é demonstrado na figura 5.13, que neste caso particular de Ciclo experimental constitui-se no próprio PAC.001 que veio compor a



Memória do Programa de Melhoria da Qualidade na STORE.

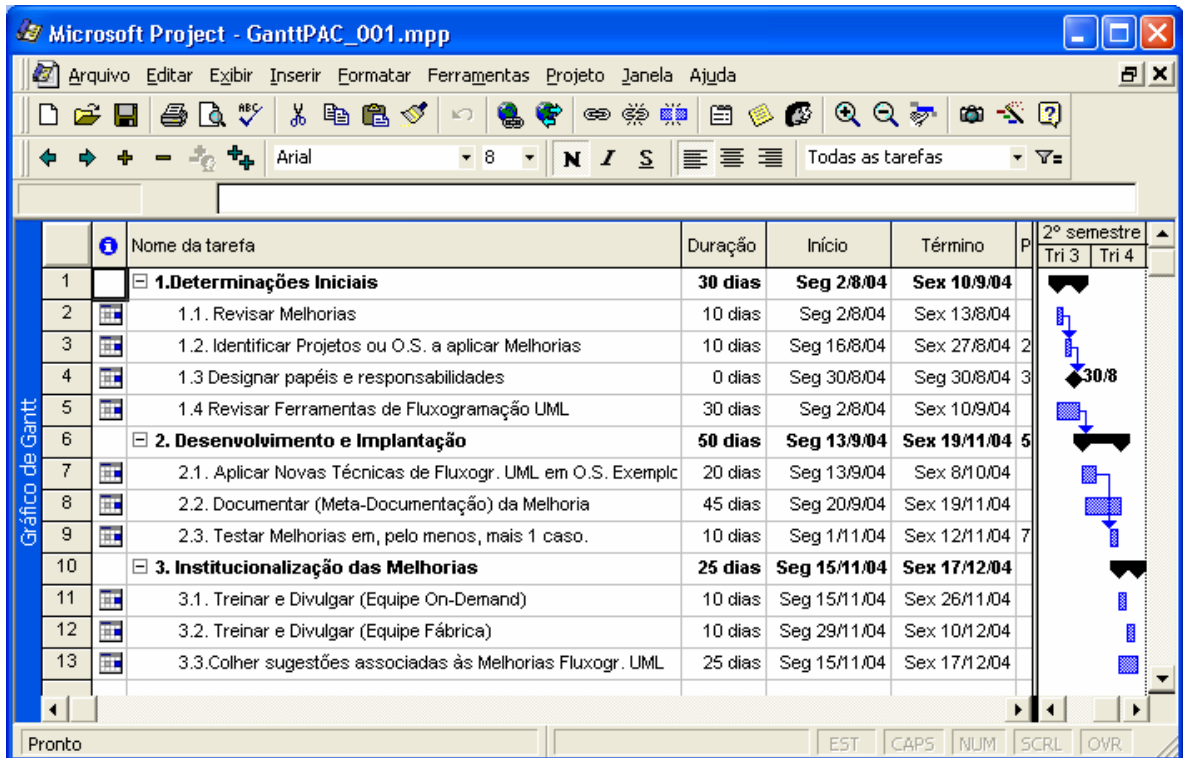
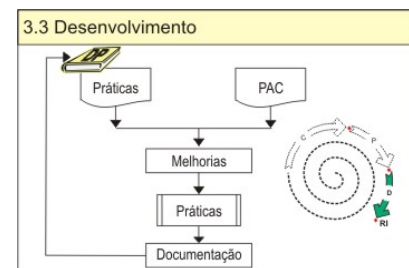


Figura 5.13. O PAC-001 Plano de Ação Circular do CMQ-SW piloto

5.5.3. PRÁTICA - O Desenvolvimento das Melhorias do Processo (Etapa 3.3)

As melhorias foram implementadas gradativamente, e o prazo de desenvolvimento acabou sendo muito mais longo do que o planejado (praticamente o triplo do tempo), pois a empresa passou por mudanças estruturais no período e o projeto perdeu momentânea prioridade.



Alguns colaboradores aprenderam a utilizar o UML e o uso da ferramenta MS-VISIO foi implementado.

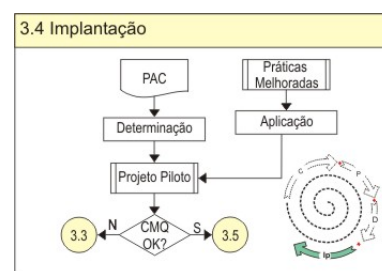
Em outubro de 2004 a pesquisa atingiu seu auge, conforme publicado no jornal interno da STORE Automação, ilustrado no ANEXO 5.07. Porém os resultados esperados de documentação do que foi feito deixou a desejar, pois:

- f) embora a GMP-Elicitação tenha sido adaptada às necessidades da STORE, circunstanciadas ao foco de melhoria nas práticas escolhidas (anexo) ela apenas foi utilizada pelo pesquisador na orientação dos trabalhos;
- g) Suas práticas, parcialmente descritas, acabaram por se desenvolver na total informalidade;

Porém o aspecto mais positivo foi a de que a exemplificação de uso do UML vingou, possibilitando que não participantes diretos na experiência passaram a utilizá-lo.

5.5.4. PRÁTICA - A Implantação das Melhorias do Processo (Etapa 3.4)

Escolheu-se um determinado projeto de um cliente multinacional e aplicou-se a fluxogramação UML em uma proposta de reestruturação de um projeto que exigia adaptação em decorrência de novas exigências do cliente ilustradas nas figuras a seguir.



PMO - Project Management Office
Diagrama de Caso de Uso

Projeto: Reestruturação
Revisão: 001

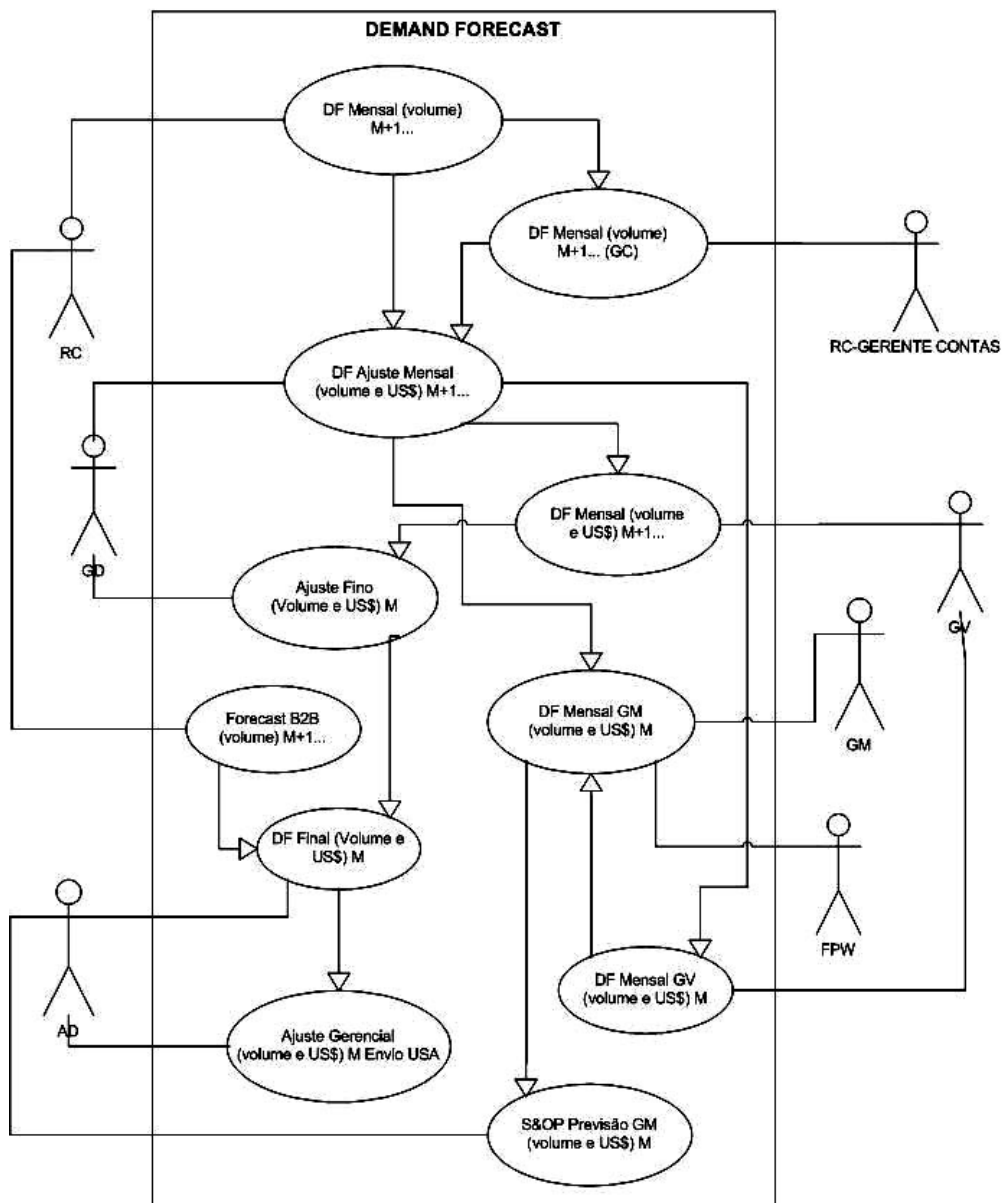


Figura 5.14. Aplicação de um Diagrama de Caso de Uso após o aprendizado do UML

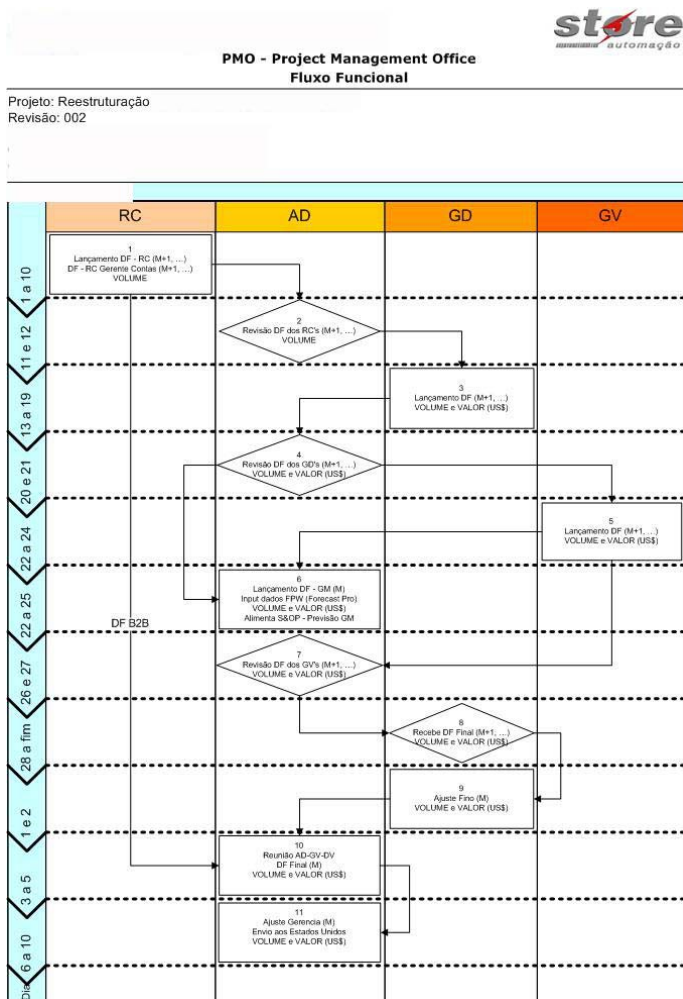
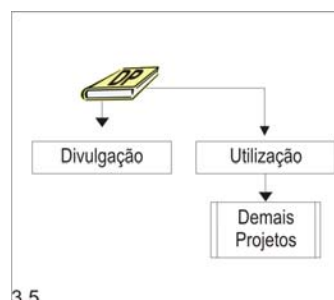


Figura 5.15. Utilização de Fluxo Funcional na Elicitação de Requisitos

5.5.5. PRÁTICA - A Institucionalização das Melhorias (Etapa 3.5).

A INSTITUCIONALIZAÇÃO demonstrou ser a etapa mais difícil para o encerramento bem sucedido do 1º. CMQ, pois, em decorrência das mudanças estruturais da empresa, havia agora, em Barra Bonita, duas grandes equipes distintas de desenvolvimento. Uma para serviços “ON-DEMAND”, e outra para criação de novas soluções de porte, denominada “FÁBRICA”, conforme ilustra o Organograma Simplificado da figura a seguir.



3.5

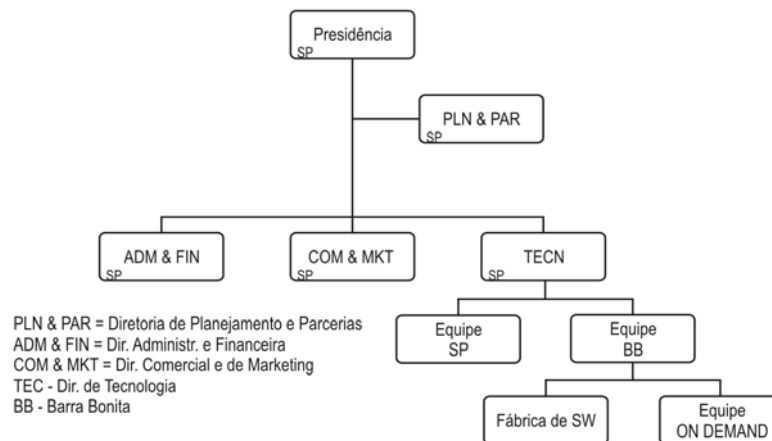


Figura 5.16. Organograma Simplificado da STORE Automação em 2005

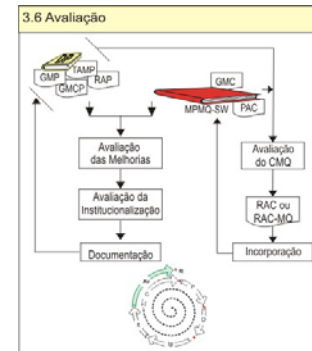
Isto implicou que a Institucionalização apenas ocorreu na equipe “ON DEMAND”, que passou a utilizar-se preponderantemente de fluxogramação, especialmente através da padronização UML, na elicitação de requisitos de todas as novas solicitações de serviços ali colocadas, mas o mesmo não aconteceu na equipe “FÁBRICA”.

Explica-se tal ocorrência em função das características peculiares de atuação de cada equipe, assim, não se justificava forçar tal implementação neste último grupo de colaboradores.

Portanto, flexibilizou-se o objetivo da GMC.001 considerando-se por encerrado o teste de implantação do CMQ-SW piloto nesse estágio de trabalho, uma vez que foi notório o proveito experimental e atendida, satisfatoriamente a intenção original de teste do MC2Q-SW, deixando que a INSTITUCIONALIZAÇÃO da melhoria implementada por este ciclo-piloto à equipe “FÁBRICA” ocorresse em ocasião futura quando a STORE acreditasse ser oportuna.

5.5.6. PRÁTICA - A Avaliação das Melhorias (Etapa 3.6)

Os melhores resultados da pesquisa-ação evidenciaram-se nesta etapa, graças à solução apresentada do indicador de Medida da Capacitação do Processo (MCP), seu correspondente visualização gráfica através do Pentágono PMCP, e do Índice de Melhoria da Capacitação do Processo (IMCP).



Considerou-se que, após a intervenção do MC2Q-SW, o processo ENG.1 evoluiu de **MCP11111 (INICIAL)** para **o MCP51111 (HOJE)**, penalizado pelo rigor das regras estipuladas pelo MC2Q-SW, que, no entanto, não houvesse o rigor das regras, poderia ser demonstrado como um **DIE₃ = 51342 (REAL)**, sendo que, em breve, através de **formalização** apropriada, o ENG.1, deve ser classificado **MCP55552 (BREVE)**, conforme as situações demonstradas à figura a seguir.

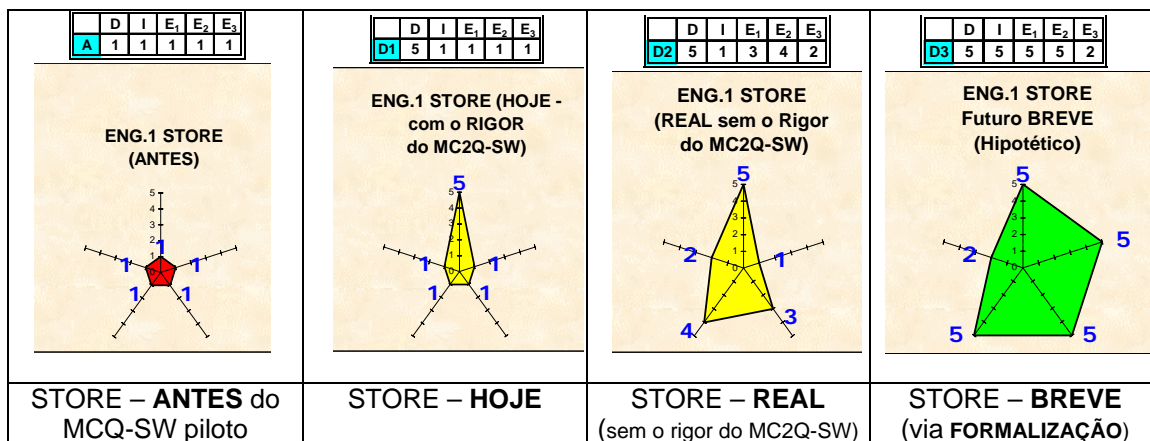


Figura 5.17. enquadramento de 4 possíveis situações da STORE face às melhorias praticadas na ELICITAÇÃO DE REQUISITOS

A avaliação das melhorias praticadas foi medida conforme se demonstra na figura 5.18, melhor visualizada pelo gráfico da figura 5.19 correspondente, que mostra as Curvas de Capacitação do Processo antes (Pa) e após (Pp) a

implementação das melhorias, e com o IMCP compreendido pelas duas curvas conforme explicado em 4.9.3.

Avaliação das MELHORIAS em PRÁTICAS após Conclusão do 1o. CMQ														
TAMP.01 - Tabela de Avaliação da Melhoria do Processo	A PRÁTICA já ocorria, no ENG.1, em out.04 em até						Hoje (jan.2006) a prática ocorre no ENG.1 em até						AVANÇO de Melhoria OCORRIDO via Práticas no ENG.1	
	ELICITAÇÃO DE	zer	até 25	até 50	até 75	até 100	ATÉ	zer	até 25	até 50	até 75	até 100		ATÉ
PRÁTICAS (ATIVIDADE / Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)	0 %	%	%	%	%		0 %	%	%	%	%			
1 - Privilegiar fluxogramação ao invés de textos		25				25					75		75	50
2 - Estabelecer e aplicar padrão de conduta a reuniões e aplicar	0					0		25					25	25
3 - Elaborar protótipo		25				25			50				50	25
4 - Promover Reuniões de Inspeção com o Cliente	0					0						100	100	100
5 - Manter um Especialista Store como Gerente do Cliente	0					0						100	100	100
6 - Prover Recursos => Ferramenta simples de fluxogramação	0					0			50				50	50
7 - Utilização de fluxogramação com notação UML	0					0				75			75	75
8 - Elaborar Dossiê de cliente	0					0		25					25	25

Figura 5.18. TAMP.01 – A Avaliação das Melhorias após o 1º. CMQ-SW na STORE

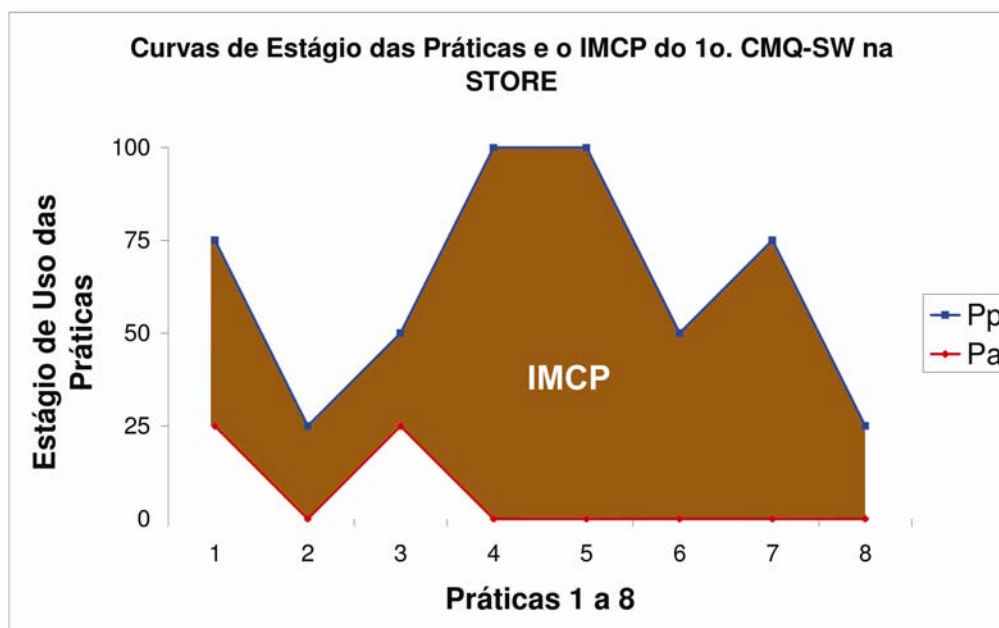


Figura 5.19. Visualização gráfica do Índice de Melhoria da Capacitação do Processo ELICITAÇÃO de REQUISITOS na STORE após a implementação do CMQ experimental

5.6. RESULTADOS

O principal resultado deste trabalho é o Método apresentado no capítulo 4 (O Método Proposto) e o resultado de sua primeira aplicação experimental circunstanciada a um estudo de caso, conforme detalhamento do capítulo 5 (A Aplicação do Método).

A aplicação e os resultados dessa avaliação prévia são entendidos como limitados ao caso. Oportuna a citação de Goode e Hatt, em Ludke e André (1986) a respeito: o caso *“pode ser similar a outros, mas é ao mesmo tempo distinto por seus contornos particulares, incidindo seu interesse, portanto, no que ele tem de particular, mesmo que, posteriormente, venham a ficar evidentes certas semelhanças com outros casos ou situações”*.

Assim, os resultados descritos detalhadamente neste capítulo 5 podem apenas servir de base comparativa aos interessados em novos testes ou aplicações do Método, não tendo a pretensão de servir a generalizações.

Para que se entenda melhor a contextualização e a representatividade dos resultados, apresentou-se uma boa descrição da empresa e do diagnóstico situacional nos itens 5.2 (Apresentação da Empresa) e 5.3 (A Primeira Fase (Diagnóstico e Capacitação)).

O método MC2Q-SW resultante é apresentado segundo sua estruturação de 3 Fases subdivididas em 13 Etapas e com a descrição dos artefatos sugeridos, resumidos a 3 essenciais, a **BP** (Base de Conhecimento dos Processos), o **DP** (Dossiê dos Processos) e a **MPMQ-SW** (Memória do Programa de Melhoria da Qualidade do Processo de *Software*). Todos os demais artefatos que documentam a utilização do método são considerados complementares ou suplementares e utilizáveis conforme a necessidade de cada caso ou situação.

O primeiro teste experimental permitiu percorrer todas as etapas do método conforme se detalhou neste capítulo 5, sendo que os resultados da **Primeira Fase (DIAGNÓSTICO E CAPACITAÇÃO)** foram o Diagnóstico Situacional e

a Proposição de Ações, aprovados sem maiores formalizações com a diretriz de alocação de um mínimo investimento possível à experimentação do MC2Q-SW; a conseqüente reorganização mínima necessária da empresa escolhida, limitada à constituição de um Grupo de Trabalho para teste do MC2Q-SW; a determinação do SPICE como o paradigma de processos e práticas a ser utilizado e adaptado, e a conseqüente constituição da Base de Processos da empresa. Também promoveu-se a capacitação dos envolvidos no que se considerou essencialmente necessário à experimentação do método.

Relativamente ao treinamento procurou-se despender o menor investimento possível sem prejuízo do indispensável a colocar em experimentação o MC2Q-SW na STORE, conforme se descreve em 5.3.2 (As Ações de Capacitação) sub-item 3, sendo oportuno observar que o que norteou essa decisão foram experiências anteriores, a exemplo da citada em Ferguson (1997), que, para a capacitação de um único técnico ou gerente no PSP (*Personal Software Process*) - o *framework* da SEI restrito ao disciplinar as atividades dos desenvolvedores - ser bem sucedida, faz-se necessário um esforço concentrado de 1,5 a 3 meses. Assim, considerando a usual impossibilidade de nossa pequena *softwarehouse* em pesquisa dispor de seu corpo técnico de 25 colaboradores por 2 meses seguidos, sem interrupção, ter-se-ia de propor um cronograma de treinamento de algumas sessões semanais que alongaria essa capacitação em, no mínimo, 6 meses.

Os resultados da **Segunda Fase (PREPARAÇÃO)** foram: a Determinação do Processo a submeter à melhoria, como sendo a **Elicitação de Requisitos** (classificado como ENG.1, pelo SPICE); a Identificação das Métricas que permitiram avaliar as melhorias implementadas, a avaliação da capacitação do processo escolhido; a identificação das melhorias possíveis e desejáveis a implementar ao processo; a montagem inicial do Dossiê de Processo (DP) através de sua GMP (Guia Mestra de Processos).

Relativamente à Determinação do Processo (ver item 4.8), aplicou-se, no caso estudado, simplesmente um método genérico, rápido e prático, adaptado à necessidade específica da empresa estudo de caso. Tal método pode ser utilizado combinadamente, caso se evidencie a necessidade de maior

rigor ou formalização na identificação dos processos, ao Método ISO / IEC 15504 – *Based Process Capability Profile for Process Improvement* (PRO2PI), técnica que incorpora características das arquiteturas contínuas e por estágios, descrito por Salviano (2004).

Relativamente à Identificação das Métricas (ver item 4.9), constatou-se que seria de pouca utilidade prática investir recursos e tempo em “medir” a maturidade de processos que se caracterizavam, neste estudo de caso, por uma pluralidade significativa de práticas dinamicamente inter-relacionadas e em constante evolução, e que exigiam múltiplas melhorias. Menor utilidade prática, ainda, teria uma simples medição estática da situação desse processo sem objetivar sua melhoria. Daí resultaram: a **MCP** (Medida de Capacitação do Processo); o **PMCP** (Pentágono da Medida de Capacitação do Processo); a **TAMP** (Tabela de Avaliação da Melhoria do Processo) e o **IMCP** (Índice de Melhoria da Capacitação do Processo).

Contribuições que podem ser notáveis são as demonstrações gráficas do **PMCP** e do **IMCP**, pois se demonstraram muitos úteis, não sendo do conhecimento do pesquisador nenhuma iniciativa antecedente com tal força ilustrativa.

Os resultados da **Terceira Fase (EXECUÇÃO)** foram: a delimitação das melhorias de interesse imediato a implementar ao processo escolhido (Elicitação de Requisitos); a determinação, o planejamento e a implementação desse primeiro Ciclo de Melhoria; a institucionalização dessas melhorias ao grupo de maior interesse da empresa (Equipe “ON-DEMAND”) e a avaliação das melhorias implementadas ao processo.

Resultados complementares, decorrentes da natureza da pesquisa-ação, principalmente face aos desafios encontrados de necessidade de simplificação e de rapidez de resultados foram a proposição do “**Modelo Evolutivo dos 3 Estágios da Organização em direção a um Programa de Melhoria da Qualidade do Processo de Software**” e da “**Síndrome da Fachada**” denominações utilizadas sem nenhuma pretensão, apenas decorrentes da tentativa de rotular situações percebidas especificamente neste estudo de caso, que permitiram explicar a influência negativa da informalidade

e da precariedade gerencial na melhoria da qualidade dos processos de *software* nesta experiência piloto.

Como primeira avaliação, considerou-se satisfatória a utilização do método pela equipe que se propôs ao trabalho, pois as melhorias praticadas anteriormente aos processos não eram planejadas, organizadas nem controladas, caracterizando a atuação *ad-hoc* e de mínimo gerenciamento.

Os resultados das melhorias foram expostos nos itens 5.5.4. (A Implantação das Melhorias do Processo) e 5.5.6 (A Avaliação das Melhorias) onde se evidencia uma nova forma de praticar a Elicitação de Requisitos através de maior visualização gráfica pelo uso de técnicas de fluxogramação UML.

No entanto, ficou prejudicada a documentação, Institucionalização e a evolução de melhorias posteriores por uma resistência focalizada à organização mínima necessária para garantir a perenidade do Programa de Melhoria da Qualidade do Processo de *Software*, que acabou por originar a explicação da situação observada como um quadro crítico denominado de “Síndrome da Fachada”.

Um resultado notório foi a quebra da inércia em empreender a melhoria formal da capacitação do processo, deixando de lado os obstáculos do temor em utilizar completamente os modelos consagrados (CMM, SPICE, MPS-Br) pelas resistências já apontadas na revisão bibliográfica, em especial ao CMM.

A clareza da importância do gerenciamento como garantia da melhoria da qualidade foi um grande ganho que repercutiu nas medidas posteriores de melhoria da qualidade dos processos, especificamente do processo Elicitação de Requisitos.

A possibilidade de avaliações e de sua visualização gráfica através das técnicas propostas pelo MC2Q-SW permitiu melhoria da qualidade do gerenciamento, como pôde ser verificado.

Assim, focando a equipe ON-DEMAND, pode-se avaliar melhor os resultados propiciados pelo MC2Q-SW de acordo com o que segue.

- a. A performance da Equipe ON-DEMAND, relativamente às melhorias do Ciclo piloto de Melhoria da Qualidade, foi avaliada mais de perto, identificando um desempenho superior ao da STORE como um todo, conforme mostra a figura 5.20 e visualizada graficamente na figura 5.21. Para melhorar tal visualização dos resultados as Práticas foram classificadas ascendentemente segundo as medições a elas atribuídas **APÓS** (coluna S) e **ANTES** (coluna M) das melhorias implementadas.

Avaliação das MELHORIAS em PRÁTICAS na Equipe ON-DEMAND												
TAMP.01 - Tabela de Avaliação da Melhoria do Processo ELICITAÇÃO DE	A PRÁTICA já ocorria, no ENG.1, em out.04 em até					Hoje (jan.2006) a prática ocorre no ENG.1 em até					AVANÇO de Melhoria OCORRIDO via Práticas no ENG.1	
	zer 0 %	até 25 %	até 50 %	até 75 %	até 100 %	ATÉ	zer 0 %	até 25 %	até 50 %	até 75 %		até 100 %
PRÁTICAS (ATIVIDADE / Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)												
1 - Privilegiar fluxogramação ao invés de textos		25				25				75		75
2 - Estabelecer e aplicar padrão de conduta a reuniões e aplicar	0					0					100	100
3 - Elaborar protótipo						0					100	100
4 - Promover Reuniões de Inspeção com o Cliente	0					0					100	100
5 - Manter um Especialista Store como Gerente do Cliente	0					0					100	100
6 - Prover Recursos => Ferramenta simples de fluxogramação	0					0					100	100
7 - Utilização de fluxogramação com notação UML	0					0					100	100
8 - Elaborar Dossiê de cliente	0	25				25					100	100

Figura 5.20. Avaliação das melhorias do processo Elicitação de Requisitos na Equipe On-Demand

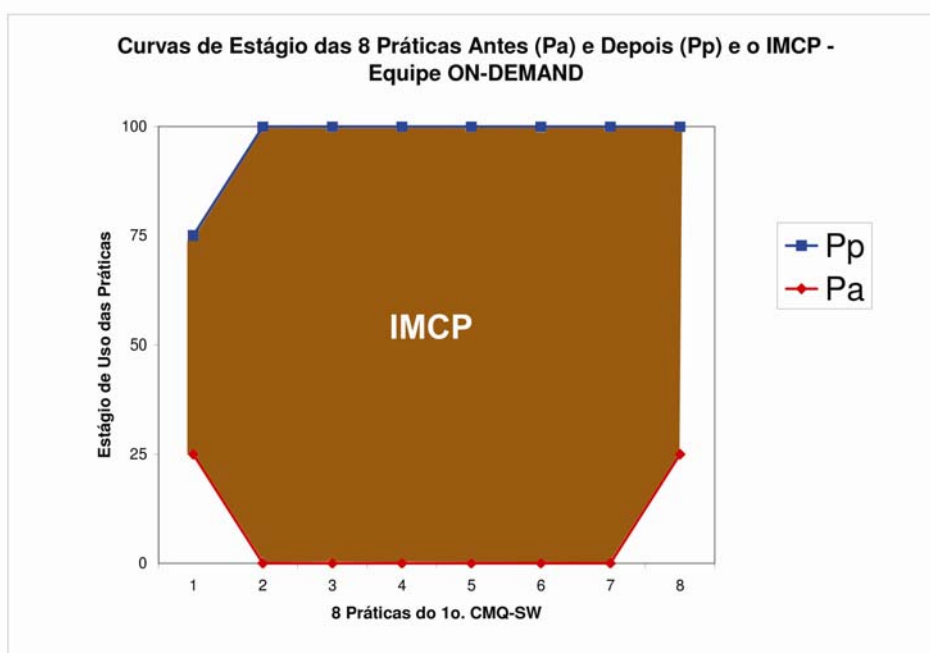


Figura 5.21. Gráfico do IMCP para a Elicitação de Requisitos (Equipe ON-DEMAND)

b. No entanto, para se ter uma boa noção de como o processo ENG1. melhorou, seriam necessárias medições mais completas. Uma primeira tentativa de melhorar essa medição, embora ainda precária, foi eleger 53 melhorias da lista original apontada por ocasião do Diagnóstico Situacional e proceder a uma avaliação focalizada equipe ON-DEMAND, cujos integrantes participaram ativamente da experiência, em um período de cerca de 14 meses após o início da experiência piloto da empresa, cujos resultados podem ser vistos nas figuras 5.22a. e 5.22b. e visualizados no gráfico da figura 5.23. A conclusão a que se chegou foi de uma melhoria significativa, de **725** pontos para **3475** pontos, conforme apontam as totalizações dos Estágios de Uso das Práticas (“Pontos Percentuais ANTES” e “Pontos Percentuais DEPOIS”) na figura 5.22b.

O aperfeiçoamento dessa medição ocorreria se houvesse sido feita uma compilação e agrupamento das práticas melhoradas para a eliminação de redundâncias, assim esta demonstração é simplesmente ilustrativa do potencial de comunicação do IMCP.

A1	B	C	D	E	F	G
2	Avalia MELHORIAS em PRÁTICAS - Equipe ON-DEMAND					
3	TAMP.01 - Tabela de Avaliação da Melhoria do Processo ELICITAÇÃO DE REQUISITOS (ENG.01)					
4	REF	0	PRÁTICAS (ATIVIDADE / Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)	A PRÁTICA já ocorria, no ENG.1, em out.04 em até x% do ideal:	A prática ocorre (em jan.2006) no ENG.1 em até y% do ideal:	AVANÇO de Melhoria OCORRIDO via Práticas no ENG.1
11	1	6	trazer o cliente à STORE na fase de elicitação e posteriores	0	0	0
36	2	31	elaborar curvas de Pareto (C3U234)	0	0	0
38	3	33	montar uma base de conhecimento de atas (C3U234)	0	0	0
56	4	51	Manter um Especialista Store como Gerente do Produto	0	0	0
59	5	54	Prover Recursos => Ferramenta CASE	0	0	0
61	6	56	Utilizar Programação Orientada a Objetos	0	0	0
62	7	57	Utilizar técnicas / metodologias XP e Agile Development Techniques	0	0	0
83	8	81	Revisões periódicas da Elicitação de Requisitos junto ao Cliente:	0	0	0
6	9	1	elaborar mapa de competências	0	25	25
12	10	7	montar uma estratégia de conquistar a simpatia do cliente com base no dossiê	0	25	25
24	11	19	montar uma base de conhecimento e necessidades dos processos de todos os clientes	0	25	25
25	12	20	usar esta base de conhecimento na proposta de novas releases do software	0	25	25
26	13	21	a cada nova reunião solicitar feedback do cliente sobre a nossa atuação, nossos problemas, nossos erros, inclusive nossas soluções	0	25	25
27	14	22	informar este feedback aos diretores envolvidos	0	25	25
28	15	23	fazer apresentações exclusivamente com base em nosso conhecimento não prometendo coisas a mais, iludindo	0	25	25
29	16	24	antes de qualquer reunião expositiva de solução ou proposta ao cliente simular a apresentação dentro da Store com os diversos envolvidos e não envolvidos	0	25	25
42	17	37	elaborar e manter base de conhecimento domínio do negócio do cliente	0	25	25
14	18	9	promover sempre segunda reunião para rever o entendimento da reunião anterior	0	50	50
31	19	26	atas de reunião	0	50	50
10	20	5	levantamentos informais junto ao cliente através de interações informais sociais	25	50	25
30	21	25	a toda solicitação/ordem de serviço deixar claro o contexto do assunto	0	75	75
40	22	35	tentar manter calma, não se afobar, não se precipitar ao encerrar a elicitação apesar de custo de levantamento, pressões diversas, etc...	0	75	75
18	23	13	idem ao que está por traz da necessidade do cliente (real causa)	25	75	50
19	24	14	elaborar protótipo	25	75	50

Figura 5.22a. Lista de Avaliação de Melhorias (Parte I) ocorridas no prazo de 14 meses em 53 Práticas da Elicitação de Requisitos na Equipe ON-DEMAND

A1	B	C	D	E	F	G	
2	Avalia MELHORIAS em PRÁTICAS - Equipe ON-DEMAND (CONTINUAÇÃO)						
3	TAMP.01 - Tabela de Avaliação da Melhoria do Processo ELICITAÇÃO DE REQUISITOS (ENG.01)						
4	REF	0	PRÁTICAS (ATIVIDADE / Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)	A PRÁTICA já ocorria, no ENG.1, em out.04 em até x% do ideal:	A prática ocorre (em jan.2006) no ENG.1 em até y% do ideal:	AVANÇO de Melhoria OCORRIDO via Práticas no ENG.1	
21	25	16	elaborar uma ata da reunião de elicitação não deixando dúvidas sobre o negócio/problema do cliente (não se inibir de demonstrar o desconhecimento ao cliente com cautela...)	25	75	50	
39	26	34	obrigatoriamente antes de propor solução conferir in loco os processos do cliente e fluxograma-los ou relata-los e documentá-los	25	75	50	
50	27	45	implantar uma interface de processamento rápido no MIS para suportar a função CRM de tracking das demandas do cliente	25	75	50	
20	28	15	idem levar sempre alternativas, proposições que conquistem o cliente ajudando a resolver seus problemas	50	75	25	
7	29	2	elaborar relatório de aderência	0	100	100	
13	30	8	elaborar e manter permanentemente atualizado um dossiê do cliente	0	100	100	
15	31	10	ao inicio de cada reunião, sempre rever a reunião anterior	0	100	100	
16	32	11	elaborar um planejamento de cada reunião	0	100	100	
37	33	32	montar uma base de conhecimento das demandas do cliente (C3U234)	0	100	100	
41	34	36	não adianta apenas documentar, mas explicar item a item o que foi levantado apresentando ao cliente	0	100	100	
55	35	50	Manter um Especialista Store como Gerente do Cliente	0	100	100	
57	36	52	Mudar comportamento => Desenhar mais Fluxogramas	0	100	100	
58	37	53	Prover Recursos => Ferramenta simples de fluxogramação	0	100	100	
60	38	55	Utilização de fluxogramação com notação UML;	0	100	100	
76	39	71	elaborar um histórico do cliente (DOSSIE)	0	100	100	
77	40	72	elaborar um dossiê de cliente (DOSSIE)	0	100	100	
78	41	73	resgatar o dossiê do cliente (DOSSIE)	0	100	100	
79	42	74	estudar o dossiê do cliente (DOSSIE)	0	100	100	
82	43	77	MANTER o dossiê do cliente (DOSSIE)	0	100	100	
9	44	4	definir fluxos de processos	25	100	75	
17	45	12	identificar no planejamento da reunião as características da empresa, contexto e stakeholders/participante	25	100	75	
70	46	65	prepara visita (ENTREV)	25	100	75	
72	47	67	registra ata de reunião (ENTREV)	25	100	75	
80	48	75	pesquisar as soluções de concorrentes (DOSSIE)	25	100	75	
23	49	18	provocar um levantamento de soluções ao problema do cliente não somente através do cliente e sim também através de outras fontes	50	100	50	
22	50	17	não deixar escapar termos essenciais ao negócio do cliente	75	100	25	
47	51	42	implantar um CRM (MIS)	75	100	25	
71	52	66	efetua visita (ENTREV)	100	100	0	
81	53	76	compilar as best practices dos sistemas concorrentes (DOSSIE)	100	100	0	
				Ptos Percents ANTES>>>	725	3475	<<<Ptos Percents DEPOIS
						479%	

Figura 5.22b. Lista de Avaliação de Melhorias (Parte II) ocorridas no prazo de 14 meses em 53 Práticas da Elicitação de Requisitos na Equipe ON-DEMAND

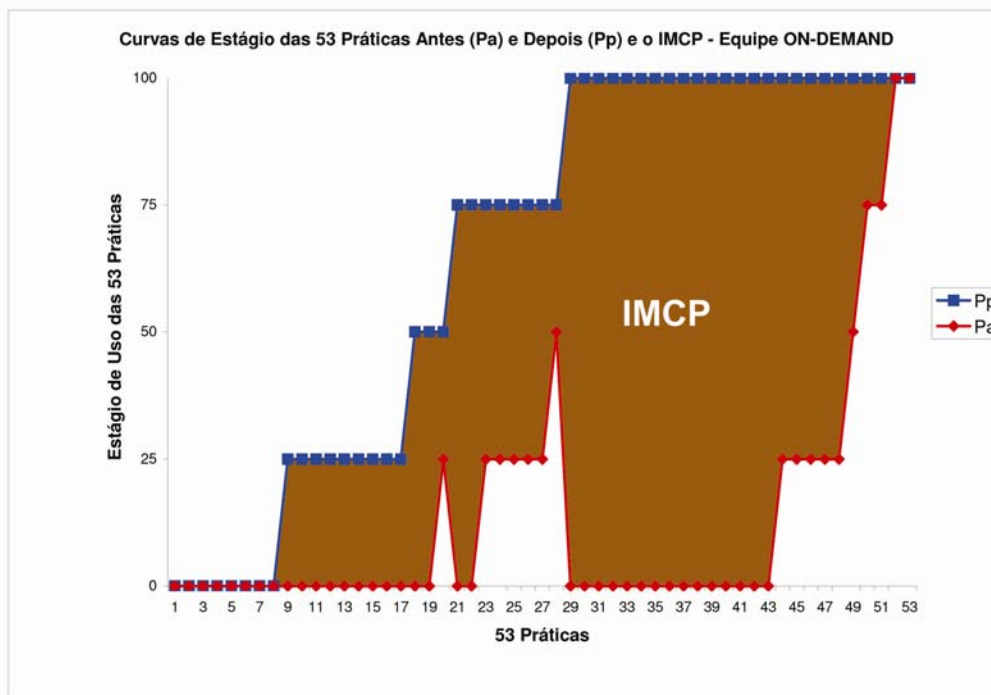


Figura 5.23. Gráfico do Índice de Melhoria da Capacitação do Processo Elicitação de Requisitos na Equipe ON-DEMAND após 14 meses de contato com o MC2Q-SW

- c. Desta maneira é possível avaliar o PMCP (Pentágono da Medida da Capacitação do Processo) Elicitação de Requisitos específico da equipe ON-DEMAND, como tendo passado da situação **MCP11111 (INICIAL)** para **MCP55443 (HOJE)** perfeitamente visualizado pela figura 5.24. Resolveu-se não atribuir **MCP55553** ao Processo Elicitação de Requisitos praticado hoje pela Equipe em questão pois a EXECUÇÃO tem sido prejudicada pela pressa em algumas situações de novos serviços de pequena amplitude (não críticas) o que ainda traz alguns problemas de comunicação entre clientes e desenvolvedores.

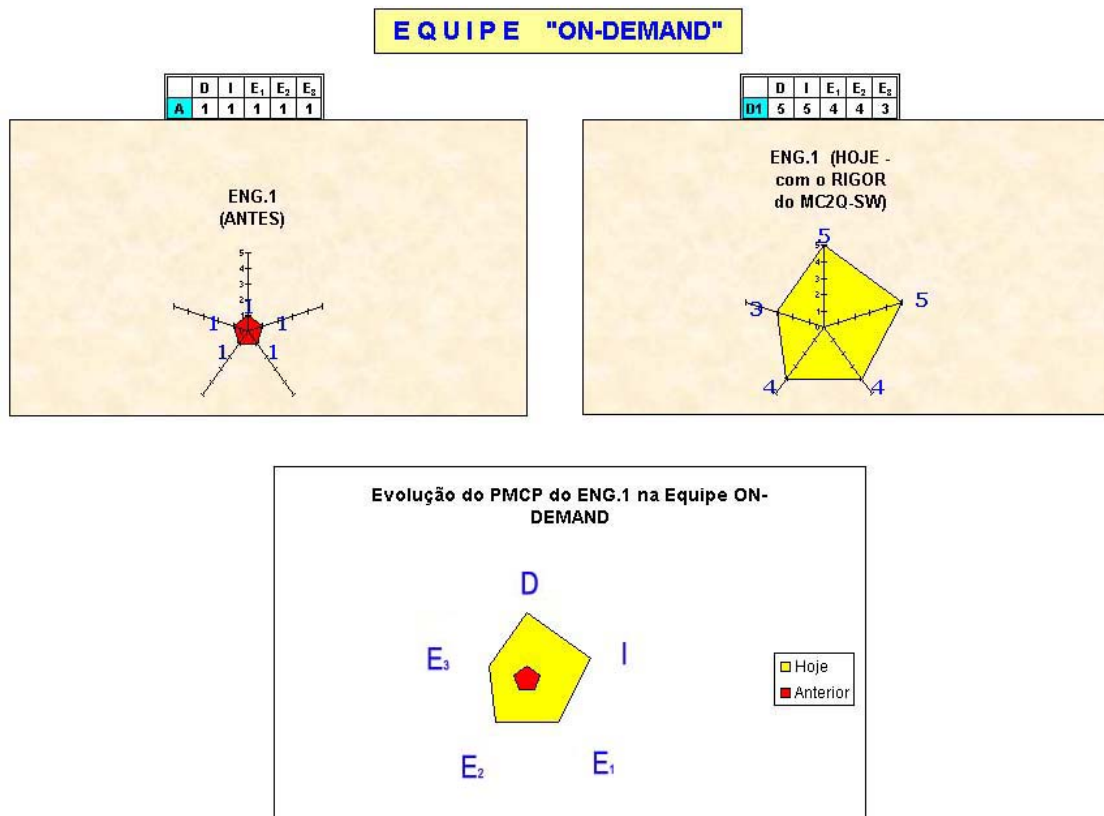


Figura 5.24. A avaliação da Melhoria da Capacitação do Processo ELICITAÇÃO DE REQUISITOS focada na equipe "ON-DEMAND" da STORE, após 14 meses de contato com o MC2Q-SW

- d. Melhorias na avaliação ainda podem ser feitas, como por exemplo o agrupamento das melhorias em categorias, como também ponderação e priorização conforme interesse, porém deixamos essa evolução para trabalhos posteriores.

A seletividade na implantação de melhorias também foi outro ganho viabilizado pela aplicação do método, pois, de forma mais transparente, as melhorias passaram a ser planejadas em deliberação, com priorização daquelas que se mostraram mais interessantes à Organização implementar.

Sintética e genericamente podemos avaliar que a melhor contribuição que o Método trouxe à STORE foi sua absorção pela Equipe "ON-DEMAND" que permitiu a evolução restrita dessa pequena célula de trabalho do Estágio INICIAL ao Estágio ORIENTADO (segundo o Modelo Evolutivo dos 3 Estágios

proposto em 4.5), ganho que, no entanto, não ocorreu na FÁBRICA, por razões de prioridade.

Assim, conservadoramente, opta-se por avaliar que a empresa estudo de caso permanece no **Estágio I**, segundo o Modelo Evolutivo dos 3 Estágios, uma vez que apenas a Equipe ON-DEMAND absorveu o conhecimento, sem que o restante da empresa houvesse se re-estruturado e se capacitado suplementarmente no sentido de implementar um PMQ-SW.

Portanto, não tendo a STORE constituído nem o GO (Grupo Organizador) nem o CQ (Comitê da Qualidade) mas apenas um Grupo de Trabalho para atuar experimentalmente com o MC2Q-SW, e, nem tendo, posteriormente à experimentação, se preocupado com essa reorganização, pode-se dizer que ela não implantou um Programa de Melhoria da Qualidade CONTÍNUO.

6. CONCLUSÃO

*“Adoro prazos.
Adoro o barulho sibilante que eles fazem quando passam...”*

*Douglas Adams
autor de “The Hitchhiker’s Guide to the Galaxy”
citado em “O Capelão do Diabo”
de Richard Dawkins.*

Este trabalho procurou contribuir com o que se acreditou ser não exatamente uma lacuna na área de Melhoria da Qualidade de *Software*, mas, sim, uma falta de ênfase e atenção a um detalhe que se considerou de suma importância: **o Método**.

Sinteticamente a idéia foi originada do princípio de que **MODELOS** destinam-se a mostrar **O QUE FAZER** e **MÉTODOS** destinam-se a mostrar **COMO FAZER**, sendo que a ênfase tem sido dada apenas aos primeiros, por, talvez, confundir-se que a determinação de “o que fazer” tem de preceder “o como fazer”.

Essa forma de agir, certamente inconsciente, tem, possivelmente, levado a maioria dos especialistas a não dar a importância ao **MÉTODO** que, no presente caso, permitiu trabalhar sobre as hipóteses que tentaram explicar o maior dilema que deixava perplexo o pesquisador, conforme discutido nas Justificativas da Pesquisa, decorrente da pesquisa bienal MCT-SEPIN (2002) (item 1.2): Por quê há tão baixa aderência (pelas *softwarehouses*) aos *frameworks* consagrados para a melhoria da qualidade de *software* (SPI) quando praticamente todas alegam reconhecer a relevância do tema?

Assim, a pesquisa bibliográfica foi direcionada a buscar na Engenharia de Sistemas e de *Software*, na área de Gestão de Negócios e na de Metodologia Científica elementos à melhor compreensão do tema.

A princípio acreditava-se que o grande obstáculo à essa “baixa aderência”, era um conjunto de razões enunciadas como:

- ✓ a complexidade, os altos custos e os longos prazos inerentes à aplicação dos referidos *frameworks*;
- ✓ a profusão de conceitos, modelos e técnicas no ramo do conhecimento;

e assim, na tentativa de oferecer uma solução, surgiu a idéia e o desenvolvimento do MC2Q-SW, como um método que permitisse às Organizações praticarem a melhoria da qualidade do processo de *software* de forma razoavelmente independente dos *frameworks* existentes, aderindo gradativamente a esses modelos conforme seu interesse, necessidade e oportunidade.

Em seqüência natural, impunha-se a necessidade do teste experimental, uma experiência prática de implantação do método em uma *softwarehouse* brasileira. Um ponto de partida na validação através de um estudo de caso.

A pesquisa-ação foi a metodologia indicada para levar a cabo o que se propôs e os resultados mostram que **o foco de atenção aos obstáculos à melhoria da qualidade de software pode ser**, (pois isto ficou evidenciado particularmente apenas neste estudo de caso), sinteticamente, **gerencial e cultural**, e não uma questão de complexidade, domínio de conhecimento técnico específico ou de altos custos.

Esta hipótese, surgida no decorrer do trabalho e que reforça a observação de Jacobson et al (1999) apontada na Introdução deste trabalho de que “A chave para a solução desse problema está em uma gerência de processos bem definida e efetiva”, mostrou-se válida neste caso. Sim, pois mesmo simplificando conceitos e técnicas, fornecendo alternativas práticas e, adicionalmente, fornecendo força de trabalho de boa qualidade e em volume considerável, o corpo gerencial da empresa estudo de caso assumiu atitude

passiva e distante ao tema. Valeu-se meramente de esforços individuais, na forma frequentemente apontada na literatura específica do tema, como que esperando que um presente, ou solução milagrosa, pudesse gerar melhoria de qualidade de forma autônoma e continuada, sem esforço gerencial.

Tal situação induziu a proposição do **Modelo Evolutivo dos 3 Estágios** e da **Síndrome da Fachada** para melhor explicá-la, e assim, o estudo de caso serviu não só de teste experimental inicial do MC2Q-SW, mas contribuiu a uma compreensão talvez ainda mais importante que esse objetivo inicial, que pode ser a razão especulada da baixa aderência.

As 4 hipóteses preliminares citadas em 3.1.3. foram comprovadas no desenvolvimento desta pesquisa-ação, na forma explicada a seguir, pois resumidamente, nesta empresa estudo de caso ficou claro que:

5. Há interesse na melhoria da qualidade na produção de *software* e forte interesse em modelos e métodos direcionados à essa melhoria e voltados à sua realidade de pequena/média empresa produtora de *software*, pois a empresa sempre esteve receptiva e aberta à experimentação;
6. Embora houvesse esse interesse, seu comprometimento real foi muito baixo, pois não investiu em um mínimo de reorganização e ações que permitissem a implantação de um Programa Contínuo de Melhoria da Qualidade do Processo de *Software*. Percebeu-se que seu interesse se limitou a se aproveitar do esforço e motivação de alguns colaboradores, que, isoladamente e sem prejuízo do atingir metas, já comprometidas com demais projetos, pudesse trazer à empresa uma melhoria geral na qualidade;
7. A disposição em investir no conhecimento e na prática de métodos de melhoria da qualidade do processo de *software* foi praticamente nula, perdendo sempre, em prioridade, aos projetos de *software* em desenvolvimento e aos serviços de manutenção e assistência técnica à sua base instalada em clientes. Pôde-se inferir que sua capacidade de dedicação esteve sempre aquém a todos os desafios que enfrenta, e, nessa situação, sua prioridade foi, em todo o período de

mais de um ano de convívio na experiência, direcionada ao que “gera caixa”, que em última análise, é o produzir e vender *software* e garantir sua operacionalização nos clientes. Pôde-se também considerar que, não só seu fôlego financeiro foi restritivo, mas, também, sua capacidade gerencial foi praticamente nula a superar o desafio de melhoria contínua da qualidade do processo de *software*.

Assim, não se percebeu, em momento algum, uma ociosidade que permitisse realocar pessoas a estudos e ações de aperfeiçoamento da melhoria da qualidade do processo de *software*, nem um gerenciamento sistemático com esse fim, situações que se supõe ocorrem com maior freqüência em empresas de maior porte;

8. Apesar da disponibilidade dos *frameworks* consagrados de melhoria da qualidade de *software*, e devido aos altos custos para a absorção de qualquer deles, muitas adaptações e inovações tornaram-se necessárias para praticar a melhoria da qualidade do processo de *software* de forma apropriada à situação desta pequena *softwarehouse*. Isso se conseguiu graças à construção de um método, o MC2Q-SW, e seus artefatos sugeridos, que conjugou e adaptou métodos de análise de problemas, de diagnósticos situacionais e de tomada de decisão e propôs medidas de avaliação com visualização gráfica esclarecedora.

A pesquisa-ação foi vivenciada em sua plenitude, pautada, permanentemente, pelo confronto das teorias consultadas e modelos em vigor (adicionados do “modelo teórico” que resultou no MC2Q-SW) com a realidade vivenciada pela empresa em sua luta diária em produzir, comercializar e atender solicitações de customização a diversos clientes em um mercado competitivo.

A verdadeira “arena” onde se equilibra a qualidade com os demais atributos exigidos de uma empresa de *software* é o mercado potencial e os clientes conquistados, e é o cliente que indica a qualidade do *software* e de serviços adicionados, por sua fidelidade.

No entanto, isto não significa que a empresa, hoje tranqüila em relação à sua carteira de clientes, não possa (a) estar em situação difícil amanhã; nem (b) melhorar ainda muito mais sua posição competitiva no sofisticado mercado de *software*. E isto tem a ver com a busca inteligente e contínua da qualidade, que reside oculta aos olhos da imensa maioria, nos processos de produção, em nosso caso nos processos de produção de *software*.

Lado a lado com os demais componentes que permitem a empresa de *software* sobreviver, vencer e liderar, a qualidade do processo de *software* neste estudo de caso particular, parece ser, no entanto, relegada a um plano secundário, fazendo-se negligenciada pelos gerentes que conduzem a produção de *software*, confiantes de seu poder de reação quando as coisas se tornarem complicadas.

A esses gerentes, e, em especial ao que se subordina a função de Planejamento Estratégico, o pesquisador alerta que as forças que equilibram as ações do mercado podem ser devastadoras e que oportunidades ao baterem à porta podem encontrá-la fechada se não houver preocupação com a qualidade do processo de *software*. Este é o caso de Clientes potenciais exigentes, como Multinacionais, Agências ou Órgãos ligados ao Governo.

Adicionalmente, observa que tudo leva a crer que, face às múltiplas exigências de desempenho gerencial, os executivos, especialmente neste estudo de caso, relegam a qualidade a um esforço puramente pessoal de seus subordinados. Não lhes dá condições, (ao mínimo, dilação de prazos em outros projetos, ou realocação de tarefas e responsabilidades) nem ferramentas que facilitem esse desafio adicional, nem premiação pelo esforço extra, donde se conclui que a empresa se eterniza no ESTÁGIO I, correndo os riscos inerentes a essa situação.

As empresas que assim se comportam podem, ainda, correr o risco de *turn-over* ou *burn-out* prejudicial de colaboradores essenciais, pelo estresse a que lhes sujeita, e pela desmotivação que lhes causa, devido ao que se poderia identificar como exploração, uma vez que seus gerentes, não assumindo o custo exigido pela qualidade, deixam-no exclusivamente absorvido pelos funcionários. Essa perda de funcionários pode ainda tornar

mais crítica a situação da empresa, ou impedi-la de se expandir, se tais colaboradores se transferem à concorrência, ou montam um novo negócio, levando consigo não só sua experiência, mas, também, a rede de relacionamentos profissionais e comerciais. Neste caso, o custo da falta de cuidado com a qualidade fica muito mais evidente e possivelmente muito mais significativo, do que a pretensa economia decorrente da atitude descrita de relegá-lo exclusivamente aos subordinados e à informalidade.

O presente trabalho permitiu dar respostas a muitas das questões e conflitos relacionados à qualidade do processo de *software* no âmbito de uma empresa brasileira de *software*, sem a pretensão de se contrapor aos *frameworks* maduros, muito ao contrário, apoiando-se bastante neles. Não pretende também incorrer no erro primário de generalizar a partir de tão pequena amostra, mas sim de mostrar como é possível praticar a melhoria da qualidade de *software* com recursos limitados, força de vontade e criatividade tirando proveito dos paradigmas consagrados, sem, no entanto escravizar-se a eles.

A conjugação de boas práticas gerenciais, desde o consagrado **POC₄**²⁵ ao processo racional de tomada de decisões (KEPNER, 1997), associados aos princípios de racionalização, estabelecidos desde o início, e à formulação da “Síndrome da Fachada” e do “Modelo Evolutivo dos 3 Estágios das Organizações”, permitiram uma importantíssima conclusão: de que qualidade do processo é uma questão gerencial, seja auto-gerenciamento pessoal, seja liderança organizacional, pela associação que têm com valores culturais, estratégias de mudança, tomada de decisões, análises de custo-benefício e de oportunidade-risco, de disciplina, de perenidade, de formalismo versus criatividade.

Assim, parece ser falta de sensibilidade empresarial e gerencial deixar a qualidade do processo ser praticada na informalidade sem estar presente na

²⁵ **GERENCIAR**, na concepção da Teoria Geral da Administração, onde Gerenciar engloba as funções Planejar, Organizar, Comandar, Coordenar e Controlar, adicionado de Comunicar, no que o pesquisador alcunha de **POC₄**.

pauta de reuniões estratégicas das organizações e no dia-a-dia dos gerentes de desenvolvimento de *software* açodados pelos prazos de projetos a cumprir.

Conclusão adicional refere-se ao perigoso alheamento gerencial em não se institucionalizar a melhoria da qualidade de *software*, através de medidas de formalização a partir de adequação da estrutura orgânico-funcional e de medidas de incentivo, premiação, valorização de carreira e análise de desempenho individual dos especialistas em *software*.

Importantes contribuições, sem dúvida, foram as medidas de melhoria da capacitação do processo, ao menos sua visualização gráfica, que permitem uma análise detalhada, rápida e objetiva, de como perceber a influência das ações sobre as práticas em processo.

Por fim, é oportuno realçar a importância do **Método** em uma cultura onde parece que se dá muito mais ênfase a ferramentas e modelos, popularmente divulgada pelo “nada se cria, tudo se copia”, que foi, em boa parte, a experiência de caso vivenciada na empresa pesquisada. Sem querer ser repetitivo, é o método que permite a conjugação de todos os elementos necessários ao atingir objetivos, assim, antecede planos, mapas, instrumentos. Fica claro que trabalhar sem método, significa, literalmente, errar, em ambas as acepções do termo: (1) incorrer em erro; (2) vaguear, espalhar-se em várias direções.

7. POSSIBILIDADES DE FUTUROS TRABALHOS

Uma vez que foi possível testar experimentalmente o MC2Q-SW e que essa experiência foi a primeira possível, com as restrições inerentes demonstradas, é recomendável que o método seja validado em outras situações, o que permitirá, certamente, melhorias. As possibilidades de trabalhos futuros percebidas ao momento são as seguintes:

- Divulgação do MC2Q-SW através de artigos em publicações especializadas, inclusive de Gestão, e submissão a congressos;
- Criação de um Guia Prático e Conciso de Aplicação do MC2Q-SW;
- Experimentação do MC2Q-SW em outras *softwarehouses*;
- Experimentação intensiva do MC2Q-SW em pelo menos uma situação de Melhoria da Qualidade de longa duração;
- Adaptação do MC2Q-SW para sua aplicação na melhoria de processos em outras indústrias, que não a de *Software*, inclusive outras áreas, como de serviços governamentais e políticas públicas.

Aos que desejarem compreender e empreender a Melhoria da Qualidade em suas áreas de interesse, e em particular a do Processo de *Software*, o pesquisador estará sempre à disposição para ajudar e aprender junto, com entusiasmo ainda maior do que quando iniciou este trabalho, pois o tema é envolvente e fecundo. O que importa não é simplesmente fazer, mas, sim, fazer bem feito, na medida em que aquilo que tiver de ser feito assim o exija. A qualificação “bem feito” circunstancia o fazer a instâncias éticas, humanas, econômicas, dentre outras, que mostra que se todos fizéssemos bem feito aquilo que nos é atribuído, ou aquilo que se espera de nós, nossa sociedade seria muito melhor.

Júlio C. Navas

navas123@uol.com.br

8. REFERÊNCIAS.

- ARRUDA; Maria N.P.; VILLAS BOAS et al - **Avaliação CMM – A Experiência do CPqD Telebrás** – Maringá, PR, WQS'98 - 1998.
- BACH, J - **Enough About Process: what We Need Are Heroes** - IEEE Software Vol. 12, No 2, February 1995.
- BACHELARD, G. - **A Epistemologia** - Lisboa, Edições 70,1984.
- BARDIN, Laurence - **Análise de Conteúdo** - Presses Universitaires de France - trad. Lisboa, Portugal, Edições 70, 2000.
- BASKERVILLE, Richard; HEJE, P. Jan - **Knowledge Capability and Maturity in Software Management.** - ACM Categories: K., K.6, K.6.3, ACM ed. IFIP, Vol. 30, No. 2, p. 26-43, 1999.
- BOEHM, B. W. - TRW Defense Systems Group - **A Spiral Model of Software Development and Enhancement** - IEEE Computer, v.21, n. 5, may, 1988.
- BOLLINGER T.; MCGOWAN, C.; - **A Critical Look at Software Capability Evaluations.** - IEEE Software Vol. 8, No. 4; July 1991
- CERVO, Amado Luiz; BERVIAN, Pedro Alcino. **Metodologia Científica.** 4ª ed., São Paulo:Makron Books, 1996.
- CHIAVENATO, I. **Introdução à Teoria Geral da Administração.** São Paulo:Makron Books, 1993.
- CMU/SEI-93-TR-17 - **Software Capability Evaluation Version 1.5 Method Description CMU/SEI-93-TR-17** - Estados Unidos: Carnegie Mellon University – Software Engineering Institute. 1993. (www.sei.cmu.edu/pub/documents/93.reports/pdf/tr17.93.pdf).
- CMU/SEI - 93-TR-24, Versao 1.1 - **CMM - Nível 1** - Modelo de Maturidade da Capabilidade de Software (CMM) - Tradução não oficial do CPqD - Telecom & IT Solutions - por GONÇALVES, M. José; VILLAS BOAS, André, 2001
- CMU/SEI - 93-TR-25, Versao 1.1 - **CMM - Nível 2** - Tradução não oficial do CPqD - Telecom & IT Solutions -por GONÇALVES, M. José; VILLAS BOAS, André, 2001
- COAD, Peter; YOURDON, Edward. **Projeto baseado em objetos.** Tradução de Vandenberg Dantas de Souza. Rio de Janeiro:Campus, 1993. p.195.
- COCKBURN, A. - **Selecting a project's methodology.** IEEE Software, jul./ago. 2000.
- COHEN, David. - **Você sabe tomar decisão?** - Revista Exame, 746 ed., ago. 2001.
- CÔRTEZ, L. M. - CHIOSSI, C. T - **Modelos de Qualidade de Software,** Ed. da Unicamp,Campinas - SP, 2001
- CURTIS, B. - **The global pursuit of process maturity.** - IEEE Software, jul./ago. 2000.

- CURTIS, W.; - **A Mature View of the CMM.** - American Programmer Vol. 7, NO. 9, September 1994.
- CURTIS, W.; **Wich Comes First, The Organization or its Processes?;** IEEE Software, nov / dec. 1998.
- DAVIS, Stanley M. - **Future Perfect.** - Massachusetts: Addison-Wesley Publishing Company, Inc, Reading, 1987.
- DEMING, W. E. **Qualidade, a revolução e a administração.** Rio de Janeiro, RJ, Editora Saraiva, 1990.
- DEMO, P. - **Introdução à Metodologia da Ciência.** São Paulo, Atlas, 1985.
- DONNA, D. K.; MASTERS, S. - **CMM – Based Appraisal for Internal Process Improvement (CBA IPI): Method Description CMU/SEI-96-TR07** - Estados Unidos: Carnegie Mellon University – Software Engineering Institute. 1996.
(www.sei.cmu.edu/pub/documents/96.reports/pdf/tr007.96.pdf).
- DRUCKER, Peter F. **As Novas Realidades.**São Paulo: Pioneira,1989.
- DRUCKER, Peter F. **O Gerente Eficaz.**Rio de Janeiro: Zahar Editores, 1974.
- EMAN, K.E.; DROUIN, J.N. e MELO, W. H. - **Spice. The theory and practice of software process improvement and capacity determination.** - Los Alamos, Califórnia, IEEE Computer Society, 1998.
- FACHIN, O . Fundamentos de Metodologia. São Paulo - Saraiva, 2001.
- FAYAD, E. Mohamed - **Software Development Process: A Necessary Evil** - Communications of the ACM, ACM Vol.40, no. 09, p. 101-103, Sept., 1997.
- FERGUSON, P.; et al. - **Introducing The Personal Software Process: Three Industry Case Studies.** - IEEE Computer, vol. 30, n.30, May 1997.
- FERGUSON, Pat; HUMPHREY, S. Watts; KHAJENOORI, Soheil; MACKE, SUSAN; MATVYA, Annette - **Results of Applying the Personal Software Process** - IEEE, 0018-9162, IEEE p. 24-31, 1997.
- FIORINI, S. T., VON Staa, A. e BAPTISTA, R.M. - **Engenharia de Software com CMM.** - Brasport, 1998.
- GANE, Chris. **Desenvolvimento rápido de sistemas.** Rio de Janeiro:LTC, 1998.
- GIL, A. C. **Como Elaborar Projetos de Pesquisa.**São Paulo: Atlas, 1999.
- HARRISON, Warren - **Whose Information Is It Anyway?** - IEEE Computer Society and the ACM, 0740-7459/03, Published by the IEEE Computer Society, p. 5-7, 2003.
- HUHNE, Leda Miranda. **Metodologia Científica – Caderno de Textos e Técnicas.** 6ª ed., Rio de Janeiro: Agir, 1995.
- HUMPHREY, W. S. - **Characterizing the Software process: A Maturity Framework** - software Engineering Institute, CMM/SEI - 87 - TR - 11, June 1987. Uma versão revisada deste relatório foi publicada na IEEE Software Vol. 5, No. 2, march 1988.
- IEEE90 - **IEEE Standard Glossary of Software Engineering**, Terminology; ANSI/

- IEEE Std 620. 12, 1990.
- ISO/IEC 15504 TR Joint Technical Committee ISO/IEC TR 15504 JTC1, Information Technology, Subcommittee SC 7, Software Engineering. - **Information technology – Software process assessment – Parts 1-9**, Suíça, 1998.
- ISO/IEC 15504-5 TR- **Software Process Assessment** - (ISO/IEC TR 15504-5:1998(E)) 1st Ed, 1998.
- ISO/IEC 15504-5.5 CD - **Information Technology - Process Assessment - Part 5: An exemplar Process Assessment Model** - (Committee Draft) ISO / IEC JTC1/SC 7 WG10 N - CD 15504-5.5, feb,16, 2004
- ISO/IEC 15504-5 - The International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC 15504 - Information Technology - **Process Assessment – Part 5: An exemplar Process Assessment Model**, 2006.
- JACOBSON I.; BOOCH G. and RUMBAUGH James - **The Unified Process** - IEEE, 0-201-57169-2, IEEE May/June 1999.
- JACOBSON, I.; BOOCH, G. E RUMBAUGH, J. **Applying CMM project planning practices to diverse environments**. IEEE. Software, jul./ago.2000.
- JOHNSON, D. L ; BRODMAN, Judith G. - **Applying the CMM to Small Organizations and Small Projects** - Proceedings of the 1998 Software Engineering Process Group Conference, Chicago, IL, march.1998.
- JURAN, J., M. A - **Qualidade desde o Projeto; Novos Passos para o Planejamento da Qualidade em Produtos e Serviços**. - Trad de Nivaldo Montigelli Jr. Brasil: Pioneira Thomson Learning. 1992.
- JURAN, J., M.; GRZYNA, F., M. **Controle da Qualidade – Conceitos, Políticas e Filosofia da Qualidade** – Volume I. Trad de Maria Cláudia de Oliveira Santos. Brasil: Mc Graw-Hill Ltda. 1991.
- KEPNER, Charles H.; TREGOE, Benjamin B. - **The New Rational Manager** - Princeton, New jersey - Princeton Research Press, 1997.
- KOHAN, Sarah - **QuickLocus: Proposta de um método de avaliação de processo de desenvolvimento de software em pequenas organizações** - Dissertação de Mestrado Profissional - Eng; da Computação - IPT -SP. 2003.
- KRUCHTEN, P. - **From Waterfall to Iterative Development - A Challenging Transition for Project Managers**, Rational software 2000, www.therationaledge.com, ago. 2003,
- KUHN, Thomas S. - **A Estrutura das Revoluções Científicas** – São Paulo, Perspectiva, 1975.
- LAUDON, K. C.; LAUDON, J. P. - **Gerenciamento de Sistemas de Informação**. - 3a. ed. Rio de Janeiro: LTC, 1999
- LOTT, C. M. - **Breathing new Life into the Waterfall model**, - IEEE Software, sep. 1997.

- LUDKE, M.; ANDRÉ, M.E.D. - **A. Pesquisa em Educação: abordagens qualitativas** - São Paulo, .Editora EPU, 1986.
- MARCONI, M.de A.; Lakatos, E.M. - **Metodologia Científica** - Editora Atlas, 3.ed.São Paulo, Atlas, 1982
- MASTERS, S.; BOTHWELL, C. **CMM Appraisal Framework, Version 1.0** - CMU/SEI-95-TR-001 - Estados Unidos, Carnegie Mellon University – Software Engineering Institute. 1995.
(www.sei.cmu.edu/pub/documents/95.reports/pdf/tr001.95.pdf - 2002).
- MCT-SEPIN, 2002 - **Qualidade e Produtividade no Setor de Software Brasileiro** – (base 2001), Ministério da Ciência e Tecnologia, Secretaria de Política de Informática, Brasília - DF, 2002.
- NAVAS, J. C, ALVES, R. F. - Anais do VI Simpósio Internacional de Melhoria do Processo de Software (SIMPROS) - **O MC2Q-SW – Um Método para a Melhoria Contínua da Qualidade de Software apoiado em CMM e SPICE - e sua Aplicação Experimental como base a Programas de Melhoria da Qualidade de Software** - São Paulo – SP, ISBN 85-7359-408-X, nov.2004.
- NBr 12207 - Norma NBr ISO /IEC 12207 -ABNT - Associação Brasileira de Normas Técnicas - Tecnologia da informação - Processos de ciclo de vida de software. (corresponde à ISO /IEC 12207, 1995), Rio de Janeiro, 1999
- Norma ISO /IEC 15504. Software process assessment. Parts 1-9, Tech. Report type 2, 1998.
- Norma NBr ISO 10005,1997 - Diretrizes para planos de qualidade, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1997
- Norma NBr ISO 10011-1,1993 - Diretrizes para auditoria de sistemas da qualidade - Parte 1 : Auditoria, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1993
- Norma NBr ISO 10011-2,1993 - Diretrizes para auditoria de sistemas da qualidade - Parte 2: Critérios para qualificação de auditores de sistemas da qualidade, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1993
- Norma NBr ISO 10011-3,1993 - Diretrizes para auditoria de sistemas da qualidade - Parte 3: Gestão de programas de auditoria, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1993
- Norma NBr ISO 10013 - Diretrizes para o desenvolvimento de manuais da qualidade, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1998
- Norma NBr ISO 9000 1, 1994 - Normas de gestão da qualidade e garantia da qualidade - Parte: 1- Diretrizes para seleção de uso, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1994
- Norma NBr ISO 9000 2, 1994 - Normas de gestão da qualidade e garantia da qualidade - Parte 2 - Diretrizes gerais para a aplicação das NBr ISO 9001, NBr ISO 9002 e NBr ISO 9003, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1994

- Norma NBr ISO 9000 3, 1994 - Normas da gestão de qualidade e garantia da qualidade - Parte 3: Diretrizes para a aplicação da NBr 9001 ao desenvolvimento, fornecimento e manutenção de software, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1994
- Norma NBr ISO 9000 4, 1994 - Normas da gestão de qualidade e garantia da qualidade - Parte 4: Guia para a gestão do programa de dependabilidade, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1994
- Norma NBr ISO 9004 -1,1994 - Gestão da qualidade e elementos do sistema da qualidade - Parte 1: Diretrizes, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1994
- Norma NBr ISO 9004 -2,1993 - Gestão da qualidade e elementos do sistema da qualidade - Parte 2: Diretrizes para serviços, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1993
- Norma NBr ISO 9004 -4,1993 - Gestão de qualidade e elementos do sistema da qualidade - Parte 4: Diretrizes para melhoria da qualidade, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1993
- Norma NBr ISO 9004-3,1994 - Gestão da qualidade e elementos do sistema da qualidade - Parte 3: Diretrizes para materiais processados, ABNT - Associação Brasileira de Normas Técnicas, Rio de Janeiro, 1994
- PÁDUA, Elisabete Matallo Marchesini de. **Metodologia da Pesquisa: abordagem teórico-prática**. 6ª. ed., Campinas:Papirus, 2000.
- PAULK, Mark C. - **How ISO 9001 compares with the CMM**. Software Engineering Institute, p. 74-83, jan. 1995.
- PAULK, Mark C. - **Using the Software CMM in Small Organizations**. Software CMM for the small project/organization. Software Engineering Institute - Carnegie Mellon University, p.1-13, 1998.
- PAULK, Mark C.; et al. - **The Capability Maturity Model: Guidelines for Improving the Software Process**. Reading, Mass, Addison-Wesley. 1995. Também disponível em ACM SIGSOFT Software Engineering Institute, Pittsburgh, p. 1-26, 1995.
- Peters James F., Pedrycz, Witold - **Engenharia de Software - Teoria e Prática** - Rio de Janeiro, Edit. Campus, 2001
- PMBOK - **Project Management Body of Knowledge** - BRASIL - MINAS GERAIS CHAPTER. Tradução livre do PMBOK 2000, V 1.0. Disponível em: <<http://www.pmimg.org.br>>.Acesso em jul.2003
- PRESSMAN, Roger S. - **Engenharia de Software**. - São Paulo: Makron Books , 1995.
- RACCOON, L. B. S. **Fifty years of progress in software engineering**. ACM SIGSOFT Software Engineering Notes, v. 22, n.1, jan. 1997.
- ROCHA, A.R., MALDONADO, J.C., Weber, K.C. - **Qualidade de Software: teoria e prática**. - São Paulo, Prentice Hall, 2001.
- RUDIO, Franz Vitor. **Introdução ao Projeto de Pesquisa Científica**. 23ª. ed.,

- Petrópolis: Editora Vozes, 1998.
- SALVIANO, Clênio F.- **Experiência de Avaliação de Processos e Planejamento da Melhoria utilizando a Futura Norma ISO/IEC 15504 (SPICE)** - Florianópolis, SC, anais do WQS'99 - Workshop da Qualidade de Software SBES'99, 1999.
- SALVIANO, Clênio F.- **Um Método para Escolha dos Processos para uma Melhoria Alinhada aos Objetivos de Negócio** - anais do Workshop de Qualidade de Software, WQS 2001 - Rio de Janeiro, págs 39-50, 2001.
- SALVIANO, Clênio F; et al.- **Towards an ISO/IEC 15504-Based Process Capability Profile Methodology for Process Improvement (PRO2PI)** - Lisbon, Proceedings of SPICE 2004: The 4th International SPICE Conference on Process Assessment and Improvement - abr.2004.
- SALVIANO, Clênio F.- **Como Iniciar uma Melhoria Viável e Necessária em uma Micro ou Pequena Empresa de Software** - Slides de apresentação de palestra no PRO-QUALITI 2005 – UFLA , Lavras, MG, 12.nov, 2005.
- SEI Appraisal Program - <http://www.sei.cmu.edu/managing/app.directory.html#cmm.based> - último acesso em ago.2004.
- SOFTTEX - **Plano Estratégico Softex 2001-2004.** - Campinas, SP, Softex, 2001
- SOMMERVILLE, Ian. - **Engenharia de software.** - Tradução de André Mauricio de Andrade. Revisão técnica Kechi Hiramã. 6. ed.São Paulo: Addison Wesley, 2003.
- SPICE - <http://www.sqi.gu.edu.au/spice>, ultimo acesso em mar.2006
- STAIR, Ralph M. - **Principios de sistemas de informação - uma abordagem gerencial.** - Tradução de Maria Lúcia Iecker Vieira e Dalton Conde de Alencar . Revisão técnica Paulo Machado Cavalheiro.2. ed. Rio de Janeiro: LTC, 1998.
- THIOLLENT, M. - **Metodologia da pesquisa-ação** - São Paulo, Cortez, 1997.
- TINGEY, M. O. - **Comparing ISO 9000 and the SEI CMM: a reference and selection guide.** - Englewood Cliffs, Prentice Hall, 1997.
- TRIPP, D. - **Pesquisa-ação: Uma Introdução Metodológica. Educação e Pesquisa.**
[online].<http://www.scielo.br/scielo.php?script=sci_arttext&pid=S151797022005000300009&lng=pt&nrm=iso>. set./dez. 2005, vol.31, no.3, p.443-466, ISSN 1517-9702, 2005
- VILLAS BOAS - **Uma Avaliação CMM de baixo custo: a abordagem do CPqD – CPqD Telecom & IT Solutions** - 2001
- WEBER, K. C. e ROCHA, A. R. C. - **Qualidade e Produtividade em Software.** 4a. ed. renovada, São Paulo:Makron Books, 2001.
- WEBER, K.C. - **Fundamentos para uma Política de Software no Brasil.** - Sociedade Softex, Abr 2000.

WIEGERS, Karl E. - **Creating a Software Engineering Culture** - New York, Dorset House Publishing, 1996.

XP - **eXtreme Programming** - <http://www.extremeprogramming.org/> , ultimo acesso em fev 2006

YOURDON, E. - **Where's the Basis for Year 2000 optimism?** - Computerworld, vol. 32, no. 7,p.68, 1998.

YOURDON, E. **Declínio e queda dos analistas e dos programadores: a salvação e novos caminhos para a produtividade e a qualidade no desenvolvimento de software.** São Paulo:Makron Books, 1995.

ZHONG, Xiaoming; MADHAVJI, H. Nazim; EMAM El Khaled - **Critical Factors Affecting Personal Software Processes** – IEEE Software, 0740-7459/00, IEEE p. 76-83, 2000.

ANEXOS

Nota Explicativa: Os anexos são numerados de acordo com o capítulo de origem (assim 5.01 refere-se ao 1º. Anexo referenciado no capítulo 5).

ANEXO 5.01

RAS.01 - Relatório de Avaliação de Situação - Apontamentos de Reunião de Eclosão de Idéias para Apuração de Problemas ou Necessidades da Equipe de Barra Bonita da STORE.

0	RAS.01 - Relatório de Avaliação de Situação - Apontamentos de Reunião de Eclosão de Idéias para Apuração de Problemas ou Necessidades da Equipe de Barra Bonita da STORE
0	
0	Problemas / Sugestões / ATIVIDADE / Processo / Artefato / etc...
1	elaborar mapa de competências
2	elaborar relatório de aderência
3	inserir atividades no MIS
4	definir fluxos de processos
5	levantamentos informais junto ao cliente através de interações informais sociais
6	trazer o cliente à STORE na fase de elicitação e posteriores
7	montar uma estratégia de conquistar a simpatia do cliente com base no dossiê
8	elaborar e manter permanentemente atualizado um dossiê do cliente
9	promover sempre segunda reunião para rever o entendimento da reunião anterior
10	ao início de cada reunião, sempre rever a reunião anterior
11	elaborar um planejamento de cada reunião
12	identificar no planejamento da reunião as características da empresa, contexto e stakeholders/participante
13	idem ao que está por traz da necessidade do cliente (real causa)
14	elaborar protótipo
15	idem levar sempre alternativas, proposições que conquistem o cliente ajudando a resolver seus problemas
16	elaborar uma ata da reunião de elicitação não deixando dúvidas sobre o negócio/problema do cliente (não se inibir de demonstrar o desconhecimento ao cliente com cautela...)
17	não deixar escapar termos essenciais ao negócio do cliente
18	provocar um levantamento de soluções ao problema do cliente não somente através do cliente e sim também através de outras fontes
19	montar uma base de conhecimento e necessidades dos processos de todos os clientes
20	usar esta base de conhecimento na proposta de novas releases do software
21	a cada nova reunião solicitar feedback do cliente sobre a nossa atuação, nossos problemas, nossos erros, inclusive nossas soluções
22	informar este feedback aos diretamente envolvidos
23	fazer apresentações exclusivamente com base em nosso conhecimento não prometendo coisas a mais, iludindo
24	antes de qualquer reunião expositiva de solução ou proposta ao cliente simular a apresentação dentro da Store com os diversos envolvidos e não envolvidos
25	a toda solicitação/ordem de serviço deixar claro o contexto do assunto
26	atas de reunião
27	considerando que há clientes que não se veem beneficiados pelo sistema ou que não conseguem extrair todas informações que gostariam
28	que nem todos os usuários acabam sendo consultados a respeito de uso, satisfação e extração de informação
29	treinar os colaboradores Store em técnicas de coleta e triagem de informações e abordagem de soluções a problemas
30	considerando que não se deve iludir aos clientes e nem morrer estressado
31	elaborar curvas de Pareto
32	montar uma base de conhecimento das demandas do cliente
33	montar uma base de conhecimento de atas

34	obrigatoriamente antes de propor solução conferir in loco os processos do cliente e fluxograma-los ou relata-los e documentá-los
35	tentar manter calma, não se afobar, não se precipitar ao encerrar a elicitación apesar de custo de levantamento, pressões diversas, etc...
36	não adianta apenas documentar, mas explicar item a item o que foi levantado apresentando ao cliente
37	elaborar e manter base de conhecimento domínio do negócio do cliente
38	preencher planilha de apontamentos de atividades (MIS)
39	planilha de levantamento de cliente (não está funcionando por que o cliente não responde)
40	suporte on-line
41	com base nesta planilha recuperar informações do sistema e cliente: stakeholder, dados de banco para suporte (on-line)
42	implantar um CRM (MIS)
43	indicar as melhorias de cada release em uma planilha disponibilizando-a no MIS (já disponível)
44	melhorar codificação das atividades
45	implantar uma interface de processamento rápido no MIS para suportar a função CRM de tracking das demandas do cliente
46	complementar no MIS o percentual de horas a completar determinada tarefa
47	efetuar uma adaptação no MIS que permita customizar etapas, fases, atividades por projeto E ainda categorizá-los
48	fazer um brain storm para o MIS
49	porque o MIS e não uma ferramenta CRM de mercado?
50	Manter um Especialista Store como Gerente do Cliente
51	Manter um Especialista Store como Gerente do Produto
52	Mudar comportamento => Desenhar mais Fluxogramas
53	Prover Recursos => Ferramenta simples de fluxogramação
54	Prover Recursos => Ferramenta CASE
55	Utilização de fluxogramação com notação UML
56	Utilizar Programação Orientada a Objetos
57	Utilizar técnicas / metodologias XP e Agile Development Techniques
58	encenação
59	reunião de preparo à apresentação ao cliente
60	reunião de planejamento da encenação
61	desenvolvimento da encenação
62	entrevistas
63	elabora questionário
64	aplica questionário
65	prepara visita
66	efetua visita
67	registra ata de reunião
68	filmar
69	fotografar
70	gravar
71	elaborar um histórico do cliente
72	elaborar um dossiê de cliente
73	resgatar o dossiê do cliente
74	estudar o dossiê do cliente
75	pesquisar as soluções de concorrentes
76	compilar as best practices dos sistemas concorrentes
77	MANTER o dossiê do cliente
78	Revisões periódicas da Elicitación de Requisitos junto ao Cliente

ANEXO 5.02

Diagnóstico Situacional e Proposta de Ação para a STORE

(Relatório conforme originalmente proposto e apresentado à Alta Administração)

Abr.2004

As atividades realizadas para apuração deste **Diagnóstico Situacional (DS)** basearam-se em entrevistas formais de coleta de informações ou atividades informais de desenvolvimento e integração com os dirigentes e demais funcionários da STORE.

Essas atividades exploraram a percepção pessoal e grupal a respeito de aspectos do cotidiano da empresa relacionados a princípios, conceitos, hábitos, padrões de comunicação e de relacionamento, processos organizacionais formais e informais que influenciam diretamente o processo de criação e manutenção de software e, portanto, fornecem indicadores do Estágio de Desenvolvimento Organizacional da Store diretamente relacionados ao processo de SW.

As principais conclusões do DS foram:

- a) Um dos maiores problemas apontados com respeito ao processo de trabalho é o da comunicação. Em relação à "**qualidade de comunicação**" somente cinco, dentre os dezoito participantes, a consideraram boa, os demais foram de opinião contrária. Tais dificuldades de comunicação abrangem um grande espectro, mas focando nos aspectos de processo de software, podemos ressaltar a ausência de um padrão de tratamento das demandas e dos processos de trabalho internos que induz os funcionários a situações tipificadas como "apagar incêndios" ou "sem tempo de respirar" e a inúmeros retrabalhos decorrentes de alterações ou complementações do originalmente especificado. Tal aspecto denuncia alta informalidade nos procedimentos e comunicação ineficiente entre os diversos grupos institucionais ou informais (o "de colaboradores (não dirigentes)", o "Gerencial", o "de São Paulo"; o "de Barra Bonita"; o "da Fábrica de SW"; o "da equipe "On Demand""; o "Técnico" e o "Administrativo");
- b) Outro grande problema é o sentimento de excesso de serviço com pressão constante para gerar resultados versus recursos limitados de tecnologia, esta última observação baseada em afirmações tais como: Não há ferramenta CASE; eles próprios têm de desenvolver o MIS (Sistema de Informações Gerenciais); não há padrões ou manuais uniformes; as ferramentas de produtividade praticamente inexistem, dentre outras afirmações;
- c) Problema igualmente grande é a percepção de pouca oportunidade de reciclagem do conhecimento e da necessidade de um constante aprendizado, principalmente pelas exigências de gerar resultados (software com qualidade e rapidez);
- d) Outro problema é a percepção da ausência de um plano de carreira que permita ao funcionário passar pelos diferentes processos de trabalho na empresa de forma organizada, favorecendo não só os interesses da empresa mas também seu crescimento profissional e sua evolução na companhia;
- e) Observou-se um número significativo de manifestações que vinculam a qualidade a uma padronização do trabalho por meio de normas estabelecidas pela empresa que devem ser seguidas por todos sem grandes graus de liberdade;
- f) Identificou-se o receio de que iniciar um Programa de Melhoria da Qualidade do Processo de Software na empresa provoque o estabelecimento de padrões e normas

de produção que impliquem em “amarras” para a criatividade²⁶, autonomia e flexibilidade no trabalho;

- g) O conceito de qualidade aparece principalmente relacionado a frases tais como satisfação dos clientes, evitar o retrabalho e estabelecimento de normas e padrões de execução do trabalho;
- h) A questão da motivação dos funcionários é outro aspecto muito enfatizado que se vinculou à questão da qualidade no sentido de que o funcionário desmotivado não apresenta condições para um trabalho de qualidade.

Para ilustrar como tais aspectos presentes no cotidiano da Store ficaram evidenciados, reproduzimos abaixo uma dramatização (dentre várias, por nós conduzidas com os funcionários separados aleatoriamente em grupos diversos) nas **Atividades de Desenvolvimento, Diagnóstico e Integração – ADDIs**.

“O cliente liga reclamando de erro em documento. Ninguém atende o telefone com medo de ser problema. Quem atende passa o recado erradamente para o programador. O programador executa o serviço, mas não o solicitado pelo cliente. O cliente liga bravo e exige que alguém vá às instalações do cliente para resolver o problema. Como é uma viagem longa e estressante o colaborador se mostra desmotivado”.

A conclusão registrada no item (b) (volume de serviço e pressão por resultados vs recursos limitados) resulta da observação de diversas situações reais e do relatado em entrevistas, reuniões e conversas informais com colaboradores diversos onde se percebeu que:

- Alguns colaboradores tiveram a iniciativa de desenvolver por si próprios o MIS (Management Information System) – um sistema que visa prover controles gerenciais dos projetos e do desempenho dos colaboradores - com um dispêndio avaliado até a presente data da ordem de US\$100mil, sendo que ainda não oferece praticamente nada além de um controle de horas por Ordens de Serviço/Projeto.

A relação custo-benefício parece muito ruim e fica ainda pior quando se percebe que falta muito a desenvolver para que ele atenda completamente o que se propõe.

Uma possível explicação para a insistência nesse desenvolvimento próprio é a crença (da equipe de Barra Bonita) de que se desgastariam muito solicitando à Alta Direção (e sem obter aprovação) a aquisição de uma ferramenta gerencial pronta (que custaria – no máximo – 25% do já investido e com aplicação imediata).

- Não há um conjunto de ferramentas de produtividade além dos simples ambientes de desenvolvimento.
- Algumas ferramentas são desejadas, como: Together ou ECO (da Borland, para geração de diagramas UML, persistência com BD e código-fonte), StarTeam (gerenciador de controle de versões com recursos adicionais de controle de alterações e work-flow) e Optimize-It (controla performance em código-fonte Java).

Foram identificadas cerca de 70 melhorias em práticas diversas, prioritárias, ou não, oportunas ou não, relacionadas ao processo de software, direta ou indiretamente, que sugerimos sejam revistas e avaliadas com a maior brevidade.

Complementarmente, mais focalizando as entrevistas conduzidas junto aos gerentes, pudemos observar que há:

- conflitos entre visões pessimistas e otimistas sobre o estágio atual e futuro da STORE;
- falta de entendimento do processo de remuneração do risco do empresário;
- falta de assumir responsabilidades (fábrica reclama mas não toma as decisões que lhes sejam pertinentes).

²⁶ O sentido de **criatividade**, se apresentou relacionado à inovação e busca de alternativas diferentes e inéditas para resolver os problemas do cotidiano.

Assim as atividades de análise situacional, em várias situações, apontam fortes indícios da necessidade de processos com padrões formalizados e melhor gerenciados. Não, forçosamente, que exigem maior atuação de gerentes (embora esta impressão não seja completamente descartada, mormente quando a grande maioria dos funcionários não tem nenhum preparo gerencial, por se concentrarem em atuação técnica de desenvolvimento de sistemas) mas sim de um tratamento mais gerenciado, nem que seja auto-gerenciado, onde cada qual deve agir de forma mais planejada, organizada, coordenada, controlada, documentada, medida e, muito importante, com canais de comunicação mais eficientes.

Tais indícios interferem, certamente, na motivação.

Pelo tudo o exposto concluímos que a STORE se enquadra no **Estágio I (Inicial ou Intuitivo)** no que diz respeito a **Programas de Melhoria da Qualidade de Software (PMQs-SW)**, segundo a terminologia adotada pelo **MC2Q-SW (Método Circular de Melhoria Contínua da Qualidade do Processo de Software)** e que o estabelecimento de um PMQ-SW na empresa exige cuidados de capacitação e decisões como seguem as sugestões enunciadas abaixo que serão detalhadas em reunião oportuna:

1. Capacitação Organizacional:

- ✓ Estabelecer Centros de resultado;
- ✓ Determinar Autonomia vs responsabilidades;
- ✓ implantar um **GO (Grupo Organizador)** nos moldes do indicado no MC2Q-SW;
- ✓ Implantar um **CQ (Comitê da Qualidade)** nos moldes do indicado no MC2Q-SW.

2. Capacitação Motivacional:

- ✓ Melhoria na Comunicação;
- ✓ Melhoria da Infra-estrutura de Tecnologia (HW e SW de Base / ferramentas de produtividade (CASE / etc);
- ✓ Melhoria das condições ambientais (instalações e mobília);
- ✓ Programa de Carreira;
- ✓ Programa de treinamento continuado;
- ✓ Programa de Trainees;
- ✓ Estabelecimento periódico de Metas por Centros de Resultado; Departamentais e Pessoais;
- ✓ Avaliação de Desempenho / Premiação por Desempenho – especialmente financeira.

3. Capacitação Técnica para a Melhoria da Qualidade do Processo, através de treinamentos, a princípio de **“capacitação fundamental”**, em:

- ✓ MC2Q-SW (*);
- ✓ Engenharia de Sistemas (especialmente Ciclo de Vida de Projetos e SPI – *Software Process Improvement*);
- ✓ CMM / CMMI;
- ✓ PSP (*Personal Software Process*) (*);
- ✓ ISO 12.207;
- ✓ SPICE;
- ✓ PMBOK;
- ✓ Xp e Agile development Tools (*);
- ✓ Inglês (*).

(*) Estes treinamentos devem ser de maior abrangência e duração, e, **permanentemente**, atualizados.

4. **Capacitação Técnica para a Melhoria da Qualidade do Processo** através da Escolha de uma **Base de Conhecimento de Processos de Software (BP)**;
5. **Decisão de iniciar os trabalhos de melhoria da qualidade de processos de SW** através dos **Ciclos de Melhoria da Qualidade do Processo de SW (CMQ-SW)**, tão logo se considere que a empresa tenha atingido o mínimo de capacitação que assegure o sucesso do PMQ-SW;
6. **Avaliação da Experiência Piloto** tão logo tenha sido considerada conclusa.

Na impossibilidade de atender intensivamente todas as sugestões, e acreditamos ser isto impeditivo, momentaneamente, sugerimos que se implante apenas uma experiência piloto de aprendizado e teste do MC2Q-SW, como um primeiro passo de implantação de um PMQ-SW, que contemple o seguinte:

1. **Capacitação Organizacional:**
 - ✓ implantar um **GO (Grupo Organizador)** nos moldes do indicado no MC2Q-SW;
 - ✓ Implantar um **CQ (Comitê da Qualidade)** nos moldes do indicado no MC2Q-SW.
2. **Capacitação Motivacional:**
 - ✓ Melhoria na Comunicação;
 - ✓ Melhoria da Infra-estrutura de Tecnologia (HW e SW de Base / ferramentas de produtividade (CASE / etc);
 - ✓ Melhoria das condições ambientais (instalações e mobília); ;
3. **Capacitação Técnica para a Melhoria da Qualidade do Processo**, através de treinamentos em:
 - ✓ MC2Q-SW;
 - ✓ PSP (*Personal Software Process*); ISO 12.207; SPICE; PMBOK (via Grupos de Estudo e premiação de curso externo aos melhores funcionários);
 - ✓ Xp e Agile development Tools (via Grupos de Estudo e premiação de curso externo aos melhores funcionários);
 - ✓ Inglês (via Grupos de Estudo e premiação de curso externo aos melhores funcionários).
4. **Capacitação Técnica para a Melhoria da Qualidade do Processo** através da Escolha de uma **Base de Conhecimento de Processos de Software (BP)**;
5. **Decisão de desenvolver uma experiência piloto de melhoria da qualidade de processos de SW** através dos CMQs-SW, tão logo se considere que a empresa tenha atingido o mínimo de capacitação que dê sustentação à experiência piloto;
6. **Avaliação da Experiência Piloto** tão logo tenha sido considerada conclusa.

Sendo o que nos cumpre por ora relatar e solicitar acatar nossa sugestão,

Subscrevemo-nos,

Atenciosamente;

Júlio C. Navas

Joyce M. A. P. Silva

À submeter a revisão por:

Roger (Gte da Fábrica); Eduardo (Diretor de Tecnologia); Aurivam (Analista Sr.); Cido (analista Sr.); e Wagner (CEO).

ANEXO 5.03

RAS.01b - Prévia à Tabela de Decisão de Determinação de Processo - Cruzamento de Problemas ou Necessidades com os PROCESSOS da BP (restringiu-se ao ENG.1 - Elicitação de Requisitos)

A1	B	C	D	E
		TAP.01 - Prévia à Tabela de Avaliação do Processo - Cruzamento de Problemas ou Necessidades com os PROCESSOS da BP (restringiu-se ao ENG.1 - Elicitação de Requisitos)		
2	0			
3	0	Problemas / Sugestões / ATIVIDADE / Processo / Artefato / etc...	F ou D	ENG.1
4	1	elaborar mapa de competências		1
5	2	elaborar relatório de aderência		1
6	3	inserir atividades no MIS		
7	4	definir fluxos de processos		1
8	5	levantamentos informais junto ao cliente através de interações informais sociais	D	1
9	6	trazer o cliente à STORE na fase de elicitação e posteriores	D	1
10	7	montar uma estratégia de conquistar a simpatia do cliente com base no dossiê	D	1
11	8	elaborar e manter permanentemente atualizado um dossiê do cliente	D	1
12	9	promover sempre segunda reunião para rever o entendimento da reunião anterior		1
13	10	ao início de cada reunião, sempre rever a reunião anterior	D	1
14	11	elaborar um planejamento de cada reunião	D	1
15	12	identificar no planejamento da reunião as características da empresa, contexto e stakeholders/participante	D	1
16	13	idem ao que está por traz da necessidade do cliente (real causa)	D	1
17	14	elaborar protótipo	D	1
18	15	idem levar sempre alternativas, proposições que conquistem o cliente ajudando a resolver seus problemas	D	1
19	16	elaborar uma ata da reunião de elicitação não deixando dúvidas sobre o negócio/problema do cliente (não se inibir de demonstrar o desconhecimento ao cliente com cautela...)	D	1
20	17	não deixar escapar termos essenciais ao negócio do cliente	D	1
21	18	provocar um levantamento de soluções ao problema do cliente não somente através do cliente e sim também através de outras fontes	D	1
22	19	montar uma base de conhecimento e necessidades dos processos de todos os clientes	D	1
23	20	usar esta base de conhecimento na proposta de novas releases do software	D	1
24	21	a cada nova reunião solicitar feedback do cliente sobre a nossa atuação, nossos problemas, nossos erros, inclusive nossas soluções		1
25	22	informar este feedback aos diretamente envolvidos		1
26	23	fazer apresentações exclusivamente com base em nosso conhecimento não prometendo coisas a mais, iludindo		1
27	24	antes de qualquer reunião expositiva de solução ou proposta ao cliente simular a apresentação dentro da Store com os diversos envolvidos e não envolvidos		1
28	25	a toda solicitação/ordem de serviço deixar claro o contexto do assunto		1
29	26	atas de reunião		1
30	27	considerando que há clientes que não se veem beneficiados pelo sistema ou que não conseguem extrair todas informações que gostariam	X	1
31	28	que nem todos os usuários acabam sendo consultados a respeito de uso, satisfação e extração de informação		1
32	29	treinar os colaboradores Store em técnicas de coleta e triagem de informações e abordagem de soluções a problemas		1
33	30	considerando que não se deve iludir aos clientes e nem morrer estressado	X	1
34	31	elaborar curvas de Pareto		1
35	32	montar uma base de conhecimento das demandas do cliente		1
36	33	montar uma base de conhecimento de atas		1
37	34	obrigatoriamente antes de propor solução conferir in loco os processos do cliente e fluxograma-los ou relata-los e documentá-los		1
38	35	tentar manter calma, não se afobar, não se precipitar ao encerrar a elicitação apesar de custo de levantamento, pressões diversas, etc...		1
39	36	não adianta apenas documentar, mas explicar item a item o que foi levantado apresentando ao cliente		1
40	37	elaborar e manter base de conhecimento domínio do negócio do cliente		1
41	38	preencher planilha de apontamentos de atividades (MIS)		
42	39	planilha de levantamento de cliente (não está funcionando por que o cliente não responde)		X1

43	40	suporte on-line		X1
44	41	com base nesta planilha recuperar informações do sistema e cliente: stakeholder, dados de banco para suporte (on-line)		X1
45	42	implantar um CRM (MIS)		
46	43	indicar as melhorias de cada release em uma planilha disponibilizando-a no MIS (já disponível)		
47	44	melhorar codificação das atividades		
48	45	implantar uma interface de processamento rápido no MIS para suportar a função CRM de tracking das demandas do cliente		
49	46	complementar no MIS o percentual de horas a completar determinada tarefa		
50	47	efetuar uma adaptação no MIS que permita customizar etapas, fases, atividades por projeto E ainda categorizá-los		
51	48	fazer um brain storm para o MIS		
52	49	porque o MIS e não uma ferramenta CRM de mercado?		
53	50	Manter um Especialista Store como Gerente do Cliente		
54	51	Manter um Especialista Store como Gerente do Produto		
55	52	Mudar comportamento => Desenhar mais Fluxogramas		
56	53	Prover Recursos => Ferramenta simples de fluxogramação		
57	54	Prover Recursos => Ferramenta CASE		
58	55	Utilização de fluxogramação com notação UML		1
59	56	Utilizar Programação Orientada a Objetos		
60	57	Utilizar técnicas / metodologias XP e Agile Development Techniques		1
61	58	encenação		1
62	59	reunião de preparo à apresentação ao cliente		1
63	60	reunião de planejamento da encenação		1
64	61	desenvolvimento da encenação		1
65	62	entrevistas		1
66	63	elabora questionário		1
67	64	aplica questionário		1
68	65	prepara visita		1
69	66	efetua visita		1
70	67	registra ata de reunião		1
71	68	filmar	D	1
72	69	fotografar	D	1
73	70	gravar	D	1
74	71	elaborar um histórico do cliente	D	1
75	72	elaborar um dossiê de cliente	D	1
76	73	resgatar o dossiê do cliente	D	1
77	74	estudar o dossiê do cliente	D	1
78	75	pesquisar as soluções de concorrentes	D	1
79	76	compilar as best practices dos sistemas concorrentes	D	1
80	77	MANTER o dossiê do cliente		1
81	78	Revisões periódicas da Elicitação de Requisitos junto ao cliente		1
82	501		F ou D	ENG.1
83	503	F = Futuramente ; D = Desejavelmente		
84	504	ENG.1 - Requirements Elicitation		
85	505	ENG.02 - System requirements analysis		
86	506	ENG.03 - System architectural design		
87	507	ENG.04 - Software requirements analysis		
88	508	ENG.05 - Software design		
89	509	ENG.06 - Software construction		
90	510	ENG.07 - Software integration		
91	511	ENG.08 - Software testing		
92	512	ENG.09 - System integration		
93	513	ENG.10 - System testing		
94	514	ENG.11 - Software installation		
95	515	ENG.12 - System and software maintenance		

ANEXO 5.04**GMP-ENG.1 - Guia Mestre do Processo ELICITAÇÃO DE REQUISITOS
(ENG.1)****GMP-ENG.1 - Requirements elicitation**

Process ID	ENG.1	Updated on	30/01/2004
Process Name	Requirements elicitation		
Process Purpose	The purpose of the Requirements elicitation process is to gather, process, and track evolving customer needs and requirements throughout the life of the product and/or service so as to establish a requirements baseline that serves as the basis for defining the needed work products. Requirements elicitation may be performed by the acquirer or the developer of the system.		
Process Outcomes	<p>As a result of successful implementation of Requirements elicitation process:</p> <ol style="list-style-type: none"> 1) continuing communication with the customer is established; 2) agreed customer requirements are defined and baselined; 3) a change mechanism is established to evaluate and incorporate changes to customer requirements into the baselined requirements based on changing customer needs; 4) a mechanism is established for continuous monitoring of customer needs; 5) a mechanism is established for ensuring that customers can easily determine the status and disposition of their requests; and 6) enhancements arising from changing technology and customer needs are identified and their impact managed. 		
Base Practices	<p>ENG.1.BP1: Obtain customer requirements and requests. Obtain and define customer requirements and requests through direct solicitation of customer and user input. NOTE 1: Requirements may be obtained through review of customer business proposals, target operating and hardware environment, and other documents bearing on customer requirements. [Outcomes: 1,4]</p> <p>ENG.1.BP2: Understand customer expectations. Ensure that both supplier and customer understand each requirement in the same way. Review with customers their requirements and requests to better understand their needs and expectations. [Outcome: 6]</p> <p>ENG.1.BP3: Agree on requirements. Obtain agreement across teams on the customer's requirements, obtaining the appropriate sign-offs by representatives of all teams and other parties contractually bound to work to these requirements. [Outcome: 2]</p> <p>ENG.1.BP4: Establish customer requirements baseline. Formalise the customer's requirements and establish as a baseline for project use and monitoring against customer needs. [Outcomes: 2,3]</p> <p>ENG.1.BP5: Manage customer requirements changes. Manage all changes made to the customer requirements against the customer requirements baseline to ensure enhancements resulting from changing technology and customer needs are identified and that those who are affected by the changes are able to assess the impact and risks and initiate appropriate change control and mitigation actions. [Outcomes: 4,5]</p> <p>NOTE 2: The tracking of requirements is handled in CFG.2 Configuration management process.</p>		

ENG.1.BP6: Establish customer query mechanism. Provide a means by which the customer can be aware of the status and disposition of their requirements changes. [Outcome: 5]
 NOTE 3: This may include joint meetings with the customer or formal communication to review the status for their requirements and requests; Refer to the process QUA.4 Joint Review.

ANEXO 5.05

TAP.01 - Tabela de Avaliação (INICIAL) do Processo ELICITAÇÃO DE REQUISITOS (ENG.1)

A1	B	C	E	F	G	H	I	J	K
		Avaliação dos 78 PROBLEMAS / NECESSIDADES ou PRÁTICAS antes do 1o. CMQ							
		TAP.01 - Tabela de Avaliação (INICIAL) do Processo ELICITAÇÃO DE REQUISITOS (ENG.01)			A PRÁTICA já ocorria, no ENG.1, em out.04 em até x% do ideal:				
		PRÁTICAS (ATIVIDADE / Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)	EN G.0	zero %	até 25%	até 50%	até 75%	até 100 %	ATÉ
2	1	elaborar mapa de competências	1						
3	2	elaborar relatório de aderência	1						
4	3	inserir atividades no MIS						100	100
5	4	definir fluxos de processos	1		25				25
6	5	levantamentos informais junto ao cliente através de interações informais sociais	1		25				25
7	6	trazer o cliente à STORE na fase de elicitação e posteriores	1						
8	7	montar uma estratégia de conquistar a simpatia do cliente com base no dossiê	1						
9	8	elaborar e manter permanentemente atualizado um dossiê do cliente	1						
10	9	promover sempre segunda reunião para rever o entendimento da reunião anterior	1						
11	10	ao inicio de cada reunião, sempre rever a reunião anterior	1						
12	11	elaborar um planejamento de cada reunião	1						
13	12	identificar no planejamento da reunião as características da empresa, contexto e stakeholders/participante	1		25				25
14	13	idem ao que está por traz da necessidade do cliente (real causa)	1		25				25
15	14	elaborar protótipo	1		25				25
16	15	idem levar sempre alternativas, proposições que conquistem o cliente ajudando a resolver seus problemas	1			50			50
17	16	elaborar uma ata da reunião de elicitação não deixando dúvidas sobre o negócio/problema do cliente (não se inibir de demonstrar o desconhecimento ao cliente com cautela...)	1		25				25
18	17	não deixar escapar termos essenciais ao negócio do cliente	1				75		75
19	18	provocar um levantamento de soluções ao problema do cliente não somente através do cliente e sim também através de outras fontes	1			50			50
20	19	montar uma base de conhecimento e necessidades dos processos de todos os clientes	1						
21	20	usar esta base de conhecimento na proposta de novas releases do software	1						
22	21	a cada nova reunião solicitar feedback do cliente sobre a nossa atuação, nossos problemas, nossos erros, inclusive nossas soluções	1						
23	22	informar este feedback aos diretamente envolvidos	1						

28	23	fazer apresentações exclusivamente com base em nosso conhecimento não prometendo coisas a mais, iludindo	1						
29	24	antes de qualquer reunião expositiva de solução ou proposta ao cliente simular a apresentação dentro da Store com os diversos envolvidos e não envolvidos	1						
30	25	a toda solicitação/ordem de serviço deixar claro o contexto do assunto	1						
31	26	atas de reunião	1						
32	27	considerando que há clientes que não se veem beneficiados pelo sistema ou que não conseguem extrair todas informações que gostariam (C ₁₂ U ₁)	1						
33	28	que nem todos os usuários acabam sendo consultados a respeito de uso, satisfação e extração de informação (C ₁₂ U ₁)	1						
34	29	treinar os colaboradores Store em técnicas de coleta e triagem de informações e abordagem de soluções a problemas (C ₁₂ U ₁)	1						
35	30	considerando que não se deve iludir os clientes e nem morrer estressado (C ₃ U ₂₃₄)	1						
36	31	elaborar curvas de Pareto (C ₃ U ₂₃₄)	1						
37	32	montar uma base de conhecimento das demandas do cliente (C ₃ U ₂₃₄)	1						
38	33	montar uma base de conhecimento de atas (C ₃ U ₂₃₄)	1						
39	34	obrigatoriamente antes de propor solução conferir in loco os processos do cliente e fluxograma-los ou relata-los e documentá-los	1		25				25
40	35	tentar manter calma, não se afobar, não se precipitar ao encerrar a elicitação apesar de custo de levantamento, pressões diversas, etc...	1						
41	36	não adianta apenas documentar, mas explicar item a item o que foi levantado apresentando ao cliente	1						
42	37	elaborar e manter base de conhecimento domínio do negócio do cliente	1						
43	38	preencher planilha de apontamentos de atividades (MIS)						100	100
44	39	planilha de levantamento de cliente (não está funcionando por que o cliente não responde)	X1						
45	40	suporte on-line	X1						
46	41	com base nesta planilha recuperar informações do sistema e cliente: stakeholder, dados de banco para suporte (on-line)	X1						
47	42	implantar um CRM (MIS)					75		75
48	43	indicar as melhorias de cada release em uma planilha disponibilizando-a no MIS (já disponível)							
49	44	melhorar codificação das atividades				50			50
50	45	implantar uma interface de processamento rápido no MIS para suportar a função CRM de tracking das demandas do cliente	1		25				25
51	46	complementar no MIS o percentual de horas a completar determinada tarefa			25				25
52	47	efetuar uma adaptação no MIS que permita customizar etapas, fases, atividades por projeto E ainda categorizá-los			25				25
53	48	fazer um brain storm para o MIS			25				25
54	49	porque o MIS e não uma ferramenta CRM de mercado?			25				25
55	50	Manter um Especialista Store como Gerente do Cliente	1						
56	51	Manter um Especialista Store como Gerente do Produto							
57	52	Mudar comportamento => Desenhar mais Fluxogramas	1						
58	53	Prover Recursos => Ferramenta simples de fluxogramação	1						
59	54	Prover Recursos => Ferramenta CASE	1						
60	55	Utilização de fluxogramação com notação UML	1						
61	56	Utilizar Programação Orientada a Objetos	1						
62	57	Utilizar técnicas / metodologias XP e Agile Development Techniques	1						

63	58	ENCENAÇÃO (ENCEN)	1		25			25
64	59	reunião de preparo à apresentação ao cliente (ENCEN)	1					
65	60	reunião de planejamento da encenação (ENCEN)	1					
66	61	desenvolvimento da encenação (ENCEN)	1					
67	62	ENTREVISTAS (ENTREV)	1					
68	63	elabora questionário (ENTREV)	1					
69	64	aplica questionário (ENTREV)	1					
70	65	prepara visita (ENTREV)	1		25			25
71	66	efetua visita (ENTREV)	1				100	100
72	67	registra ata de reunião (ENTREV)	1		25			25
73	68	filmar	1					
74	69	fotografar	1					
75	70	gravar	1					
76	71	elaborar um histórico do cliente (DOSSIE)	1					
77	72	elaborar um dossiê de cliente (DOSSIE)	1					
78	73	resgatar o dossiê do cliente (DOSSIE)	1					
79	74	estudar o dossiê do cliente (DOSSIE)	1					
80	75	pesquisar as soluções de concorrentes (DOSSIE)	1		25			25
81	76	compilar as best practices dos sistemas concorrentes (DOSSIE)	1				100	100
82	77	MANTER o dossiê do cliente (DOSSIE)	1					
83	78	Revisões periódicas da Elicitação de Requisitos junto ao Cliente	1					
84	90	<p>Preencher com 1 quando a PRÁTICA está relacionada ao Processo em questão</p> <p>Indicar em quantos % a prática já era utilizada no processo em foco (ELICITAÇÃO) em out.04.</p> <p>Preencha com 0, 25, 50, 75 ou 100 na célula apropriada conforme seja o caso.</p> <p>Exemplo:</p> <p>Se a utilização do UML na ELICITAÇÃO (ENG.1) era nula, então preencha 0 na célula F60 mesmo que o uso do UML fosse total em demais PROCESSOS como, hipoteticamente, no ENG.5 (SW Design).</p>						
85	90							
86	90							
87	90							
88	90							
89	90							
90	90							
91	90							

ANEXO 5.06

GMCP-ENG.1 - Guia da Melhoria da Capabilidade do Processo ELICITAÇÃO DE REQUISITOS (ENG.1)

A1	B	C	D	E	F	G	H	I	J	K
		Classificação de PROBLEMAS / NECESSIDADES ou PRÁTICAS antes do 1o. CMQ								
2	0									
3	0	GMCP-ENG.01 - Guia da Melhoria da Capabilidade do Processo ELICITAÇÃO DE REQUISITOS (ENG.01)				A PRÁTICA já ocorria, no ENG.1, em out.04 em até x% do ideal:				
4	0									
5	0	PRÁTICAS (ATIVIDADE / Processo / Decisão / Artefato / etc... que visa melhorar o PROCESSO em foco)	C => U; D; F	EN G.0 1	zero %	até 25%	até 50%	até 75%	até 100 %	ATÉ
32	27	considerando que há clientes que não se veem beneficiados pelo sistema ou que não conseguem extrair todas informações que gostariam (C ₁₂ U ₁)	C ₁	1						0
33	28	que nem todos os usuários acabam sendo consultados a respeito de uso, satisfação e extração de informação (C ₁₂ U ₁)	C ₂	1						0
35	30	considerando que não se deve iludir os clientes e nem morrer estressado (C ₃ U ₂₃₄)	C ₃	1						0
9	4	definir fluxos de processos	U ₀	1		25				25
13	8	elaborar e manter permanentemente atualizado um dossiê do cliente	U ₀	1	0					0
19	14	elaborar protótipo	U ₀	1		25				25
41	36	não adianta apenas documentar, mas explicar item a item o que foi levantado apresentando ao cliente	U ₀	1	0					0
57	52	Mudar comportamento => Desenhar mais Fluxogramas	U ₀	1	0					0
58	53	Prover Recursos => Ferramenta simples de fluxogramação	U ₀	1	0					0
60	55	Utilização de fluxogramação com notação UML	U ₀	1	0					0
77	72	elaborar um dossiê de cliente (DOSSIE)	U ₀	1	0					0
34	29	treinar os colaboradores Store em técnicas de coleta e triagem de informações e abordagem de soluções a problemas (C ₁₂ U ₁)	U ₁	1	0					0
78	73	resgatar o dossiê do cliente (DOSSIE)	U ₁	1	0					0
79	74	estudar o dossiê do cliente (DOSSIE)	U ₁	1	0					0
82	77	MANTER o dossiê do cliente (DOSSIE)	U ₁	1	0					0
83	78	Revisões periódicas da Elicitação de Requisitos junto ao Cliente:	U ₁	1						0
36	31	elaborar curvas de Pareto (C ₃ U ₂₃₄)	U ₂	1	0					0
37	32	montar uma base de conhecimento das demandas do cliente (C ₃ U ₂₃₄)	U _m	1	0					0
38	33	montar uma base de conhecimento de atas (C ₃ U ₂₃₄)	U _n	1	0					0
39	34	obrigatoriamente antes de propor solução conferir in loco os processos do cliente e fluxograma-los ou relata-los e documentá-los	U _q	1		25				25
67	62	ENTREVISTAS (ENTREV)	U _r	1	0					0
68	63	elabora questionário (ENTREV)	U _r	1	0					0
69	64	aplica questionário (ENTREV)	U _r	1	0					0
70	65	prepara visita (ENTREV)	U _r	1		25				25
71	66	efetua visita (ENTREV)	U _r	1					100	100
72	67	registra ata de reunião (ENTREV)	U _r	1		25				25
76	71	elaborar um histórico do cliente (DOSSIE)	U _s	1	0					0
80	75	pesquisar as soluções de concorrentes (DOSSIE)	U _t	1		25				25
81	76	compilar as best practices dos sistemas concorrentes (DOSSIE)	U _u	1					100	100
6	1	elaborar mapa de competências			0					0
7	2	elaborar relatório de aderência		1	0					0

8	3	inserir atividades no MIS						100	100
10	5	levantamentos informais junto ao cliente através de interações informais sociais		1		25			25
11	6	trazer o cliente à STORE na fase de elicitação e posteriores		1	0				0
12	7	montar uma estratégia de conquistar a simpatia do cliente com base no dossiê		1	0				0
14	9	promover sempre segunda reunião para rever o entendimento da reunião anterior		1	0				0
15	10	ao início de cada reunião, sempre rever a reunião anterior		1	0				0
16	11	elaborar um planejamento de cada reunião		1	0				0
17	12	identificar no planejamento da reunião as características da empresa, contexto e stakeholders/participante		1		25			25
18	13	idem ao que está por traz da necessidade do cliente (real causa)		1		25			25
20	15	idem levar sempre alternativas, proposições que conquistem o cliente ajudando a resolver seus problemas		1		50			50
21	16	elaborar uma ata da reunião de elicitação não deixando dúvidas sobre o negócio/problema do cliente (não se inibir de demonstrar o desconhecimento ao cliente com cautela...)		1		25			25
22	17	não deixar escapar termos essenciais ao negócio do cliente		1				75	75
23	18	provocar um levantamento de soluções ao problema do cliente não somente através do cliente e sim também através de outras fontes		1		50			50
24	19	montar uma base de conhecimento e necessidades dos processos de todos os clientes		1	0				0
25	20	usar esta base de conhecimento na proposta de novas releases do software		1	0				0
26	21	a cada nova reunião solicitar feedback do cliente sobre a nossa atuação, nossos problemas, nossos erros, inclusive nossas soluções		1	0				0
27	22	informar este feedback aos diretores envolvidos		1	0				0
28	23	fazer apresentações exclusivamente com base em nosso conhecimento não prometendo coisas a mais, iludindo		1	0				0
29	24	antes de qualquer reunião expositiva de solução ou proposta ao cliente simular a apresentação dentro da Store com os diversos envolvidos e não envolvidos		1	0				0
30	25	a toda solicitação/ordem de serviço deixar claro o contexto do assunto		1	0				0
31	26	atas de reunião		1	0				0
40	35	tentar manter calma, não se afobar, não se precipitar ao encerrar a elicitação apesar de custo de levantamento, pressões diversas, etc...		1	0				0
42	37	elaborar e manter base de conhecimento domínio do negócio do cliente		1	0				0
43	38	preencher planilha de apontamentos de atividades (MIS)						100	100
44	39	planilha de levantamento de cliente (não está funcionando por que o cliente não responde)		X1	0				0
45	40	suporte on-line		X1	0				0
46	41	com base nesta planilha recuperar informações do sistema e cliente: stakeholder, dados de banco para suporte (on-line)		X1	0				0
47	42	implantar um CRM (MIS)						75	75
48	43	indicar as melhorias de cada release em uma planilha disponibilizando-a no MIS (já disponível)			0				0
49	44	melhorar codificação das atividades				50			50
50	45	implantar uma interface de processamento rápido no MIS para suportar a função CRM de tracking das demandas do cliente		1		25			25
51	46	complementar no MIS o percentual de horas a completar determinada tarefa				25			25

ANEXO 5.07
INFORMATIVO STORE.

Nº 02 Outubro de 2004



informe**ativo**

4 Tecnologia
PMQ na Store

5 Clientes
BASF



7 Entrevistas
Daniela
Luiz Antônio
João Paulo

6 Mercado
Onde atua o
STORE/EADI

1 Editorial

2 Palavra do Presidente

3 Em Andamento

8 Minha área

9 Fatos e Atos

10 Aniversários

11 Humor

12 Comunidade

O InformeAtivo é uma realização da Store Automação. Responsável: Roger Dias Barbosa • Projeto Gráfico: Mariana Lima Scaliza.

O PMQ na Store continua a todo vapor!

O Programa de Melhoria da Qualidade baseia-se em estudos desenvolvidos pela Fábrica de Software em conjunto com uma dissertação de mestrado em defesa na UNIMEP – Universidade Metodista de Piracicaba, a partir de modelos consagrados como o CMM (Capability and Maturity Model) e o SPICE (Software Process Improvement & Capability dEtermination) e do conjunto de melhores práticas compiladas no PMBOK (Project Management Body Of Knowledge).

Graças a uma metodologia oriunda da dissertação de mestrado e adaptada às peculiaridades da Store e de seus clientes, focamos a área relativa à Engenharia de Software.

O PMQ tem uma característica contínua, com metas de curto, médio e longo prazo, sendo que as metas mais imediatas são o enquadramento da Fábrica de Software no nível 2 do CMM, seguida do nível 3 como meta de médio prazo.

No entanto, adotamos uma abordagem orientada pela norma ISO/IEC 15.504, onde a ênfase recai sobre a capacidade dos processos conforme as conveniências de cada organização.

O projeto se iniciou em meados de 2003 com a iniciativa da Diretoria Técnica e adquiriu um reforço externo quando Júlio César Navas, ex-técnico e executivo de Informática em grandes grupos nacionais e multinacionais, hoje atuando como consultor em projetos de sistemas e de melhoria da qualidade, empenhado em obter o título de Mestre em Ciência da Computação pela Universidade Metodista de Piracicaba, juntou-se ao nosso time.

Seu tema de estudo, "Um Método para Melhoria do Processo de Desenvolvimento de Software aplicando conceitos de CMM, SPICE e a norma ISO 12207", encaixava-se perfeitamente em nosso foco de atenção, assim começamos um trabalho conjunto que tem se mostrado altamente produtivo, pois de ambas as partes há uma forte e constante interação e intercâmbio de conhecimento.

“Assim começamos um trabalho conjunto que tem se mostrado altamente produtivo.”

GLOSSÁRIO

Por razões de uso de terminologia já corrente no ramo do conhecimento e na aplicação do controle de qualidade relacionado ao *software*, julgou-se apropriado apresentar os termos conforme constam de traduções utilizadas pela própria ABNT ou conforme constam na bibliografia referenciada por este trabalho. Assim, o glossário a seguir, embora obtido de diversas fontes, utilizou primordialmente o formato e grande parte dos termos do próprio glossário da ABNT obtido do site ao final desta explicação, com ligeiras adaptações e formatação mais prática.

Evitou-se utilizar neste trabalho alguns termos apropriados da língua inglesa em adaptações semânticas que possam causar estranheza, como é o caso das palavras com sufixo “dade” costumeiramente utilizadas em substituição ao sufixo “ção”. No entanto, por constarem diversos desses termos na bibliografia consultada, preferimos mantê-los neste glossário no intuito de facilitar o entendimento do assunto a quem aqui buscar referências.

É o caso, como exemplos, dos termos: analisabilidade (qualidade de ser analisável); modificabilidade (qualidade de modificável, ou capacidade de modificação); extensibilidade (qualidade de extensível, ou capacidade de extensão); interoperabilidade (qualidade de interoperacionável, ou capacidade de interoperação); recuperabilidade (qualidade de recuperável, ou capacidade de recuperação); usabilidade (qualidade de usável, ou, simplesmente, capacidade de uso), alguns dos quais já são adotados em normas publicadas pela ABNT, referenciadas ao lado do verbete neste glossário.

Cabe ressaltar que se evitou o uso do termo CAPABILIDADE DO PROCESSO [CMM - Nível 1, 2001] entendido como sinônimo de CAPACITAÇÃO DO PROCESSO, pois, embora amplamente utilizado coloquialmente há vários anos pela comunidade de especialistas e pesquisadores envolvidos com a melhoria da qualidade do processo de

construção de *software*, pode causar estranheza decorrente da não existência do verbete CAPABILIDADE em dicionário;

Possivelmente o uso de tais termos, que já se fazem presentes na bibliografia utilizada, tenha escapado à última revisão em alguma parte deste trabalho, pelo qual o pesquisador, sem nenhuma intenção de novação, pede desculpas.

(fonte: <http://www.celepar.gov.br/abntsoftware/ce10101/glo10101.html> em 31.ago.2003)

Termos	DEFINIÇÕES	DOCUMENTOS REFERIDOS
A		
Accuracy	Acurácia	
Acordo	(Agreement) A definição de termos e condições sob a qual o relacionamento de trabalho entre as partes deverá ser conduzido.	ISO/IEC 12207 e NBR 12207
Acurácia	(Accuracy) Atributos do software que evidenciam a geração de resultados ou efeitos corretos ou conforme acordados.	ISO/IEC 9126 e NBR 13596
ad hoc	esporádico	
Adaptabilidade	(Adaptability) Atributos do software que evidenciam sua capacidade de ser adaptado a ambientes diferentes especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para esta finalidade pelo software considerado.	ISO/IEC 9126 e NBR 13596
Adaptability	Adaptabilidade	
Adequação	(Suitability) Atributos do software que evidenciam a presença de um conjunto de funções e sua apropriação para as tarefas especificadas.	ISO/IEC 9126 e NBR 13596
Aderência	(Compliance)	ISO/IEC 14598-1
Agreement	Acordo	
Analisabilidade	(Analysability) Atributos do software que evidenciam o esforço necessário para diagnosticar deficiências ou causas de falhas, ou para identificar partes a serem modificadas.	ISO/IEC 9126 e NBR 13596
Analysability	Analisabilidade	
Aplicabilidade	aplicação ou qualidade de aplicável, ou, ainda, capacidade de aplicação.	
Application Software	Software Aplicativo	
Apreensibilidade	(Learnability) Atributos do software que evidenciam o esforço do usuário para aprender sua aplicação (por exemplo: controle de operação, entradas, saídas).	ISO/IEC 9126 e NBR 13596

Aquisição	(Acquisition) O processo de obtenção de um sistema, produto de software ou serviço de software.	ISO/IEC 12207 e NBR 12207
Artefato	(artifact) elemento real, concreto, que encerra conhecimento, informação ou dado (ex: relatório, programa, sistema, manual, documento, arquivo, gravação sonora).	
artifact	A tangible output, such as a work product, produced from the execution of an implemented process.	
assessed capability	The output of one or more recent, relevant process assessments conducted in accordance with the provisions of this International Standard.	
Assessment	Julgamento	
Atributo	(Attribute) Uma propriedade mensurável, física ou abstrata, de uma entidade. Nota: Atributos podem ser internos ou externos.	ISO/IEC 14598-1
Attribute	Atributo	
Audit	Auditoria	
Auditoria	(Audit) Processo conduzido por uma pessoa autorizada com o objetivo de prover um julgamento independente de produtos e processos de software, a fim de avaliar a conformidade com seus requisitos.	ISO/IEC 12207 e NBR 12207
Avaliação	(Evaluation) Uma determinação sistemática do grau de atendimento de uma entidade em relação aos critérios para ela estabelecidos.	ISO/IEC 12207 e NBR 12207
Avaliação de qualidade	(Quality Evaluation) Exame sistemático do quanto uma entidade é capaz de atender requisitos especificados. [ISO 8402:1994] Nota: Os requisitos podem ser formalmente especificados, por exemplo quando: Um produto é desenvolvido para um usuário específico sob um contrato; Um produto é desenvolvido para um usuário não específico, ou seja especificado por uma organização de desenvolvimento. Os requisitos também podem ser mais gerais, como quando um usuário avalia produtos com o objetivo de comparação e seleção.	ISO/IEC 14598-1
B		
base practice	A software engineering or management activity that directly addresses the purpose of a particular process and contributes to the creation of its output. A base practice is an essential activity of a particular process.	
base practice existence	A judgement, within the process context, of whether a base practice is implemented and produces some output.	
Benchmark	padrão de desempenho; (a) ponto num índice que é importante, e que pode ser usado para comparar com outros valores; (b) programa usado para testar o desempenho de software ou hardware ou um sistema;	
Benchmarking	provação padrão; teste de um sistema ou programa com uma prova padrão. comentário: a mesma tarefa ou programa é dado para diferentes sistemas e seus resultados e velocidades de trabalho são comparados.	

C		
CAE	computer-aided engineering	
capability level	A set of common features (i.e. generic practices) that works together to provide a major enhancement in the capability to perform a process.	
Capabilidade	capacitação ou capacidade de capacitação	CMM - Nível 1 (2001)
Capabilidade do Processo	entende-se como sinônimo de CAPACITAÇÃO DO PROCESSO.	CMM - Nível 1 (2001)
Capacidade para ser instalado	(Installability) Atributos do software que evidenciam o esforço necessário para sua instalação num ambiente especificado.	ISO/IEC 9126 e NBR 13596
Capacidade para substituir	(Replaceability) Atributos do software que evidenciam sua capacidade e esforço necessário para substituir um outro software, no ambiente estabelecido para esse outro software.	ISO/IEC 9126 e NBR 13596
Capacitação do Processo	é a capacidade intrínseca de um processo reproduzir seus resultados de forma consistente, durante ciclos múltiplos de operação.	JURAN (1992)
Características de qualidade de software	(Software quality characteristics) Um conjunto de atributos de um produto de software, através do qual sua qualidade é descrita e avaliada. Uma característica de qualidade de software pode ser detalhada em múltiplos níveis de subcaracterísticas.	ISO/IEC 9126 e NBR 13596
CEO	chief executive officer	
Changeability	Modificabilidade	
Check program	Problema de teste	
Checklist	Lista de Verificação	
CIO	chief information officer	
Clause	Seção	
Clause	Subseção	
Cliente	(Customer)	ISO/IEC 14598-1
CMM	(Capability Maturity Model) - modelo de maturidade de capacitação de organizações para o software criado pelo Instituto de Engenharia de Software (SEI) que descreve os elementos de um processo efetivo de desenvolvimento, implantação e manutenção de software.	
CMMI®	Capability Maturity Model Integration	
Cobertura de teste	(Test coverage) A extensão em que os casos de teste dos requisitos de um sistema ou produto de software são testados.	ISO/IEC 12207 e NBR 12207
competence	The work performance that results from effectively applying skills, knowledge and personal attributes.	
competency	The skills, knowledge and personal attributes that enable effective work performance.	
Completion	Encerramento	
Completeness	(Completeness)	ISO/IEC 12119 e NBR 12119
Completeness	Completeness	

Compliance	Aderência ou conformidade	
Compliance	Conformidade(1)	
Comportamento em relação ao tempo	(Time behaviour) Atributos do software que evidenciam seu tempo de resposta, tempo de processamento e velocidade na execução de suas funções.	ISO/IEC 9126 e NBR 13596
Comportamento em relação aos recursos	(Resource behaviour) Atributos do software que evidenciam a quantidade de recursos usados e a duração de seu uso na execução de suas funções.	ISO/IEC 9126 e NBR 13596
Confiabilidade	(Reliability) Conjunto de atributos que evidenciam a capacidade do software de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo estabelecido.	ISO/IEC 9126 , NBR 13596 e 12119
Configuração	(Configuration) A maneira na qual o hardware e o software de um sistema de processamento de informação são organizados e interconectados. (ISO 2382-1:1993)	ISO/IEC 12119 e NBR 12119
Conformance	Conformidade(2)	
Conformidade(1)	(Compliance) Atributos do software que fazem com que o mesmo esteja de acordo com as normas, convenções ou regulamentações previstas em leis e descrições similares, relacionadas à aplicação.	ISO/IEC 9126 e NBR 13596
Conformidade(2)	(Conformance) Atributos do software que o tornam consonante com padrões ou convenções relacionados à portabilidade.	ISO/IEC 9126 e NBR 13596
Contract	Contrato	
Contrato	(Contract) Um acordo realizado entre duas partes, respaldado pela lei, ou acordo interno similar restrito a uma organização, para o fornecimento de serviços de software ou para o fornecimento, desenvolvimento, produção, operação ou manutenção de um produto de software.	ISO/IEC 12207 e NBR 12207
Critério de julgamento de qualidade de software	(Software quality assessment criteria) O conjunto de regras e condições definidas e documentadas que são usadas para decidir se a qualidade total de um produto de software específico é aceitável ou não. A qualidade é representada pelo conjunto de níveis de pontuação associados ao produto de software.	ISO/IEC 9126 e NBR 13596
critical mission system	Sistema de missão crítica	
Custom made software product	Produto de software sob encomenda	
Customer	Cliente	

D		
Dados	(Data) Representação reinterpretabil da informação de maneira formalizada, adequada para comunicação, interpretação ou processamento. (ISO 2382-1:1993)	ISO/IEC 12119 e NBR 12119
Dados de teste	(Test data) Dados utilizados em um problema de teste. (ISO 2382-8:1986)	ISO/IEC 12119 e NBR 12119
Data	Dados	
Data Medium de software	Meio de dados	e NBR 13596
Defeito	(Fault) Um passo, processo ou definição de dados incorretos em um programa de computador. Nota: Esta definição é mesma da norma IEEE 610.12-1990.	ISO/IEC 14598-1
defined process	The operational definition of a set of activities for achieving a specific purpose. A defined process is characterized by standards, procedures, training, tools, and methods.	
Descontinuação	(Retirement) Cancelamento do suporte ativo pela organização de operação e manutenção, substituição total ou parcial por um novo sistema, ou instalação de um sistema atualizado.	ISO/IEC 12207 e NBR 12207
Descrição de produto	(Product description) Um documento exposto as propriedades de um pacote de software, com o principal objetivo de auxiliar os potenciais compradores na avaliação da adequação do produto antes de sua aquisição. Nota: Este termo é mais específico que o termo "descrição de sistema" (system description), na ISO/IEC 2382-20. O objetivo da descrição do produto abrange também a "informações gerais" (cover information) da ISO 9127. A descrição de produto não é uma especificação, ela atende a um objetivo diferente.	ISO/IEC 12119 e NBR 12119
Descrição de tarefas	(Statement of work) Um documento utilizado pelo adquirente como um meio para descrever e especificar as tarefas a serem executadas conforme o contrato.	ISO/IEC 12207 e NBR 12207
Desenvolvedor	(Developer) Uma organização que executa atividades de desenvolvimento (incluindo análise de requisitos, projeto, testes até a aceitação) durante o processo de ciclo de vida do software.	ISO/IEC 14598-1, ISO/IEC 12207 e NBR 12207
Design	Projeto	
Destinatários Documento de requisitos	(Recipient) (Requirements document) Um documento contendo quaisquer combinações de recomendações, requisitos ou regulamentações a serem atendidas por um pacote de software. Nota: Exemplos: normas ergonômicas ou técnicas, lista de requisitos (ou modelo de especificação de requisitos) de um grupo (ex.: setor de marketing, associação de usuário ou técnica), uma lei ou um decreto.	ISO/IEC 14598-1 ISO/IEC 12119 e NBR 12119
E		
Efficiency	Eficiência	

Eficiência	(Efficiency) Conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas.	ISO/IEC 9126 , NBR 13596 e 12119
Encerramento	(Completion)	ISO/IEC 14598-1
enhanced capability	A capability greater than current assessed capability, justified by a credible process improvement programme.	
Enhancement	Melhoria	
Escala	(Scale) Um conjunto de valores com propriedades definidas. Nota: Exemplos de escala são: uma escala nominal que corresponde a um conjunto de categorias; uma escala ordinal que corresponde a um conjunto ordenado de pontos de escala; uma escala de intervalos que corresponde a uma escala ordenada com pontos de escala eqüidistantes; e uma escala de proporção (ratio scale) que tem pontos de escala eqüidistantes e também um zero absoluto. Métricas usando escalas ordinais ou nominais produzem dados qualitativos e métricas usando escalas de intervalos ou de proporção (ratio scales) produzem dados quantitativos.	ISO/IEC 14598-1
Escala Proporcional (ratio scale)	(Ratio Scale)	ISO/IEC 14598-1
Estabilidade	(Stability) Atributos do software que evidenciam o risco de efeitos inesperados ocasionados por modificações .	ISO/IEC 9126 e NBR 13596
Estrutura	(Framework)	ISO/IEC 14598-1
Etapa	(Step) Quando forem macro-atividades	ISO/IEC 14598-1
Etapa	Neste trabalho ETAPA é usado no sentido conjunto de Processos que permite o atingimento de metas intermediárias à conclusão de uma Fase (O MC2Q-SW se constitui de 13 etapas);	MC2Q-SW
Evaluation	Avaliação	
Evaluation module	Módulo de avaliação	
Explicitada	(Stated) Substituindo estabelecida.	ISO/IEC 14598-1
Extendability	Extensibilidade	
extended process	A process which differs from any process contained in part 2 of this International Standard, either by having additional base practices defined for an existing process or being an entirely new process. An extended process should conform to the requirements for laid down in Annex A in part 2 of this International Standard.	
Extensibilidade	(Extendability) qualidade de extensível, ou capacidade de extensão	ISO/IEC 14598-1
Extent	O quanto	
External measure	Medida externa	
External quality	Qualidade externa	
F		

Failure	Falha	
Falha	(Failure) O término da capacidade de um item de executar uma função requeridas ou a sua incapacidade de executá-las dentro de limites previamente especificados. (ISO CD 15026:19??)	ISO/IEC 14598-1
Fase	Neste trabalho: conjunto de Etapas que permite o atingimento de uma meta importante ao método (O MC2Q-SW se constitui de 3 fases);	MC2Q-SW
Fault	Defeito	
Fault tolerance	Tolerância a falhas	
Features	Particularidades	NBR 13596
Feedback	(Feedback)	ISO/IEC 14598-1
Firmware	Hardware que contém os dados e um programa de computador que não podem ser alterados no ambiente do usuário. Os dados e o programa de computador contidos no firmware são classificados como software; os circuitos que contém os dados e o programa de computador são classificados como hardware. A combinação de um dispositivo de hardware e instruções ou dados de computador que residem como um software somente para leitura no dispositivo de hardware. Este software não pode ser diretamente modificado por um programa.	ISO/IEC 9126 e NBR 13596 ISO/IEC 12207 e NBR 12207
Fog index	(Fog index) É um método comprovado de análise de material escrito para verificar a sua facilidade de leitura e compreensão.	ISO/IEC 14598-1
Formalidade	Neste trabalho: bom gerenciamento e adequada documentação de procedimentos em relação à Melhoria da Qualidade do Processo.	
Fornece, Outorga	(Gives) Verificar o contexto utilizando alternativas para "dá".	ISO/IEC 14598-1
Fornecedor	(Supplier) Uma organização que firma um contrato com o adquirente para fornecimento de um sistema, produto de software ou serviço de software conforme os termos do contrato. Notas: 1 O termo "fornecedor" é sinônimo de contratado, produtor, vendedor ou distribuidor. 2 O adquirente pode designar uma parte de sua organização como fornecedor.	ISO/IEC 14598-1, ISO/IEC 12207 e NBR 12207
Framework	Estrutura	
Função	(Function) A implementação de um algoritmo em um programa com o qual o usuário ou o programa pode realizar toda uma tarefa ou parte dela. Notas: 1 Uma função não precisa ser necessariamente acionável pelo usuário (por exemplo, backup automático ou salvamento automático de dados).	ISO/IEC 12119 e NBR 12119

	os dados de testes; o procedimento; o comportamento esperado do sistema. Nota de Tradução: O termo "Guia de Teste" foi escolhido porque o termo "Caso de Teste" está sendo usado de acordo com autores como Pressman, Schach, Hetzel, Myers e entidades como o IEEE, significando apenas o conjunto de entradas de teste e saídas esperadas.	
Guidance	Orientação	
I		
IEEE	Institute of Electrical and Electronics Engineers	
Implied needs	Necessidades implícitas	
Indicador	(Indicator) Uma medida que pode ser utilizada para estimar ou prever outra medida. Notas: 1 A medida prevista pode, ou não, ser da mesma característica de qualidade de software. 2 Indicadores podem ser utilizados tanto para estimar atributos de qualidade de software, quanto para estimar atributos do processo de produção. Eles são medidas indiretas dos atributos.	ISO/IEC 14598-1
Indicator	Indicador	
Indirect measure	Medida indireta	
Informalidade	Neste trabalho: gerenciamento e documentação de procedimentos precários ou inexistentes em relação à Melhoria da Qualidade do Processo.	
Installability Inteligibilidade	Capacidade para ser instalado (Understandability) Atributos do software que evidenciam o esforço do usuário para reconhecer o conceito lógico e sua aplicabilidade.	ISO/IEC 9126 e NBR 13596
Interface	(Interface) Limite compartilhado entre duas unidades funcionais, definido por características funcionais, características de interconexão física comuns, características de sinal, e outras características, quando apropriado. (ISO/IEC 2382-9:1984, sem a Nota)	ISO/IEC 12119 e NBR 12119
Interface de usuário	(User interface) Uma interface que permite que as informações sejam passadas entre um usuário e componentes de hardware ou de software de um sistema computacional. (ANSI/IEEE Std 610.12-1990)	ISO/IEC 12119 e NBR 12119
Intermediário de compra (<i>procurer</i>)	(Procurer)	ISO/IEC 14598-1
Intermediate software product	Produto de software intermediário	
Internal measure	Medida interna	
Internal quality	Qualidade interna	

Interoperabilidade	(Interoperability) Atributos do software que evidenciam sua capacidade de interagir com sistemas especificados.	ISO/IEC 9126 e NBR 13596
Interoperability	Interoperabilidade	
ISO/IEC 14598	Série ISO/IEC 14599	
IT	information technology	
Item de configuração	(Configuration item) Uma entidade dentro de uma configuração que satisfaz uma função de uso final, e que pode ser identificada de forma única em um determinado ponto de referência.	ISO/IEC 12207 e NBR 12207
Item que não será entregue	(Non-deliverable item) Hardware ou produto de software cuja entrega não é exigida em contrato, mas pode ser utilizado no desenvolvimento do produto de software.	ISO/IEC 12207 e NBR 12207
J		
JAD	(Joint Application Design) - Projeto de Aplicações em Conjunto - técnica de estabelecimento de especificações de projetos desenvolvida pela IBM em que, através de reuniões com os agentes essenciais ao projeto, usuários ou não, conduzidas por um especialista na técnica, estabelecem-se projetos, prioridades e demais elementos fundamentais que direcionarão o trabalhos de desenvolvimento de sistemas em uma organização.	
Julgamento	(Assessment) Ação de aplicar critérios de julgamento específicos e documentados a um produto, pacote ou módulo de software específico, com o propósito de determinar sua aceitação ou liberação para uso.	ISO/IEC 9126 e NBR 13596
K		
L		
Learnability	Apreensibilidade	
Level of performance	Nível de desempenho	
Liberação (release)	(Release) Uma versão particular de um item de configuração que é colocada à disposição para um propósito específico (por exemplo, liberação para teste).	ISO/IEC 12207 , e NBR 12207
Life cycle model	Modelo de ciclo de vida	
Linha básica (baseline)	(Baseline) Uma versão formalmente aprovada de um item de configuração, independente de mídia, formalmente definida e fixada em um determinado momento durante o ciclo de vida do item de configuração.	ISO/IEC 12207 e NBR 12207
Lista de Verificação	(Checklist)	ISO/IEC 14598-1
M		
Maintainability	Manutenibilidade	
Maintainer	Mantenedor	
Maintenance	Manutenção	
Management	Gestão	
Mantenedor	(Maintainer) Uma organização que executa atividades de manutenção.	ISO/IEC 14598-1, 1,

		ISO/IEC 12207 e NBR 12207
Manual de manutenção de programa	(Program maintenance manual) Documento que fornece toda a informação necessária para a manutenção de um programa. (ISO/IEC 2382-20:1990)	ISO/IEC 12119 e NBR 12119
Manutenção	(Maintenance) Parte da manutenção de sistemas que está relacionada à modificação do pacote de software.	ISO/IEC 12119 e NBR 12119
Manutenção de sistema	(System maintenance) Alteração de um sistema para corrigir defeitos, para melhorar o desempenho, ou para adaptar o sistema às mudanças de ambiente ou de requisitos. (ISO/IEC 2382-20:1990)	ISO/IEC 12119 e NBR 12119
Manutenibilidade	(Maintainability) Conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no software.	ISO/IEC 9126 , NBR 13596 , ISO/IEC 12119 e NBR 12119
Maturidade	(Maturity) Atributos do software que evidenciam a frequência de falhas por defeitos no software.	ISO/IEC 9126 e NBR 13596
Maturity	Maturidade	
May	Pode	
May not	Não deve	
Measure (noun)	Medida	
Measure (verb)	Medir	
Measurement	Medição	
Medição	(Measurement) Ação de aplicar uma métrica de qualidade de software a um produto de software específico. O uso de uma métrica para atribuir um valor a partir de uma escala (o qual pode ser um número ou categoria) a um atributo de uma entidade. Nota: Medição pode ser qualitativa quando utilizar-se de categorias. Por exemplo, alguns atributos importantes de produtos de software, como a linguagem de um programa fonte (ADA, C, COBOL, etc.) são categorias qualitativas.	ISO/IEC 9126 e NBR 13596 ISO/IEC 14598-1
Medida	(Measure (noun)) Número ou categoria atribuído a um atributo de uma entidade através de uma medição.	ISO/IEC 14598-1
Medida direta	(Direct measure) Uma medida de um atributo que não depende da medida de qualquer outro atributo.	ISO/IEC 14598-1
Medida externa	(External measure) Uma medida indireta de um produto derivada de medidas do comportamento do sistema do qual ele faz parte. Notas: 1 O sistema inclui qualquer associação de hardware, software (seja personalizado ou de prateleira) e usuários.	ISO/IEC 14598-1

	<p>2 O número de falhas encontrados durante o teste é uma medida externa do número de defeitos no programa, porque o número de falhas é contado durante a operação de um sistema computacional executando um programa.</p> <p>3 Medidas externas podem ser utilizadas para avaliar atributos de qualidade de maneira mais próxima aos objetivos do projeto.</p>	
Medida indireta	<p>(Indirect measure) Uma medida de um atributo a qual é derivada de medidas de um outro ou de vários outros atributos.</p> <p>Nota: Uma medida externa de um atributo de um sistema de computação (como o tempo de resposta a uma entrada de usuário) é uma medida indireta de atributos do software, uma vez que a medida será influenciada por atributos do ambiente de computação, bem como por atributos do software.</p>	ISO/IEC 14598-1
Medida interna	<p>(Internal measure) Uma medida do próprio produto, seja direta ou indireta.</p> <p>Nota: O número de linhas de código, medidas de complexidade, o número de defeitos encontrados em um walkthrough e o Fog Index são todas medidas internas feitas no próprio produto.</p>	ISO/IEC 14598-1
Medir	(Measure (verb)) Fazer uma medição.	ISO/IEC 14598-1
Meio de dados	(Data Medium) Material no qual os dados podem ser armazenados e do qual podem ser recuperados. (ISO 2382-1:1993)	ISO/IEC 12119 e NBR 12119
Melhoria	(Enhancement)	ISO/IEC 14598-1
Método	processo organizado, lógico e sistemático de trabalhar, ou o "caminho para chegar a um fim".	AURÉLIO (1986)
Método de teste	(Test method) Procedimento técnico especificado para executar um teste. (ISO/IEC Guide 2:1991)	ISO/IEC 12119 e NBR 12119
Metric	Métrica	
Métrica	<p>(Metric) O método e a escala de medição definidos.</p> <p>Notas:</p> <p>1 Métricas podem ser internas ou externas, e diretas ou indiretas.</p> <p>2 Métricas incluem métodos para categorização de dados qualitativos.</p>	ISO/IEC 14598-1
Métrica de qualidade	(Software quality metric) Um método e uma escala quantitativa que podem ser usados para determinar o valor que uma particularidade (feature) recebe em um produto de software específico.	ISO/IEC 9126
Modelo	representação ou fonte de inspiração do que se deve executar, como forma ou padrão a ser imitado por sua perfeição.	HOUAISS (2006)
Modelo de ciclo de vida	(Life cycle model) Uma estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema desde a definição de seus requisitos até o término de seu uso.	ISO/IEC 12207 e NBR 12207

Modelo de propósito geral	(General-purpose model)	ISO/IEC 14598-1
Modelo de qualidade	(Quality model) O conjunto de características e os relacionamentos entre elas, que fornecem a base para a especificação dos requisitos de qualidade e para a avaliação de qualidade.	ISO/IEC 14598-1
Modificabilidade	(Changeability) Atributos do software que evidenciam o esforço necessário para modificá-lo, remover seus defeitos ou adaptá-lo a mudanças ambientais.	ISO/IEC 9126 e NBR 13596
Módulo de avaliação	(Evaluation module) Um pacote de tecnologia de avaliação para uma característica ou subcaracterísticas de qualidade de software específicas.	ISO/IEC 14598-1
Monitoração	(Monitoring) Um exame da situação das atividades de um fornecedor e dos seus resultados, efetuado pelo adquirente ou uma terceira parte.	ISO/IEC 12207 e NBR 12207
Monitoring	Monitoração	
N		
Não deve	(May not) Conforme Diretivas 3 da ABNT	
Necessidades implícitas	(Implied needs) Necessidades que podem não terem sido explicitadas, mas estão presentes quando a entidade é utilizada em condições particulares. Nota: Necessidades implícitas são necessidades reais que podem não ter sido documentadas.	ISO/IEC 14598-1
Níveis de Pontuação	(Rating Level)	ISO/IEC 14598-1
Nível de desempenho	(Level of performance) O grau em que as necessidades são satisfeitas, representado por um conjunto específico de valores para as características de qualidade.	ISO/IEC 9126 e NBR 13596
Nível de pontuação	(Rating level) Uma faixa de valores numa escala para permitir que o software seja classificado (pontuado) de acordo com as necessidades explícitas ou implícitas. Níveis de pontuação adequados podem ser atribuídos às diferentes visões de qualidade de usuários, gerentes e equipe de desenvolvimento. Estes níveis são chamados de níveis de pontuação. Um ponto de escala em uma escala ordinal o qual é usado para categorizar uma escala de medição. Notas: 1 O nível de pontuação permite que o software seja classificado (pontuado) de acordo com as necessidades explícitas ou implícitas. 2 Níveis de pontuação apropriados podem ser associados a diferentes visões de qualidade, por exemplo: visão dos usuários, dos gerentes ou dos desenvolvedores e equipes de desenvolvimento.	ISO/IEC 9126 e NBR 13596 ISO/IEC 14598-1
Non-deliverable item	Item que não será entregue	
Normatizar	(Standardisation)	ISO/IEC 14598-1

O		
O quanto	(Extent)	ISO/IEC 14598-1
Off-the-shelf product	Produto de prateleira	
Off-the-shelf software	Software de prateleira	
OMT	(Object Modeling Technique) - Técnica cujo objetivo é descrever a estrutura estática de um sistema em termos dos objetos que o constituem e de suas relações	
OOP	Object Oriented Project	
Operability	Operacionalidade	
Operacionalidade	(Operability) Atributos do software que evidenciam o esforço do usuário para sua operação e controle da sua operação.	ISO/IEC 9126 e NBR 13596
Operador	(Operator) Uma organização que opera o sistema.	ISO/IEC 12207 e NBR 12207
Operator	Operador	
organizational unit	That part of an organization that is the subject of an assessment. An organizational unit deploys one or more processes that have a coherent process context (q.v.) and operates within a coherent set of business goals. An organizational unit is typically part of a larger organization, although in a small organization, the organizational unit may be the whole organization. An organizational unit may be, for example: a specific project or set of (related) projects; a unit within an organization focused on a specific lifecycle phase (or phases) such as acquisition, development, maintenance or support; a part of an organization responsible for all aspects of a particular product or product set.	
Orientação	(Guidance)	ISO/IEC 14598-1
P		
Package documentation	Documentação de pacote	
Pacote de software	(Software package) Conjunto completo e documentado de programas fornecido a diversos usuários para uma aplicação ou função genérica. (ISO/IEC 2382-20:1990, sem a Nota)	ISO/IEC 12119 e NBR 12119
Particularidades	(Features) São propriedades identificadas de um produto de software que podem ser relacionadas às características de qualidade.	ISO/IEC 9126
Passo	(Step) Quando forem atividades elementares	ISO/IEC 14598-1
Path length	Comprimento do caminho	
Pedido de proposta	(Request for proposal (tender)) Um documento utilizado pelo adquirente como meio para divulgar aos potenciais fornecedores sua intenção de adquirir um sistema, produto de software ou serviço de software especificado.	ISO/IEC 12207 e NBR 12207
Perspectiva do usuário	(User's perspective)	ISO/IEC 14598-1

Plano de teste ou plano de avaliação e teste de sistema	(Test plan, system test and evaluation) Plano que estabelece requisitos detalhados, critérios, metodologia geral, responsabilidades e planejamento geral para testes e avaliação de um sistema. (ISO 2382-20:1990)	ISO/IEC 12119 e NBR 12119
Pode	(May) Conforme Diretivas 3 ABNT	
Pontuação	(Rating) Ação de mapear o valor medido ao nível de pontuação apropriado. Usado para determinar o nível de pontuação obtido pelo software em uma característica de qualidade específica. A ação de mapear o valor medido em comparação ao nível de pontuação apropriado dentro da escala. Utiliza-se para determinar o nível de pontuação do software para uma característica de qualidade específica.	ISO/IEC 9126 e NBR 13596 ISO/IEC 14598-1
Portabilidade	(Portability) Conjunto de atributos que evidenciam a capacidade do software de ser transferido de um ambiente para outro.	ISO/IEC 9126 , NBR 13596, ISO/IEC 12119 e NBR 12119
Portability	Portabilidade	
practice	A software engineering or management activity that contributes to the creation of the output (work products) of a process or enhances the capability of a process.	
Prática	(practice) é um conjunto de ações ou decisões com a finalidade de atender determinados objetivos;	
Predict	Prever	
Prever	(Predict)	ISO/IEC 14598-1
Problema de teste	(Check program) Problema com uma solução conhecida usada para determinar se uma unidade funcional está operando corretamente. (ISO 2382-8:1983)	ISO/IEC 12119 e NBR 12119
process	A set of activities [ISO/IEC12207-1] or A statement of purpose and an essential set of practices (activities) that address that purpose.	
Process	Processo	
process capability	The range of expected results that can be achieved by following a process. (See process performance for contrast.) [CMM Version 1.1 - CMU/SEI-93-TR-25].	
process category	A set of processes addressing the same general area of activity. The process categories address five general areas of activity: customer-supplier, engineering, project, support, and organization.	
process performance	A measure of the actual results achieved by following a process. (See process capability for contrast.) [CMM Version 1.1 - CMU/SEI-93-TR-25].	
process purpose	A summary description of the intent or functional objectives of a process and its base practices.	

Processo	(Process) Um conjunto de atividades inter-relacionadas, que transforma entradas em saídas. Nota: O termo "atividades" engloba a utilização de recursos. [Veja NBR ISO 8402/1994, 1.2] Neste trabalho preferiu-se adaptar esta definição para: PROCESSO é o conjunto de práticas necessárias a criar ou modificar artefatos;	ISO/IEC 12207 e NBR 12207
Processo de Software	Conjunto de práticas necessárias à construção e manutenção de software, segundo se compreende a partir dos trabalhos de Watts Humphrey nas décadas de 1980 e 1990 a partir de sua experiência com a IBM e para a criação do CMM no SEI (Software Engineering Institute).	
Procurer	Intermediário de compra; corretor; despachante.	
Product description	Descrição de produto	
Produto de prateleira	(Off-the-shelf product) Produto já desenvolvido e disponível para utilização na forma em que se encontra ou com modificação.	ISO/IEC 12207 e NBR 12207
Produto de software	(Software product) Uma entidade de software disponível para liberação a um usuário. O conjunto de programas de computador, procedimentos e possível documentação e dados associados. (ISO/IEC 12207:1995) (aguardar resultado da votação e verificar tradução da 12207 Nota: Produtos incluem produtos intermediários, e produtos destinados a usuários como desenvolvedores ou mantenedores e equipes de desenvolvimento e manutenção.	ISO/IEC 9126 e NBR 13596 ISO/IEC 14598-1, ISO/IEC 12207 e NBR 12207
Produto de software intermediário	(Intermediate software product) Um produto do processo de desenvolvimento de software que é utilizado como entrada para para iniciar um outro estágio do processo de desenvolvimento de software. Nota: Em alguns casos um produto intermediário também pode ser um produto final.	ISO/IEC 14598-1
Produto de software sob encomenda	(Custom made software product)	ISO/IEC 14598-1
Program	Programa (de computador)	
Program maintenance manual	Manual de manutenção de programa	
Programa (de computador)	(Program) ou (computer program) Unidade sintática que está em conformidade com as regras de uma linguagem de programação particular, composta de declarações, comandos ou instruções necessárias para executar uma certa função ou tarefa, ou ainda solucionar um problema. (ISO/IEC 2382-1:1993)	ISO/IEC 12119 e NBR 12119
Projeto	(Design)	ISO/IEC 14598-1
Projeto	é o conjunto de resultados obtidos em todo o ciclo de desenvolvimento de um software aplicativo, desde sua concepção até sua implantação e testes, excluindo-se os procedimentos de manutenção	

Purchaser	Comprador	
Q		
Qualidade (1)	<p>(Quality) A totalidade das características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas (NBR ISO 8402/1993)</p> <p>A totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas (ISO 8402 : 1994).</p> <p>Notas:</p> <p>1- Numa situação contratual, ou numa área regulamentada, tal como na área de segurança nuclear, as necessidades são especificadas, enquanto que em outras áreas, convém que necessidades implícitas sejam identificadas e definidas (ISO 8402 : 1994, note 1).</p> <p>2 Na série ISO/IEC 14598 a entidade relevante é um produto de software.</p>	<p>ISO/IEC 9126</p> <p>e NBR 13596</p> <p>ISO/IEC 14598-1</p>
Qualidade (2)	<p>é a totalidade dos desempenhos em função das características de um produto ou serviço que se sustenta em sua possibilidade efetiva para atender às necessidades especificadas ou implícitas.</p> <p>Neste trabalho preferiu-se adaptar esta definição para: QUALIDADE é a totalidade de características de uma entidade ou de uma decisão que lhe confere a capacidade de satisfazer suas necessidades ou requisitos explícitos e implícitos, entendendo-se por entidade, neste trabalho, uma "Organização" (uma instituição de direito público ou privado) ou um "Produto" (tais como: sistemas, programas, artefatos) ou um serviço (tais como: processos ou práticas).</p>	<p>JURAN & GRYNA (1991)</p>
Qualidade de <i>software</i>	<p>(Software quality) A totalidade das características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas.</p>	<p>ISO/IEC 9126</p> <p>e NBR 13596</p>
Qualidade em uso	<p>(Quality in use) O quanto um produto, utilizado por usuários específicos, atende as necessidades desses usuários para que eles atinjam atingindo metas especificadas com eficácia, produtividade e satisfação, num contexto de uso definido.</p>	<p>ISO/IEC 14598-1</p>
Qualidade externa	<p>(External quality) O quanto um produto satisfaz necessidades explícitas e implícitas quando utilizado em condições especificadas.</p>	<p>ISO/IEC 14598-1</p>
Qualidade interna	<p>(Internal quality) A totalidade dos atributos de um produto que determinam sua capacidade para satisfazer satisfazer necessidades explícitas e implícitas quando utilizado em condições especificadas.</p> <p>Notas:</p> <p>1 O termo "qualidade interna", utilizado na ISO/IEC 14598 em contraste ao termo "qualidade externa", possui essencialmente o mesmo significado de "qualidade" na NBR/ISO 8402.</p>	<p>ISO/IEC 14598-1</p>

	2 O termo "atributo" é utilizado com o mesmo significado que o termo "característica" no item 4.21, sendo da mesma forma que na NBR 13596 (ISO/IEC 9126) o termo "característica" é utilizado num sentido mais específico.	
Qualificação	(Qualification) O processo de demonstrar se uma entidade é capaz de atender os requisitos especificados. [Veja NBR ISO 8402/1994, 2.13]	ISO/IEC 12207 e NBR 12207
Qualification	Qualificação	
Qualification requirement	Requisito de qualificação	
Qualification testing	Teste de qualificação	
qualified assessor	An individual who has attained the qualifications for carrying out process assessments, as defined in part 6 of this International Standard.	
Quality	Qualidade	
Quality assurance	Garantia da qualidade	
Quality in use	Qualidade em uso	
Quality model	Modelo de qualidade	
R		
Rastreabilidade	(Traceability)	ISO/IEC 14598-1
Rating	Pontuação	
Rating Level	Níveis de Pontuação	
Rating level	Nível de pontuação	
Ratio Scale	Escala Proporcional (ratio scale)	
Recipient	Destinatários	
Recoverability	Recuperabilidade	
Recuperabilidade	(Recoverability) Atributos do software que evidenciam sua capacidade de restabelecer seu nível de desempenho e recuperar os dados diretamente afetados, em caso de falha, e o tempo e esforço necessários para tal.	ISO/IEC 9126 e NBR 13596
Relatório de testes	(Test report) Documento que apresenta resultados de testes e outras informações relevantes para um teste. (ISO/ IEC Guide 2:1991)	ISO/IEC 12119 e NBR 12119
Release	Liberação (release)	
Reliability	Confiabilidade	
Replaceability	Capacidade para substituir	
Request for proposal (tender)	Pedido de proposta	
Requirements document	Documento de requisitos	
Requisito de qualificação	(Qualification requirement) Um conjunto de critérios ou de condições que, quando atendido, qualifica um produto de software quanto à conformidade às suas especificações e quanto à sua utilização no seu ambiente alvo.	ISO/IEC 12207 e NBR 12207
Resource behaviour	Comportamento em relação aos recursos	

Retirement	Descontinuação	
Reuse	Reutilizar	
Reutilizar	(Reuse)	ISO/IEC 14598-1
Rotina utilitária, programa utilitário	(Utility routine, utility program) Rotina (um programa computacional) que fornece serviços gerais, freqüentemente necessários. Exemplos: uma rotina de entrada, um programa de diagnóstico, um programa trace, um programa sort. (ISO/IEC 2382-7:1989)	ISO/IEC 12119 e NBR 12119
S		
SAD	(DSS) (Decision Support System) Sistema de Apoio à Decisão	
Scale	Escala	
Seção	(Clause)	ISO/IEC 14598-1
Security(1)	Segurança	
Security(2)	Segurança de acesso	
Segurança	(Security(1)) A proteção de informações e dados de modo que pessoas ou sistemas não autorizados não possam lê-los ou modificá-los e que pessoas ou sistemas autorizados não tenham acesso negado a eles.	ISO/IEC 12207 e NBR 12207
Segurança de acesso	(Security(2)) Atributos do software que evidenciam sua capacidade de evitar o acesso não autorizado, acidental ou deliberado a programas e dados.	ISO/IEC 9126 e NBR 13596
SEI	(Software Engineering Institute)	
Série ISO/IEC 14598	(ISO/IEC 14598)	ISO/IEC 14598-1
Serviço de software	(Software service) Execução de atividades, trabalho ou obrigações relacionados ao produto de software tais como seu desenvolvimento, manutenção e operação.	ISO/IEC 12207 e NBR 12207
Shall	Deve	
Should	Convém que	
SIGPLAN	ACM Special Interest Group for Programming Languages	
Sintetizados	(Summarised) Substitui resumos	ISO/IEC 14598-1
Sistema	(System) Um conjunto integrado que consiste de um ou mais processos, hardware, software, recursos e pessoas, capaz de satisfazer uma necessidade ou objetivo definido.	ISO/IEC 12207, NBR 12207 e ISO/IEC 14598-1
Sistema computacional	(Computer system)	ISO/IEC 14598-1
Sistema em operação	(Working system)	ISO/IEC 14598-1
Software	(Software) Programas, procedimentos, regras e qualquer documentação associada, pertinente à operação de um sistema computacional. O conjunto completo ou apenas uma parte, dos programas, procedimentos, regras e documentação associada de um sistema de processamento de informação. (ISO/IEC 2382-1:1993)	ISO/IEC 9126, NBR 13596,

	Nota: Software é uma criação intelectual que independe do meio no qual é armazenado.	ISO/IEC 12119 e NBR 12119 ISO/IEC 14598-1
Software Aplicativo	(Application Software) um sistema aplicado a um negócio ou ramo de atividade	
Software de prateleira	(Off-the-shelf software)	ISO/IEC 14598-1
Software de sistema	(System software) Software independente da aplicação que suporta a execução de um software aplicativo. (ISO/IEC 2382-20:1990)	ISO/IEC 12119 e NBR 12119
Software package	Pacote de software	
Software para consumidor final	(Consumer Software)	ISO/IEC 14598-1
Software product	Produto de software	
Software quality	Qualidade de software	
Software quality assessment criteria	Critério de julgamento de qualidade de software	
Software quality characteristics	Características de qualidade de software	
Software quality metric	Métrica de qualidade	
Software service	Serviço de software	
Software unit	Unidade de software	
SPI	(Software Process Improvement) - melhoria dos processos de software. Forma para que se tenha a garantia da qualidade na organização.	
SPICE	(Software Process Improvement and Capability dEtermination) - modelo de qualidade de software criado pela ISO (ISO/IEC TR 15504) com o objetivo de desenvolver um padrão internacional de avaliação de processos de desenvolvimento de software.	
sponsor (process capability determination)	The organization, part of an organization or person initiating a process capability determination.	
SPU	(Software Producing Unit) - Unidade Produtora de Software. Setor, dentro da organização, responsável pelo desenvolvimento de software.	
Stability	Estabilidade	
standard process	The operational definition of the basic process that guides the establishment of a common process in an organization. It describes the fundamental process elements that is expected to be incorporate into any defined process. It also describes the relationships (e.g., ordering and interfaces) between these process elements. (See defined process.)	
Standardisation	Normatizar	
Stated	Explicitada	

Statement of work	Descrição de tarefas	
Step	Etapa	
Step	Passo	
Subseção	(Clause)	ISO/IEC 14598-1
Suitability	Adequação	
Summarised	Sintetizados	
Supplier	Fornecedor	
SW-CMM	Capability Maturity Model for Software	
System	Sistema	
System maintenance	Manutenção de sistema	
System software	Software de sistema	
T		
Tarefa	(Work task) Resultado esperado num contexto de trabalho. (ISO 6385:1981)	ISO/IEC 12119 e NBR 12119
target capability	That process capability which the process capability determination sponsor judges will represent an acceptable process risk to the successful implementation of the specified requirement.	
Test	Teste	
Test case	Guia de teste (test case)	
Test coverage	Cobertura de teste	
Test data	Dados de teste	
Test method	Método de teste	
Test plan, system test and evaluation	Plano de teste ou plano de avaliação e teste de sistema	
Test report	Relatório de testes	
Testabilidade	(Testability) Atributos do software que evidenciam o esforço necessário para validar o software modificado. A extensão em que um teste objetivo e factível pode ser projetado para determinar se um requisito é atendido.	ISO/IEC 9126 e NBR 13596 ISO/IEC 12207 e NBR 12207
Testability	Testabilidade	
Teste	(Test) Operação técnica que consiste na constatação de uma ou mais características de um dado produto, processo ou serviço, de acordo com um procedimento especificado. (ISO/IEC Guide 2:1991)	ISO/IEC 12119 e NBR 12119
Teste de qualificação	(Qualification testing) Teste, conduzido pelo desenvolvedor e testemunhado pelo adquirente (quando apropriado), para demonstrar que o produto de software atende as suas especificações e está pronto para utilização no seu ambiente alvo.	ISO/IEC 12207 e NBR 12207
Third-party evaluation	Avaliação de terceira-parte	
Time behaviour	Comportamento em relação ao tempo	

Tolerância a falhas	(Fault tolerance) Atributos do software que evidenciam sua capacidade em manter um nível de desempenho especificado nos casos de falhas no software ou de violação nas interfaces especificadas.	ISO/IEC 9126 e NBR 13596
Top Three levels	Três níveis superiores	
Traceability	Rastreabilidade	
Três níveis superiores	(Top Three levels)	ISO/IEC 14598-1
U		
UML	(Unified Modeling Language) Um conjunto de técnicas de diagramação.	
Understandability	Inteligibilidade	
Unidade de software	(Software unit) Uma parte de código compilável separadamente.	ISO/IEC 12207 e NBR 12207
Unidade funcional	(Functional unit) Entidade de hardware, ou de software ou de ambos, capaz de executar um propósito específico. (ISO 2382-1:1993)	ISO/IEC 12119 e NBR 12119
Usabilidade	(Usability) Conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários.	ISO/IEC 9126, NBR 13596, ISO/IEC 12119 e NBR 12119
Usability	Usabilidade	
User	Usuário	
User documentation	Documentação de usuário	
User interface	Interface de usuário	
User's perspective	Perspectiva do usuário	
Using	Utilizando	
Usuário	(User) Um indivíduo que usa o produto de software para executar uma função específica. Nota: Usuários podem ser operadores, pessoas que recebem o resultado do software, ou desenvolvedores e mantenedores equipes de desenvolvimento e manutenção de software. Um indivíduo ou organização que utiliza um sistema em operação para executar uma função específica. Nota: O usuário pode executar outros papéis, tais como adquirente, desenvolvedor ou mantenedor.	ISO/IEC 14598-1 ISO/IEC 12207 e NBR 12207
Utility routine, utility program	Rotina utilitária, programa utilitário	
Utilizando	(Using)	ISO/IEC 14598-1
V		

Validação	(Validation) Confirmação, por exame e fornecimento de evidência objetiva, de que os requisitos específicos, para um determinado uso pretendido, são atendidos. Nota: 1 Nas atividades de projeto e desenvolvimento, a validação se refere ao processo de examinar um produto para determinar sua conformidade com as necessidades do usuário. 2 A validação é feita normalmente no produto final, sob condições de operação definidas, podendo contudo tornar-se necessária em fases anteriores. 3 O termo "validado" é usado para designar o estado após a validação. 4 "Validações múltiplas" podem ser realizadas se existirem diferentes usos pretendidos. [ISO 8402: 1994]	ISO/IEC 14598-1, ISO/IEC 12207 e NBR 12207
Validation	Validação	
Verificação	(Verification) Confirmação, por exame e fornecimento de evidência objetiva, do atendimento aos requisitos especificados. Nota: 1 Nas atividades de projeto e desenvolvimento, a verificação refere-se ao processo de examinar o resultado de dada atividade para determinar a sua conformidade com os requisitos estabelecidos para a mesma atividade. 2 O termo "verificado" é utilizado para designar o estado após a verificação. [NBR ISO 8402: 1993]	ISO/IEC 14598-1, ISO/IEC 12207 e NBR 12207
Verification	Verificação	
Versão	(Version) Uma instância identificada de um item. Nota: Modificações em uma versão de produto de software, resultando em uma nova versão, requerem uma ação de gerência de configuração.	ISO/IEC 12207 e NBR 12207
Version	Versão	
well-defined process	A process with inputs, entry criteria, tasks, validation, outputs, and exit criteria that are documented, consistent, and complete.	
Work system	Contexto de trabalho	
Work task	Tarefa	
Working system	Sistema em operação	