

Universidade Federal de Santa Catarina
Programa de Pós-graduação em Engenharia de Produção

Fabricio Paulo Lopes

BENCHMARKING PARA EMPRESAS DE SOFTWARE:
DESENVOLVIMENTO E APLICAÇÃO DE UM MODELO DE
REFERÊNCIA

Dissertação de Mestrado

Florianópolis, SC

2008

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Fabricio Paulo Lopes

***BENCHMARKING PARA EMPRESAS DE SOFTWARE:
DESENVOLVIMENTO E APLICAÇÃO DE UM MODELO DE
REFERÊNCIA***

Dissertação apresentada ao programa de Pós-graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de mestre em Engenharia de Produção.

Orientadora: Profa. Aline França de Abreu, Ph.D.

Florianópolis, SC

2008

Fabricio Paulo Lopes

***BENCHMARKING PARA EMPRESAS DE SOFTWARE: DESENVOLVIMENTO E
APLICAÇÃO DE UM MODELO DE REFERÊNCIA***

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

Florianópolis, 04 de julho de 2008.

Prof. Antônio Sérgio Coelho, Dr.
Coordenador do PPGEF

Banca Examinadora:

Profª. Aline França de Abreu, Ph.D.
Orientadora

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

Prof. Nelson Casarotto Filho, Dr.
Membro

Prof. Álvaro Guillermo Rojas Lezana, Dr.
Membro

Natalino Uggioni, Msc.
Membro

Aos meus pais, Irene e Adelino, que me deram o que não tiveram, permitindo minha chegada até aqui.

À minha esposa, Karina, que me apoiou e orientou em cada momento, trazendo a motivação necessária para a realização deste trabalho.

Agradecimentos

A Deus, pela proteção e pela vida.

À professora Aline França de Abreu, pela orientação desta pesquisa.

Ao Instituto Euvaldo Lodi de Santa Catarina e seus excelentes profissionais, meus amigos,
por todo apoio e troca de conhecimentos e experiências.

Às empresas que abriram suas portas às aplicações-piloto deste trabalho, pela visão e pelo
reconhecimento e incentivo à pesquisa científica brasileira.

Aos irmãos Elaine e Fernando, ao Ícaro e ao Marcelo, pelo carinho e amizade.

Aos meus amigos, pelas conversas e pelos poucos, porém bons momentos de distração
durante o período de mestrado.

À Universidade Federal de Santa Catarina e seus professores, ao Núcleo de Estudos em
Inovação, Gestão e Tecnologia de Informação – IGTI – e ao Programa de Pós-graduação em
Engenharia de Produção – PPGE.

A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Resumo

A indústria de software tornou-se, em apenas algumas décadas de existência, um setor vital para o mundo. Porém, os problemas pelos quais teve que passar, principalmente relacionados à qualidade de seus produtos e processos, fizeram surgir diversos modelos e normas para a disseminação de melhores práticas de engenharia de software, iniciativas e estudos em busca da qualidade, sempre focados mais nos aspectos técnicos e pouco nos aspectos empresariais. O presente trabalho teve como objetivo o desenvolvimento de um método para realização de *benchmarking* em empresas da indústria de software, consistindo num processo de avaliação por indicadores de práticas e resultados (*performance*) e num modelo de relatório de análise de resultados. O método baseia-se em melhores práticas referenciadas em normas e modelos do setor de software, que tratam sobre engenharia e qualidade de produtos e processos de software, bem como em indicadores de gestão organizacional, de projetos, de inovação e de qualidade, adaptados dos métodos Benchstar e *Benchmarking* Industrial, amplamente utilizados pelo Instituto Euvaldo Lodi de Santa Catarina (IEL/SC) em indústrias tradicionais. O modelo de ferramenta desenvolvido foi aplicado, em escala piloto, em três empresas de software de Santa Catarina, que apresentaram resultados distintos devido a suas diferentes características. No entanto, alguns pontos comuns foram verificados, como as deficiências nas práticas em gestão da inovação e gestão organizacional ou os pontos fortes na *performance* em gestão da qualidade e organizacional. Os resultados permitiram validar o potencial do método para a avaliação e comparação das empresas do setor de software.

Palavras-chave: *Benchmarking*. Gestão de empresas de software. Qualidade e engenharia de software.

Abstract

The software industry has become, in few decades of existence, a vital segment for the world. But the problems for which he had to overcome, mainly related to the quality of their products and processes, have led to many models and standards for the dissemination of best practices of software engineering, several initiatives and studies in search of quality, focused more on technical aspects than on the business one. This work was aimed at developing a method for benchmarking companies in the software industry, consisting of a assess process through practices and performance indicators, and of a model report to results analysis. The method is based on best practices referenced in models and standards of software industry, about engineering and quality of software products and processes as well as indicators of organizational, project, innovation and quality management, adapted from the Benchstar and Benchmarking Industrial methods, widely applied by Euvaldo Lodi Institute of Santa Catarina (IEL/SC) in traditional industries. The developed tool was applied, in pilot scale, in three software companies of Santa Catarina, presenting different results due to their different characteristics. However, some common points have been verified, as the deficits in practice of innovation management and organizational management, as well as the strengths in quality and organizational management performance. The results validate the model as a potential method for evaluating and comparing companies in the software industry.

Keywords: *Benchmarking. Management of software companies. Software engineering and quality.*

Lista de Figuras

Figura 1 – Áreas avaliadas no <i>Benchmarking</i> Industrial (IEL/SC, 2005a)	16
Figura 2 – Conhecimento dos modelos de qualidade de software (BRASIL, 2005a).....	25
Figura 3 – Utilização de normas e modelos de melhoria do processo de software (BRASIL, 2005a).....	25
Figura 4 - Modelo Cascata para desenvolvimento de software (PRESSMAN, 1995).....	33
Figura 5 - Modelo Espiral (adaptado de FERREIRA, 2002)	34
Figura 6 - Dimensões do ciclo de vida RUP (IBM, 2004 apud LUIZ, 2006).	36
Figura 7 – Modelo de ciclo de vida de software - ISO 12207 (adaptado de ABNT, 1998).....	52
Figura 8 – Interação entre processos de gerenciamento de projeto (adaptado de PMI, 2004).64	
Figura 9 – Modelo de avaliação do <i>benchmarking</i> em empresas de software (adaptado de IEL/SC, 2005b).....	79
Figura 10 – Etapas de aplicação do <i>Benchmarking</i> para Empresas de Software	81
Figura 11 - Estrutura dos indicadores desenvolvidos.....	87
Figura 12 - Resultados das aplicações-piloto: índices de práticas e <i>performance</i> por área ...	109

Lista de Tabelas

Tabela 1 – Dez maiores empresas de software do mundo.....	21
Tabela 2 – Características e Sub-características de qualidade de software.....	49
Tabela 3 – Atributos exigidos por nível de capacidade na ISO/IEC 15504.....	56
Tabela 4 – Agenda solicitada para reunião de consenso durante aplicação-piloto	106
Tabela 5 – Resultados das aplicações-piloto: índices de Práticas e <i>Performance</i> por área....	108

Lista de Quadros

Quadro 1 – Normas e modelos de qualidade de software	48
Quadro 2 – Inovação: vantagens e desvantagens das pequenas e médias empresas.	72
Quadro 3 – Áreas de avaliação do modelo e sua origem no Benchstar e <i>Benchmarking</i> Industrial.....	78
Quadro 4 – Indicadores, princípios pesquisados e suas fontes.....	84
Quadro 5 – Organização dos indicadores nas seções do questionário	87
Quadro 6 – Novos indicadores e seções do questionário de <i>benchmarking</i>	89

Sumário

1	Introdução.....	1
1.1	Problema de Pesquisa	1
1.2	Objetivos.....	3
1.2.1	Objetivo Geral	3
1.2.2	Objetivos Específicos	3
1.3	Justificativa e Relevância da Pesquisa.....	3
1.4	Metodologia de Pesquisa	6
1.5	Estrutura do Trabalho	7
2	<i>Benchmarking</i> de Gestão Empresarial.....	10
2.1	O que é <i>Benchmarking</i>	10
2.1.1	Origem e Conceitos	10
2.1.2	Tipos de <i>Benchmarking</i>	12
2.1.3	Fases do Processo	12
2.1.4	Os Benefícios do <i>Benchmarking</i>	13
2.2	Modelos de <i>Benchmarking</i> do IEL/SC	14
2.2.1	O <i>Benchmarking</i> Industrial	15
2.2.2	O Benchstar	17
2.2.3	O <i>Benchmarking</i> para a Indústria de Software	18
3	Indústria de Software: Caracterização e Gestão	20
3.1	A Indústria de Software no Mundo.....	20
3.2	A Indústria de Software no Brasil.....	23
3.3	Gestão em empresas de Software	26
3.4	Engenharia de Software	29
3.5	Processo de Desenvolvimento de Software	31
3.5.1	Modelo Cascata ou Clássico	32
3.5.2	Modelo de Prototipação ou Evolutivo	33
3.5.3	Modelo Espiral.....	34

3.5.4	<i>Rational Unified Process (RUP)</i>	35
3.5.5	<i>Enterprise Unified Process (EUP)</i>	39
3.6	Técnicas para Desenvolvimento de Software	41
3.6.1	Análise Estruturada.....	41
3.6.2	Engenharia de Software Auxiliada por Computador	43
3.7	Qualidade de Software.....	44
3.7.1	Engenharia de Requisitos.....	45
3.7.2	Normas e Modelos de Qualidade de Software.....	46
3.7.2.1	ISO/IEC 9126.....	49
3.7.2.2	ISO/IEC 25051 (NBR ISO/IEC 12119).....	50
3.7.2.3	ISO/IEC 12207.....	51
3.7.2.4	CMMI-SW	53
3.7.2.5	ISO/IEC 15504.....	55
3.7.3	Métricas de software.....	57
3.7.4	Gerenciamento do Nível de Serviço	59
3.8	Gerenciamento de Projetos	61
3.8.1	O Ciclo de Vida e Processos dos Projetos	63
3.8.2	Áreas de Conhecimento em Gerenciamento de Projetos	65
3.8.3	O Gerenciamento de Projetos de Software	66
3.9	Inovação e Competitividade no Setor de Software.....	68
3.9.1	Inovação.....	68
3.9.2	Aptidões Estratégicas para a Inovação.....	69
3.9.3	Inovação e Competitividade em PMEs Tradicionais	71
3.9.4	A Inovação em PMEs de Base Tecnológica	73
3.9.5	Inovação em Empresas de Software	73
4	Método de <i>Benchmarking</i> da Indústria de Software	77
4.1	Modelo de avaliação	77
4.2	Processo de Aplicação	80
4.3	Questionário e Indicadores	81

4.3.1	Novos Indicadores	88
4.3.1.1	Engenharia de Software	89
4.3.1.2	Qualidade de Software	92
4.3.1.3	Gerenciamento de Projetos	93
4.3.2	Adequação dos Indicadores Existentes	97
4.3.2.1	Fábrica e Equipamento.....	97
4.3.2.2	Tempos de Ciclo	97
4.3.2.3	Fornecedores	98
4.3.2.4	Gestão Organizacional	99
4.3.2.5	Gestão da Qualidade	99
4.3.2.6	Avaliação de Desempenho da Empresa	100
4.3.2.7	Desenvolvimento de Novos Produtos	101
5	Aplicação-Piloto e Verificação do Método	105
5.1	Aplicações-Piloto.....	105
5.1.1	Caracterização das Empresas	105
5.1.2	Processo de Aplicação	105
5.1.3	Análise de Resultados	107
5.1.4	Conclusões dos Resultados das Aplicações	114
5.2	Opinião das Empresas.....	115
5.3	Contribuições	117
6	Conclusões e Recomendações	121
6.1	Conclusões	121
6.2	Limitações e Recomendações	122
	Referências Bibliográficas.....	124
	Apêndices	134
	Apêndice 1 – Tabela de indicadores por área: resultado da aplicação-piloto	135
	Apêndice 2 – Questionário de <i>benchmarking</i> desenvolvido.....	137

1 Introdução

1.1 *Problema de Pesquisa*

A atividade de desenvolvimento de software é caracterizada por um passado ainda muito recente, que nasceu juntamente com o surgimento dos primeiros computadores na década de 40. É dependente, dessa forma, da indústria da computação e, indiretamente, da indústria eletrônica, que permitiu o desenvolvimento de máquinas com poder de armazenamento e processamento cada vez maiores, fundamentais para o funcionamento e aperfeiçoamento do software.

O negócio da indústria de software, mais próximo do modelo que conhecemos hoje, com empresas especificamente dedicadas ao desenvolvimento de programas que fazem os computadores (e hoje em dia diversos equipamentos eletrônicos) funcionarem, surgiu a partir da década de 60. Segundo Gutierrez e Alexandre (2004), antes desse período os computadores já eram vendidos com o software básico instalado, e o desenvolvimento de aplicações específicas passou a ser feito pelos próprios compradores, facilitados pelo surgimento de linguagens de programação de alto nível como o Fortran, em 1957. O surgimento de “empresas prestadoras de serviços de programação”, como citam as autoras, ocorreu somente a partir de 1960, com empresas que, além de desenvolver e fornecer software, também prestavam serviços, como processamento de dados e gerenciamento de projetos.

Assim, comparada com outras indústrias tradicionais surgidas a partir da revolução industrial no século XVIII, a indústria de software pode ser considerada recente, com cerca de meio século de existência. Por ser jovem, é uma indústria ainda em desenvolvimento e amadurecimento do processo de produção e gestão, após ter passado por momentos críticos em seus poucos anos de vida, durante o período chamado por muitos de “crise do software”.

A crise do software iniciou-se nos anos 70, à medida que a demanda por software crescia, surgindo diversos problemas de produtividade no seu desenvolvimento, relacionados à falta de mão-de-obra qualificada e à racionalização das atividades tal qual um processo produtivo tradicional (PONDÉ, 1993 apud DUARTE, 2003). Embora a existência da crise do software entre os anos 70 e 90 seja defendida por muitos autores, sua superação não é consenso: enquanto alguns defendem que a crise acabou (TIGRE, 1987 apud DUARTE, 2003; GLASS, 2000a; GLASS, 2000b), outros defendem que seus problemas relacionados a qualidade e

gerenciamento, ainda persistem (MEJRI et al., 2003; NOGUEIRA, 2003; MACHADO; BURNETT, 2002; FERREIRA, 2002; DOLCI; BECKER, 1999).

Pressman (1995) chega a questionar o termo “crise de software”, já que uma crise remete-se a algo momentâneo, a um evento decisivo, enquanto que para o autor a característica duradoura dos problemas enfrentados pelo setor de software remete mais a uma “aflição crônica”.

A grande complexidade envolvida no processo de desenvolvimento de softwares contribui para a dificuldade em termos de resultados satisfatórios, tanto na gestão dos processos dessas empresas, quanto nas próprias características intrínsecas aos produtos. Esta crise ou aflição crônica do software nada mais é do que a dificuldade em manter a qualidade de produtos e processos das empresas de software, o que tem demandado investimentos na definição de modelos por parte de diferentes instituições e empresas ao redor do mundo, como a própria *International Organization for Standardization* (ISO), a *International Electrotechnical Commission* (IEC), o *Software Engineering Institute* (SEI), a IBM ou a Microsoft, entre diversas outras.

Como consequência, uma diversidade de modelos e normas de referência de qualidade passou a surgir e a coexistir, alguns de forma mais integrada e outros nem tanto, resultando numa diversidade de abordagens e modelos de diversas instituições. Tais modelos e normas desenvolvidos apresentam como foco principal de estudo as questões técnicas relacionadas à qualidade e engenharia de produtos e processos de software, sendo que não existe um modelo capaz de permitir um processo de *benchmarking* de práticas investidas e resultados obtidos na gestão em empresas de software.

Mejri et al. (2003) citam que, apesar de sua considerável contribuição, a grande variedade de metodologias e técnicas têm encontrado dificuldades em serem amplamente utilizadas. Kaushik (2008), por sua vez, defende que o conjunto de modelos disponíveis é grande e confuso, exigindo ajuda para as empresas na determinação das normas e modelos mais benéficas.

Além da diversidade de modelos de diferentes fontes, os mesmos são focados em escopos específicos. Alguns focam na qualidade dos produtos de software, outros na qualidade do processo de software, ou na gestão da qualidade, etc., não havendo um modelo específico para a avaliação de um negócio de software como um todo, de forma integrada e com visão geral, não apenas técnica. A existência de um modelo abrangente de *benchmarking* para o setor, que avaliasse práticas de referência e seus resultados de forma abrangente nas diversas áreas de

gestão de uma empresa de software, serviria como fonte de orientação e comparação do nível de preparo e competitividade das empresas dentro do setor de software, potencializando o levantamento e disseminação de informações importantes de forma rápida e eficiente.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver um método para *benchmarking* de práticas e *performance* em empresas de software, que as oriente, estratégica e operacionalmente, no seu processo de gestão.

1.2.2 Objetivos Específicos

A fim de alcançar o objetivo geral, são propostos os seguintes objetivos específicos:

- a. Revisar literatura sobre práticas de referência, técnicas e gerenciais, em empresas de software (qualidade, organização, inovação, desenvolvimento de software, gerenciamento de projetos);
- b. Definir modelo de referência de indicadores de práticas e *performance* de gestão em empresas de software;
- c. Desenvolver um método para estudo de *benchmarking* no setor de software, baseando-se no modelo de referência;
- d. Realizar aplicações-piloto do método em empresas de software.

1.3 Justificativa e Relevância da Pesquisa

Conforme mencionado anteriormente, a indústria de software, induzida principalmente pela chamada crise do software ocorrida nas últimas décadas, passou a contar com uma diversidade de modelos de referência de qualidade, com uma diversidade de abordagens e modelos de diferentes instituições, focados em escopos específicos: qualidade dos produtos de software, qualidade do processo de software, ou engenharia de software, não havendo um modelo abrangente para a avaliação e *benchmarking* de um negócio de software de forma integrada.

Nesse cenário, disputam o entendimento e a adoção, por parte das empresas de todo o mundo, instituições como a própria ISO (*International Organization for Standardization*), com normas como a ISO/IEC 9126 e a ISO/IEC 14598 sobre características de qualidade de produtos de software, a ISO/IEC 25051 focada em software de prateleira, ou a ISO/IEC 15504 sobre qualidade do processo de desenvolvimento de software. Outras instituições são o *Software Engineering Institute (SEI)*, do Departamento de Defesa dos EUA, do qual desponta o modelo chamado *Capability Maturity Model (CMM)*, o *European Software Institute (ESI)*, *Office of Government Commerce (OGC)*, a IBM, o *Institute of Electrical and Electronics Engineers (IEEE)* ou a própria Microsoft. A Associação Brasileira de Normas Técnicas (ABNT), por sua vez, segue os passos da ISO, criando versões brasileiras das normas internacionais, como a NBR ISO/IEC 9126, sobre qualidade do produto de software ou a NBR ISO/IEC 15504 sobre qualidade em processos de ciclo de vida de software.

Esses modelos, no entanto, não são suficientes por si só para uma avaliação completa das empresas de software, tampouco permitem a realização de estudos de *benchmarking* entre empresas desse setor com um escopo abrangente de gestão para a excelência.

A norma ISO/IEC 12207, por exemplo, é uma das mais abrangentes no escopo ao estabelecer uma estrutura comum com processos, atividades e tarefas para todo o ciclo de vida de software, incluindo processos fundamentais, como o desenvolvimento de software, processos de apoio, como a garantia da qualidade, e processos organizacionais, como a gerência do ciclo de vida do software. No entanto, a norma estabelece uma estrutura padrão muito focada em práticas do ciclo de vida do software e não é voltada para uma avaliação dessas práticas, tampouco para a avaliação de resultados obtidos. Além disso, a norma apresenta um conteúdo pesado, principalmente para aplicação em empresas de menor porte, sendo um processo que exige tempo, esforço e adaptações.

A norma ISO/IEC 15504 (ver 3.7.2.5) veio com o propósito de permitir a avaliação da maturidade de processos de tecnologia da informação. Embora apresente a vantagem de integrar diferentes padrões existentes, sendo mais flexível, também acaba requerendo considerável esforço, tempo e experiência para sua aplicação prática nas empresas, sendo que, segundo o Ministério da Ciência e Tecnologia (MCT), mais de 75% das empresas de software de pequeno porte não a conhecem (ANACLETO, 2004).

Embora as normas ISO/IEC 12207 e 15504 possam ser aplicadas juntas para uma avaliação de processo em empresas de software, além de não ser um processo tão simples e prático, a avaliação de maturidade de processo não permite a comparação de indicadores de resultados

obtidos (*performance*) com os processos existentes, o que seria alcançado com uma ferramenta de *benchmarking* de práticas e *performance*.

Além disso, o escopo de avaliação dessas normas acaba deixando de fora aspectos importantes da gestão empresarial, como é o caso da gestão da inovação, que é competência estratégica para empresas de software. A inovação é uma necessidade não somente nas empresas maiores, mas principalmente em pequenas e médias empresas (PMEs), que é a maioria dos casos no setor de software brasileiro. Segundo a Associação para Promoção da Excelência do Software Brasileiro – SOFTEX (2002), 82% das empresas de software brasileiras no ano 2000 eram microempresas. Estas devem explorar suas características, como a flexibilidade e agilidade para agir inovadoramente e sobreviver frente à concorrência das grandes empresas, que possuem maiores escalas e disponibilidade de recursos.

O método de *benchmarking*, proposto no presente trabalho, vem permitir uma avaliação da gestão das empresas de software por meio de um processo rápido e simples, que adote indicadores que meçam a adoção de práticas e indicadores que meçam os resultados obtidos pelas empresas, ou seja, sua *performance*, não estando restrita à avaliação dos processos de ciclo de vida de software, mas incluindo indicadores referentes a todas as áreas de gestão: organizacional, de inovação, da qualidade e de projetos.

Apesar da importância do setor de software para o Brasil, poucos estudos completos foram feitos e publicados no país, sendo que estes, quando ocorrem, são focados em qualidade de software ou de processos de desenvolvimento de software, bem como estudos de caracterização do setor, havendo uma carência de uma ferramenta que avalie o desempenho dessas empresas com um viés não somente técnico, mas também de negócio.

Em estudo da SOFTEX (2002), o Brasil surgia como o sétimo mercado de software do mundo, com movimentação estimada em US\$ 7,7 bilhões referente a produtos e serviços de software no ano de 2001, estando bem próximo à China e Índia, com US\$ 7,9 bilhões e US\$ 8,2 bilhões, respectivamente. Atualmente, segundo a Associação Brasileira das Empresas de Software – ABES, o mercado brasileiro de software e serviços ocupou a 13ª posição do *ranking* mundial em 2006, tendo movimentado aproximadamente 9,09 bilhões de dólares (ABES, 2007).

Apesar desse grande mercado, o Brasil exporta pouco e apresenta dificuldades em despontar como um dos *players* mundiais, sendo um dos motivos, segundo Araújo e Meira (2004) apud Araújo (2003), a falta de uma estratégia nacional para o setor. O desenvolvimento da indústria

de software depende também da direção em que as políticas enxergam os ciclos. No Brasil, especialmente, as políticas pensadas e voltadas para o setor enxergaram os ciclos em uma direção: de muita tecnologia (ênfase no ciclo de vida), algo de *marketing* e vendas, e quase nada de negócios (ARAÚJO; MEIRA, 2004).

Na visão de Araújo e Meira (2004) a indústria de software brasileira está muito direcionada a *clusters* de competência da indústria e serviços tradicionais nacionais, como o setor eletromecânico, o setor bancário, o varejo e o governo, não tendo sido desenvolvidas alternativas para soluções internacionais. Porém o setor não tem alternativa a não ser a internacionalização, acompanhando a própria globalização dos setores tradicionais para os quais fornecem.

O método proposto no presente trabalho também apresenta o potencial de ajudar na internacionalização das empresas do setor de software no Brasil, o que tem sido a busca de instituições como o IEL/SC, o Sebrae, o SOFTEX e o próprio governo. A identificação dos pontos fracos e orientação de melhores práticas a serem buscadas, por meio de uma ferramenta de *benchmarking* baseada em modelos mundialmente aceitos, orientam investimentos para a melhoria da competitividade das empresas, inclusive no âmbito internacional.

Desta forma, ressalta-se a importância de um método que defina um modelo de referência para a avaliação da gestão no nível individual das empresas, que não foque somente os aspectos técnicos, mas também os de negócio, orientando as empresas em direção às melhores práticas e resultados que garantam a excelência. Além disso, à medida que as empresas submetam-se à avaliação do método, um valioso banco de dados poderá ser formado, permitindo a condução de processos de *benchmarking*, assim como aumentando a disponibilidade de dados da indústria de software nacional, que podem subsidiar projetos de fomento ao setor.

1.4 Metodologia de Pesquisa

Segundo Silva e Menezes (2001, p. 20), pesquisa “é um conjunto de ações, propostas para encontrar a solução para um problema, que têm por base procedimentos racionais e sistemáticos”.

O conjunto de ações propostas nesta pesquisa, conforme apresentado anteriormente em 1.2, visam à criação de uma solução de *benchmarking* para empresas de software, tendo por base procedimentos de pesquisa bibliográfica, construção de um modelo e, por fim, aplicação-piloto do método desenvolvido em empresas de software.

A presente pesquisa é de natureza aplicada, que segundo Silva e Menezes (2001), objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos, envolvendo verdades e interesses locais, neste caso, referente ao setor de software. A presente pesquisa tem ainda uma abordagem qualitativa e exploratória, o que, segundo Silva e Menezes (2001) apud Gil (1991), visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou a construir hipóteses. Envolve levantamento bibliográfico ou entrevistas com pessoas que tiveram experiências práticas com o problema pesquisado.

A primeira etapa da pesquisa consistiu em um estudo bibliográfico sobre as melhores práticas de gestão e desenvolvimento de software com base nos mais variados modelos, métodos, normas, pesquisas e artigos científicos referentes ao assunto. Essa etapa contribuiu para levantar quais as práticas de referência da indústria de software, que servirão de base para a elaboração de um questionário que constituirá a ferramenta básica para o processo de avaliação e *benchmarking* de empresas do setor.

Na segunda etapa, o funcionamento da ferramenta foi avaliado na prática, por meio de três aplicações-piloto em empresas de desenvolvimento de software. Esta avaliação consistiu em testar o método de aplicação e a ferramenta em si, verificando seu potencial e validando o resultado final de análise das empresas.

1.5 Estrutura do Trabalho

O capítulo 1 traz uma apresentação do tema do presente trabalho, onde após uma breve introdução sobre o problema de pesquisa são apresentados os objetivos geral e específicos que orientam o desenrolar da dissertação. A relevância da pesquisa também é abordada no capítulo, onde são apresentadas as justificativas para o estudo. A metodologia de pesquisa antecipa os passos seguidos com o objetivo de se atingir o que foi proposto, nos métodos e seqüência necessários.

No capítulo 2 temos uma revisão bibliográfica sobre o tema *benchmarking*, apresentando seus principais conceitos, fases, tipos e benefícios para a gestão empresarial. Também são

apresentados, nesse capítulo, dois métodos utilizados pelo Instituto Euvaldo Lodi de Santa Catarina para realização de *benchmarking* em empresas da indústria de transformação tradicional: o *Benchmarking Industrial*, para empresas de médio e grande porte, e o *Benchstar*, para empresas de menor porte. Ambos os métodos foram utilizados como referencial para o desenvolvimento do método para *benchmarking* na indústria de software, escopo deste trabalho.

O capítulo 3 inicia-se com uma apresentação da indústria de software, uma vez que se trata do público-alvo do método a ser desenvolvido e que guarda importantes diferenças da indústria tradicional. Em seguida, destaca-se a seção que introduz o tema da gestão em empresas de software, que será avaliado pelo método de *benchmarking*, objeto deste trabalho.

O capítulo traz ainda uma revisão sobre engenharia de software e qualidade de software, desde os modelos tradicionais de processo de desenvolvimento dos produtos desse setor, até os modelos unificados mais recentes, com suas diversas técnicas de desenvolvimento defendidas mundialmente no setor de software, bem como as diversas normas e modelos, métricas e outras práticas referentes à qualidade de software.

No capítulo 3 é explorado ainda o gerenciamento de projetos e sua importância para o setor de software, com uma revisão sobre os principais conceitos e práticas e sua relação com o processo de desenvolvimento de produtos de software. O capítulo é encerrado com os conceitos de inovação e aptidões estratégicas para o seu alcance, com o objetivo de se apresentar a importância da gestão da inovação para a competitividade das empresas, sendo uma das áreas de avaliação do modelo de *benchmarking* proposto.

O capítulo 4 aborda o desenvolvimento do método de *benchmarking* para a indústria de software, foco do presente trabalho. O capítulo inicia pela apresentação do modelo de avaliação e etapas do processo de *benchmarking* proposto. Em seguida, são apresentados os indicadores do questionário, em duas partes: na primeira são apresentados os indicadores desenvolvidos especificamente para o setor de software, com base nas pesquisas do presente trabalho; na segunda parte são apresentadas as adequações efetuadas nos indicadores do *Benchmarking Industrial* e *Benchstar*, adaptados à realidade da indústria de software.

No capítulo 5 é apresentada a etapa de verificação do método de *benchmarking* pela aplicação-piloto em três empresas de software. São exploradas neste capítulo as aplicações em si, descrevendo as empresas, o processo, a análise de resultados e o *feedback* dos respondentes, bem como as contribuições dessas aplicações-piloto para a consolidação da

versão final do questionário e método de aplicação do *benchmarking* para empresas de software.

Por fim, o capítulo 6 traz as conclusões de todo o trabalho realizado, abordando sua validade, viabilidade, contribuições e limitações, sendo o capítulo finalizado com as devidas recomendações para trabalhos de pesquisa e estudos futuros.

2 Benchmarking de Gestão Empresarial

Expõem-se, a seguir, os conceitos por trás de um processo de *benchmarking*, com os quais se pretende demonstrar a importância e o potencial que uma ferramenta para *benchmarking* de indicadores de gestão no setor de software pode apresentar.

A conceituação é seguida pela apresentação de dois métodos constantes no portfólio do IEL/SC, conforme citado anteriormente: o *Benchmarking* Industrial e o Benchstar. Esses métodos são a base para o desenvolvimento de um modelo específico para empresas de software, proposta do presente trabalho.

2.1 O que é Benchmarking

2.1.1 Origem e Conceitos

No início do século XX, Henry Ford teve uma idéia que revolucionou a manufatura moderna e a história automotiva mundial. Alguns meses após observar como os empregados cortavam a carne de carcaças penduradas em ganchos atados a uma correia aérea, durante visita a uma casa de abate de Chicago, Ford criou a primeira linha de montagem de automóveis do mundo (IEL/SC, 2005a). Embora não soubesse, a prática de Ford poderia hoje ser chamada de *benchmarking*, uma vez que adaptou uma melhor prática observada externamente, independente de ser originária de um setor diferente, resultando em desempenho significativamente superior para seu negócio.

Após a Segunda Guerra Mundial, a indústria japonesa passou a utilizar a prática de observar as características, métodos e produtos da indústria americana com o objetivo de alavancar seu próprio desempenho, abalado pelo resultado da guerra. Aliada à sua cultura *Dantotsu*, de “lutar para se tornar o melhor dos melhores”, essa prática japonesa resultou no crescimento de sua indústria, no surgimento de práticas como a Produção Puxada e no destaque mundial do Japão entre os países industrializados (AZEVEDO, 2001; IEL/SC, 2005a).

O primeiro relato oficial do uso sistematizado de *benchmarking* como ferramenta gerencial data do final da década de 1970, embora a prática conte com exemplos anteriores, como os da experiência pós-guerra da indústria japonesa. Através da disseminação do estudo realizado

pela Xerox Corporation, que buscou nos anos 70 conhecer as práticas empresariais japonesas, o termo *benchmarking* difundiu-se pelo mundo.

A experiência da Xerox ocorreu quando as empresas japonesas estavam iniciando suas vendas de máquinas fotocopadoras no mercado norte-americano com preços inferiores e variedade de modelos superior às empresas locais. O cenário desafiador preocupava importantes executivos americanos, entre eles os da Xerox, que passaram a ter interesse em conhecer o sistema japonês de produção e como alcançavam tal *performance* competitiva (IEL/SC, 2005a).

Os trabalhos iniciais da Xerox foram focados no conhecimento do produto dos concorrentes a partir de engenharia reversa, pela qual se desmontava peça a peça o produto, constituindo o que se chama de *benchmarking* de produto. Em continuidade a essa prática, a Xerox passou a investigar também os processos produtivos e gerenciais de outras empresas, tornando a prática de *benchmarking* algo muito mais amplo e proveitoso.

A partir da experiência da Xerox, seu executivo David Kearns, definiu o *benchmarking* como um “processo contínuo de medição de produtos, serviços e práticas em relação aos mais fortes concorrentes, ou às empresas reconhecidas como líderes em suas indústrias” (CAMP, 1998, p. 8). Segundo Seibel (2004), *benchmarking* é o processo pelo qual ocorre a comparação com padrões de referência chamados *benchmarks*, termo este originado da área de agrimensura quando da definição de pontos de referência para medições, posteriormente trazido para o ambiente de negócios.

Ou seja, *benchmarking* pode ser definido como um processo de busca e comparação de referenciais, com o qual se pretende mudar as práticas e produtos de uma empresa de maneira estruturada e assim obter um melhor desempenho, ganhando vantagem competitiva. Uma das principais vantagens do *benchmarking* é o estabelecimento de metas baseadas em práticas externas das melhores empresas, e não apenas na melhoria a partir de níveis internos e tendências com base no passado. Quando se tem por base somente os níveis internos, pode-se incorrer no erro de se estar defasado em relação aos resultados do mercado.

É importante atentar para o fato de que o processo de *benchmarking* não fica restrito a um processo de cópia. Ou seja, *benchmarking* não é cópia e sua condução pressupõe a adaptação de melhores práticas observadas (sejam elas de concorrentes diretos ou não) para a realidade da empresa que está realizando o *benchmarking*. Utilizando o conceito de Camp (1998), é um exercício de correção de rota, e não de seguir atrás dos concorrentes. A correção e redefinição

de rota pressupõem inovações para surpreender e superar a concorrência. Massa e Testa (2004) enfatizam, por exemplo, que o *benchmarking* não é um processo de imitação, pois busca práticas e *performances* através de comparação fora dos limites da empresa, permitindo a aquisição de conhecimento tácito e explícito que, aliados aos conhecimentos existentes na empresa, promovem novos conhecimentos que levam a inovações.

2.1.2 Tipos de *Benchmarking*

O processo de *benchmarking* pode ser classificado, quanto à abordagem, em quatro tipos principais (BENDELL et al., 1998 apud DEY, 2002; ZAPELINI, 2002):

1. *Benchmarking* interno, quando realizado entre diferentes departamentos dentro de uma mesma empresa;
2. *Benchmarking* competitivo, quando diferentes empresas de um mesmo setor trocam informações, de maneira acordada e estruturada, sendo um dos mais difíceis pela concorrência direta existente entre as empresas;
3. *Benchmarking* funcional, onde empresas de diferentes setores podem comparar processos funcionais específicos, como por exemplo, estocagem; e
4. *Benchmarking* genérico, que envolve a comparação entre diferentes processos de negócio em diferentes indústrias, conferindo oportunidades para grandes inovações e melhorias.

Spendolini (1993) apud Cardoso e Souza (1999), entretanto, resume os tipos de *benchmarking* a apenas três, uma vez que considera os tipos funcional e genérico como maneiras diferentes de referir-se à mesma abordagem.

2.1.3 Fases do Processo

Segundo Camp (1998) apud Zapelini (2002), um processo típico de *benchmarking* pode ser dividido em cinco fases distintas, cada qual com seus passos:

1. Planejamento: consiste no estudo prévio e estruturação do processo, para que o mesmo atinja seus objetivos da melhor forma possível. Seus passos consistem na identificação do objeto que será alvo do processo, na identificação das organizações passíveis de comparação e na determinação do método de coleta de dados e a própria coleta em si;
2. Análise: é a comparação dos dados coletados, procurando evidenciar os pontos fortes e fracos da organização frente às organizações pesquisadas. Seus passos consistem na

determinação da lacuna ou *gap* de diferença entre as empresas comparadas, bem como na projeção de futuros níveis de desempenho.

3. Integração: é o planejamento da incorporação das melhores práticas ou produtos verificados no processo de *benchmarking*, contribuindo para a melhoria interna na empresa. Tem como passos a comunicação e obtenção de aceitação dos marcos de referência e o estabelecimento de metas funcionais para sua incorporação.

4. Ação: trata-se da incorporação e posterior acompanhamento das melhores práticas planejadas na fase anterior, sendo implementada pelos passos de desenvolvimento do plano de ação, sua implementação, monitoramento de resultados e recalibração dos marcos de referência externos.

5. Maturidade: é a obtenção de resultados competitivos a partir da implementação das ações. Para esta fase ser alcançada, a empresa deve incorporar o processo de *benchmarking* de forma sistemática no seu processo de gestão.

2.1.4 Os Benefícios do *Benchmarking*

A seguir, resumem-se os principais benefícios que o processo de *benchmarking* pode trazer para as empresas, a partir das citações de Camp (1998) e Balm (1995) apud Zapelini (2002):

- Permite que melhores práticas de qualquer empresa sejam incorporadas;
- Motiva e estimula os profissionais envolvidos, uma vez que exige criatividade na implementação das descobertas;
- Inibe a resistência às mudanças, que é menor quando as novas idéias vêm de bons exemplos externos, aumentando a credibilidade do processo de melhoria;
- Identifica avanços tecnológicos antes impensados em sua área de atuação;
- Amplia as experiências dos indivíduos, trazendo conhecimento para a organização;
- Melhora os indicadores adotados pelas empresas;
- Permite estabelecer metas confiáveis e eficazes;
- Permite identificar pontos fortes e fracos.

2.2 Modelos de Benchmarking do IEL/SC

O Instituto Euvaldo Lodi de Santa Catarina (IEL/SC), instituição pertencente ao Sistema Federação das Indústrias de Santa Catarina, apresenta entre suas competências a experiência e métodos para a condução de processos de *benchmarking* na indústria brasileira, trabalhando no papel de agente neutro com a função de intermediar a comparação da gestão de empresas com as melhores de seu setor de interesse, sejam nacionais ou internacionais.

Tal experiência surgiu com um projeto de parceria que criou o método de *benchmarking Made in Brazil*, que posteriormente resultou no desenvolvimento do atual modelo chamado *Benchmarking Industrial*, apresentado com maior detalhe na seção 2.2.1.

O sucesso do IEL/SC como instituição detentora do método e intermediadora em aplicações de *benchmarking* na indústria tradicional de transformação, resultou em diversas parcerias com a Universidade Federal de Santa Catarina (UFSC) com o objetivo de disseminar os benefícios do método para a realidade de diferentes setores, não passíveis de uso do método original do *Benchmarking Industrial*.

Azevedo (2001) desenvolveu em sua dissertação uma ferramenta para *benchmarking* em Instituições de Educação Tecnológica seguindo a linha metodológica do *Benchmarking Made in Brazil*, precursor do *Benchmarking Industrial*.

Seibel (2004) explorou em sua tese o processo e os resultados de aplicação do *Benchmarking Made in Brazil* aplicados em 278 empresas industriais exportadoras de Santa Catarina.

Zapellini (2002) e Santos (2005) seguiram a idéia de utilização do método para o setor de educação, criando modelos de avaliação para Programas de Pós-Graduação e para Instituições de Ensino Superior, respectivamente.

Mazo (2003) adaptou o método do *Benchmarking Industrial*, destinado a empresas de médio e grande porte, para a realidade de micro e pequenas empresas da indústria tradicional, resultando no segundo método de *benchmarking* amplamente aplicado pelo IEL/SC: o Benchstar. O método do Benchstar é apresentado com mais detalhes na seção 2.2.2.

Casarotto (2002), por sua vez, desenvolveu uma ferramenta de *benchmarking* para avaliação de redes de empresas na indústria da construção civil, utilizando como base os modelos do Benchstar, de cadeia de valor e rede de empresas.

O potencial dos métodos do *Benchmarking Industrial* e Benchstar, comprovado por meio das aplicações feitas pelo IEL/SC na indústria brasileira, bem como pelos trabalhos supracitados,

foi fator motivador de sua utilização como modelos de ferramenta e método de *benchmarking* de gestão para a indústria de software. Para o melhor entendimento dos métodos, faz-se uma apresentação do *Benchmarking* Industrial e do Benchstar nas seções a seguir.

2.2.1 O *Benchmarking* Industrial

Para empresas industriais de médio e grande porte, o IEL/SC dispõe do método chamado *Benchmarking* Industrial, amplamente difundido no mundo que avalia e compara indicadores de práticas e *performance* de gestão, dispondo assim de um banco de dados mundial, e difundida no Brasil por meio do Programa Melhores Práticas para Excelência Industrial – PMPEI. O *Benchmarking* Industrial permite avaliar a gestão da empresa como um todo, incluindo indicadores para diversas áreas de gestão da empresa.

O *Benchmarking* Industrial surgiu em 1993 na Inglaterra, quando consultores da *London Business School* (LBS), em cooperação com um grupo de consultoria da IBM, desenvolveram uma ferramenta para avaliar a indústria européia em relação ao padrão denominado Classe Mundial, em aspectos como custo, qualidade, flexibilidade e atendimento ao cliente. Em 1994, a partir do desenvolvimento de um método de *benchmarking*, criou-se o programa denominado *Made in Europe* cujo objetivo era medir o nível de práticas nível classe mundial implantadas e *performance* operacional resultante da adoção destas práticas em empresas industriais européias. A partir desta iniciativa, outros países, como Suíça, Austrália, Bélgica, Dinamarca, França, África do Sul, Suécia e Estados Unidos, se interessaram em participar do programa. Em cada um destes países foram criados programas coordenados por entidades locais, com o objetivo de comparar suas empresas com as líderes mundiais e posicionar a indústria em relação ao padrão classe mundial (IEL/SC, 2005a).

Em 1996, o IEL/SC estabeleceu uma cooperação internacional com o *International Institute for Management Development* (IMD) para a utilização, no Brasil, do método e de seu banco de dados, lançando o projeto *Benchmarking Made in Brazil* no ano de 1997, o qual, após diversas melhorias, inclusão de novas áreas de avaliação e incorporação de novas empresas e países ao banco de dados, tornou-se o método chamado *Benchmarking* Industrial, difundido por todo o Brasil (IEL/SC, 2005a).

O método do *Benchmarking* Industrial baseia-se na pontuação de indicadores de práticas e *performance* por uma equipe composta por colaboradores da empresa que aplica o método e

consultores facilitadores do IEL/SC. O modelo baseia-se na adoção de práticas consideradas de Classe Mundial com o objetivo de se alcançar uma *performance* de excelência, que leve à competitividade internacional das empresas.

Indicadores de práticas representam o conjunto de ferramentas gerenciais e tecnológicas implantadas na empresa, como por exemplo, o nível de automação, sistema da qualidade ou engenharia simultânea. Indicadores de *performance* avaliam os resultados mensuráveis obtidos pela empresa, preferencialmente resultantes da adoção das práticas, como por exemplo a rotatividade dos estoques, satisfação dos clientes ou índice de defeitos (SEIBEL, 2004; MAZO, 2003).

Os indicadores do *Benchmarking* Industrial permitem a avaliação da empresa e comparação com o banco de dados em sete áreas de avaliação de gestão, conforme apresenta a Figura 1 a seguir.

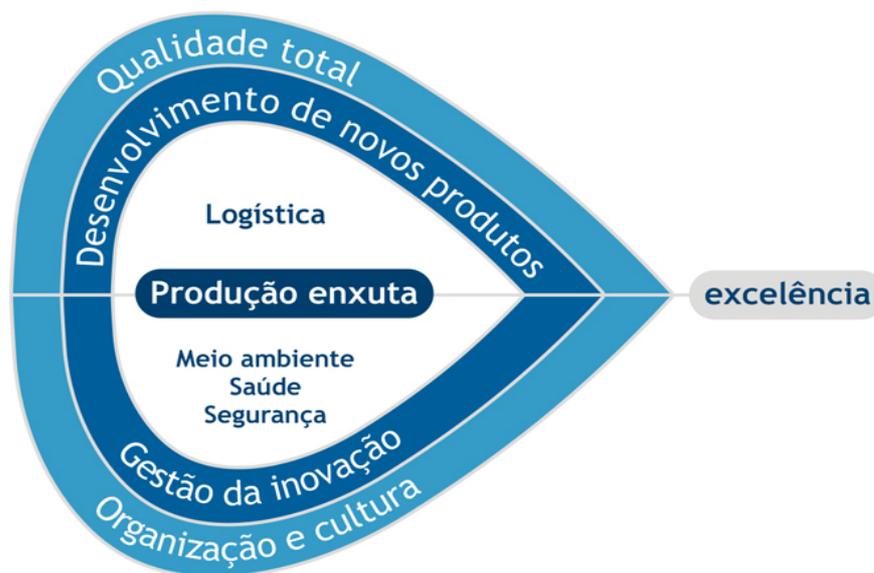


Figura 1 – Áreas avaliadas no *Benchmarking* Industrial (IEL/SC, 2005a)

O método do *Benchmarking* Industrial utiliza como ferramenta de coleta de dados um questionário, que traz os indicadores de práticas e *performance* divididos em seções, de acordo com seu tema, bem como uma seção de orientações para preenchimento e uma seção de caracterização da empresa, que levanta dados para a qualificação das empresa avaliada.

O processo de aplicação do *Benchmarking* Industrial inicia-se pelo envio do questionário pelo IEL/SC à empresa a ser avaliada. Esta, por sua vez, monta uma equipe interna de *benchmarking* composta por colaboradores de diferentes áreas da empresa, com o objetivo de

pontuar cada um dos indicadores. Montada a equipe, a mesma deve responder previamente todo o questionário como forma de preparação para a etapa seguinte, a de reunião de consenso.

Para a reunião de consenso um facilitador do IEL/SC visita a empresa e, juntamente com a equipe de *benchmarking* da mesma, revisa indicador a indicador a avaliação e compreensão dos indicadores respondidos, etapa essa realizada em 1 dia. A pontuação consensada dos indicadores irá gerar o índice percentual de práticas e *performance* em cada uma das sete áreas de avaliação, cujos resultados, comparados com um banco de dados com informações de diversas empresas, são apresentados à empresa avaliada por meio de um relatório de resultados de *benchmarking*.

No relatório de resultados são apresentados os pontos fortes da empresa e as principais oportunidades para melhoria, além da distância a ser percorrida pela mesma para atingir o padrão dos líderes mundiais do setor industrial de interesse.

Um ponto importante do método do *Benchmarking* Industrial diz respeito à confidencialidade dos dados, que garante o sigilo de toda e qualquer informação que possa identificar individualmente as empresas constantes no banco de dados. O sigilo dessas informações garante o interesse da participação das empresas no processo, resultando numa ferramenta com dados atuais e em quantidade representativa da indústria mundial.

O desenvolvimento de um método com essas características, específica para empresas do setor de software, representa uma fonte importante de informações que orientará tanto as empresas, individualmente, na busca da excelência, bem como fornecerá dados importantes para o desenvolvimento do setor no Brasil.

2.2.2 O Benchstar

O Benchstar é o método de *benchmarking* oferecido pelo IEL/SC a empresas de pequeno porte de setores tradicionais da indústria de transformação. O Benchstar surgiu de uma idéia de se adaptar o método do *Benchmarking* Industrial, utilizado desde 1997, para aplicação em empresas de micro e pequeno porte. O método, então aplicado em médias e grandes empresas, foi totalmente revisado e adaptado à realidade de empresas menores, ocorrendo, em meados de 2000, as primeiras aplicações do Benchstar em oito empresas de confecções do consórcio de exportação da cidade de Brusque, em Santa Catarina (MAZO, 2003; IEL/SC, 2005b).

Assim como no *Benchmarking* Industrial, o Benchstar utiliza indicadores de práticas e *performance* na avaliação das empresas, sendo que as áreas avaliadas referem-se à gestão das empresas como um todo, com um foco dado ao sistema de produção industrial. As áreas de avaliação que formam o modelo do Benchstar são: Sistema de produção, Gestão da produção, Gestão da qualidade e Gestão organizacional. O Benchstar, no entanto, não apresenta em seu modelo áreas de avaliação específicas para Gestão da inovação, Desenvolvimento de novos produtos, Logística e Meio ambiente, saúde e segurança, a exemplo do *Benchmarking* Industrial.

O processo de aplicação do Benchstar em uma empresa segue as mesmas etapas verificadas no *Benchmarking* Industrial, tendo apenas como diferença a duração da visita e reunião de consenso, que ocorre apenas em 4 horas de trabalho, devido ao menor tamanho das instalações das pequenas empresas, bem como ao menor número de indicadores do método.

2.2.3 O *Benchmarking* para a Indústria de Software

O IEL/SC, frente à importância do setor de software para a competitividade da indústria catarinense, tem atuado em projetos como a Plataforma de Tecnologia da Informação e Comunicação (PLATIC) e em outros projetos relacionados ao setor de software, em parceria com instituições como a Financiadora de Estudos e Projetos (Finep) e incubadoras de empresas de Santa Catarina. Apesar de suas competências em *benchmarking*, o IEL/SC não dispõe de um método de *benchmarking* para empresas de base tecnológica específico ao setor de software, sendo seu desenvolvimento uma necessidade para o estudo do setor e para a avaliação do impacto de projetos nos quais está envolvido, medindo a evolução da indústria de software catarinense e brasileira.

A carência desse método, reunindo num modelo único os referenciais descritos em diversos modelos e normas do setor de software, foi o ponto de partida do presente trabalho, no qual se buscou um método para *benchmarking* de indicadores de práticas (investimentos) e indicadores de *performance* (resultados), específicos para a indústria de software, porém não limitados ao seu produto ou ao seu processo de desenvolvimento. O método permitirá às empresas, na medida em que forem avaliadas, conduzirem um processo de *benchmarking* primeiramente em relação a padrões de referência existentes e tidos como de excelência, e num segundo momento, um *benchmarking* entre empresas, permitindo a uma determinada

organização verificar seu posicionamento competitivo e diferença em relação a empresas líderes melhor pontuadas.

Um estudo censitário feito sobre 699 empresas de software associadas à Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) em 2001 mostrou que 77,4% delas eram de micro e pequeno porte (SOFTEX, 2001). Outra publicação da SOFTEX afirma que 82% das empresas de software brasileiras, no ano 2000, eram microempresas (SOFTEX, 2002). Diante deste quadro, adotou-se como premissa que a realidade das empresas de software brasileiras tenha características comuns à realidade das pequenas empresas tradicionais: seja pelo porte, disponibilidade de recursos, dificuldades na gestão, entre outras características.

Utilizou-se dessa forma o Benchstar como base principal para o desenvolvimento do método de *benchmarking* do setor de software, verificando quais indicadores se adequavam a ambas as realidades, quais necessitavam ser adaptados e quais eram inexistentes no Benchstar e precisaram ser elaborados com base em modelos de referência do setor de software. Além disso, utilizaram-se indicadores do *Benchmarking* Industrial não abordados no modelo Benchstar, a saber, referentes às áreas de Gestão da Inovação e Desenvolvimento de Novos Produtos, também aplicáveis ao setor de software.

Um processo de *benchmarking* pode se dar por diversas ferramentas ou meios, entre eles questionários, entrevistas, visitas técnicas, pesquisas na internet, congressos, etc. Os três primeiros normalmente resultam em maior potencial para melhorias e inovações entre os participantes, uma vez que envolvem a ciência e autorização do processo por todas as partes e comparações detalhadas frente ao que se deseja. Por outro lado, eles apresentam maior dificuldade quando se trata de *benchmarking* competitivo, pela cultura existente nas empresas de restringir informações próprias que possam significar sua vantagem ou conferir vantagem aos competidores.

Nesse sentido, a adoção do Benchstar e do *Benchmarking* Industrial como modelos para o método desenvolvido, vai ao encontro de dispor do potencial de um *benchmarking* via questionário e entrevista, para o qual a existência do IEL/SC como instituição mediadora e centralizadora dos dados dá credibilidade e viabilidade, contribuindo para a adesão de empresas ao processo, já que a intermediação por instituições isentas pode ser a saída para facilitar a condução de processos de *benchmarking* e potencializar os resultados das mesmas.

3 Indústria de Software: Caracterização e Gestão

3.1 A Indústria de Software no Mundo

De acordo com Araújo e Meira (2004), a evolução da sociedade baseada na indústria, característica do século XX, para a chamada sociedade do conhecimento do século XXI, foi marcada pelo surgimento de uma nova indústria, que se mostrou elemento fundamental para o desenvolvimento econômico e social nessa transição: a indústria de software.

A atividade de desenvolvimento de software surgiu na segunda metade do século XX, juntamente com o desenvolvimento da tecnologia da computação e da indústria eletrônica. O desenvolvimento de software como indústria, no entanto, passou a se verificar na metade da década de 1960, quando o desenvolvimento do software desvinculou-se da fabricação do hardware e quando surgiram as primeiras empresas dedicadas ao desenvolvimento e comercialização de software, principalmente nos Estados Unidos, impulsionadas fortemente por demandas de projetos estratégicos do governo daquele país (GUTIERREZ; ALEXANDRE, 2004).

Desde então, os números do setor mostram sua importância: um estudo feito pela *Organisation for Economic Co-Operation and Development* – OECD (2006) com base em 250 empresas de tecnologia da informação e comunicação (TIC) de todo o mundo, mostrou que o faturamento das empresas dos setores de software e serviços de tecnologia da informação (TI) foi o que apresentou os maiores crescimentos dentre os setores TIC, com aumento de 8% ao ano de 2000 a 2005. No que diz respeito aos investimentos em Pesquisa & Desenvolvimento (P&D), o estudo da OECD mostra que o setor de software foi o que mais investiu frente aos setores de TIC pesquisados, com uma média de quase 15% do faturamento investidos em P&D no ano de 2005.

Gutierrez e Alexandre (2004) citam os números publicados pela revista *Software Magazine* nos quais as receitas das 500 maiores empresas de software chegavam a US\$ 289,7 bilhões em 2002. Ainda segundo as autoras, a grande dinâmica do mercado, devido às constantes evoluções tecnológicas ou da própria indústria do software, favorece o surgimento de novas pequenas empresas, que se estabelecem sem muito investimento inicial. Se por um lado há esse favorecimento, por outro há enormes barreiras devido a características monopolistas e de grande concentração no setor, favorecendo aquisições das empresas pequenas pelas grandes.

Um estudo da OECD (2006) mostra que dentre as dez maiores empresas de software pesquisadas, oito estão sediadas nos Estados Unidos (ver Tabela 1). Em termos de balança comercial dos negócios de software, o estudo mostra que os Estados Unidos aparecem em segundo lugar com US\$ 3,03 e 1,24 bilhões de exportação e importação, respectivamente, em 2004, ficando atrás apenas da Alemanha, com US\$ 3,21 e 1,81 bilhões de exportação e importação, respectivamente (OECD, 2006).

Tabela 1 – Dez maiores empresas de software do mundo

	País	Faturamento 2000	Faturamento 2005	Investimento P&D 2000	Investimento P&D 2005
Microsoft	EUA	22 956	39 788	3 772	6 184
Oracle	EUA	10 231	11 799	1 010	1 481
SAP	Alemanha	7 562	9 563	1 170	1 323
Softbank	Japão	3 927	7 737
CA	EUA	6 094	3 530	1 110	690
Electronic Arts	EUA	1 420	3 129	256	633
Symantec/Veritas	EUA	746	2 583	108	332
Intuit	EUA	1 037	2 038	166	305
Amdocs	EUA	1 118	1 918	75	132
Adobe Systems	EUA	1 226	1 885	240	351

Obs.: valores em milhões de dólares.

Fonte: OECD (2006)

Para complementar o atual quadro mundial dos *players* do setor de software, deve-se citar o papel que recentemente está sendo representado por economias emergentes no cenário dessa indústria, como é o caso da Índia, China e até mesmo do Brasil. Estudo da OECD (2006) mostra que economias como a China, Brasil, África do Sul e Índia investiram 7.940, 3.566, 2.369 e 1.908 milhares de dólares em 2005, respectivamente, no setor.

A Índia se destaca hoje por seu sucesso na exportação de software, sendo que seu bom desempenho é resultado de investimentos iniciados na década de 90, quando o governo, em parceria com a iniciativa privada local, criou incentivos à qualificação de empresas de software inclusive com certificações internacionais e capacitação de mão-de-obra para o setor, sempre objetivando a exportação (SAUR, 2004).

Como resultado desses investimentos, em 2003 o faturamento das empresas indianas de software chegou a 1,4% do total do setor no mundo. Além disso, entre 2002 e 2003, suas receitas cresceram 29%, enquanto o faturamento total das empresas norte-americanas apresentou uma expansão de apenas 4% no mesmo período (GUTIERREZ; ALEXANDRE, 2004).

Devido ao foco na exportação, a maior parte das receitas apresentadas pela Índia são provenientes de clientes estrangeiros, referentes a serviços do chamado *outsourcing*: a terceirização do desenvolvimento de software. Essa estratégia passa a exigir das empresas a profissionalização do processo de desenvolvimento de software e sua certificação de acordo com modelos e normas internacionais de qualidade.

Quando se fala em países emergentes na economia mundial, não se pode deixar de lado a China, que também tem sido destaque no setor de software. Segundo o estudo da OECD (2006), o setor de software movimentou na China em 2005, de acordo com o próprio Ministério da Informação Industrial e a Associação da Indústria de Software Chinesa, cerca de US\$ 49 bilhões. Apesar disso, o software chinês não é focado na exportação como é o da Índia: embora as exportações chinesas tenham crescido 28% ao ano, seu valor é divulgado como de US\$ 3,6 bilhões de exportação em 2005, enquanto o mercado doméstico cresceu em média 56% ao ano de 2003 a 2005, com um volume de faturamento mais de dez vezes maior. Assim, a indústria chinesa de software chama a atenção pelo potencial de seu crescente mercado interno.

Pelo exposto acima, pode-se perceber a importância crescente da indústria de software em todo o mundo, tanto para os países desenvolvidos quanto para aqueles em desenvolvimento. O software caminha junto com a evolução tecnológica e está ligado a muitas das atividades executadas no dia-a-dia do homem, seja no funcionamento de microcomputadores, processamento e armazenamento de dados de grandes empresas e governos, funcionamento de brinquedos, celulares, utensílios domésticos ou de aparelhos cuja confiabilidade do software embutido é vital, como em aeronaves ou equipamentos médicos.

Apesar da evolução do software e da importância que tem adquirido, seu desenvolvimento muitas vezes é um processo difícil e cheio de problemas, fato que levou a cunhar a expressão “Crise do Software”. Brock et al. (2003) apud Nakajima (2004) mostram que pesquisas realizadas em 2002 apontam que apenas 34% dos projetos de desenvolvimento foram finalizados no tempo e custo previstos, com todas as funções e características originalmente especificadas. Outros 51% foram finalizados, porém com custo e tempo acima da proposta

original e os 15% dos projetos de software restantes foram cancelados antes de serem finalizados ou nunca foram implementados.

A autora ainda cita um relatório divulgado em 2002 pelo *National Institute of Science and Technology* (NIST), que mostra que o custo anual dos defeitos de software nos Estados Unidos era da ordem de US\$ 59,5 bilhões (HALLEM; PARK; ENGLER, 2003 apud NAKAJIMA, 2004).

Os números acima apresentados, representando ou não uma verdadeira crise do setor de software, cuja existência foi combatida por alguns e defendida por muitos, alertaram ao mundo sobre a necessidade de se desenvolverem modelos que criassem uma base para a gestão da qualidade no processo de desenvolvimento de software e também no próprio produto de software, assunto que será explorado mais adiante neste capítulo.

3.2 A Indústria de Software no Brasil

Segundo estudo da SOFTEX (2002), a indústria de software brasileira se desenvolveu a partir de uma série de incentivos que surgiram com a abertura do mercado de software em 1992, que culminou de um lado na entrada de empresas internacionais no Brasil e no conseqüente aumento da concorrência, mas que por outro lado resultou na necessidade do desenvolvimento e melhoria da competitividade do setor de software no país.

A indústria de software brasileira, assim como a de países emergentes como Índia e China, tem apresentado nos últimos anos uma evolução muito grande, o que é refletido pelos números do setor. Segundo o Ministério da Ciência e Tecnologia (BRASIL, 2005a), a receita com direitos autorais nas exportações brasileiras de software alcançou em 2004 a soma de US\$ 24,9 milhões, um valor dez vezes maior que as exportações registradas em 1999. O volume total comercializado no mercado externo atingiu US\$ 100 milhões com produtos e serviços de software brasileiros.

No entanto, o volume do mercado externo de software brasileiro ainda é considerado muito tímido, sendo foco do setor ainda o mercado interno, que movimentou, segundo dados do SOFTEX (2002), cerca de US\$ 7,7 bilhões em 2001. Sendo assim, a indústria brasileira de software possui uma grande oportunidade em aproveitar seu desenvolvimento e competitividade adquiridos internamente para lançar-se no mercado internacional, com o

objetivo de tornar-se um grande fornecedor do crescente mercado mundial, como fez, por exemplo, a Índia.

Um estudo apresentado pelo MCT (BRASIL, 2005a) mostrou que a maioria das empresas brasileiras de software possui um porte pequeno ou micro, com 77% das empresas de TI empregando menos de 50 funcionários efetivos em seu quadro, embora análises sobre o faturamento não tenham sido apresentadas no estudo. Esse quadro de funcionários apresenta formação relativamente superior a outros setores tradicionais da indústria, sendo uma característica de empresas de base tecnológica: 17,6% têm graduação em informática, 5,2% pós-graduação *lato sensu*, 2,7% são mestres e 0,4% doutores (BRASIL, 2005a).

Outros dados revelam ainda a pouca maturidade gerencial que permeia o setor de software brasileiro, altamente baseado em competências ainda restritas a questões técnicas: em 2004 apenas 35% das empresas de TI pesquisadas no estudo do MCT (BRASIL, 2005a) elaboravam planos estratégicos e de negócios com revisão periódica sistemática.

Qualidade Brasileira de Software

O conhecimento dos modelos de qualidade de software pelas empresas do país tem evoluído nos últimos anos, o que reflete a exigência mundial crescente com relação ao maior controle sobre o desenvolvimento dos produtos do setor, com vistas a evitar a manifestação de sintomas da chamada crise do software.

O MCT, em seu estudo realizado com as empresas brasileiras de TI (BRASIL, 2005a), verificou o conhecimento acerca dos modelos de qualidade tanto de processo de desenvolvimento de software quanto de produtos de software. Foram pesquisados, por exemplo, os modelos CMM, CMMI, NBR ISO/IEC 12207 e ISO/IEC 15504, relacionados à qualidade do processo de software, e as normas NBR ISO/IEC 9126-1, NBR ISO/IEC 14598 e NBR ISO/IEC 12119 relacionadas a qualidade de produtos de software. O resultado, que pode ser visto na Figura 2, mostra que os modelos existentes já são bem conhecidos entre as empresas, embora sua adoção ainda seja muito restrita (Figura 3).

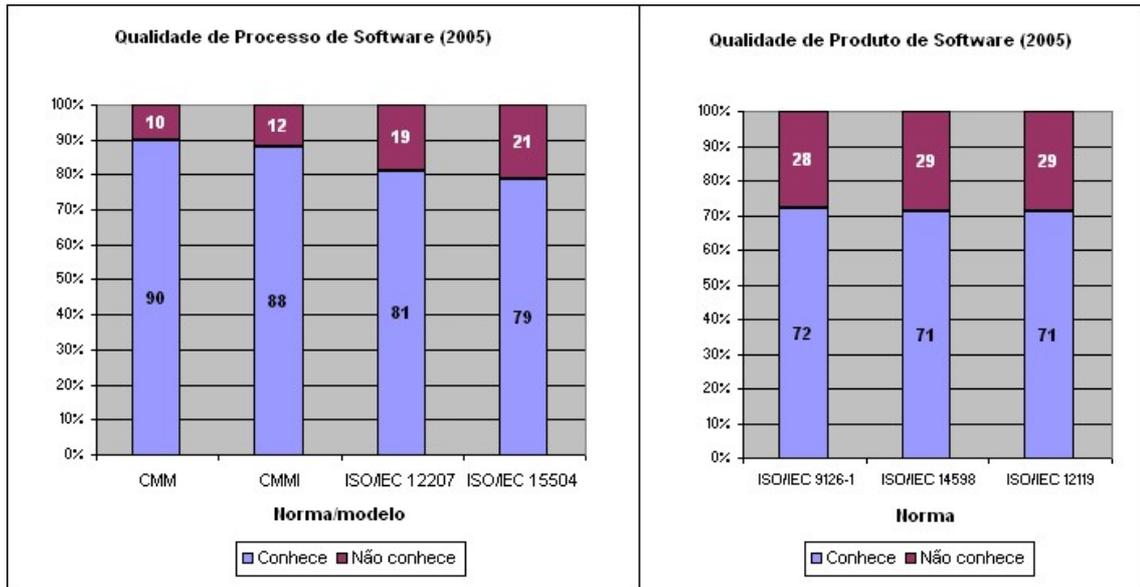


Figura 2 – Conhecimento dos modelos de qualidade de software (BRASIL, 2005a)

A Figura 3 nos mostra que, embora a maioria das empresas pesquisadas conheça os modelos e normas de qualidade do processo de software, 63% das empresas conhecem mas não adotam o modelo¹.

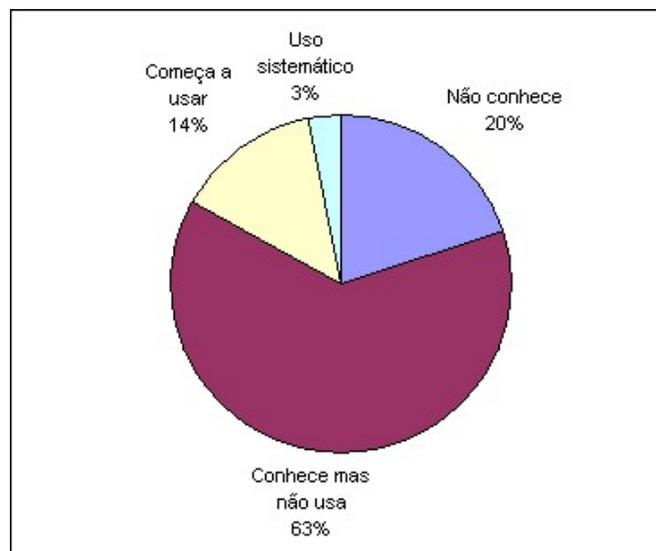


Figura 3 – Utilização de normas e modelos de melhoria do processo de software (BRASIL, 2005a)

Um relatório da Secretaria de Política de Informática (SEPIN) do MCT elaborado em 2006 mostra os números da adoção dos modelos CMM e CMMI no Brasil. Segundo o relatório, até agosto de 2006, obtiveram qualificação CMM 49 empresas de software brasileiras, sendo 40

¹ Os dados referentes à qualidade de produto de software não foram mostrados na pesquisa, mas devem ser ainda menos adotados, a julgar que mais empresas desconhecem as normas em comparação à qualidade de processo, conforme gráficos da Figura 2.

no nível de maturidade 2, oito no nível 3 e uma no nível 4; quanto ao CMMI, apenas 21 empresas apresentaram qualificação oficial, sendo onze no nível de maturidade 2, quatro no nível 3 e seis empresas no nível 5 (BRASIL, 2006).

A julgar pelo número de empresas desenvolvedoras de software no Brasil, que segundo a SOFTEX (2002) no ano de 2000 chegou a 2.398 empresas de todos os portes, os números de qualificadas CMM/CMMI mostram-se ainda muito tímidos, da ordem de 1 a 2% do setor.

O fato da limitada adoção dos modelos e normas de qualidade de software no Brasil, aliado aos números apresentados anteriormente sobre a pouca maturidade gerencial do setor, mostram que a indústria de software brasileira tem ainda um longo caminho a percorrer antes de se inserir fortemente nesse mercado global. O desenvolvimento de uma ferramenta ampla de *benchmarking*, que reúna práticas gerenciais e de qualidade de software, vem ao encontro do interesse dessa indústria nacional, justificando mais uma vez o investimento em um modelo de *benchmarking* para o setor.

3.3 Gestão em empresas de Software

O sucesso competitivo de qualquer empresa inserida num mercado caracterizado pela livre concorrência depende fundamentalmente de suas práticas gerenciais adotadas e dos resultados obtidos com essas práticas. Com as empresas da indústria do software não é diferente: a gestão empresarial é tão fundamental quanto se dispor de um bom produto. Aliás, é nessa questão específica em que muitas empresas de software acabam falhando, ao colocar seus resultados muito atrelados somente às características técnicas de seus produtos, ignorando o aspecto da gestão da empresa como um todo, sem a qual, na verdade, dificilmente se consegue atingir a materialização do conceito de um bom produto.

A dificuldade de sobrevivência de muitas empresas de base tecnológica, dentre elas as de software, vem do fato de terem sido fundadas a partir de uma boa idéia e excelentes conhecimentos acerca de um produto, a partir de pessoas com um perfil técnico excelente, porém sem muitas habilidades gerenciais, resultando em fracos processos de gestão empresarial que podem inclusive afetar a qualidade do processo de desenvolvimento dos produtos e do próprio produto em si.

Além da importância da qualidade de software, para a competitividade de empresas de software é imprescindível a preocupação com questões como a gestão organizacional, estratégias empresariais, gestão da inovação, gerenciamento de serviços e qualidade total.

Kubota (2007) realizou um estudo para mostrar a importância do impacto da gestão no resultado das empresas de software. Segundo o autor, um dos problemas dessa indústria brasileira diz respeito à falta de visão de negócios e de *marketing*, aliada à formação técnica dos empresários. Um dos resultados de seu estudo apontou, por exemplo, que a parceria e reconhecimento de clientes, ao lado da gestão da qualidade de software, contribuem para a rentabilidade das empresas de software pesquisadas.

Ahmed, Capretz e Sheikh (2007) investigam os fatores organizacionais chave para as empresas que trabalham sob o modelo chamado Linha de Produto de Software (*Software Product Line*), citando a importância de aspectos como estrutura organizacional, papéis e responsabilidades, aprendizagem organizacional, gestão de mudanças, cultura e comprometimento, segundo os autores, muito pouco discutidos na literatura de software.

Em seu estudo, Ahmed, Capretz e Sheikh (2007) identificaram que a cultura organizacional com bom relacionamento entre gerentes e empregados contribui para o bom desempenho das empresas de linha de produto de software: se as decisões são tomadas somente centralizadas na administração, tende a ocorrer resistência por parte dos colaboradores; se há uma cultura de abertura, todos dão opinião e sugestões que levam à inovação.

O comprometimento organizacional também é positivamente correlacionado ao desempenho das empresas de linha de produto de software: os gestores devem criar estratégias de promoção e sistemas de recompensa que aumentem o comprometimento dos colaboradores. A aprendizagem organizacional também é fator de sucesso para essas empresas do setor de software, caracterizando-se tanto pela aprendizagem dos empregados quanto da própria organização: seja por meio de treinamentos regulares, seja pela captura de conhecimentos externos referentes aos clientes, concorrentes, ambiente externo e segmentos de mercado (AHMED; CAPRETZ; SHEIKH, 2007).

Em outro estudo, Ahmed e Capretz (2007) afirmam que a orientação para o mercado é fundamental para prover informações sobre as necessidades e requisitos dos clientes, assim como as informações sobre os concorrentes são importantes para novas funcionalidades nos produtos desenvolvidos. Os autores afirmam ainda que o planejamento estratégico é essencial

para a definição dos objetivos da organização e para o alinhamento dos recursos a esses objetivos.

Tais aspectos devem ser avaliados nas empresas de software em geral e não somente para o regime de linhas de produto de software, uma vez que o processo de desenvolvimento de software baseia-se fundamentalmente nas pessoas, na forma como se organizam, sua cultura, seu conhecimento e visão.

A definição de uma estratégia empresarial e a orientação das ações da empresa em sua direção também devem ser preocupação das empresas de software. Definir visão e objetivos de longo prazo, com acompanhamento de metas através de indicadores, deve evitar que as empresas foquem na exploração (em curto prazo) de produtos específicos, por dominarem sua tecnologia (atitude típica das empresas de base tecnológica), esquecendo das novas oportunidades e da necessidade da inteligência competitiva para o correto posicionamento estratégico. A gestão participativa, com todos participando e entendendo a estratégia da empresa, é fundamental para essa indústria baseada em capital intelectual bem qualificado.

Essa idéia é corroborada por Burgelman et al. (2004) ao afirmarem que estratégia é inerentemente uma função da quantidade e qualidade das capacidades de uma empresa. Sem capacidade, a estratégia não tem força e, por outro lado, capacidade sem estratégia não tem rumo. Segundo os autores, a estratégia articula a forma pela qual são exploradas as oportunidades para criação das capacidades de uma empresa.

Uma das estratégias que as empresas de software não podem deixar de lado diz respeito à inovação. A gestão da inovação, embora pareça marginal, principalmente em fabricantes de software customizado, é responsável por soluções criativas, eficientes e mais rápidas para os produtos e clientes de software. Para tal, as empresas devem preocupar-se com estratégias de inovação, inteligência competitiva, gestão de conhecimento, bem como com a gestão organizacional, que permitem a aquisição, geração e difusão de conhecimento e, conseqüentemente, obtenção de soluções.

Não se pode esquecer a importância da competência em gerenciamento de projetos, que desempenha o papel de um planejamento e controle da produção para as empresas de software. De fato, as preocupações tradicionais do gerenciamento de projetos clássico são alinhadas às da qualidade de processo de desenvolvimento de software, haja vista o foco no atendimento de prazos e orçamentos, definição de escopo dos produtos, gestão de mudanças, qualidade do produto, entre outros.

Os fatores-chave para a gestão das empresas apresentados nessa seção foram incorporados à ferramenta de *benchmarking* desenvolvida no presente trabalho, por meio de indicadores como os da seção Gestão Organizacional do questionário. A gestão organizacional é avaliada no modelo de *benchmarking* pelo nível de declaração e compartilhamento da visão, missão e metas; pelo planejamento e análise das metas, feito com a participação e consentimento de todos na empresa; pelo estilo de administração não centralizador e motivador da participação e satisfação dos colaboradores; pelas políticas de incentivo e desenvolvimento de pessoal; e pela orientação para o cliente, conhecimento da concorrência e planejamento de *marketing*. Outros indicadores, em seções como Gestão da Inovação e Gestão de Projetos, também avaliam importantes fatores aqui discutidos, sendo que são apresentados com maior detalhe nas seções 3.8 e 3.9.

3.4 Engenharia de Software

A Engenharia de Software tornou-se uma disciplina essencial para a estruturação que caracterizou o amadurecimento do setor de software mundial. Com o surgimento da indústria do software, bem como, conseqüentemente, dos problemas que caracterizaram a crise do software, houve a necessidade do desenvolvimento dessa área da engenharia, preocupada com o desenvolvimento dos produtos e o atendimento das especificações dos clientes e dos projetos, levando ainda ao surgimento da disciplina chamada Qualidade de Software (DUARTE, 2003; FERREIRA, 2002; PRESSMAN, 2002 apud NOGUEIRA, 2003).

Uma das definições mais claras sobre a engenharia de software talvez tenha sido dada por Maffeo (1992) apud Otero (2002), segundo o qual a engenharia de software é:

Uma área interdisciplinar que engloba vertentes tecnológica e gerencial, visando abordar de modo sistemático (modular) os processos de construção, implantação e manutenção de produtos de software, com qualidade assegurada por construção, segundo cronogramas e custos previamente definidos (MAFFEO, 1992 apud OTERO, 2002, p. 23).

Nessa definição fica evidente a intenção da engenharia de software de cuidar tanto da parte técnica, relacionada a modelos e técnicas de desenvolvimento de produtos de software, quanto da parte gerencial do projeto de desenvolvimento, alocando os recursos da melhor maneira possível para se definir e atingir corretamente os requisitos de produto e projeto. Fica evidente

ainda, na definição do autor, a preocupação com os principais problemas dos projetos de desenvolvimento de software, a saber, o cumprimento de prazos e custos, o que ratifica o surgimento da engenharia de software como necessidade frente à crise do software.

Já a IEEE *Computer Society* define engenharia de software como sendo a aplicação de uma abordagem sistemática, disciplinada e quantificada ao desenvolvimento, operação e manutenção de software, ou seja, a aplicação da engenharia ao software. (IEEE, 2004, p. 1-1). Além disso, como se pode observar no *Software Engineering Body of Knowledge* (SWEBOK), guia de engenharia de software da IEEE, a disciplina de engenharia de software engloba todo conhecimento, ferramentas e métodos necessários a todo o ciclo de vida do software, desde a definição de requisitos até as tarefas de manutenção (IEEE, 2004).

O escopo do SWEBOK, que se refere a engenharia de software como todo conhecimento, ferramentas e métodos necessários a todo o ciclo de vida do software, pode ser complementado com a idéia de Garcia (2003), o qual afirma que a engenharia de software deve estar equipada com métodos e técnicas adequados aos princípios que irão trazer qualidade ao processo e produto de software. Além disso, segundo o autor, o estudo e definição do conjunto adequado de método e técnicas a ser adotado trata-se da metodologia de desenvolvimento de software, também no escopo da engenharia de software.

Para entender perfeitamente o que, na prática, é abordado pela engenharia de software, faz-se necessário a compreensão dos seguintes termos: método, técnicas, ferramentas e metodologia.

Roque (1998), após estudo de diversos autores, verifica que os termos metodologia e método não apresentam um consenso em sua definição. A autora chega à conclusão de que o termo *metodologia* é mais abrangente, significando um conjunto recomendado de filosofias, fases, procedimentos, técnicas, regras, ferramentas, documentação, gerenciamento e treinamento para o desenvolvimento de um sistema (MADDISON apud ROQUE, 1998). Já os *métodos*, segundo Garcia (2003), são linhas gerais que regulam previamente uma série de operações que se deve realizar em vista de um resultado determinado. A partir desse conceito, podem-se traduzir os métodos como sendo procedimentos sistematizados para se atingir um resultado determinado.

Já ferramentas e técnicas são os meios pelos quais se põe em prática um método. Segundo Roque (1998, p. 54), “uma *técnica* é um modo de fazer uma atividade particular no processo de desenvolvimento de sistemas [...] Cada técnica pode envolver o uso de uma ou mais *ferramentas* que representam alguns dos artefatos usados no desenvolvimento de sistemas”.

Assim, sumariza-se a abordagem sistemática da engenharia de software englobando o estudo do processo de desenvolvimento (definindo os *métodos* adequados) e a execução do projeto de software (definindo as *técnicas* e *ferramentas*), caracterizando-se todo o processo pela definição e utilização de uma *metodologia* por uma empresa, no seu processo de desenvolvimento de software, que seja mais adequada a sua realidade.

3.5 Processo de Desenvolvimento de Software

Freiberg (2004) apud Anacleto (2004, p. 17) define o Processo de Software como sendo “uma estrutura para as atividades que são necessárias para a construção de software de alta qualidade”.

Segundo discurso de Kolence para o Comitê Científico da OTAN, uma metodologia de desenvolvimento de software é composta pelo conhecimento e entendimento do que um programa é e pelo conjunto de métodos, procedimentos e técnicas pelo qual é desenvolvido o programa. Assim, torna-se óbvio que as técnicas e os problemas de gerenciamento de software são inter-relacionados por meio das metodologias existentes (NATO, 1969, p. 24).

Baseando-se nas definições expostas acima e na seção anterior, pode-se concluir que o processo de desenvolvimento de software está relacionado à definição de uma metodologia cujas regras gerais ditarão todo o conjunto de atividades, ferramentas e técnicas necessárias à obtenção de um software de qualidade de forma eficiente.

O processo de desenvolvimento de software, ou processo de software, não deve estar focado apenas nas atividades técnicas do desenvolvimento, como definição de requisitos, arquitetura ou testes, mas deve ser mais abrangente e envolver atividades organizacionais e gerenciais como manutenção e suporte ao cliente (ANACLETO, 2004).

Os processos de desenvolvimento de software são, digamos assim, visualizados e melhor gerenciados através dos chamados Modelos de Processo de Desenvolvimento de Software, ou Modelos de Processo de Software, definidos por Acuña et al. (2000) apud Anacleto (2004, p. 17) como “uma representação abstrata da arquitetura, projeto ou definição do processo de software, que descreve [...] a organização dos elementos de um processo”.

Existem hoje diversos modelos de processo de software que contribuem para a estruturação do método de desenvolvimento adotado nas empresas, desde o modelo tradicional em cascata,

que foi o precursor e cuja evolução gerou diversos outros modelos (evolutivo, espiral etc), até modelos mais avançados, que possuem avaliação de qualidade embutida (CMMI, ISO/IEC 12207, entre outros).

Convém ressaltar que não são esgotados todos os modelos e métodos neste trabalho, não sendo explorados, por exemplo, os métodos de desenvolvimento ágil de software (*agile software development*), como o *Extreme Programming* (XP) ou o Scrum. Tais métodos possuem ainda algumas limitações, como por exemplo, a dificuldade de utilização em grandes empresas e projetos (ZANATTA, 2004; SOARES, 2004), a ausência de análise de risco (SOARES, 2004) e a limitada divisão de tarefas e responsabilidades em grandes projetos (VASCO; VITHOFT; ESTANTE, 2005). Os métodos de desenvolvimento ágil utilizam muitas das melhores práticas aqui apresentadas, porém outros aspectos, como a falta de planejamento (foco nos requisitos e necessidades atuais) ou pouca preocupação com a documentação ou definição de um escopo inicial, são ainda muito controversos e alvo de críticas.

3.5.1 Modelo Cascata ou Clássico

O modelo em cascata foi o primeiro modelo que tentou organizar as etapas do processo de desenvolvimento de software de forma a estruturá-lo. Sua grande contribuição foi a de precursor em prever as atividades iniciais de projeto antes da implementação, do código de programação, chamando a atenção para a importância dos estudos e atividades prévias de viabilidade e levantamento de requisitos.

Esse modelo serviu como base para diversos modelos aperfeiçoados que surgiram posteriormente (BONA, 2002), tendo suas fases se tornado etapas básicas para muitos modelos atualmente utilizados. Porém, a principal deficiência e crítica do modelo cascata recaíram sobre sua rigidez, uma vez que se trata de um processo sem realimentações e com necessidades de aprovações de documentação em cada uma das fases, dificultando alterações de requisitos ou outras mudanças necessárias ao longo do processo de desenvolvimento (REZENDE, 1999; SOARES, 2004).

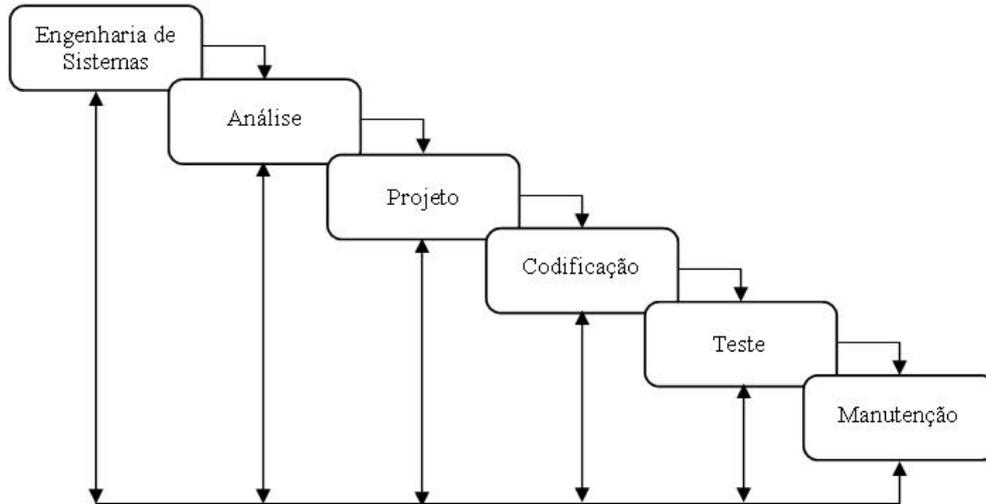


Figura 4 - Modelo Cascata para desenvolvimento de software (PRESSMAN, 1995)

Pressman (1995) apud Ferreira (2002) aponta, como algumas limitações do modelo cascata: a diversidade do processo real, que dificilmente segue uma seqüência rígida como a do modelo; a dificuldade do cliente explicitar todas as necessidades já, e somente, no início do processo; a inexistência de versões preliminares para a validação do cliente, que ocorre somente no final (quando pode ser tarde para a correção de erros); e seu custo elevado.

3.5.2 Modelo de Prototipação ou Evolutivo

A principal evolução em relação ao modelo precursor em cascata está na preocupação com o *feedback* e interação constante com o usuário/cliente ao longo de todo o ciclo de vida do desenvolvimento do software.

Nesse modelo, após uma etapa prévia de levantamento de requisitos, desenvolve-se uma versão piloto do software em curtíssimo tempo. Essa versão inicial do produto retornará ao cliente e servirá como base para o aprofundamento da discussão sobre as especificações e requisitos do software. A partir daí, utiliza-se o modelo cascata para o desenvolvimento de uma segunda versão do software, que quando concluída passa por uma etapa de avaliação e refinamento com a participação do cliente. A liberação do software ocorre apenas após uma aprovação completa, por parte do cliente, de uma das versões desenvolvidas (MELNIKOFF, 1992; OTERO, 2002; BONA, 2002).

Sendo assim, a grande vantagem do modelo evolutivo é este ciclo de realimentação e *feedback* com o cliente, que dispõe de uma base concreta (as versões-protótipo) para avaliar e ajustar o projeto de acordo com suas necessidades.

3.5.3 Modelo Espiral

Boehm, em 1986, sugeriu um modelo evolucionário para o desenvolvimento de software, baseado em uma seqüência de fases que geram versões incrementais (REIS, 2001). O modelo espiral pode ser considerado uma evolução dos modelos de prototipação e de cascata, agregando etapas de análise de risco que ajudarão na decisão de continuidade do desenvolvimento e na eliminação de riscos ao sucesso do projeto (OTERO, 2002; BONA, 2002).

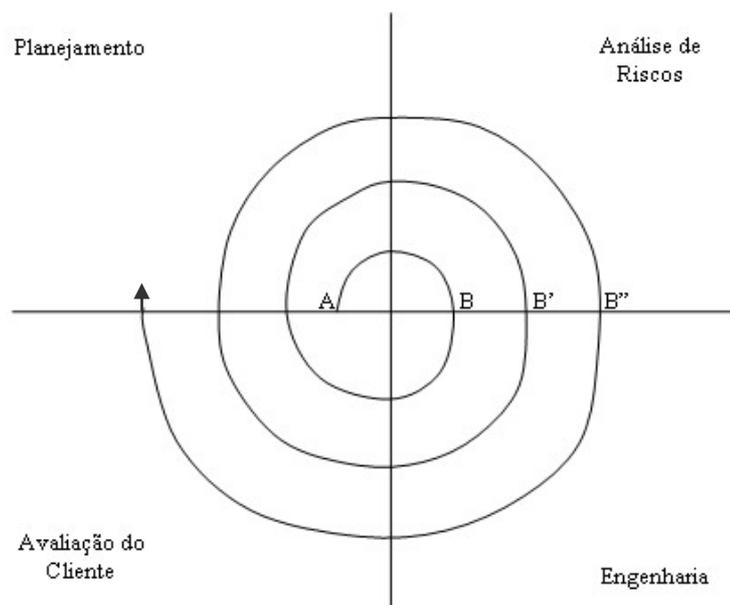


Figura 5 - Modelo Espiral (adaptado de FERREIRA, 2002)

Os pontos B, B' e B'' da Figura 5 representam as sucessivas análises de continuidade ou não do processo de desenvolvimento, com base nas informações coletadas em etapas anteriores, como se fossem portões (ou *gates*) de aprovação do projeto.

Segundo Ferreira (2002), uma particularidade do modelo em espiral frente ao modelo de prototipação está na idéia de evolução e aproveitamento das versões anteriores, sendo que o modelo de prototipação (seção 3.5.2) defende o descarte do protótipo.

3.5.4 Rational Unified Process (RUP)

O *Rational Unified Process* (RUP) é o precursor e, por isso, o mais difundido representante do chamado Processo Unificado de desenvolvimento de software. Criado pela empresa *Rational Software Corporation* (hoje pertencente à IBM), é um modelo de processo de desenvolvimento de software que agrega em sua arquitetura diversas técnicas e boas práticas de engenharia de software, como os casos de uso, o *Unified Modeling Language* (UML) e a orientação a objetos, defendendo o desenvolvimento iterativo e incremental (FERREIRA, 2002; AMBLER, 2005; VASCO; VITHOFT; ESTANTE, 2005).

Segundo Ferreira (2002), as principais vantagens deste modelo consistem na redução de riscos devido a *feedbacks* prematuros, na flexibilidade para alterações de requisitos e na melhoria na qualidade de software. Ambler (2005) afirma que o RUP provê diversos mecanismos como iterações de curto prazo com metas bem definidas e pontos de decisão ao fim de cada fase, contribuindo com a visibilidade gerencial ao longo do processo de desenvolvimento.

Podem-se reunir as características e descrições do RUP feitas por diversas fontes (KRUCHTEN, 2001; FERREIRA, 2002; AMBLER, 2005; LUIZ, 2006) numa descrição do *Rational Unified Process* completa e resumida, exposta a seguir.

O RUP reúne as idéias de gerenciamento de projetos, envolvimento multidisciplinar interno e externo, e engenharia simultânea num modelo de processo de desenvolvimento de software que pode ser resumido pelo esquema da Figura 6.

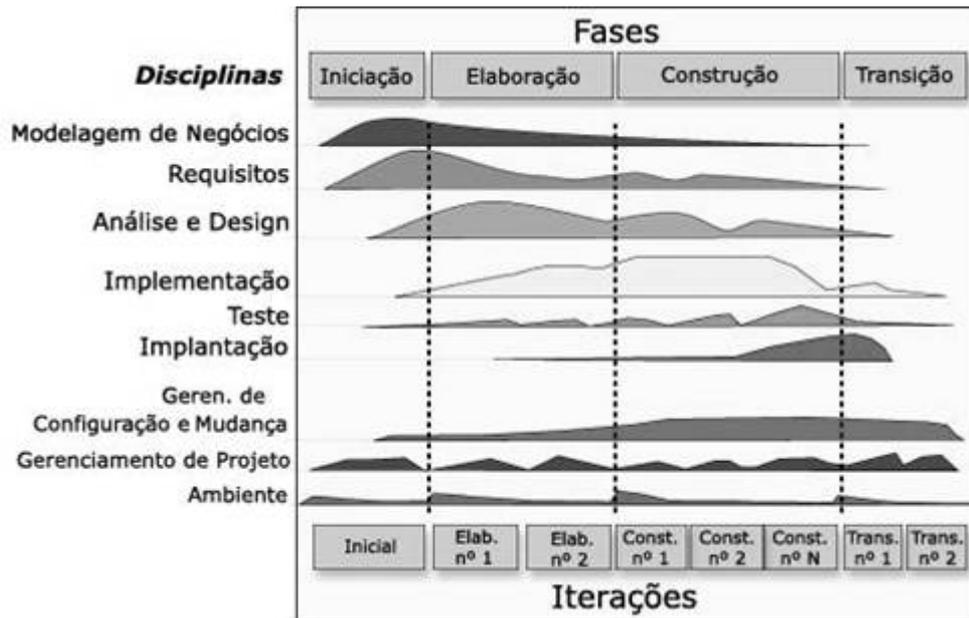


Figura 6 - Dimensões do ciclo de vida RUP (IBM, 2004 apud LUIZ, 2006).

O modelo é dividido nas dimensões horizontal (ou dinâmica) e vertical (ou estática). A dimensão horizontal é composta pelas perspectivas gerenciais e técnicas, que se desenvolvem ao longo do tempo; já a dimensão vertical indica as disciplinas que descrevem o trabalho a ser feito em termos de atividades, responsáveis, artefatos e fluxos de trabalho (*workflows*).

As perspectivas gerenciais e técnicas ocorrem simultaneamente ao longo do tempo, enfatizando o caráter multidisciplinar de integração necessário ao desenvolvimento do software. A perspectiva gerencial é explicada por Ferreira (2002) através de quatro fases:

- a. Concepção ou iniciação: Estabelecimento do negócio para o sistema a ser desenvolvido. Tem como objetivo principal decidir o avanço para a fase de Elaboração. Rocha e Belchior (2004) acrescentam o tratamento dos riscos dos casos de negócio.
- b. Elaboração: Tem como objetivo principal a análise de domínio do problema. Estabelece uma base arquitetural e a avaliação de riscos.
- c. Construção: Desenvolvimento incremental de um produto de software completo e pronto para a transição para os usuários.
- d. Transição: Colocação do software disponível aos usuários.

A perspectiva gerencial pode ocorrer mais de uma vez, evento que quando ocorre chama-se evolução, que gera novas gerações do ciclo de desenvolvimento gerencial, beneficiando-se assim da característica incremental do processo.

Já a perspectiva técnica da dimensão horizontal reflete as iterações de atividades dentro de cada fase da perspectiva gerencial, contribuindo para a liberação de subprodutos aperfeiçoados a cada fase.

Na dimensão vertical, estão representadas as disciplinas, que segundo Luiz (2006, p. 8), são “coleções de atividades relacionadas que fazem parte de um contexto comum em um projeto”. As disciplinas podem ser divididas em nove tipos: seis relacionados à engenharia de software e três ao suporte. Ferreira (2002) apresenta as disciplinas como se tratando dos fluxos de trabalho, ou *workflows*, porém Luiz (2006) baseia-se na versão 2003 do modelo de ciclo de vida da IBM para afirmar que tanto as disciplinas quanto os fluxos de trabalho são componentes do processo, no entanto os fluxos de trabalho representam a seqüência de atividades das disciplinas. Alcântara (2001) apud Ferreira (2002) descreve as disciplinas do RUP da seguinte forma:

- *Disciplinas de engenharia de software:*
 - a. Modelagem do negócio – envolve o entendimento da estrutura e dinâmica da organização cliente, garantindo que clientes, usuários e desenvolvedores tenham a mesma visão das necessidades da organização para a qual será desenvolvido o sistema.
 - b. Requisitos – envolve a definição dos requisitos do sistema e de como gerenciar escopo e mudanças de requisitos.
 - c. Análise e projeto – envolve a tradução dos requisitos numa especificação que descreve como implementar o sistema. A *Unified Modeling Language (UML)* (ver 3.6) é usada como ferramenta nessa etapa.
 - d. Implementação – envolve o desenvolvimento de código: classes, objetos, teste de unidades e integração de subsistemas.
 - e. Teste – envolve a verificação do sistema como um todo, com testes de integração e conformidade com os requisitos especificados.

- f. Distribuição (ou implantação) – envolve o empacotamento, distribuição, instalação e treinamento para os usuários, assim como o planejamento e condução de beta-testes, realizados pelo usuário em seu próprio ambiente.
- *Disciplinas de suporte:*
 - a. Gerência de projeto – envolve o gerenciamento de riscos, planejamento e acompanhamento do projeto.
 - b. Gerência de configuração e mudanças – envolve o gerenciamento dos artefatos gerados durante o desenvolvimento.
 - c. Configuração do ambiente – envolve a organização do ambiente de trabalho para a equipe do projeto e a configuração do RUP para o projeto.

Além das disciplinas, a dimensão vertical é composta ainda pelos papéis que são desempenhados pelos integrantes alocados à equipe de desenvolvimento, pelas atividades que compõem cada papel de cada integrante (o papel de analista de sistema, por exemplo, envolve atividades como definição de requisitos, modelagem de casos de uso, etc), composta ainda pelos fluxos de trabalho (seqüência das atividades executadas para a produção de um resultado do projeto) e pelos artefatos, que são os subprodutos de entregas intermediárias do processo de desenvolvimento do software.

A existência das duas dimensões, horizontal e vertical, no modelo do RUP, dá ao mesmo as vantagens das práticas de engenharia simultânea, amplamente discutidas nos projetos de desenvolvimento de produtos tradicionais. Como mostra a Figura 6, as diversas atividades das disciplinas do RUP trespassam todas as fases da dimensão horizontal, o que é importante para a qualidade dos resultados e sinergia entre diferentes equipes.

Toda a arquitetura e funcionamento do RUP baseia-se em melhores práticas de engenharia de software. Desde o início da existência do modelo, a *Rational Software Corporation* divulgava seis melhores práticas principais que norteavam o RUP, a saber: desenvolvimento iterativo, gerenciamento de requisitos, uso de arquiteturas baseadas em componentes, modelagem visual, verificação contínua da qualidade e gerenciamento de mudanças (KRUCHTEN, 2001). No entanto, segundo Ambler (2005), em outubro de 2005 a IBM (atual proprietária do RUP) divulgou uma lista atualizada das melhores práticas de engenharia de software para o RUP, explicadas a seguir:

- Adaptação do Processo: um modelo de processo de desenvolvimento de software deve ser escolhido com base nas características e porte dos projetos da empresa, adaptando-se o modelo, e não o contrário.
- Balanço de prioridades dos envolvidos: as partes envolvidas (*stakeholders*) – usuários finais, gerentes do projeto, programadores, clientes externos – têm diferentes prioridades e necessidades, muitas vezes conflitantes. Deve-se tomar uma abordagem evolucionária ao longo do desenvolvimento fazendo com que os envolvidos participem ativamente para a convergência ao atendimento de todas as necessidades do projeto.
- Colaboração entre times: fornecer ferramentas e ambiente de trabalho que potencializem a efetiva colaboração e comunicação entre os integrantes do time de desenvolvimento. Isto vai desde o layout que aproxime as pessoas, a adoção de ferramentas de compartilhamento de informações, até práticas de compartilhamento de produtos de trabalho entre times.
- Demonstração iterativa de resultados: planejar iterações no projeto nas quais se possa entregar partes do produto do software antecipadamente, obtendo *feedbacks* antecipados importantes para as correções de rota do projeto.
- Elevação do nível de abstração: utilizar ferramentas de modelagem que elevem o nível de visão e entendimento do projeto, visualizando os processos através de arquiteturas que levarão ao atendimento de requisitos.
- Foco contínuo em qualidade: trazer a disciplina da qualidade total para dentro do processo de desenvolvimento de software, fazendo com que todos, e não somente o time de testes, sejam responsáveis pela qualidade do software.

3.5.5 Enterprise Unified Process (EUP)

De acordo com Ambler (2007), o *Enterprise Unified Process* (EUP) é um novo modelo de processo unificado de software, a exemplo do RUP, com o escopo estendido para as etapas do processo além da entrega do software ao cliente.

Assim, o EUP acrescenta duas novas fases além daquelas apresentadas no RUP: *Produção* e *Descontinuação* (*retirement*) do software.

A fase de *Produção* do software, embora possa parecer relacionada à codificação, na verdade refere-se à etapa posterior à implantação do software no ambiente do usuário cliente. A fase de *Produção*, segundo Ambler (2007), consiste em manter o sistema funcionando no ambiente para o qual é destinado, exigindo para tal a preocupação com relação a atividades de suporte e manutenção, finalizando no momento de decisão de descontinuação do sistema de software.

A fase de *Descontinuação* inicia-se a partir da decisão de se substituir o sistema existente, seja por uma nova versão ou outro produto completamente diferente. Essa fase geralmente ocorre paralelamente ao desenvolvimento e/ou implantação de um novo sistema, exigindo planejamento e coordenação entre os dois projetos (AMBLER, 2007).

Além de novas fases, o EUP traz oito novas disciplinas não abordadas no RUP. Sete disciplinas empresariais (*enterprise disciplines*) estendem o escopo das disciplinas do RUP para todos os processos organizacionais da empresa, isto é, tira o foco de um projeto específico de software para uma visão da empresa como um todo, considerando seu modelo, estratégias e seus diferentes projetos, de forma integrada. As sete disciplinas empresariais são: Modelagem de Negócios da Empresa, Gerenciamento de Portifólio de TI, Arquitetura da Empresa, Estratégia de Reutilização, Gestão de Pessoas, Administração da Empresa e Melhoria de Processo de Software. A oitava disciplina adicional corresponde a Operações & Suporte, fortemente ligadas às fases de Produção e Descontinuação descritas anteriormente (AMBLER, 2007).

Os pontos abordados na seção 3.5 representam muito dos modelos e práticas essenciais para o setor de desenvolvimento de software, tendo sido discutidos e elaborados ao longo dos últimos 30 ou 40 anos. Tais pontos foram incorporados ao método de *benchmarking* desenvolvido neste trabalho, em forma de seus indicadores de avaliação de Engenharia de Software e Desenvolvimento de Produtos: a adoção de um modelo de processo de desenvolvimento, adaptado à empresa, estruturando as atividades, ferramentas e técnicas para a obtenção de um software de qualidade; a simultaneidade no processo de engenharia, com equipes interdisciplinares do início ao fim do projeto; o envolvimento externo e a iteratividade durante o processo de desenvolvimento, contribuindo para os contínuos *feedback* e gerenciamento de requisitos do cliente e visibilidade gerencial ao longo do processo; o planejamento e a contínua realização de testes nos sistemas desenvolvidos; a importância do planejamento e condução da implantação do produto no cliente como atividade integrante do

processo de desenvolvimento do software; *feedback* de campo; e a visão com foco contínuo em qualidade. Todos estes pontos são desdobrados principalmente nos indicadores descritos na seção 4.3.1.1.

3.6 Técnicas para Desenvolvimento de Software

Conforme visto no início deste capítulo, as técnicas são entendidas como o modo de se pôr em prática as orientações definidas pelos métodos, sendo um apoio para a escolha de ferramentas para a execução dos projetos aplicando os conceitos de engenharia de software.

Será apresentada sucintamente, a seguir, uma série de técnicas comumente encontradas na literatura e amplamente adotadas e defendidas pelas empresas de software em todo o mundo.

Rezende (1999) observa que as técnicas possuem similaridade entre si e que as particularidades são poucas: um processo de Análise Estruturada assemelha-se a um *objeto* da Análise Orientada para Objetos, ou a uma *essência* da Análise Essencial, ou um *requisito funcional* da Análise Orientada à Qualidade.

3.6.1 Análise Estruturada

Segundo Gane e Sarson (1986) apud Rezende (1999), a Análise Estruturada é um conjunto de ferramentas orientado para a construção de uma especificação estruturada para projetos, seja ela relacionada à estrutura lógica ou física.

A Análise Estruturada tem como princípio a divisão dos sistemas em módulos e submódulos, aumentando a produtividade, legibilidade e flexibilidade dos sistemas ao explorar as vantagens da modularidade e apresentação gráfica. Sua importância está também na construção de um modelo lógico do sistema em desenvolvimento antes de sua implementação física, reduzindo custos de alterações e correções futuras (REZENDE, 1999).

A Análise estruturada é a técnica básica que rege o funcionamento de diversas técnicas e ferramentas que foram surgindo e evoluindo sob seu princípio. Ela compreende a modelagem de dados, que estrutura e organiza o fluxo de informações de um sistema real, representado graficamente pela Diagramação de Software, como por exemplo o Diagrama de Fluxo de Dados. A Análise Estruturada evoluiu, ainda, segundo Garcia (2003), para a ênfase nos Diagramas de Relacionamento, para a Análise Essencial e para a Orientação a Objetos, cuja

técnica mais difundida foi a UML (*Unified Modeling Language* ou Linguagem de Modelagem Unificada).

A técnica da Análise Essencial consiste em levantar os requisitos verdadeiros de um projeto, orientando para que não sejam confundidos com requerimentos irrelevantes para o negócio, ou muito ligados a uma tecnologia de aplicação específica, o que seriam os chamados requisitos falsos (REZENDE, 1999).

Já a abordagem Orientada a Objetos preocupa-se primeiramente em identificar os objetos a partir do domínio da aplicação, para depois encaixar as funções nestes objetos (MARTINS e JÚNIOR, 1999). As vantagens em termos da qualidade da Orientação a Objetos consistem principalmente na maior confiabilidade do sistema, que por ser baseado em Classes de Objetos são menos suscetíveis a erros, a robustez diante de circunstâncias imprevistas, a manutenibilidade devido a códigos de fácil compreensão e mudanças, e a reusabilidade, como por exemplo, reutilizando classes ou funções existentes em diferentes sistemas.

A Linguagem de Modelagem Unificada (UML), ou *Unified Modeling Language*, é um método que procura sistematizar o uso das linguagens orientadas a objeto, de forma padronizada, através de linguagem de modelagem visual. Segundo Rumbaugh, Jacobson e Booch (1999), a UML é uma linguagem de propósito geral para a especificação, visualização, construção e documentação de artefatos de sistemas de software, que capta suas informações a respeito da estrutura estática e seu comportamento dinâmico.

Rumbaugh, Jacobson e Booch (1999) enfatizam que a UML não é uma linguagem de programação de software, mas que seu benefício está em poder ser usado em conjunto com algumas ferramentas que fornecem geradores de código a partir da UML, gerando códigos numa variedade de linguagens existentes.

Apesar de todos esses benefícios do legado da modelagem da Análise Estruturada, verificou-se que sua notação, originalmente desenvolvida para ser uma técnica descritiva manual, não se adaptaria à modelagem de projetos muito grandes, o que tornaria os modelos de sistemas com portes atualmente vistos impossíveis de serem entendidos e de terem um volume de documentação facilmente mantido. Surgem, dessa forma, as ferramentas CASE (*Computer-Aided Software Engineering*), ou Engenharia de Software Auxiliada por Computador, cujos recursos de desenho são a base para modelar o sistema, automatizando uma série de atividades durante a modelagem, como a construção dos elementos e de seus relacionamentos

e geração de documentação, com uma coleção de ferramentas em um *layout* organizado que auxiliam na consistência e exatidão dos modelos (PRESSMAN, 1995).

3.6.2 Engenharia de Software Auxiliada por Computador

A Engenharia de Software Auxiliada por Computador (CASE – *Computer-Aided Software Engineering*) nada mais é do que software no auxílio à fabricação de software. A ISO/IEC 14102 (ISO, 1995, p. 4) define a ferramenta CASE como “um produto de software que pode dar assistência aos engenheiros de software por meio de suporte automatizado às atividades do ciclo de vida do software”.

Segundo Pressman (1995), as ferramentas CASE englobam um banco de dados que integra as seguintes categorias de ferramentas de desenvolvimento de software: Planejamento de sistemas de informações, Gerenciamento de projeto, Apoio/suporte, Análise e projeto, Programação, Integração e testes, Prototipação, Manutenção e Estrutura (*framework*).

Segundo o autor, as ferramentas CASE ainda estão num patamar evolutivo inicial, assim como eram as ferramentas CAD/CAE/CIM na década de 70 para a indústria de transformação tradicional. O principal ponto a se desenvolver fica por conta da automatização da geração de código de software, a partir da análise e projetos iniciais, o que tornaria o CASE muito mais poderoso.

As técnicas revisadas na seção 3.6 foram incorporadas a dois indicadores no método desenvolvido para o *benchmarking* das empresas de software: *Análise e Modelagem de Arquitetura de Software*, cujo cenário de referência traz as melhores técnicas como a análise estruturada, essencial e orientada a objetos, através de linguagem de modelagem unificada; e o indicador *Ferramentas de Apoio à Engenharia de Software*, pelo qual empresas que adotam apenas ferramentas básicas de codificação, documentação e comunicação, não integradas, enquadram-se no cenário de pior pontuação, enquanto aquelas que exploram todo o potencial de gerenciamento, automação, organização e integração dos elementos do processo de desenvolvimento, via CASE, recebem pontuação de referência.

3.7 Qualidade de Software

Marini (2002) apresenta duas definições de qualidade que se enquadram perfeitamente ao ambiente e produto do desenvolvimento de softwares. Primeiro apresenta a definição de Juran que define a qualidade como sendo uma adequação ao uso do ponto de vista do cliente. Segundo o autor, esta definição é complementada pela definição de Falconi, segundo o qual um produto ou serviço de qualidade é aquele que atende perfeitamente, de forma confiável, segura e no tempo certo, às necessidades do cliente.

Um software, como qualquer outro produto, precisa ser desenvolvido seguindo um processo com qualidade e deve apresentar, ao final de seu desenvolvimento, características que atendam às necessidades dos usuários, o que, sob a ótica do conceito de “adequação ao uso”, significa “possuir qualidade”. Além disso, os fatores confiabilidade, segurança e atendimento ao tempo (seja de entrega ou processamento de dados ou execução de funções, por exemplo) são fundamentais para softwares com qualidade.

Blaschek (1996, p. 109) define a qualidade de sistemas de informação como sendo um conjunto de características que um sistema deve possuir. No entanto, o autor ressalta que esse conjunto de características tem importância dependente da aplicação e do contexto do sistema, tendo-se que levantar quais características são pertinentes a um sistema antes de sua qualidade ser avaliada.

Nakajima (2004, p. 29) complementa essa idéia, ao afirmar que “a qualidade depende do contexto do ambiente de negócios em que o software será utilizado, do domínio da aplicação”, e alerta ainda que a definição de qualidade depende de quem a define: enquanto que para os desenvolvedores uma codificação elegante pode ser uma qualidade, para o usuário são qualidades a facilidade, funcionalidade ou o desenvolvimento rápido, independente de como foi codificado o software (NAKAJIMA, 2004).

Segundo os conceitos apresentados, a qualidade de software depende fortemente das necessidades dos usuários e seus ambientes de aplicação. Dessa forma, a definição correta dessas necessidades, traduzidas em requisitos, tem impacto direto sobre a qualidade dos produtos e satisfação dos usuários. Essa definição de requisitos é explorada mais a seguir, na seção 3.7.1 – Engenharia de Requisitos.

Pressman (2002) apud Nakajima (2004) detalha ainda alguns pontos importantes na definição de qualidade de software: é a satisfação de padrões de desenvolvimento documentados, que garantam a satisfação ao desempenho e aos requisitos funcionais explícitos e implícitos, já

que há características implícitas que, assim como um bom processo de desenvolvimento, não são avaliados diretamente pelo usuário, mas que impactam fortemente no resultado final do software.

Segundo este conceito, fica claro a necessidade de se distinguir a qualidade de produto de software, que diz respeito às características do software em si, da qualidade do processo de software, que está relacionada à gestão da qualidade durante todo o processo de seu desenvolvimento, com vistas a garantir um produto com qualidade.

Para auxiliar as empresas de software a incluir gestão da qualidade em seus processos e produtos, surgiram diversos modelos e normas que orientam na identificação das características de qualidade, sua avaliação e implementação com o objetivo de se atingir um alto grau de maturidade na qualidade dos processos. Esses modelos e normas são apresentados na seção 3.7.2 – Normas e Modelos de Qualidade de Software. Ao mesmo tempo, surgiu a necessidade de se desenvolverem métricas para a medição da qualidade dos softwares, verificando a efetividade da aplicação dos modelos, o que é explorado na seção 3.7.3 – Métricas de software.

3.7.1 Engenharia de Requisitos

A engenharia de requisitos tem importância vital para o resultado da qualidade dos produtos de software, na medida em que é a disciplina responsável por se obter, de forma correta e eficiente, os requisitos que traduzem as necessidades dos clientes.

Kotonya e Sommerville (1998) apud Rodrigues (2006, p.5) definem requisito como sendo “uma declaração de um serviço ou uma restrição que deve ser implementada em um sistema”. Nesse contexto, segundo Rodrigues (2006), a engenharia de software é responsável pela definição de métodos, ferramentas e técnicas que visam à obtenção de uma documentação de requisitos, requisitos esses que representem adequadamente as necessidades de todos os *stakeholders* e padrões de qualidade.

A engenharia de requisitos, segundo Martins (2001), vem substituir a chamada “análise de sistemas”, uma vez que esta última não engloba todas as atividades envolvidas no tratamento de requisitos de um sistema de software de forma apropriada. A engenharia de requisitos pode ser dividida nas seguintes atividades (MARTINS, 2001):

- Elicitação: etapa inicial na qual os desenvolvedores reúnem-se com os usuários com o objetivo de fazer um levantamento de suas necessidades. É a etapa mais complexa, uma

vez que exige a competência dos desenvolvedores em identificar inclusive necessidades não explícitas ou não conhecidas dos usuários, como as relacionadas à organização, processos de negócio e domínio de aplicação.

- **Análise:** envolve o estudo dos dados levantados na elicitação, verificando a real necessidade dos requisitos definidos, sua consistência para que seja isento de contradições, se os requisitos são completos e nenhuma restrição tenha sido omitida, bem como verificando a viabilidade ou possibilidade de atendimento às necessidades, frente a restrições de tempo e custo.
- **Especificação:** trata-se da documentação dos requisitos analisados, servindo como forma de organização e disseminação das informações e necessidades, a todos os *stakeholders*. Técnicas como a diagramação e linguagens unificadas (apresentadas na seção 3.6) são utilizadas para essa atividade.
- **Validação:** certificação de que os requisitos acordados e especificados representam uma descrição aceitável do sistema a ser construído (KOTONYA e SOMMERVILLE, 1998 apud MARTINS, 2001). É basicamente uma análise final das especificações, verificando omissões, ambigüidades ou outros problemas.
- **Gerenciamento:** atividade que deve ser realizada ao longo de todo o desenvolvimento do sistema de software, gerenciando todas as modificações necessárias e seus impactos nos requisitos em si e nas dependências e relacionamentos entre os mesmos.

Algumas normas tratam diretamente sobre os requisitos de software, como por exemplo a ISO/IEC 9126, a ISO/IEC 12207 e IEEE 830. A ISO/IEC 9126 fornece bases para a especificação de requisitos de software a partir do momento que explora as características da qualidade dos produtos de software, qualidades essas internas, externas e em uso. A ISO/IEC 12207 descreve diversas preocupações que se deve ter nas diferentes etapas do ciclo de vida (processo de desenvolvimento) de software. A IEEE 830, por sua vez, descreve boas práticas para a especificação de requisitos de software.

3.7.2 Normas e Modelos de Qualidade de Software

Nesta seção, pretende-se expor, resumidamente, informações básicas sobre as principais normas e modelos atualmente aceitos pela indústria de software mundial e nacional. Não é objetivo aqui detalhar o conteúdo desses padrões de forma exaustiva, mas sim mostrar o

escopo e principais linhas dos mesmos, servindo como base de melhores práticas e modelos de avaliação para a ferramenta de *benchmarking* para o setor.

Segundo Kaushik (2008), bons resultados no desenvolvimento de sistemas e software dependem da investigação de processos, bem como da qualidade, custo e efetividade de seus produtos. Para tal, modelos e normas para avaliação desses processos têm se multiplicado. O autor distingue seis categorias de modelos, dentre as quais as principais são descritas a seguir (KAUSHIK, 2008):

- Normas e diretrizes (*guidelines*): estabelecem o que deve ser feito em um contrato, mas boa parte são recomendações de boas práticas gerais que podem ser adaptadas pelas partes conforme a necessidade. Incluem as séries ISO e normas comerciais como *Electronics Industry Association* (EIA).
- Modelos de melhoria de processo e métodos de avaliação interna: definem as características de bons processos, porém sem prescrever como devem ser desempenhados. Inclui o CMM e Software Engineering CMM (SE-CMM).
- Modelos de ciclo de vida de engenharia de software: especificam os processos em todo o ciclo de vida mas, ao contrário dos CMMs, não provêm requisitos que devem ser atendidos em cada fase. Inclui a ISO/IEC 12207.

O Quadro 1 traz uma lista dos principais documentos encontrados relacionados à qualidade de produto e processo de software, com uma breve descrição do seu conteúdo.

Norma/Modelo	Descrição
CMMI-SW	<i>Capability Maturity Model Integration for Software</i> . Modelo da SEI (<i>Software Engineering Institute</i>) para avaliação da qualidade do processo de desenvolvimento de software.
IEEE 830	<i>Recommended Practice for Software Requirements Specifications</i> (Boas práticas para Especificação de Requisitos).
IEEE P1061	<i>Standard for Software Quality Metrics Methodology</i> (produto de software).
NBR ISO/IEC 12119	Tecnologia de informação - Pacotes de software - Teste e requisitos de qualidade (versão brasileira da ISO/IEC 25051).
NBR ISO/IEC 12207	Tecnologia de informação - Processos de ciclo de vida de software (versão brasileira da ISO/IEC 12207).
NBR ISO/IEC 14102	Tecnologia de informação - Orientação para avaliação e seleção de ferramentas CASE (versão brasileira da ISO/IEC 14102).
NBR ISO/IEC 14143-1	Tecnologia de informação - Medição de software - Medição de tamanho funcional - Parte 1: Definição de conceitos.

Norma/Modelo	Descrição
NBR ISO/IEC 14598	Requisitos, recomendações e guias para a avaliação de produtos de software, baseada na utilização prática da norma ISO/IEC 9126. (Versão brasileira da ISO/IEC 14598)
NBR ISO/IEC 9126-1	Engenharia de software - Qualidade de produto - Parte 1: Modelo de qualidade (Versão brasileira da ISO/IEC 9126).
NBR/ISO 9001	Sistemas de gestão da qualidade - Requisitos. Especifica requisitos para um sistema de gestão da qualidade quando uma organização necessita demonstrar sua capacidade para fornecer de forma coerente produtos que atendam aos requisitos do cliente e requisitos regulamentares aplicáveis.
NBR ISO/IEC 15504	Avaliação de processo de desenvolvimento de software, publicada em fevereiro de 2008 (versão brasileira da ISO/IEC 15504 - SPICE).

Quadro 1 – Normas e modelos de qualidade de software

Devido à grande discussão acerca da qualidade aplicada à indústria de software e a seus produtos, muito em função do aumento da complexidade dos sistemas e também da chamada Crise do Software, instituições como a ISO/IEC, IEEE e SEI (internacionalmente) e ABNT (nacionalmente) passaram a discutir profundamente os modelos e padrões pelos quais se poderia aplicar os conceitos da qualidade a um produto complexo e não físico, como é o caso do software. Toda essa discussão deu origem a normas e modelos como os listados no Quadro 1.

Conforme mencionado anteriormente, a aplicação dos conceitos da qualidade à área de software exigiu a criação de modelos de qualidade aplicáveis a esses produtos. A ISO 9001, por exemplo, traz orientações para a implementação de um sistema da qualidade, que pode constituir-se como um passo inicial para a implementação de um modelo específico de qualidade de software.

Embora a norma ISO 9001 não seja específica para o setor de software, ela traz diretrizes gerais aplicáveis às empresas do setor, sendo que modelos específicos devem complementá-la, como CMMI-SW ou SPICE, ou as normas relacionadas a software: para a qualidade de processo de desenvolvimento de software, temos a ISO/IEC 12207 e sua versão brasileira; voltadas à qualidade de produto de software, há como principais normas a ISO/IEC 9126, ISO/IEC 25051, ISO/IEC 14598, e suas versões brasileiras.

Serão abordados, a seguir, os principais modelos e normas mundialmente difundidos e aceitos.

3.7.2.1 ISO/IEC 9126

A ISO/IEC 9126 teve sua primeira versão publicada em 1991 e é uma das mais antigas da área de qualidade de software, sendo que a versão 1991 foi substituída pela atual versão 2001, a ISO/IEC 9126-1:2001 (ISO, 2007). Sua versão brasileira foi publicada em agosto de 1996 como NBR 13596, e substituída em julho de 2003 pela NBR ISO/IEC 9126-1.

Esta norma descreve um modelo de qualidade do produto de software, composto de duas partes: a) qualidade interna e qualidade externa e b) qualidade em uso. A primeira parte do modelo especifica seis características para qualidade interna e externa, as quais são por sua vez subdivididas em sub-características. Estas sub-características são manifestadas externamente, quando o software é utilizado como parte de um sistema computacional, e são resultantes de atributos internos do software (ABNT, 2007).

A ISO/IEC 9126 foi a primeira norma que desenvolveu um modelo da qualidade específico para o setor de software, atualmente apresentado na parte 1 da norma. Segundo Azuma (1996, p. 148), um modelo da qualidade é um modelo que desdobra a qualidade em um conjunto de características e seus relacionamentos, fornecendo as bases para a especificação de requisitos da qualidade e para a avaliação da qualidade de um produto.

O modelo da qualidade apresentado define seis características de qualidade interna e externa de software como produto: Funcionalidade; Confiabilidade; Usabilidade; Eficiência; Manutenibilidade; e Portabilidade. Cada uma dessas características são divididas em sub-características, conforme tabela a seguir (ABNT, 2003):

Tabela 2 – Características e Sub-características de qualidade de software

Características da qualidade interna e externa	Sub-características	
Funcionalidade	Adequação	Segurança de acesso
	Acurácia	Conformidade da funcionalidade
	Interoperabilidade	
Confiabilidade	Maturidade	Recuperabilidade
	Tolerância a falhas	Conformidade da confiabilidade
Usabilidade	Inteligibilidade	Atratividade
	Apreensibilidade	Conformidade da usabilidade
	Operacionalidade	
Eficiência	Comportamento em	Conformidade da

	relação ao tempo	eficiência
	Utilização de recursos	
Manutenibilidade	Analisabilidade	Testabilidade
	Modificabilidade	Conformidade da manutenibilidade
	Estabilidade	
Portabilidade	Adaptabilidade	Capacidade para substituir
	Capacidade para ser instalado	Conformidade da portabilidade
	Coexistência	

Fonte: ABNT, 2003

Já com relação ao Modelo da Qualidade em Uso, a ISO/IEC 9126 define quatro características que representam os atributos da qualidade: *eficácia*, *produtividade*, *segurança* e *satisfação*. Esses atributos representam a qualidade do ponto de vista do usuário, e são dependentes das qualidades interna e externa do software (ABNT, 2003).

3.7.2.2 ISO/IEC 25051 (NBR ISO/IEC 12119)

A ISO/IEC 25051 foi publicada em 2006, em substituição à ISO/IEC 12119, publicada inicialmente em 1994 e que originou a atual versão brasileira, a NBR ISO/IEC 12119, intitulada “Tecnologia de informação - Pacotes de software - Teste e requisitos de qualidade”, publicada em outubro de 1998.

A ISO/IEC 25051 define os requisitos da qualidade específicos para os chamados Pacotes de Software, ou software de prateleira (do inglês, COTS - *Commercial Off-The-Shelf*), com base nos requisitos e características da qualidade descritos na ISO/IEC 9126. Além disso, define os requisitos para a documentação de testes (plano, descrição e resultados de testes), podendo ser utilizada por fornecedores, órgãos certificadores, laboratórios de teste de software ou compradores de software (ISO, 2007).

Os requisitos da qualidade definidos na ISO/IEC 25051 referem-se às características da descrição do produto como item integrante do pacote, bem como à documentação do usuário (orientações de utilização) e da qualidade do software em si. A norma exige a disponibilidade de uma descrição das características do produto, com indicações das características de sua qualidade, inclusive qualidade em uso, conforme modelos da ISO/IEC 9126. A documentação deve ser completa, correta e consistente, de fácil compreensão e acesso ao público-alvo do pacote de software.

Além disso, a norma traz os requisitos para a documentação dos testes realizados pelo fabricante ao longo do desenvolvimento do software, com todo seu plano, descrições e resultados, bem como traz instruções destinadas a instituições certificadoras de como avaliar a conformidade do software com as informações fornecidas pelo fabricante nas descrições, documentações do usuário e dos testes.

3.7.2.3 ISO/IEC 12207

A ISO/IEC 12207 – Tecnologia de informação - Processos de ciclo de vida de software – foi publicada internacionalmente em 1995, possuindo dois adendos (2002 e 2004), sendo que está atualmente em processo de revisão (ISO, 2007). Sua versão brasileira foi publicada em 1998, sob o mesmo título e com o código NBR ISO/IEC 12207 (ABNT, 2007).

Esta norma estabelece uma nomenclatura padrão para as diversas etapas, atividades e tarefas constituintes de um processo de desenvolvimento de software, durante todo o seu ciclo de vida, com o objetivo de servir de base comum para a indústria de software. Este modelo pode ser aplicado na aquisição de softwares ou sistemas que contêm software, bem como de serviços de software, e ainda no fornecimento, desenvolvimento, operação e manutenção de produtos de software (ABNT, 2007).

Embora a norma estabeleça um padrão de nomenclatura e modelo para o processo de desenvolvimento de software, a mesma, como ocorre com a maioria das normas, não é prescritiva no sentido de detalhar como pôr em prática as atividades e tarefas que fazem parte do modelo, nem tampouco é um modelo de avaliação, papel este desempenhado pela norma ISO/IEC 15504.

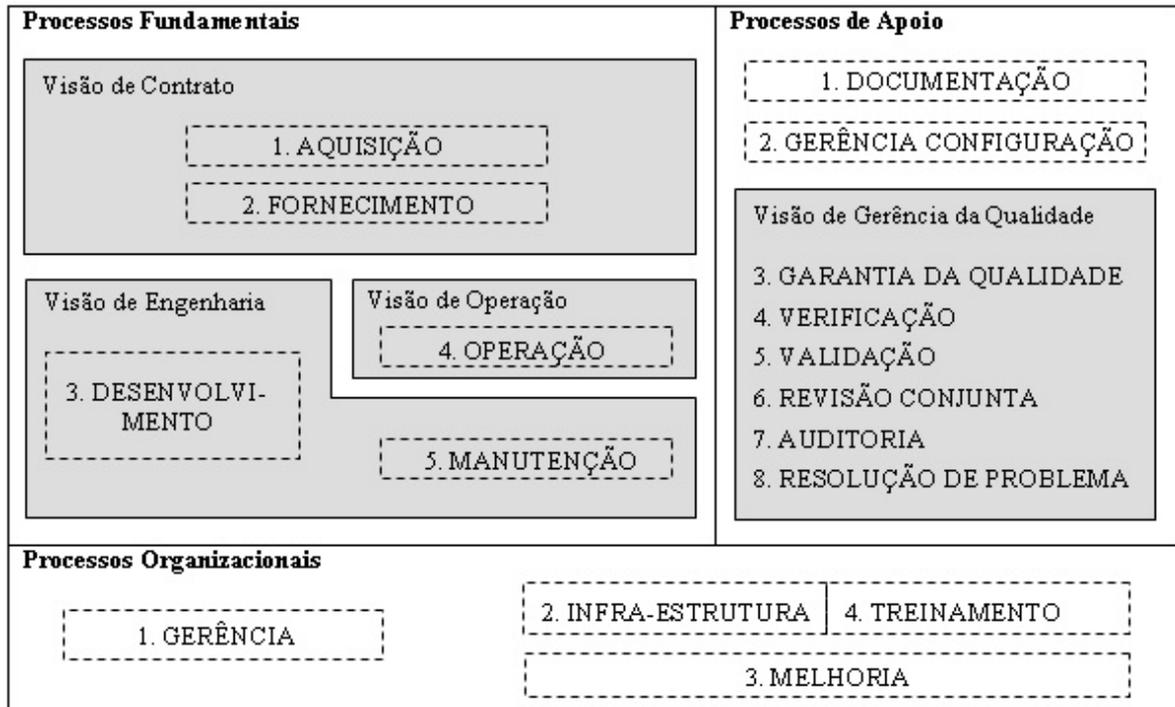


Figura 7 – Modelo de ciclo de vida de software - ISO 12207 (adaptado de ABNT, 1998)

Conforme visto na figura acima, o modelo de ciclo de vida da ISO/IEC 12207 categoriza os diversos processos que devem estar mapeados em toda organização envolvida com o desenvolvimento de produtos de software em três categorias:

- Processos Fundamentais: pode-se traduzir como sendo os processos diretamente ligados à produção do produto, e que devem ser estabelecidos a cada projeto de desenvolvimento de um sistema de software.
- Processos de Apoio: são aqueles que darão suporte à execução dos processos fundamentais. Assim como os fundamentais, os processos de apoio devem ser estabelecidos a cada novo projeto.
- Processos Organizacionais: são aqueles ligados ao funcionamento da organização como um todo, não estando ligados diretamente a um projeto de software, como ocorre com os processos anteriores. Trata-se de gerir a organização e seus processos, garantindo a infra-estrutura necessária ao funcionamento e melhoria contínua dos processos e projetos da empresa.

A norma exige que os processos (ver processos na Figura 7) estejam implementados na organização, ou seja, que haja uma sistemática que garanta sua execução, quando aplicável,

utilizando para isso modelos, contratos, planos, ferramentas e métodos, documentação e acompanhamento de resultados, para cada processo.

3.7.2.4 CMMI-SW

O CMMI-SW (*Capability Maturity Model Integration for Software*) surgiu a partir de diferentes modelos de avaliação de maturidade de processo existentes. Esses diferentes modelos, chamados CMM, que passaram a ser desenvolvidos a partir de 1991, eram focados cada um em uma disciplina específica, dentre as quais os modelos para sistemas de engenharia, engenharia de software, aquisição de software, gestão e desenvolvimento de pessoas e desenvolvimento integrado de produtos. A existência de diferentes modelos CMM apresentava dificuldades de implementação nas empresas, demandando tempo e elevado custo para a aplicação dos modelos de forma integrada, ao longo de toda a organização (SEI, 2002).

A necessidade de integrar os diferentes modelos existentes fez com que diversas organizações (industriais, governamentais e o próprio SEI) se reunissem para desenvolver o chamado CMMI (*Capability Maturity Model Integration*), que deveria englobar os conceitos dos modelos CMM já existentes, bem como deveria estar alinhado à ISO/IEC 15504 em desenvolvimento (modelo SPICE), que já discutia a avaliação de processo de software através de comitês técnicos na ISO (SEI, 2002).

O modelo de CMMI a que se chegou pode ser utilizado por qualquer empresa, ajudando a definir objetivos e prioridades para a melhoria de seus processos, sejam eles de desenvolvimento, aquisição ou manutenção de produtos ou serviços, com objetivo de atingir níveis de maturidade mais elevados (SEI, 2002). As disciplinas integrantes do CMMI também podem ser escolhidas pela empresa, sendo que para o setor foco deste trabalho dispõe-se do CMMI-SW – CMMI para Engenharia de Software, que cobre o desenvolvimento, operação e manutenção de software, de forma sistemática, disciplinada e quantificada.

O modelo CMMI trabalha com o conceito de Áreas de Processo, que são os grupos de práticas relacionadas a 4 categorias: gerenciamento de processo, gerenciamento de projeto, engenharia e suporte (SEI, 2002).

O CMMI pode ter duas formas de representação: contínua ou por estágios. A representação contínua utiliza níveis de capacidade para medir a evolução dos processos, enquanto que a representação por estágios utiliza níveis de maturidade (SEI, 2002).

A avaliação dos níveis de capacidade do modelo contínuo se aplica a cada Área de Processo, e cada nível corresponde ao alcance de uma meta genérica e seu conjunto de práticas. Já no modelo por estágios, os níveis de maturidade se aplicam a toda a organização, e cada nível corresponde a um conjunto predefinido de Áreas de Processo (SEI, 2002). A representação que será apresentada neste trabalho refere-se à de estágios, uma vez que é a representação derivada do modelo CMM originalmente relacionado à engenharia de software, o SW-CMM.

Os níveis de maturidade, que são os estágios evolutivos em direção à melhoria de processo, são resumidos, a partir da SEI (2002), da seguinte forma:

- Nível de maturidade 1 – Inicial: a organização não possui práticas de gestão bem estabelecidas nem um ambiente estável ao desenvolvimento de processos. Apresentam atuação reativa e falta de planejamento.
- Nível de maturidade 2 – Gerenciado: projetos gerenciados, incluindo requisitos, produtos e serviços, com processos planejados, executados, medidos e controlados. Compromissos são estabelecidos entre as partes interessadas, com revisões para *feedback* e validação das entregas. Os padrões de processo são definidos a cada novo projeto.
- Nível de maturidade 3 – Definido: processos (de desenvolvimento, manutenção, gestão) são bem definidos e compreendidos, descritos em forma de padrões, procedimentos, ferramentas e métodos, que são bem documentados e revisados continuamente. A principal diferença em relação ao nível 2 é que o conjunto de padrões da organização é utilizado em todos os projetos e são também descritos com maior nível de detalhes e rigor.
- Nível de maturidade 4 – Quantitativamente gerenciado: qualidade e *performance* de processo são gerenciados quantitativamente, através de metas, acompanhamento e tratamento estatístico, verificando causas de variações e formando repositório de dados para apoio à tomadas de decisão futuras.
- Nível de maturidade 5 – Em otimização: foco na melhoria contínua da *performance* de processo eliminando causas comuns de variações de resultado, com objetivos mensuráveis estabelecidos e revisados continuamente. A principal diferença para o nível 4 é que melhorias nos processos são conseguidas de forma incremental ou por inovações, com a participação de todos na organização.

Segundo a SEI (2002), as organizações podem atingir melhorias progressivas na maturidade de seus processos primeiramente através de estabilidade nos projetos e, em seguida, estendendo a uma melhoria contínua a todos os processos da organização, usando dados tanto qualitativos como quantitativos para a tomada de decisão.

3.7.2.5 ISO/IEC 15504

A iniciativa da ISO ao elaborar a ISO/IEC 15504 teve como objetivo criar uma norma que abranja a avaliação da maturidade de todo o processo de desenvolvimento de software, a exemplo do que já é feito por meio de modelos como o SW-CMM. A idéia foi de se utilizar dos conceitos já oficializados pela ISO/IEC 12207 (ver seção 3.7.2.3), juntamente com o modelo de avaliação do SW-CMM, o que resultou numa norma que possui as características de avaliação de nível de maturidade próximas às do modelo contínuo do CMMI-SW, que se baseou na própria ISO/IEC 15504, quando ainda estava em desenvolvimento pelo projeto então chamado SPICE. No Brasil, foi publicada em fevereiro de 2008 e passou a ser válida a partir de março.

Os níveis evolutivos de capacidade que podem ser atingidos e medidos na ISO/IEC 15504 são equivalentes aos cinco níveis do CMMI-SW: nível 0 – Processo Incompleto (não desempenhado); nível 1 – Processo Realizado (desempenhado informalmente); nível 2 – Processo Gerenciado (planejado e rastreado); nível 3 – Processo Estabelecido (bem definido); nível 4 – Processo Previsível (quantitativamente controlado); e nível 5 – Processo em Otimização (melhorando continuamente).

Cada nível de capacidade está atrelado ao atendimento de Atributos de Processo, que podem ser considerados práticas exigidas das empresas para a caracterização do amadurecimento de seus processos internos. A ISO/IEC 15504 definiu limites de realização de cada atributo em um determinado processo avaliado, em formato percentual, sendo:

- Não-realizado – até 15%
- Parcialmente realizado – acima de 15% até 50%
- Em grande parte realizado – acima de 50% até 85%
- Totalmente realizado – acima de 85% até 100%

A medição da realização de cada atributo em um determinado processo irá gerar, segundo a norma, o perfil daquele processo, cujo nível de capacidade poderá ser medido pela comparação com os limites estabelecidos conforme a Tabela 3 (ISO, 2003).

Os níveis de capacidade apresentados fazem parte da Dimensão de Capacidade da ISO/IEC 15504, que representa o modelo de medição da norma com base em atributos que indicam a capacidade de um determinado processo em atingir seus resultados. No entanto, a determinação de quais processos precisam estar estabelecidos é dada pela outra dimensão da norma, a de Processo, relacionada a quais processos precisam ser implementados para a boa prática de engenharia de software (MOREAU apud NAKAJIMA, 2004).

Tabela 3 – Atributos exigidos por nível de capacidade na ISO/IEC 15504

Atributos de Processo	Níveis de Capacidade				
	1	2	3	4	5
Resultado do processo	>50%	>85%	>85%	>85%	>85%
Gerenciamento do resultado		>50%	>85%	>85%	>85%
Gerenciamento do produto de trabalho		>50%	>85%	>85%	>85%
Definição de processo			>50%	>85%	>85%
Desdobramento de processo			>50%	>85%	>85%
Medição de processo				>50%	>85%
Controle de processo				>50%	>85%
Inovação de processo					>50%
Otimização de processo					>50%

Fonte: adaptado de ISO, 2003

No entanto, a ISO (2003) relaciona, nesta norma, a Dimensão de Processo a um chamado Modelo de Referência de Processo, que pode ser qualquer modelo adotado pela empresa que apresente descrições de etapas de processo em conformidade com as exigências da norma, sendo este modelo ou uma produção de uma comunidade de interesse, ou uma norma nacional ou internacional, ou ainda especificações publicamente disponíveis. Como exemplo de aplicação, a parte cinco da ISO/IEC 15504 (ISO, 2006) utiliza como Modelo de Referência de Processo o ciclo de vida de software descrito na ISO/IEC 12207, embora outros possam ser adotados.

3.7.3 Métricas de software

Conforme visto anteriormente, a qualidade de um produto ou processo de software depende de diversas características relacionadas ao que se espera obter ao final do desenvolvimento, para o atendimento das necessidades e expectativas, explícitas ou não, dos usuários clientes e outros *stakeholders* do processo.

Tais características foram exploradas em maior detalhe em normas como a ISO 9126, que as dividiu em sub-características e atributos da qualidade, com o objetivo de facilitar e sistematizar a avaliação prática da qualidade dos produtos de software.

A avaliação, no entanto, exige a definição de um método e uma escala de medição, o que compõe, segundo a ABNT (2003), uma Métrica. Segundo Marini (2002), criou-se, então, uma área de estudo à parte dentro da Qualidade de Software conhecida como Métricas de Software, com o objetivo de definir a medição numérica, de forma mais precisa, de uma determinada característica.

Segundo Pressman (1995), essa medição proporcionada pelas métricas, quando relacionadas à engenharia de software, deve ser conduzida pelas seguintes razões:

- Indicar a qualidade do produto.
- Avaliar a produtividade das pessoas.
- Avaliar os ganhos de produtividade e qualidade pela aplicação de métodos e ferramentas de software.
- Formar uma linha de base para estimativas.
- Justificar adoção de novas ferramentas ou treinamentos.

Pressman (1995) cita ainda o nível de dificuldade na medição de algumas medidas que somente podem ser avaliadas de forma indireta, como é o caso de funcionalidade de software, qualidade, complexidade, eficiência ou confiabilidade. Já os atributos que permitem medidas diretas são relativamente mais fáceis de avaliar, como por exemplo, o número de linhas de código como indicador do custo e esforço de desenvolvimento de um produto de software.

As medidas tradicionais relacionadas à qualidade de software mais amplamente difundidas estão relacionadas à funcionalidade, usabilidade, manutenibilidade e confiabilidade, que inclusive fazem parte do modelo da ISO/IEC 9126 e são citadas por Pressman (1995), embora este último use o termo “corretitude” para a funcionalidade ou cite especificamente a

integridade, que é abordada na ISO/IEC dentro da característica Funcionalidade, através da sub-característica Segurança de Acesso.

Pressman (1995, p. 70) cita que a corretude “é o grau em que o software executa a função que é dele exigida”, sendo que se utiliza como medida mais comum os defeitos por KLOC (10^3 *lines of code* ou mil linhas de código), sendo entendido como defeito o não atendimento às especificações.

Porém, mesmo as medidas diretas utilizadas em métricas como a descrita acima, consideradas de fácil levantamento pelo autor, apresentam controvérsia e não são consenso na engenharia de software. Isso porque características como a quantidade de linhas de código de um software (LOC ou KLOC) dependem de diversos fatores, como a linguagem de programação adotada, o tamanho ou experiência da organização, a complexidade do problema a ser resolvido ou a escolha de técnicas e ferramentas para o processo (PRESSMAN, 1995).

Medidas de confiabilidade como MTBF (*Mean Time Between Failure* ou Tempo médio entre falhas) têm mais correlação com as expectativas e necessidades dos usuários do que medidas de quantidade de falhas por linhas de código, já que o usuário não está preocupado se o software possui ou não muitas linhas de programação, mas sim, se o mesmo demora a apresentar problemas. Métricas apresentadas como sugestão pela norma ISO/IEC 9126, partes 2 e 3, adequam-se a este conceito, ao apresentar, por exemplo, as seguintes métricas:

- Funcionalidade $\approx [1 - (\text{n}^\circ \text{ de funções faltantes detectadas})] / (\text{n}^\circ \text{ de funções descritas nas especificações})$ ou $\approx [1 - (\text{n}^\circ \text{ de funções implementadas incorretamente})] / (\text{n}^\circ \text{ de funções descritas nas especificações})$.
- Confiabilidade $\approx (\text{n}^\circ \text{ de falhas detectadas em revisões}) / (\text{n}^\circ \text{ estimado de falhas com base em dados históricos ou modelo de referência})$ ou $\approx [1 - (\text{n}^\circ \text{ de travamentos em falhas})] / (\text{n}^\circ \text{ de falhas ocorridas})$;

Pressman (1995) também foca as necessidades dos usuários ao definir métricas como:

- Confiabilidade $\approx \text{MTBF} = \text{MTTF} + \text{MTTR}$, onde MTTF é o tempo médio entre ocorrência de falhas e MTTR é o tempo médio de reparo.
- Disponibilidade = $[\text{MTTF} / (\text{MTTF} + \text{MTTR})] \times 100\%$.

Além da importância imediata e pontual da utilização de métricas para a qualidade no desenvolvimento de um produto de software, auxiliando na localização de defeitos, problemas de legibilidade, etc., destaca-se a importância crescente da utilização das métricas subsidiando dados históricos para análises estatísticas. Essa prática é citada pelo *Software Engineering Body of Knowledge – SWEBOOK* (IEEE, 2004), bem como por Pressman (1995) que utiliza o termo Garantia Estatística de Qualidade para atendimento à Garantia da Qualidade de Software (SQA – *Software Quality Assurance*).

A manutenção de dados históricos permite que a organização passe a monitorar o comportamento de diversas características em diferentes desenvolvimentos de produtos ao longo dos anos, permitindo a análise estatística dessas informações para o auxílio à tomada de decisões quanto aos projetos atuais e futuros. A identificação de padrões de comportamento, através de análises como Pareto, gráficos de dispersão e distribuição, testes estatísticos ou modelos de regressão e tendência, auxilia em definições relativas a esforço e foco de testes, problemas mais prováveis por tipo de aplicação desenvolvida, estimativa de recursos e prazos necessários em novos projetos, entre outras.

3.7.4 Gerenciamento do Nível de Serviço

No atual mercado de tecnologia da informação, tem sido verificada cada vez mais uma tendência de terceirização, por parte de grandes empresas, de serviços de TI. Por meio do chamado *outsourcing*, essas empresas têm repassado também o desenvolvimento de seus sistemas de software para empresas especializadas, mantendo seu foco em seu verdadeiro negócio e usufruindo os serviços especializados das empresas que têm seu foco no desenvolvimento de software em si. Segundo Carvalho e Pires (2007), o mercado global de terceirização de TI foi de US\$ 607 bilhões em 2004 e desde então apresenta uma taxa de crescimento de 5,6% ao ano. Embora não se tenham números recentes e precisos, no Brasil a terceirização é um mercado importante, a exemplo do que ocorre no mercado global. Segundo Gutierrez e Alexandre (2004), o mercado de *outsourcing* brasileiro expandiu-se 9,3% (em dólares) entre 2001 e 2002, com grande parte do mercado sob domínio de grandes fornecedores cuidando principalmente de infra-estrutura e gestão de aplicativos.

Devido à grande difusão da terceirização, inclusive entre diferentes empresas, sentiu-se a necessidade de instrumentos que regessem essa coordenação e troca de informações, bem como que conferissem a gestão das expectativas e níveis de serviço desejados por ambas as

partes. Surgem, dessa forma, os Acordos de Nível de Serviço (ANS), também chamados *Service Level Agreement (SLA)*.

Um Acordo de Nível de Serviço é um instrumento contratual que visa a gestão da qualidade e quantidade a partir do entendimento claro do conjunto de compromissos de ambas as partes. Serve, dessa forma, como ferramenta de comunicação e gestão de conflitos, definindo claramente os critérios de avaliação dos serviços prestados (GOMES, FALBO e MENEZES, 2005).

Os ANS's são instrumentos para o chamado Gerenciamento de Níveis de Serviço (*Service Level Management – SLM*), que possui os objetivos acima descritos e que deve gerir os projetos de desenvolvimento de software, foco deste trabalho, ao longo de todo o ciclo de desenvolvimento. O Gerenciamento de Níveis de Serviço está presente em modelos como o ITIL (*Information Technology Infrastructure Library*) – Biblioteca de Infra-estrutura de Tecnologia da Informação – ou o CMMI-SW (*Capability Maturity Model Integration for Software Engineering*), entre outros.

O ITIL é um método de gerenciamento de sistemas de informação que tem por objetivo o alinhamento operacional da TI às estratégias da empresa. Em termos de gerenciamento de serviços de TI, o ITIL divide-se em dois principais tipos de processos: de entrega e de suporte de serviços. O Gerenciamento de Níveis de Serviço faz parte do processo de Entrega de Serviços, correspondendo ao processo que busca garantir e monitorar os acordos de nível de serviço estabelecidos, indicando o que compõe cada serviço e o respectivo nível de qualidade necessário (SANTOS, 2006).

O gerenciamento de níveis de serviço também pode ser encontrado, implicitamente, no CMMI-SW, atrelado à prática genérica de Envolvimento das partes interessadas (*stakeholders*) relevantes. No nível de maturidade quatro, por exemplo, essa prática genérica exige que as empresas envolvam os clientes no estabelecimento de objetivos e prioridades quanto à qualidade e *performance* (quantitativos) do projeto, sua medição, identificação e gestão de riscos no cumprimento dos objetivos de qualidade e *performance*, bem como na definição de ações corretivas necessárias ao cumprimento destes.

O ANS é um importante instrumento para o gerenciamento dos níveis de serviço em empresas desenvolvedoras de software, que precisa estar alinhado às necessidades estratégicas dos clientes e ser continuamente revisado e consensado entre as partes, tornando-se a principal garantia do gerenciamento da qualidade do produto e processo de software.

Para este fim, além do completo estabelecimento do escopo, dos níveis desejados, objetivos, critérios, riscos e responsabilidades, constantes no ANS, é fundamental a definição de indicadores e métricas para o acompanhamento dos níveis de serviço de forma objetiva. Gomes, Falbo e Menezes (2005) enfatizam a importância de mecanismos e relatórios automatizados para esse acompanhamento, cujos público-alvo e tipo de informações contidas devem determinar a frequência de geração. Além de contribuir para a comunicação com os clientes, o acompanhamento contínuo permite a definição de planos de ações corretivas e a melhoria contínua dos níveis de serviço atingidos.

A revisão apresentada em toda a seção 3.7 levanta questões importantes com relação à qualidade de software, constantes de diferentes modelos e normas desenvolvidos para o setor. As práticas principais foram incorporadas à ferramenta de *benchmarking* desenvolvida, por meio principalmente (mas não somente) dos indicadores das seções do questionário referentes a Gestão da Qualidade (ver 4.3.2.5) e Qualidade de Software (ver 4.3.1.2). A importância da visão da qualidade presente em toda a organização; a pesquisa das necessidades dos clientes; a completa engenharia de requisitos; o uso de métricas para a busca e medição das características de qualidade dos produtos de software tais quais funcionalidade, confiabilidade, usabilidade, eficiência e manutenibilidade; a documentação de produtos e testes; a gestão de fornecedores e seus produtos; a satisfação dos clientes; o controle de mudanças de configuração; o *feedback* de campo por meio de informações do produto em uso; a maturidade dos processos e redução no tempo do ciclo de desenvolvimento; e o gerenciamento do nível de serviço, são todos pontos fundamentais abordados pelo método de *benchmarking* para o setor de software.

3.8 Gerenciamento de Projetos

Conforme visto até o momento, muito do sucesso das empresas de software recai sobre práticas de gerenciamento de projetos de seus produtos (ver seção 3.5.5).

Torna-se importante levantar o que aqui se define como projeto, que segundo Ferreira (2002, p. 79) trata-se de “um conjunto de esforços que visam à realização de um intento, esboço ou desígnio, ou seja, abrange a fase de execução do que foi delineado”.

No conceito apresentado no PMBOK (PMI, 2004, p. 5), um projeto é “um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo”, sendo que este conceito acrescenta o caráter temporal dos projetos, significando que possuem início e fim definidos, bem como acrescenta o caráter da singularidade de produtos, serviços ou resultados como sendo entregas exclusivas, ou seja, o produto de um projeto nunca será igual ao de outro, por mais semelhantes que possam ser.

A definição do Guia PMBOK representa exatamente os projetos de desenvolvimento de software, uma vez que os produtos das empresas de software são únicos, considerando-se inclusive a customização de softwares existentes para as necessidades de cada novo cliente, que deverão ter seus requisitos levantados, adaptações implementadas e implantação do software no ambiente dos usuários, caracterizando um projeto exclusivo a ser gerenciado. A produção de um software sempre será algo exclusivo, mesmo no desenvolvimento de softwares de prateleira, uma vez que a produção, no caso de produtos de software, refere-se à etapa de desenvolvimento, diferentemente de produtos tradicionais manufaturados. Para estes últimos, a produção é a materialização do produto desenvolvido, enquanto que os softwares apenas necessitam, após seu desenvolvimento, serem gravados e replicados em mídias ou distribuídos pela internet ou outros meios.

Os projetos, devido a suas características acima expostas e importância para o negócio das organizações, sejam elas de software ou não, exigiram o desenvolvimento de uma disciplina que explorasse os conceitos e definisse modelos e melhores práticas para seu estabelecimento eficiente dentro das organizações. Surge, dessa forma, a disciplina de gerenciamento de projetos, amplamente difundida e discutida no mundo.

Uma das entidades que se destacou mundialmente foi o *Project Management Institute* (PMI), que desenvolveu o Guia PMBOK, reunindo um conjunto de conhecimentos em gerenciamento de projetos, já em sua terceira edição. Segundo o PMBOK, “o gerenciamento de projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos” (PMI, 2004, p. 8).

Diante do conceito exposto, fica mais que clara a importância do gerenciamento de projetos para o setor de software. Anacleto (2004) cita, por exemplo, que no processo de software, a ISO 12207 (ver seção 3.7.2.3) também considera, além dos processos fundamentais como análise de requisitos ou testes, os processos organizacionais e gerenciais, como por exemplo, a gerência de projetos de software.

3.8.1 O Ciclo de Vida e Processos dos Projetos

O processo completo de um determinado projeto, que como definido anteriormente implica um início e fim, começa desde a verificação da necessidade de se resolver um problema determinado, seja ele desenvolver um novo software, aprimorar uma versão de um software existente etc., até o momento em que se entrega o produto final do projeto, definido em seu escopo.

Porém, esse ciclo completo geralmente não costuma durar menos que seis meses em projetos do porte típico da indústria de software. Os projetos, além de exigirem longos prazos, exigem também o gerenciamento de muitos recursos, interesses ou riscos, dentre outras preocupações, ao longo de todo o seu ciclo de vida.

Para se ter um melhor controle sobre o gerenciamento de um projeto ao longo de sua existência, faz-se necessária sua divisão em fases e processos, cada qual com suas características e atividades. Essa organização auxiliará na definição de prioridades e focos a serem dados a determinadas atividades, recursos e práticas, com o objetivo de se ter um projeto dentro de prazo, custo e escopo que garantam sua qualidade total.

O PMI (2004) divide o projeto em três fases principais: inicial, intermediária(s) e final (ver Figura 8). Essas divisões, embora simples, são divisões macro que ajudam as empresas a projetar o montante de recursos que deverão ser mobilizados ao longo do tempo; uma falha de planejamento aqui poderá atrasar significativamente o projeto, aumentar seus custos ou simplesmente impedir sua conclusão.

Assim, o custo das mudanças é baixo na fase inicial dos projetos, sendo crescente até a fase final. Essa característica enfatiza a importância de se gerenciar muito bem a participação dos *stakeholders* na fase inicial, evitando mudanças ao longo do desenvolvimento do projeto que levem ao não cumprimento das metas de custo.

As fases inicial, intermediária e final de um projeto são, ainda segundo o PMI (2004), compostas por cinco grupos de processos de gerenciamento: iniciação, planejamento, execução, monitoramento e controle, e encerramento (ver Figura 8):

- Iniciação: processos correspondentes às definições e autorização de um projeto.

- Planejamento: planejamento e detalhamento dos objetivos e ações necessárias para o cumprimento do escopo do projeto.
- Execução: processos de mobilização dos recursos necessários ao gerenciamento do projeto.
- Monitoramento e controle: medição regular do progresso para monitorar desvios e planejar ações corretivas e preventivas.
- Encerramento: formalização da aceitação do produto final do projeto.

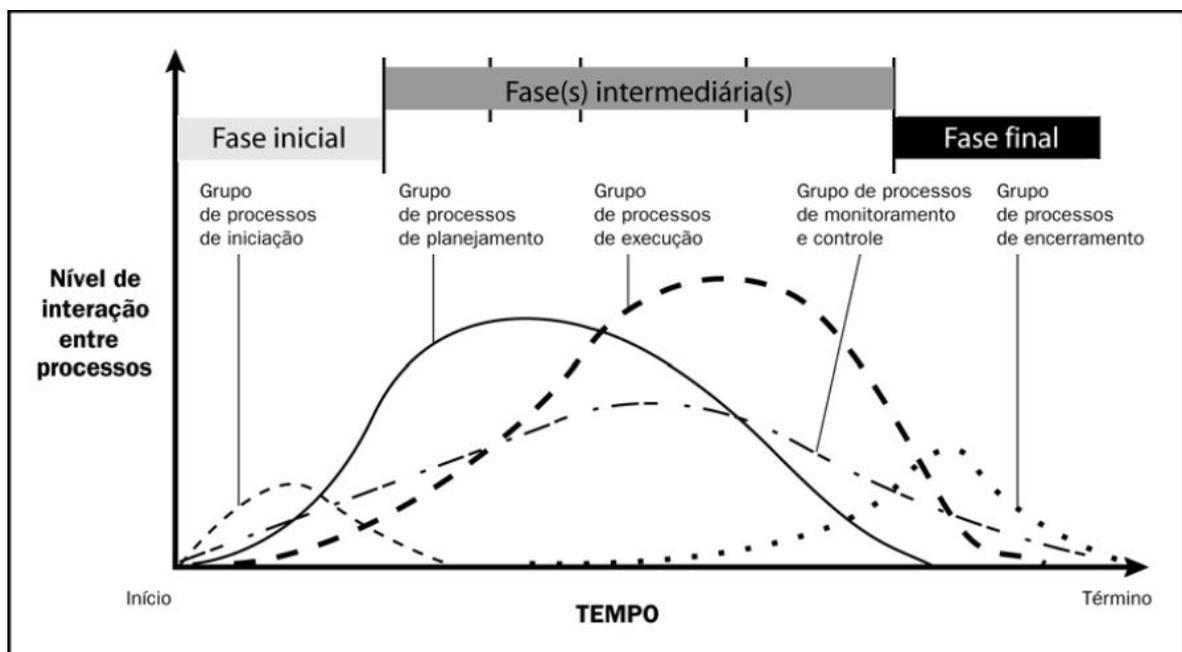


Figura 8 – Interação entre processos de gerenciamento de projeto (adaptado de PMI, 2004)

O PMI (2004) enfatiza ainda a alta influência das partes interessadas (*stakeholders*) nas fases iniciais do projeto, que se reduz até a fase final, porém sem nunca deixar de existir. O sucesso de um projeto depende muito da comunicação e envolvimento com todos os interessados, do início ao fim, sendo especialmente importante no início, onde são definidas as características fundamentais do produto, que definirão todo o projeto.

3.8.2 Áreas de Conhecimento em Gerenciamento de Projetos

Todos os processos de gerenciamento de projetos, que fazem parte dos cinco grupos abordados na seção anterior, estão relacionados a nove áreas de conhecimento (PMI, 2004). Essas áreas de conhecimento servem como uma orientação de quais preocupações as empresas devem ter no momento de adotar melhores práticas para o gerenciamento de seus projetos.

A seguir, uma breve descrição dessas áreas de conhecimento, segundo o PMI (2004):

5. **Gerenciamento de Integração:** diz respeito aos processos, nos diferentes grupos de processo e fases de projeto, que visam identificar, definir, combinar, unificar e coordenar todos os processos e atividades em todo o projeto. Inclui o desenvolvimento do termo de abertura, da declaração preliminar de escopo, do plano de gerenciamento do projeto, gestão da execução e monitoramento, controle de mudanças e encerramento do projeto.

6. **Gerenciamento de Escopo:** diz respeito aos processos e atividades de gerenciamento que irão garantir que o projeto contenha todo o trabalho necessário para atingir seu produto desejado, nem mais, nem menos. Inclui a criação da estrutura analítica do projeto, bem como planejamento, verificação e controle de escopo.

7. **Gerenciamento de Tempo:** engloba os processos necessários para concluir o projeto no prazo determinado, como a definição de atividades e seu seqüenciamento, duração, recursos por atividade e cronograma.

8. **Gerenciamento de Custo:** engloba os processos necessários para concluir o projeto dentro do orçamento aprovado, como por exemplo, planejamento, estimativa, orçamentação e controle de custos.

9. **Gerenciamento de Qualidade:** são os processos necessários para se obter, ao final do projeto, um produto com a qualidade esperada pelo cliente e partes interessadas. Envolve definição de políticas da qualidade, garantia da qualidade e controle total da mesma, com atividades de melhoria contínua dos processos.

10. **Gerenciamento de Recursos humanos:** diz respeito aos processos que organizam e gerenciam a equipe do projeto, incluindo planejamento, contratações e desenvolvimento de pessoal.

11. **Gerenciamento das Comunicações:** processos com o objetivo de garantir a geração, distribuição, armazenamento, recuperação e destinação final de toda informação relacionada ao projeto.

12. **Gerenciamento de Riscos:** identificar, analisar, agir, monitorar e planejar o gerenciamento de riscos (evento ou condição incerta que pode afetar, positiva ou negativamente, o projeto).

13. **Gerenciamento de Aquisições:** são os processos necessários para a aquisição de produtos que precisam ser obtidos fora da equipe de projeto, incluindo planejamento de compras, seleção de fornecedores e administração de contratos.

3.8.3 O Gerenciamento de Projetos de Software

Os modelos de processo de desenvolvimento de software, até o momento explicitados, têm algumas características em comum entre si e com a disciplina de gerenciamento de projetos.

O próprio modelo inicial em cascata apresenta uma semelhança com as fases tradicionais do gerenciamento de projetos descritas no *Project Management Body of Knowledge* – PMBOK (PMI, 2004): iniciação, planejamento, execução, monitoramento e controle e encerramento. As fases iniciais de engenharia de sistemas e análise/especificação de requisitos do modelo cascata assemelham-se às duas fases iniciais do modelo PMBOK. As fases de projeto, codificação e testes no modelo cascata assemelham-se à execução e monitoramento/controle no PMBOK. As fases finais de liberação e manutenção do modelo cascata correspondem, no PMBOK, ao processo de encerramento do projeto.

No entanto, uma das áreas de conhecimento do gerenciamento de projetos do guia PMBOK, o gerenciamento de riscos, só surge com destaque a partir do modelo espiral, que prevê fases de análise de riscos em cada ciclo de desenvolvimento que ajudarão na decisão de continuidade do desenvolvimento e na eliminação de riscos negativos ao projeto.

De fato, os modelos posteriores ao espiral passam a adotar a melhor prática de gerenciamento de riscos, como por exemplo o RUP. A disciplina de gerência de projetos, integrante da dimensão vertical do RUP, é onde se encontram as atividades de desenvolvimento do plano de gestão dos riscos, identificação e avaliação de riscos, que são atribuições do gerente de projeto.

Porém, assim como exemplificado para o gerenciamento de riscos, as diversas áreas de conhecimento para gerenciamento são descritas no PMBOK com maior detalhamento do que visto nos modelos de desenvolvimento de software. Para Rocha e Belchior (2004), o PMBOK oferece detalhamento de descrições das entradas, ferramentas e técnicas sugeridas e saídas, uma vez que seu foco é específico no gerenciamento de projetos. Entretanto, o autor cita a desvantagem do guia PMBOK não ser um modelo específico para software.

Ressalta-se aqui, dessa forma, que os modelos para processo de desenvolvimento de software devem estar aliados a práticas de referência em gerenciamento de projetos descritas pelo guia PMBOK, dada a importância do gerenciamento para o sucesso dos projetos de software, englobando gerenciamento de integração, escopo, tempo, custo, qualidade, recursos humanos, comunicação, riscos e aquisições.

Pressman (1995) concorda com a importância das várias etapas de gerenciamento de projetos para o desenvolvimento de software. O sucesso, segundo o autor, está diretamente relacionado a diversas preocupações como: a compreensão do escopo do trabalho (projeto e produto final); o gerenciamento dos riscos envolvidos, fazendo sua análise e acompanhamento de impactos sobre custo e prazo; a estimativa e gerenciamento dos recursos necessários, incluindo hardware, software e pessoal; o planejamento das atividades por meio de redes PERT/CPM e cronogramas, com marcos de referência para acompanhamento; o esforço despendido, forma pela qual o autor refere-se à importância do gerenciamento de custos.

As tarefas de planejamento do projeto devem gerar um Plano de Projeto de Software, que fornece informações multidisciplinares básicas sobre todas as áreas de conhecimento nas diversas fases do projeto, auxiliando na definição e controle dos resultados de cada etapa do processo, contribuindo para o gerenciamento do projeto de software (PRESSMAN, 1995).

Pereira et al. (2008) analisaram diversos estudos que permitiram enumerar os fatores de sucesso e falhas em projetos: requisitos dos usuários, tempo de entrega, orçamento, satisfação do cliente, metas de negócio, qualidade dos produtos, confiabilidade, manutenibilidade, trabalho em equipe e satisfação da equipe de desenvolvimento e do gerente de projetos. Esses fatores vêm enfatizar a importância das áreas de conhecimento em gerenciamento de projetos para o sucesso no desenvolvimento dos produtos de software, ao lado da preocupação com aspectos da engenharia de software, gestão da qualidade e gestão organizacional.

Os principais pontos da disciplina de gerenciamento de projetos foram considerados numa seção do questionário desenvolvido para o método de *benchmarking* do setor de software. Na seção Gerenciamento de Projetos (ver 4.3.1.3) foram elaborados indicadores para as seguintes práticas e seus resultados: estrutura organizacional da empresa, se voltada a projetos; gerenciamento de riscos, tempo, custo, escopo e comunicação; uso de ferramenta de apoio ao gerenciamento de projetos; e o cumprimento de prazos, custos e funcionamento da comunicação.

3.9 Inovação e Competitividade no Setor de Software

O presente capítulo traz os conceitos de inovação e aptidões estratégicas para o seu alcance com o objetivo inicial de se apresentar a importância da gestão da inovação para a competitividade das empresas. Em especial, trata-se aqui das empresas de pequeno e médio porte, partindo de uma abordagem geral da inovação nas empresas tradicionais desse porte, passando pelas empresas de base tecnológica e finalizando com as empresas do setor de software. O enfoque em pequeno e médio porte é justificado pelo tamanho das empresas de software brasileiras, predominantemente pequenas e médias (ver seção 3.2), embora as considerações feitas apliquem-se tanto à competitividade de pequenas e médias quanto às grandes empresas.

3.9.1 Inovação

A palavra inovação tem sido amplamente utilizada e difundida no meio acadêmico, empresarial e social de forma geral, sob diversas perspectivas e, sendo assim, com diferentes significados. No entanto, ela remete sempre às mudanças exigidas num ambiente de contínuas novidades, concorrência crescente, necessidades e expectativas mutantes e informações disponíveis de forma cada vez mais rápida, em grande volume e variedade, acessíveis a todas as pessoas.

Schumpeter (1934) já abordava a questão inovação em uma de suas obras mais conhecidas, a Teoria do Desenvolvimento Econômico, escrita em 1912, na qual cita que novas combinações de materiais e forças de produção, que surgem de forma descontínua ao longo do tempo, caracterizam o que o autor chama de desenvolvimento pela “destruição criativa”.

Inovar virou sinônimo de sucesso, uma vez que pesquisas apontam que as empresas que utilizam a inovação para melhorar ou estabelecer diferenças nos seus processos e produtos, demonstram um desempenho superior quando comparado aos seus concorrentes, no que se refere à participação de mercado, lucratividade e crescimento (TIDD et al., 2001 apud SPALDING, 2006).

Tornatzky e Fleischer (1990) recorrem à origem latina na palavra *novus*, citando que inovação possui um grande número de significados relacionados, sendo muitas vezes alternativamente definida como “a introdução de algo novo” ou “uma nova idéia, método ou dispositivo”. Os autores citam que o conceito é situacional, exigindo considerações sobre o quanto novo é novo para ser considerado inovação, ou por quanto tempo de uso ou de conhecimento algo permanece inovador. Resumem, ao final, o termo “inovação tecnológica” como sendo a introdução e desenvolvimento, situacionais, de ferramentas, artefatos e dispositivos baseados em conhecimento, pelos quais as pessoas expandem e interagem com seu ambiente.

A definição de inovação adotada no Manual de Oslo pela OECD (2005, p. 46), por sua vez, mostra-se mais abrangente: “é a implementação de produtos (bens ou serviços) ou processos novos ou significativamente melhorados, novos métodos de *marketing* ou novos métodos organizacionais como práticas de negócio, organização da área de trabalho ou relacionamentos externos”.

Esse conceito vai ao encontro das áreas organizacionais propícias a mudanças, apresentadas por Mañas (2001) apud Marochi (2003), que são o empreendimento (mudanças no produto, serviço, mercado ou negócios), a estrutura (realocação de pessoal, hierarquias, unidades de negócios), a tecnologia (mudanças em processos, equipamentos) e o comportamento (mudanças de atitudes e habilidades das pessoas).

3.9.2 Aptidões Estratégicas para a Inovação

Conforme já comentado anteriormente, o processo de inovação ocorre pela geração de novos produtos, processos, estratégias de marketing, formas organizacionais ou de gestão, a partir da utilização de uma matéria-prima essencial: o conhecimento. Desta forma, boa parte do sucesso na implementação de um processo de gestão da inovação recai sobre a gestão do conhecimento e informação, sobre a criação de um ambiente instigador da troca sistematizada

de informações, seja entre o ambiente interno e o externo, ou entre áreas, grupos e pessoas dentro do ambiente interno à organização.

Devido à importância, para o processo de inovação, da obtenção do conhecimento, sua disseminação e multiplicação, gerando novos conhecimentos aplicados à introdução de novos produtos, processos e outras inovações para as empresas, é de enorme contribuição explorar os conceitos de aptidões estratégicas introduzidos por Leonard-Barton (1998). Segundo a autora, as aptidões estratégicas são obtidas pela criação e utilização de conhecimentos que trarão um diferencial, uma vantagem competitiva permanente para a empresa frente a seus concorrentes.

As aptidões estratégicas, por seu caráter diferencial e resultante de uma evolução de conhecimento gradual e de longo prazo, não podem ser imitadas pelos concorrentes, diferentemente das aptidões suplementares, que adicionam valor às aptidões estratégicas (como por exemplo canais de distribuição), e das aptidões habilitadoras, que são as aptidões mínimas necessárias para se concorrer num mercado, mas que não garantem vantagem (LEONARD-BARTON, 1998).

As aptidões estratégicas ocorrem em quatro dimensões numa empresa (LEONARD-BARTON, 1998):

- conhecimento e qualificações dos colaboradores (capital humano);
- sistemas técnicos físicos (bancos de dados, *softwares*, equipamentos): competência técnica acumulada ao longo do tempo;
- sistemas de gestão (sistemas de instrução, remuneração e incentivo): criam os canais por onde o conhecimento flui e é acessado;
- valores e normas (cultura organizacional): determinam quais conhecimentos devem ser buscados, quais atividades devem ser encorajadas.

Essas dimensões dão uma boa orientação das esferas para as quais as organizações precisam estar atentas com o objetivo de transformar uma empresa numa organização inovadora. O caminho, que passa por todas as esferas, pode começar com a geração de uma cultura organizacional que promova valores e normas que valorizem a abertura, a experimentação, a sugestão e implementação de idéias, o desenvolvimento dos indivíduos e de um ambiente dinâmico. Este ambiente, por sua vez, deve ser instrumentado por um sistema de gestão adequado ao tratamento dos resultados de tal comportamento, com sistemas capazes de tratar

as idéias geradas, patrocinar o desenvolvimento das pessoas e gerenciar a execução dos projetos inovadores. As qualificações e conhecimentos, assim como os sistemas físicos, serão resultado da evolução das dimensões anteriores, onde as pessoas com os valores e normas aceitos pela organização serão responsáveis pela criação das aptidões individuais e coletivas, resultando em resolução de problemas, geração de conhecimento, inovação dos produtos e dos próprios sistemas técnicos físicos, que evoluirão e constituirão um importante patrimônio das aptidões da empresa.

3.9.3 Inovação e Competitividade em PMEs Tradicionais

Porter (1993) afirma que as empresas criam vantagem competitiva através da inovação ao perceberem novas e melhores maneiras de competir numa indústria e levando-as ao mercado. Ainda segundo o autor, as inovações influem na vantagem competitiva quando os rivais não se dão conta da nova maneira de competir, não querem ou não são capazes de reagir a ela.

Normalmente, a estratégia competitiva das empresas de micro e pequeno porte, e em menor grau das empresas de médio porte, está baseada na competição por baixos custos, uma vez que tentam explorar suas vantagens sobre as grandes empresas no que diz respeito a uma estrutura enxuta e com menores custos. Conforme estudo da ANPEI (2004), a maioria das empresas nacionais aponta como importante atividade inovativa a aquisição de máquinas e equipamentos pelo fato de ser fundamental, para as pequenas empresas, a modernização de processos para fabricar a preços competitivos.

A inovação como estratégia competitiva de concorrência por custos, que se manifesta nas PMEs de forma mais freqüente pela aquisição de equipamentos, apesar de conferir à empresa, em maior ou menor grau, um aprendizado na tecnologia de fabricação, não caracteriza uma fonte de vantagem competitiva duradoura e difícil de copiar, uma vez que as novas máquinas e equipamentos também estarão à disposição da concorrência.

Para isso, Porter (1993) cita as vantagens competitivas de ordem superior, mais duráveis, baseadas em tecnologias de processos protegidas por direito de propriedade, diferenciação baseada em produtos ou serviços excepcionais, a reputação ou imagem da marca e as relações com os clientes.

Essas inovações, no entanto, enfrentam barreiras para a implantação nas PMEs devido a dificuldades financeiras na obtenção de recursos, aversão aos riscos inerentes à inovação,

dificuldades gerenciais e de capital intelectual, falta de estratégia e rede de relacionamentos. No entanto, uma das características das PMEs, a de proximidade com o cliente, pode ser explorada como vantagem para a adoção de tais inovações em relação a empresas de maior porte.

Deitos (2002), com base em pesquisas de diversos autores, apresenta um quadro com as vantagens e desvantagens das pequenas e médias empresas em relação à inovação, reproduzida no Quadro 2 a seguir.

	Vantagens	Desvantagens
Organização e Gestão	Ausência de burocracia Dinamismo Estrutura flexível Comunicação interna informal e rápida	Dificuldade de acesso a fontes de informação externa
Recursos humanos	Adaptabilidade	Escassez de pessoal altamente qualificado
Mercado	Capacidade de reação Habilidade de penetração em nichos de mercado	Baixa presença em mercados internacionais
Financiamento		Escassez de recursos Dificuldade de acesso ao capital de risco Dependência de crédito bancário
Fonte: Deitos (2002, p. 38).		

Quadro 2 – Inovação: vantagens e desvantagens das pequenas e médias empresas.

A inovação, como se vê, é a força motriz da competitividade das empresas, inclusive para aquelas de pequeno e médio porte, que precisam buscar o desenvolvimento de certos fatores que, aliados a algumas de suas vantagens estruturais em relação a grandes empresas, tendem a conferir vantagem para a sobrevivência no longo prazo.

3.9.4 A Inovação em PMEs de Base Tecnológica

As barreiras enfrentadas pelas PMEs tradicionais na adoção e implantação da inovação diferem um pouco da realidade existente em PMEBTs devido a suas características. As dificuldades de escassez de capital intelectual, por exemplo, comum nas empresas tradicionais, não aparece, em princípio, de forma muito restritiva nas empresas de base tecnológica, intensivas em capital intelectual. Seguindo a mesma linha, normalmente são empresas com maior grau de gerenciamento e com estratégias possivelmente mais bem definidas, com investimentos reservados para P&D em volumes consideravelmente maiores que aqueles destinados nas empresas tradicionais, embora a disponibilidade de recursos ainda se manifeste como um limitante.

Além disso, as PMEBTs mantêm vantagens típicas de pequenas e médias empresas tradicionais relativas ao porte, como a de proximidade com o cliente e flexibilidade, que garantem vantagens extras para as PMEBTs na adoção e implementação de tais inovações em relação a empresas de maior porte.

Deitos (2002) enfatiza que tanto as PMEs tradicionais quanto as de base tecnológica dependem da existência de um processo de capacitação científica e tecnológica que amplie o potencial de inovação das mesmas. No entanto, cita que nas “empresas de alta tecnologia”, que possuem a inovação tecnológica como seu objeto de negócio, as “inovações radicais constituem um tema relevante, pois estas empresas, atuando em segmentos industriais emergentes, têm o potencial de deflagrar processos importantes de inovação radical.” (KRUGLIANSKAS, 1996 apud DEITOS, 2002, p. 36). As inovações radicais exigem um esforço muito maior em pesquisa e desenvolvimento (P&D), destacando a importância destas atividades para as PMEBTs em relação às empresas tradicionais de pequeno e médio porte.

3.9.5 Inovação em Empresas de Software

As empresas de base tecnológica da indústria de software também compartilham das mesmas características descritas na seção anterior. Sua vantagem competitiva geralmente não está baseada numa estratégia de custos reduzidos, como ocorre nas pequenas empresas tradicionais, mas sim, em vantagens competitivas nomeadas por Porter (1993) como de ordem superior. Segundo o autor, as vantagens competitivas de ordem superior são de longo prazo e baseadas em tecnologias de processos protegidas por direito de propriedade, diferenciação

baseada em produtos ou serviços excepcionais, em reputação ou imagem da marca e em relações com os clientes.

A obtenção dessas vantagens competitivas exige das empresas conhecimentos e capacidades mais avançados, como pessoal treinado e especializado, capacidade técnica interna e relações estreitas com os clientes. Essas vantagens geralmente dependem de um histórico de investimento constante e cumulativo em instalações físicas, aprendizado, Pesquisa e Desenvolvimento (P&D) e comercialização (PORTER, 1993).

A necessidade do investimento constante e cumulativo é especialmente importante na indústria de software, não focado em instalações físicas, mas sim na introdução contínua de inovações nos produtos, sejam elas incrementais ou radicais, dependendo do seu setor de atuação.

Ahmed e Capretz (2007) citam como fator-chave para empresas de linha de produto de software o estreito relacionamento de trabalho com os clientes, que permite que os desenvolvedores melhorem o desempenho e funcionalidades dos produtos sucessivamente, a cada novo desenvolvimento.

Duarte (2003) explica que a inovação contínua é necessária primeiro pela característica de bem durável verificada em muitos produtos de software, sendo que as inovações criam demandas nos antigos clientes, além de conquistar novos; segundo, é um setor que está intimamente ligado à evolução do setor de informática, onde as rápidas mudanças tecnológicas exigem novos sistemas e aplicativos; terceiro, a característica de intangibilidade do software, eliminando seu custo de reprodução, facilita a cópia de inovações ou a pirataria por parte de empresas concorrentes.

A prática de registro de direitos autorais contra a cópia de novos produtos no setor de software, muitas vezes mostra-se pouco eficiente devido à facilidade em se trabalhar rotinas, sejam novas ou combinações de existentes, que permitem que empresas simplesmente copiem funcionalidades de produtos de empresas concorrentes. Essa dificuldade de apropriação do desenvolvimento tecnológico exige estratégia de inovação contínua, fazendo com que as empresas inovadoras do setor estejam sempre na frente (VELOSO, 2002 apud DUARTE, 2003).

A capacidade técnica interna, através de pessoal treinado e especializado, também confere vantagens competitivas de ordem superior. Corresponde às dimensões de conhecimento e de sistemas técnicos físicos, defendidos por Leonard-Barton (1998), como conferentes de

aptidões estratégicas (ver seção 3.9.2). São características obrigatórias às empresas de base tecnológica, sobretudo as de software, na obtenção de vantagens competitivas por meio da inovação.

Neste sentido, o desenvolvimento de pessoal (capacitação dos recursos humanos) e relacionamento com agentes externos à empresa (parcerias com universidades, centros de pesquisa, empresas de hardware ou até mesmo com clientes), constituem práticas fundamentais para a geração e transferência de conhecimento, tanto tácito quanto explícito, que irão resultar em novos produtos e processos. Essas práticas conferem o aprendizado, defendido por Porter, para a obtenção de vantagens de ordem superior.

Internamente à empresa torna-se fundamental um ambiente inovativo, ou seja, com uma cultura de abertura e encorajamento a idéias inovadoras, com um sistema que permite a comunicação e troca de conhecimento entre as diferentes áreas da empresa, com sistema de recompensa e apoio (inclusive financeiro) às inovações de produto e processo. Isto envolve a preocupação com as áreas organizacionais citadas por Mañas (2001) apud Marochi (2003) na seção 3.9.1, buscando inovações no comportamento, estrutura organizacional, empreendimento e tecnologia.

A importância das atividades de P&D para a inovação defendidas por Porter (1993) como fonte de vantagens competitivas de longo prazo, exige que se discuta a especificidade do papel da pesquisa e desenvolvimento nas empresas da indústria de software. Seu processo de P&D se confunde com o próprio processo de produção dos seus produtos, ou seja, seu processo de desenvolvimento, já que a reprodução deste produto com características intangíveis exige pouco esforço das empresas. Tal característica dificulta inclusive a adoção da medida de investimentos em P&D como indicador de comparação de inovação entre empresas do setor de software.

Porém, nem todo desenvolvimento de software pode ser classificado como uma atividade de P&D. A OECD (2005) classifica como atividade de P&D todo trabalho criativo, realizado de forma sistemática, ou uso do conhecimento para gerar novas aplicações, sendo que aquelas atividades de desenvolvimento com menor grau de novidade não são classificadas como pesquisa e desenvolvimento. Esse conceito é complementado pela pesquisa PINTEC 2003 (BRASIL, 2005b), que cita que o desenvolvimento de software caracteriza-se como atividade interna de P&D somente quando envolve um avanço tecnológico ou científico.

Pode-se concluir que, embora as empresas de software dependam basicamente de grandes investimentos no processo de desenvolvimento de seus produtos, nem sempre estes investimentos constituem uma atividade de P&D. O processo de desenvolvimento, quando focado na melhoria incremental de produtos já existentes, não garante a geração de vantagem competitiva de ordem superior, aproximando a empresa de seus concorrentes pela falta de diferenciais, salvo características específicas dos mercados.

Duarte (2003) explora ainda uma diferença verificada entre empresas de software atuantes no mercado horizontal e vertical. As empresas que atuam no mercado horizontal (desenvolvedoras de software generalizado) têm maior dependência de atividades de P&D sobre a área de informática, enquanto as que atuam no mercado vertical (aplicações específicas para determinados setores) dependem muito mais do conhecimento no ramo específico de seus clientes, sendo que os usuários clientes são fundamentais no fornecimento de informações para o desenvolvimento de soluções customizadas.

Esses pontos vêm enfatizar a importância, para a gestão da inovação no setor de software, das seguintes melhores práticas geradoras de vantagem competitiva de ordem superior: investimentos na formação do capital intelectual através do desenvolvimento de pessoal; manutenção de um ambiente inovativo; investimentos no sistema técnico, com uma política explícita de busca de novas tecnologias, sejam em produtos ou em processo (método) produtivo; ampliação de relacionamento externo à empresa, promovendo geração de conhecimento e conseqüentemente de inovação por meio de parcerias com entidades, fornecedores ou clientes (este último fundamental para o segmento vertical); estratégias e planejamento para introdução de inovações significativas e radicais, de forma sistemática, viabilizadas por atividades de pesquisa e desenvolvimento.

4 Método de *Benchmarking* da Indústria de Software

Conforme apresentado na seção 2.2.3, o método de *benchmarking* para a indústria de software, resultado da presente pesquisa, teve sua estrutura fundamentalmente baseada nos modelos do Benchstar, devido à característica de pequeno porte da maioria das empresas de software brasileiras, bem como no modelo do *Benchmarking* Industrial, naqueles aspectos não avaliados pelo Benchstar.

Os modelos do Benchstar e *Benchmarking* Industrial baseiam-se centralmente no conceito de Sistema Produtivo Classe Mundial, que utiliza a definição ampla de sistema produtivo, segundo o qual a administração da produção abrange áreas como recursos humanos, gestão de fornecedores e desenvolvimento de produtos (SEIBEL, 2004; MAZO, 2003). Desta forma, os modelos desses métodos preocupam-se tanto com a perspectiva técnica do processo produtivo, quanto com a perspectiva gerencial e estratégica de negócios vista na gestão da qualidade, na gestão da organização e cultura, na gestão da inovação e desenvolvimento de produtos.

Adotou-se para o modelo de avaliação do *benchmarking* da indústria de software o mesmo conceito amplo, segundo o qual, para que o desempenho de excelência de toda a organização seja obtido, deve-se procurar a adoção de melhores práticas em todos os critérios de gestão, não restritos à produção. Sob essa perspectiva, as melhores práticas foram pesquisadas acerca dos tópicos vistos na revisão bibliográfica dos capítulos anteriores, englobando questões técnicas de software há muito discutidas no meio acadêmico-empresarial do setor, como a engenharia de software, bem como englobando a gestão de seus projetos de desenvolvimento, a qualidade de software, a gestão da inovação e a gestão organizacional.

4.1 Modelo de avaliação

O modelo de avaliação adotado é baseado principalmente no método de *benchmarking* para pequenas empresas, chamado Benchstar, bem como no método chamado *Benchmarking* Industrial, utilizados pelo IEL/SC, conforme apresentado na seção 2.2.3.

As áreas de avaliação que formam o modelo do Benchstar para a indústria tradicional são: sistema de produção, gestão da produção, gestão da qualidade e gestão organizacional. No entanto, as empresas-alvo desses métodos são pertencentes à indústria tradicional de

transformação, ou seja, com desenvolvimento e produção caracterizados por processos distintos de criação e reprodução física de produtos tangíveis.

Já o produto de software é caracterizado pela intangibilidade, exceto pelas mídias físicas através das quais os mesmos são reproduzidos e distribuídos eventualmente (como CDs, DVDs ou memórias *flash* portáteis), havendo, no entanto, outros meios como a internet que dispensam as mídias. Há ainda, de forma particular, os softwares embarcados, que são instalados em equipamentos (como celulares, tocadores de mp3, televisores, eletrodomésticos, equipamentos de telecomunicações etc.). O presente método de *benchmarking*, entretanto, foca a produção do software, sendo que empresas que produzem software embarcado podem aplicar o método para avaliar a parte de software, possuindo ainda a possibilidade de aplicarem a parte de fábrica do Benchstar tradicional (ou do *Benchmarking* Industrial) para avaliarem a produção de seu *hardware*.

Devido a essas diferenças, as áreas de avaliação do Benchstar e *Benchmarking* Industrial precisaram ser adaptadas para as empresas de software. No Benchstar, as áreas Sistema de Produção e Gestão da Produção dão lugar à área de *Gestão da Produção*, que passa a conter os indicadores sobre engenharia de software. No *Benchmarking* Industrial, a área de Desenvolvimento de Novos Produtos diz respeito, na indústria de software, ao próprio processo produtivo, sendo que seus indicadores estarão abordados na nova área de *Gestão da Produção*, citada anteriormente e que focará os aspectos técnicos da engenharia de software, e em outra área chamada *Gestão de Projetos*, que contém indicadores que focarão o gerenciamento dos projetos de desenvolvimento.

As áreas não citadas acima (*Gestão Organizacional*, *Gestão da Inovação* e *Gestão da Qualidade*) permanecem no questionário, sofrendo revisões em alguns indicadores específicos que as formam com o objetivo de melhor se adequar à realidade das empresas de software.

<i>Benchmarking</i> Software	Benchstar	<i>Benchmarking</i> Industrial
Gestão da Produção	Gestão da Produção Sistemas de Produção	-
Gestão da Produção Gestão de Projetos	-	Desenvolvimento de Novos Produtos
Gestão da Inovação	-	Gestão da Inovação
Gestão da Qualidade	Gestão da Qualidade	-
Gestão Organizacional	Gestão Organizacional	-

Quadro 3 – Áreas de avaliação do modelo e sua origem no Benchstar e *Benchmarking* Industrial

O Quadro 3 apresenta as áreas de avaliação do método de *Benchmarking* para Empresas de Software e as áreas dos métodos *Benchstar* e *Benchmarking Industrial* que as originaram.

A Figura 9 apresenta o novo modelo de avaliação do *Benchmarking* para Empresas de Software.

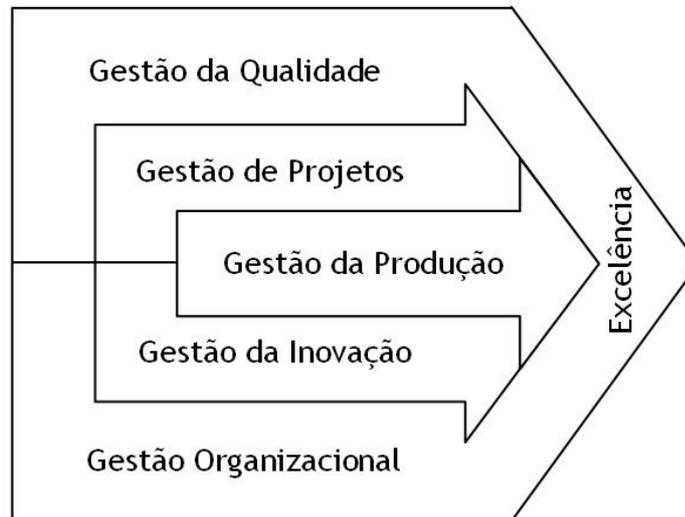


Figura 9 – Modelo de avaliação do *benchmarking* em empresas de software (adaptado de IEL/SC, 2005b)

O modelo apresenta a gestão da produção no centro, representando a aplicação da engenharia de software no dia-a-dia da produção de produtos de software na empresa, ou seja, trata-se do processo de desenvolvimento de novos produtos de software em si, devido às características de intangibilidade do setor.

No seu entorno, e influenciando diretamente a gestão da produção diária de software, apresentam-se a gestão de projetos e gestão da inovação, que devem ser preocupações além daquelas diárias de curto prazo, mas representam as práticas que devem guiar a empresa com uma visão mais de médio e longo prazo: não adianta a empresa aplicar muito bem as técnicas de engenharia de software no dia-a-dia se o escopo do projeto estiver errado, ou se o produto final estiver ultrapassado.

Como última esfera, a gestão organizacional deve formar uma base sólida para que o restante da empresa funcione bem e de forma sistemática e duradoura, com a mesma organizada e com pessoas capacitadas e dispostas a buscar os objetivos da empresa. Além disso, a gestão da qualidade forma um “guarda-chuva” que envolve todas as atividades da empresa, representando a gestão da qualidade total.

Cada uma dessas áreas de avaliação inclui as melhores práticas que, quando adotadas, devem em tese levar aos resultados, ou seja, a uma *performance* proporcional ao nível de práticas adotadas. As setas do modelo da Figura 9, por fim, representam que a adoção de todas as práticas nas áreas de gestão levam ao alcance de boa *performance* em cada área, que levam, conseqüentemente, ao alcance da excelência, ou seja, uma posição de liderança ou *benchmark* em seu mercado.

4.2 Processo de Aplicação

O método de *benchmarking* para a indústria de software foi desenvolvido de modo a permitir o mesmo processo geral de aplicação, salvo questões particulares de duração, adotado nos métodos Benchstar e *Benchmarking* Industrial. Como será visto na seção 4.3, o processo de *benchmarking* utiliza-se de um questionário como ferramenta principal, que contém os indicadores que serão avaliados na empresa.

A etapa inicial, realizada já após a concordância da empresa em realizar o *benchmarking*, corresponde ao envio das orientações e do questionário à empresa a ser avaliada, que monta uma equipe interna de *benchmarking* composta por gerentes e colaboradores de diferentes áreas da empresa. A equipe responde previamente todo o questionário como forma de auto-avaliação de preparação para a reunião de consenso.

Para a reunião de consenso, um facilitador visita a empresa e, juntamente com a equipe interna de *benchmarking*, revisa a pontuação e compreensão dos indicadores respondidos pela equipe interna. A exemplo de como ocorre no Benchstar, a reunião de consenso possui uma estimativa de duração de 4 horas.

A pontuação consensada dos indicadores gera então os índices percentuais de práticas e *performance* em cada área de avaliação, e estes geram os índices gerais de práticas e *performance*. Uma descrição mais detalhada dos cálculos é apresentada na seção 5.1.3, uma vez que foram melhor definidos na etapa de aplicação-piloto do método. Todos os resultados são apresentados à empresa avaliada por meio de um relatório de resultados de *benchmarking*.

A seqüência de etapas de aplicação do processo de *Benchmarking* para Empresas de Software está representada pelo esquema da Figura 10.

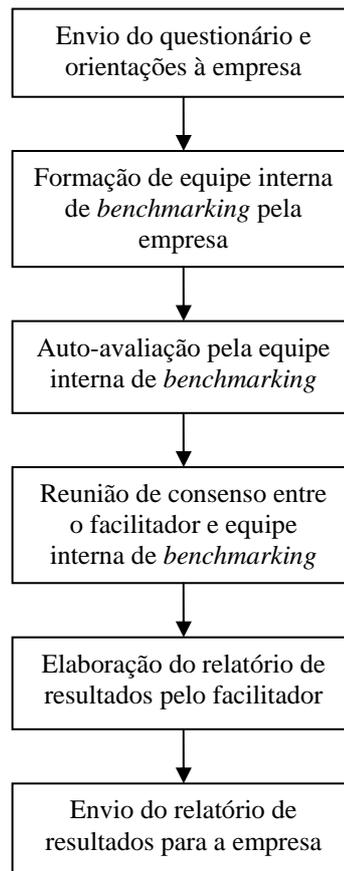


Figura 10 – Etapas de aplicação do *Benchmarking* para Empresas de Software

O relatório de resultados de *benchmarking* é enviado à empresa por correio eletrônico, sendo que o mesmo consiste na apresentação gráfica dos índices num gráfico radar, suas descrições na tabela dos índices das áreas avaliadas e na tabela de pontuação dos indicadores, além dos comentários e análises elaborados pelo facilitador.

4.3 Questionário e Indicadores

A ferramenta para o método de *benchmarking* desenvolvido no presente trabalho consiste basicamente num questionário. O mesmo é composto por perguntas iniciais, com o objetivo de identificar e caracterizar a empresa estudada, e por indicadores, com o objetivo de medir em qual nível as empresas adotam as melhores práticas para o setor de software e quais os resultados (*performance*) obtidos com a adoção das práticas.

Os indicadores de práticas e *performance* que compõem a ferramenta de *benchmarking* são apresentados num questionário dividido em seções distintas das áreas de avaliação do modelo da Figura 9, o que contribui para que os respondentes das empresas não vinculem suas respostas e pontuações do questionário diretamente aos índices das áreas de avaliação final, evitando vieses nas respostas dadas.

O desenvolvimento dos indicadores baseou-se nos princípios encontrados nos diversos modelos, normas e trabalhos científicos, apresentados nos capítulos anteriores. A seguir, um quadro resume os princípios que foram mapeados, suas fontes e os indicadores da ferramenta de *benchmarking* a eles relacionados.

Princípios	Fonte	Indicadores
Modelo de processo de desenvolvimento de software	Anacleto (2004) Ferreira (2002) Garcia (2003) IEE (2004) Kruchten (2001) NBR ISO/IEC 12207 (1998)	Modelo de Processo de Desenvolvimento
Processo iterativo de desenvolvimento de software	Ambler (2005) Ferreira (2002) Kruchten (2001) Luiz (2006) Reis (2001)	Iteratividade no Desenvolvimento Envolvimento externo*
Análise essencial e estruturada Modelagem unificada Linguagem estruturada e orientada a objetos Elevação do nível de abstração	Ambler (2005) Garcia (2003) Kruchten (2001) Martins e Júnior (1999) Pressman (1995) Rezende (1999)	Análise e Modelagem de Arquitetura de Software
Análise de requisitos Engenharia de requisitos	NBR ISO/IEC 9126 (2003) NBR ISO/IEC 12207 (1998) Martins (2001) Rodrigues (2006)	Engenharia de Requisitos Orientação para o Cliente*** Geração de conceitos de produtos inovadores*

Princípios	Fonte	Indicadores
Testes de Software	Ferreira (2002) Garcia (2003) Kruchten (2001) NBR ISO/IEC 12119 (1998) NBR ISO/IEC 12207 (1998) Pressman (1995)	Testes de Software
Implantação de software	Kruchten (2001)	Implantação do Produto no Cliente
Suporte e manutenção de software	Ambler (2007) NBR ISO/IEC 12207 (1998)	Suporte e Manutenção <i>Feedback</i> de Campo*
Maturidade dos processos de software	ISO/IEC 15504 (2004) Nakajima (2004) Quintella e Rocha (2006) SEI (2002)	Maturidade dos Processos Tempo de ciclo de desenvolvimento **
Ferramentas de Apoio à Engenharia de Software (CASE)	ISO/IEC TR 14471 (1999) ISO/IEC 14102 (1995) Pressman (1995)	Ferramentas de Apoio à Engenharia de Software (CASE)
Qualidade de produto de software	ISO/IEC 14598 (1999) NBR ISO/IEC 9126 (2003) Pressman (1995)	Funcionalidade dos Produtos Confiabilidade Usabilidade Eficiência Manutenibilidade Métricas de Software Reclamações de Clientes***
Documentação de software	NBR ISO/IEC 12119 (1998) NBR ISO/IEC 12207 (1998)	Documentação dos Produtos
Medição e métricas de Software	IEEE (2004) Marini (2002) NBR ISO/IEC 9126 (2003) Pressman (1995)	Métricas de Software Medidas de Desempenho ****

Princípios	Fonte	Indicadores
Gerenciamento de projetos	Ferreira (2002) NBR ISO/IEC 12207 (1998) PMI (2004) Pressman (1995) Rocha e Belchior (2004)	Planejamento de Projetos Organização para Projetos Gerenciamento de Riscos Gerenciamento de Tempo Gerenciamento de Custos Gerenciamento de Escopo Gerenciamento de Comunicação Ferramenta de Apoio ao Gerenciamento de Projeto** Cumprimento de Prazos Cumprimento de Custos Funcionamento da Comunicação Controle de Mudanças*
Gerenciamento de Níveis de Serviço	Gomes, Falbo e Menezes (2005); Santos (2006)	Gerenciamento do Nível de Serviço
Aptidões estratégicas	Leonard-Barton (1998)	Maturidade dos Processos Modelo de Processo de Desenvolvimento Desenvolvimento de pessoal**** Estilo de administração*** Ambiente inovativo** Estratégia de tecnologia de produto e processo***
Estratégia de inovação e vantagem competitiva	Burgelman (2004) Deitos (2002) Porter (1993) Tornatzky e Fleicher (1990)	Estratégia de tecnologia de produto e processo*** Relação com universidades e centros de pesquisa* Estratégia de obtenção de novas tecnologias* Geração de conceitos de produtos inovadores* Desenvolvimento de pessoal**** Introdução de novos produtos**
Legenda: (*) Indicador do <i>Benchmarking</i> Industrial sem adaptação (**) Indicador do <i>Benchmarking</i> Industrial com adaptação (***) Indicador do <i>Benchstar</i> sem adaptação (****) Indicador do <i>Benchstar</i> com adaptação		

Quadro 4 – Indicadores, princípios pesquisados e suas fontes

Os indicadores do questionário são de dois tipos diferentes: indicadores de práticas e indicadores de *performance*. Os de práticas representam o conjunto de ferramentas gerenciais e tecnológicas implantadas na empresa, como por exemplo, ferramentas CASE ou modelagem de software. Indicadores de *performance* avaliam os resultados mensuráveis obtidos pela empresa, como por exemplo o cumprimento de prazos de projetos, funcionalidade de produtos ou satisfação dos clientes. A distinção entre esses dois tipos de indicador é feita somente no relatório final de resultados, sendo que no questionário os tipos não são discriminados nem separados, estando os indicadores apenas organizados em seções.

Para a organização do questionário em seções, utilizaram-se como base as seções existentes nos modelos do Benchstar e *Benchmarking* Industrial, de acordo com a aplicabilidade ao setor de software. A seguir, apresenta-se um quadro que mostra as seções definidas e seus escopos, que orientaram a distribuição dos indicadores no questionário do método desenvolvido.

Seção do questionário	Escopo	Indicadores
Gestão Organizacional	Ferramentas e práticas de gestão para orientar os recursos da empresa em direção aos objetivos estratégicos e posicionamento no mercado. Origem no Benchstar.	Declaração e compartilhamento da visão (GO1) Planejamento e análise das metas (GO2) Estilo de administração (GO3) Participação dos empregados (GO4) Política de incentivos (GO5) Desenvolvimento de pessoal (GO6) Orientação para o cliente (GO7) Rotatividade da mão-de-obra (GO8) Conhecimento da concorrência (GO9) Planejamento de marketing (GO10) Sistemas de informação (GO11)
Gestão da Qualidade	Ferramentas e práticas de gestão para garantir e monitorar a qualidade em todo o processo. Origem no Benchstar.	Visão da qualidade (GQ1) Qualidade produtos entregues subcontratados (GQ2) Reclamação de clientes (GQ3) Fornecedores e subcontratados (GQ4) Pesquisa das necessidades dos clientes (GQ5)
Avaliação de Desempenho	Monitoramento estratégico dos resultados físico-financeiros. Origem no Benchstar.	Satisfação do cliente (AD1) Variação nas vendas (AD2) Medidas de desempenho (AD3) Lucratividade (AD4) Prazo de entrega dos subcontratados (AD5)

Seção do questionário	Escopo	Indicadores
Inovação	Práticas e <i>performance</i> da gestão estratégica da inovação, gerando conhecimento e transformando-o em novas aplicações. Origem: <i>Benchmarking Industrial</i> .	Geração de conceitos de produtos inovadores (IN1) Planejamento do ciclo de vida do produto (IN2) Capacidade de inovação (IN3) Introdução de novos produtos (IN4) Relacionamento universidades e centros pesquisa (IN5) Estratégia de obtenção novas tecnologias (IN6) Estratégia de tecnologia de Produto e Processo (IN7) Ambiente inovativo (IN8)
Desenvolvimento de Produtos	Ferramentas e práticas para a implementação da inovação no processo produtivo. Origem: <i>Benchmarking Industrial</i> .	Simultaneidade no processo de engenharia (DP1) Envolvimento externo (DP2) Controle de mudanças (DP3) <i>Feedback</i> de campo (DP4) Projeto para produção e manutenção (DP5) Tempo de ciclo de desenvolvimento (DP6)
Engenharia de Software	Ferramentas e práticas de referência para a fabricação de software.	Modelo de Processo de Desenvolvimento (ES1) Iteratividade no Desenvolvimento (ES2) Análise e Modelagem de Arquitetura de Software (ES3) Engenharia de Requisitos (ES4) Testes de Software (ES5) Implantação do Produto no Cliente (ES6) Suporte e Manutenção (ES7) Maturidade dos Processos (ES8) Ferramentas de Apoio à Engenharia de Software (ES9)
Qualidade de Software	Monitoramento da qualidade dos produtos de software.	Funcionalidade dos Produtos (QS1) Confiabilidade (QS2) Usabilidade (QS3) Eficiência (QS4) Manutenibilidade (QS5) Documentação dos Produtos (QS6) Métricas de Software (QS7)
Gerenciamento de Projetos	Práticas e <i>performance</i> da gestão dos projetos de desenvolvimento de software.	Organização para Projetos (GP1) Gerenciamento de Riscos (GP2) Gerenciamento de Tempo (GP3) Gerenciamento de Custos (GP4)

Seção do questionário	Escopo	Indicadores
		Gerenciamento de Escopo (GP5) Gerenciamento de Comunicação (GP6) Ferramenta de Apoio ao Gerenciamento de Projeto (GP7) Gerenciamento do Nível de Serviço (GP8) Cumprimento de Prazos (GP9) Cumprimento de Custos (GP10) Funcionamento da Comunicação (GP11)

Quadro 5 – Organização dos indicadores nas seções do questionário

A Figura 11 mostra a estrutura dos indicadores desenvolvidos, que seguiu a estrutura dos indicadores dos métodos do IEL/SC: uma nota explicativa à esquerda esclarece a questão a ser avaliada, enquanto os três cenários evolutivos permitem a pontuação das empresas numa escala de 1 a 5. As pontuações 2 e 4 correspondem a situações intermediárias, quando não se alcança totalmente as exigências do cenário de pontuação superior. A mesma estrutura é adotada para todos os indicadores do questionário.

Modelo de Processo de Desenvolvimento	1	2	3	4	5
Existem hoje diversos modelos de processo de software que contribuem para a estruturação do método de desenvolvimento adotado, desde o modelo tradicional em cascata (limitado, por ser seqüencial e sem iterações), que foi o precursor e cuja evolução gerou diversos outros modelos (evolutivo, espiral, unificado/EUP/RUP etc). Um modelo de processo de desenvolvimento de software foi escolhido com base nas características e porte dos projetos da empresa?	Não adota um modelo. Trabalho realizado conforme a necessidade diária e experiência da equipe.		São utilizadas partes de modelos, de forma incompleta, com partes do processo ainda não sistematizadas.		Empresa adota um modelo para todo o ciclo de vida do software, que orienta a adoção de métodos, técnicas e melhores práticas de engenharia de software.

Figura 11 - Estrutura dos indicadores desenvolvidos

A seguir, são detalhados e comentados os novos indicadores desenvolvidos a partir do estudo do presente trabalho, que formarão a ferramenta de *benchmarking* para empresas de software ao lado de indicadores adaptados do Benchstar e *Benchmarking Industrial* (ver seção 4.3.2). Os códigos colocados entre parêntesis, ao lado do título dos indicadores mencionados, são os códigos dos indicadores no questionário do *benchmarking* desenvolvido.

4.3.1 Novos Indicadores

Os novos indicadores de práticas e *performance* desenvolvidos para a indústria de software foram divididos em três seções de questionário, que representam as importantes disciplinas para o setor, ainda não abordadas no modelo anterior do Benchstar. São elas: *Engenharia de Software*, *Qualidade de Software* e *Gerenciamento de Projetos*. O quadro a seguir mostra os indicadores que pertencem a cada uma dessas seções do questionário.

Engenharia de Software (ES)
Modelo de Processo de Desenvolvimento (ES1)
Iteratividade no Desenvolvimento (ES2)
Análise e Modelagem de Arquitetura de Software (ES3)
Engenharia de Requisitos (ES4)
Testes de Software (ES5)
Implantação do Produto no Cliente (ES6)
Suporte e Manutenção (ES7)
Maturidade dos Processos (ES8)
Ferramentas de Apoio à Engenharia de Software (CASE) (ES9)
Qualidade de Software (QS)
Funcionalidade dos Produtos (QS1)
Confiabilidade (QS2)
Usabilidade (QS3)
Eficiência (QS4)
Manutenibilidade (QS5)
Documentação dos Produtos (QS6)
Métricas de Software (QS7)
Gerenciamento de Projetos (GP)
Organização para Projetos (GP1)
Gerenciamento de Riscos (GP2)
Gerenciamento de Tempo (GP3)
Gerenciamento de Custos (GP4)
Gerenciamento de Escopo (GP5)
Gerenciamento de Comunicação (GP6)
Ferramenta de Apoio ao Gerenciamento de Projeto (GP7)
Gerenciamento do Nível de Serviço (GP8)
Cumprimento de Prazos (GP9)

Cumprimento de Custos (GP10)
Funcionamento da Comunicação (GP11)

Quadro 6 – Novos indicadores e seções do questionário de *benchmarking*

4.3.1.1 Engenharia de Software

A seção de Engenharia de Software reúne indicadores referentes às melhores práticas relacionadas aos modelos e técnicas para a produção de software, através de levantamento nos diversos modelos, normas e conceitos pesquisados, conforme Quadro 4 apresentado anteriormente.

Essa seção do questionário inicia-se pela avaliação do indicador *Modelo de Processo de Desenvolvimento (ES1)* pelo fato de ser a adoção destes modelos um fator fundamental para a adoção das diversas práticas de engenharia de software. Os modelos de processo de software contribuem para a estruturação do método de desenvolvimento adotado, desde o modelo tradicional em cascata (limitado, por ser seqüencial e sem iterações), que foi o precursor e cuja evolução gerou diversos outros modelos (evolutivo, espiral, unificado/EUP/RUP etc). O cenário de pior pontuação nesse indicador reflete empresas que trabalham de forma empírica e sem sistematização, evoluindo para o melhor cenário no qual a empresa adota um modelo de processo de desenvolvimento para todo o ciclo de vida do software, orientando métodos, técnicas e práticas de engenharia de software. A nota explicativa do indicador lembra ao avaliador que um modelo deve ser escolhido com base nas características e porte dos projetos da empresa.

O indicador seguinte avalia a *Iteratividade no Desenvolvimento (ES2)*, que consiste na liberação de versões antes da conclusão final do software, contribuindo para a avaliação pelo cliente de subprodutos aperfeiçoados a cada fase. Como visto na seção 3.5, diversos modelos de processo de software passaram com o tempo a apresentar a iteratividade como boa prática, já que sua adoção no desenvolvimento de software apresenta vantagens na redução de riscos devido a *feedbacks* prematuros, importantes para as correções de rota do projeto. A avaliação do indicador vai desde um cenário de desenvolvimento de software seqüencial e sem iteratividade e, conseqüentemente, sem interatividade com o cliente, até o melhor cenário onde as iterações são planejadas e executadas em todas as fases dos projetos, entregando versões preliminares dos produtos para avaliação do cliente.

O indicador *Análise e Modelagem de Arquitetura de Software (ES3)* avalia se a empresa adota linguagens e ferramentas que permitem a elevação do nível de abstração na análise e modelagem do software, o que significa ter uma visão geral e completa dos requisitos do cliente/usuários sem se deixar limitar por questões técnicas e funcionais antes de se entrar na programação do código em si, o que representaria o pior cenário. O melhor cenário utiliza linguagens unificadas de modelagem que englobam diferentes técnicas, inclusive análise essencial, diagramas de dados, módulos e relacionamentos (como por exemplo a UML), que permitem uma análise estruturada e orientada ao objeto.

Conforme apresentado na seção 3.7.1, a importância do indicador *Engenharia de Requisitos (ES4)* reside no fato de sua importância para a qualidade do software, já que diz respeito ao uso de técnicas sistemáticas e repetíveis para garantir que os requisitos do sistema estejam completos, consistentes e sejam atendidos. Sua menor pontuação corresponde a um cenário onde os requisitos são incompletos, mal definidos e com poucos usuários envolvidos na sua definição e validação. No cenário de referência, de maior pontuação, a empresa leva em conta o modelo organizacional e de negócio do cliente para a definição dos requisitos, que abordam proteção, ergonomia e interface, e baseiam-se na ISO 9126. Além disso, há um gerenciamento de requisitos durante todo o processo, para o qual contribui a rastreabilidade de todos os requisitos, controlada pela empresa.

Outra prática fundamental na engenharia de software diz respeito aos testes, que permitem a identificação de falhas e controle de qualidade do produto em desenvolvimento. Desde os primeiros modelos de processo de desenvolvimento de software, como o de Cascata (seção 3.5.1), os testes já apareciam como controle sobre componentes e módulos de software e de integração de sistemas. Normas como a NBR ISO/IEC 12119, que cuida da documentação dos testes em pacotes de software como forma de garantir aos compradores a qualidade dos produtos, e a ISO/IEC 12207, que orienta sobre testes em diversas etapas do ciclo de vida de desenvolvimento de software, mostram a importância dessa prática, que deve, no entanto, ser planejada, já que os testes podem se tornar procedimentos demorados e caros, exigindo um *trade-off* entre esforço vs. eficiência. O indicador *Testes de Software (ES5)* avalia justamente a preocupação com o planejamento e condução dos testes, que devem ser realizados em todas as etapas de desenvolvimento: de unidades, integração destas e validação do sistema, para cada requisito definido, inclusive com testes alfa e beta realizados pelos usuários.

Porém todas as práticas de engenharia de software acima não garantem por si só uma boa *Implantação do Produto no Cliente (ES6)*, cuja prática o indicador com este nome tenta

avaliar. Uma transição de um sistema antigo (ou inexistente) para o sistema de software desenvolvido, sem que haja o devido planejamento e preparação do ambiente e dos usuários, pode ter impactos negativos no ambiente de hardware, software e de aceitação/satisfação dos usuários. Uma estratégia de implantação, com treinamento dos usuários, material de suporte e reuniões de acompanhamento de implantação, são fundamentais para a boa aceitação e funcionamento do novo sistema de software no cliente.

Mas o processo de desenvolvimento de software não se encerra na entrega do produto final ao cliente. Além do suporte à implantação do sistema (indicador anterior), o processo pode englobar o suporte à operação do sistema de software no dia-a-dia do ambiente do usuário, bem como a manutenção, tanto para correção quanto para melhorias. O indicador *Suporte e Manutenção (ES7)* mede desde um cenário ruim de empresas incapazes do fornecimento de suporte e manutenção, até um cenário de referência que reflete a existência de planos de suporte e manutenção bem definidos e acordados com os clientes, incluindo definição de períodos de serviço, responsabilidades das partes, migrações e descontinuações de produtos de software.

O indicador *Maturidade dos Processos (ES8)* avalia em qual grau a empresa se beneficia da adoção de modelos, fazendo com que os mesmos sejam adotados e orientem o dia-a-dia dos processos de desenvolvimento, seu gerenciamento e melhoria. A maturidade de processos foi discutida em modelos como o CMMI-SW, ISO/IEC15504 (SPICE) e OPM3, vistos em seções anteriores, nos quais um cenário de excelência em maturidade são caracterizados por processos melhorados continuamente, com análise contínua das causas dos problemas e com a implementação de inovações permitindo saltos de melhoria acima da média.

Essa seção do questionário se encerra com o indicador que avalia as *Ferramentas de Apoio à Engenharia de Software (CASE) (ES9)* utilizadas pelas empresas, discutidas na seção 3.6.2. Ferramentas básicas de codificação (compiladores, editores, depuradores), além de ferramentas de apoio, por exemplo, à documentação, correio eletrônico e gerenciamento de banco de dados, figuram o cenário básico das empresas de software, que devem avançar para melhores cenários com a utilização de ferramentas de análise e projeto estruturados (método SA/SD), de codificação de quarta geração e orientadas ao objeto, além de ferramentas automatizadas de integração e testes. Ferramentas de prototipação e simulação de sistemas, com ambiente integrado (i-CASE) que permite integração e coordenação de diversas ferramentas CASE, compartilhando informações e reduzindo o esforço de gerenciamento, figuram um cenário mais avançado, pouco visto ainda nas empresas de software atualmente.

4.3.1.2 Qualidade de Software

A seção de Qualidade de Software traz basicamente indicadores que irão medir a *performance*, ou seja, o resultado de qualidade do produto obtido pelas empresas com a adoção das melhores práticas de engenharia de software e de gestão empresarial, de forma indireta. Os indicadores baseiam-se fortemente nas características da qualidade de software da norma ISO/IEC 9126, conforme apresentado na seção 3.7.2.1, além de abordarem Documentação dos Produtos e Métricas de Software.

O primeiro indicador avalia a *Funcionalidade dos Produtos (QS1)*. A funcionalidade de produtos de software é uma característica da qualidade do produto relacionada à capacidade do mesmo prover funções que atendam às necessidades implícitas e explícitas do usuário. Para apresentar bons resultados, os produtos devem apresentar as funções adequadas aos requisitos funcionais, com acurácia de resultados ou efeitos, com o grau de precisão necessário. Além disso, devem apresentar segurança contra vazamento e/ou perda de dados importantes e boa interação com outros sistemas do ambiente do usuário. Neste indicador, avalia-se o quê o software faz para atender aos requisitos, enquanto os indicadores seguintes avaliam de que forma esses requisitos são atendidos.

O indicador de *Confiabilidade (QS2)* avalia se o produto de software foi desenvolvido para evitar falhas durante seu funcionamento, o que pode ser verificado pela frequência com que uma falha no software causa queda ou parada de todo o ambiente de produção, ou métricas como o tempo médio entre falhas (MTBF)? Essas medidas de confiabilidade devem ser monitoradas e trabalhadas ao longo do desenvolvimento do software, para que a empresa controle essa característica da qualidade em todo o processo. Num cenário de referência, os produtos de software, em caso de falhas, recuperam desempenho e dados de forma satisfatória.

A *Usabilidade (QS3)* é outra característica da qualidade avaliada por um dos indicadores dessa seção do questionário. Usabilidade é a capacidade do software em ser compreendido, incluindo o período de aprendizagem e de operação normal. Depende muito da documentação (incluindo manuais de usuário), da ergonomia do seu layout e indicações do que está executando, do que pode ou irá executar. O cenário de pior pontuação reflete produtos de Software difíceis de entender e usar, com falta de ergonomia e com dificuldades para os usuários em cancelar ou desfazer operações indesejadas. O melhor cenário exige software com layout de tela, acesso a funções, menus e organização que facilitam o aprendizado do usuário, que entende o que o software executa e como pode ser usado para tarefas específicas.

O próximo indicador, *Eficiência (QS4)*, diz respeito à capacidade do produto de software apresentar um bom desempenho, mesmo quando muitos recursos são utilizados, dentro das condições de uso especificadas pelo cliente. Para a pontuação do indicador, é necessário que a empresa avalie métricas de tempo de resposta dos softwares, observando o desempenho dos mesmos em condições severas de uso, dentro das condições especificadas.

O indicador de *Manutenibilidade (QS5)* avalia a capacidade do software de ser corrigido, melhorado ou adaptado. Inclui a facilidade com que o mesmo é analisado em busca de causas de falhas, a possibilidade de inclusão de funcionalidades e a estabilidade apresentada após a manutenção. Pouco esforço na análise, execução e testes de mudanças no software, são avaliados com maior pontuação neste indicador.

A documentação é um componente essencial dos produtos de software, tanto para o cliente, que precisa de detalhes sobre o conteúdo e qualidade do produto que está adquirindo, quanto para a empresa desenvolvedora, que precisará de informações documentadas para manutenções ou reutilizações de código futuras, por exemplo. Por isso a importância da avaliação do indicador *Documentação dos Produtos (QS6)* no questionário de *benchmarking*, onde as empresas precisam apresentar documentação de software com informações detalhadas, consistentes, completas e de fácil compreensão para o público-alvo, incluindo informações sobre avaliações de qualidade e resultado de testes sobre seus produtos.

O indicador *Métricas de Software (QS7)* tem por objetivo avaliar em que nível as empresas de software adotam métricas, que são formas de se quantificar características específicas dos softwares, avaliando sua qualidade por meio de um método e uma escala de medição. É importante não somente a utilização das métricas na obtenção de medidas de software, mas também, a utilização dos resultados para a melhoria contínua dos produtos e processos da empresa. Assim, o cenário de referência exige a existência de um modelo que sistematize a utilização de métricas bem definidas, que por sua vez alimentam bancos de dados usados para análise estatística de problemas, resultando em melhoria do processo de desenvolvimento de software.

4.3.1.3 Gerenciamento de Projetos

Por fim, a nova seção Gerenciamento de Projetos contempla indicadores de práticas e *performance* relacionados às principais disciplinas necessárias ao planejamento, monitoramento, controle e encerramento de projetos de desenvolvimento de software (ver seção 3.8.3 e capítulo 3.8).

O indicador *Organização para Projetos (GP1)* foca a importância da organização hierárquica da empresa para o desenvolvimento dos projetos. Estruturas organizacionais clássicas, departamentalizadas, representadas no cenário de menor pontuação nesse indicador, não são organizadas em sintonia com os projetos, sendo que podem resultar em divergência de interesses, recursos e objetivos. As organizações matriciais, de pontuação intermediária, por sua vez, já estão um pouco mais voltadas aos objetivos dos projetos, mas ainda há limitação de atuação integrada por dificuldades de interação entre setores e de autoridade/influência dos gerentes de projeto. O cenário recomendado é o de estrutura por projetos, que organiza a empresa de forma alinhada aos objetivos dos projetos, dando autoridade a gerentes com perfil adequado de liderança, negociação, comunicação, conhecimento técnico-administrativo e relacionamento.

Outro indicador, *Gerenciamento de Riscos (GP2)*, refere-se a uma das disciplinas mais importantes e, no entanto, ainda pouco praticadas pelas empresas. O objetivo do gerenciamento de riscos é o de aumentar a probabilidade e influência de riscos positivos e diminuí-las no caso de riscos negativos. Modelos unificados de desenvolvimento de software, como o RUP, por exemplo, exigem o constante gerenciamento dos riscos em todas as etapas do processo. A iteratividade, caracterizada pela liberação de versões antecipadas do software, deve priorizar os maiores riscos identificados, que serão tratados já nas etapas iniciais do desenvolvimento do software. O melhor cenário nesse indicador remete a empresas com planejamento efetivo de resposta aos riscos, executado para evitá-los (negativos) ou explorá-los (positivos), além de monitoramento e controle de riscos residuais e eventuais riscos secundários.

A maior preocupação e um dos maiores problemas enfrentados em gerenciamento de projetos diz respeito ao cumprimento de prazos, por isso a importância do indicador *Gerenciamento de Tempo (GP3)*. Empresas que se destacam nessa prática apresentam análise estatística de dados históricos de métricas de software, que permitem estimar durações confiáveis para diferentes tipos de projeto. Além disso, gestão de recursos alocados, monitoramento e controle sistemáticos permitem bom cumprimento de prazos.

Outro item que, ao lado dos prazos, é foco de preocupações nos projetos diz respeito ao cumprimento de custos, para o qual é fundamental um bom *Gerenciamento de Custos (GP4)*. Custos bem estimados impactam na viabilidade do projeto e decisão de aceitação do cliente. Gerenciamento ineficaz de custos pode levar a prejuízos da empresa de software e/ou insatisfação dos clientes. Uma empresa com um bom cenário de gerenciamento de custos

apresenta estimativas confiáveis a partir de dados históricos de diferentes projetos, além de um controle rigoroso de custos ao longo da execução.

O gerenciamento do escopo permite garantir que o projeto contenha todo o trabalho necessário para atingir seu produto desejado, nem mais, nem menos. Inclui a criação da estrutura analítica do projeto (EAP), bem como o planejamento, a verificação e o controle de escopo. Por isso a importância do indicador *Gerenciamento de Escopo (GP5)*, uma vez que alterações não controladas ao longo do projeto resultam em problemas de custo, prazo e não atendimento de requisitos do cliente sobre o produto, como visto no cenário intermediário do indicador, refletindo a realidade ainda de muitas empresas. Num cenário ideal, o escopo do produto e do projeto bem definidos, consensados com as partes interessadas e com alterações controladas ao longo de todo o ciclo do projeto.

O *Gerenciamento de Comunicação (GP6)* dos projetos tem o objetivo de garantir a geração, distribuição, armazenamento, recuperação e destinação final de toda informação relacionada ao projeto. Segundo um estudo de *benchmarking* em gerenciamento de projetos realizado pelo PMI no Brasil (PMI, 2006), os problemas de comunicação ficaram com a terceira maior relevância entre os problemas de projetos nas empresas pesquisadas. Apesar disso, apenas 73% das empresas de telecomunicações e tecnologia da informação pesquisadas citou o item comunicação como um aspecto abordado pela metodologia de gerenciamento de projetos adotada. E embora as empresas considerem a comunicação a segunda habilidade mais valorizada nos gerentes de projetos (atrás de liderança), a comunicação foi a deficiência dos gerentes mais citada pelas empresas. Devido a essa importância, o indicador *Gerenciamento de Comunicação (GP6)* atribui maior pontuação às empresas onde o planejamento das comunicações avalia todos os requisitos e a tecnologia das comunicações, para que todas as informações necessárias sejam repassadas no momento e formato necessário às partes interessadas, com gerenciamento contínuo da matriz de comunicação, registros e relatórios do projeto.

O gerenciamento da comunicação foi avaliado em sua *performance* por meio da elaboração do indicador *Funcionamento da Comunicação (GP11)*, no qual o pior cenário reflete falhas constantes de comunicação interferindo no andamento e qualidade dos projetos, ao contrário do cenário de referência, no qual uma matriz de comunicações é seguida à risca, com melhoria contínua de relatórios e processos de comunicação. Nestas empresas, a comunicação deve ser completa e em tempo, o que contribui para o gerenciamento dos projetos e satisfação das partes interessadas.

O indicador *Ferramenta de Apoio ao Gerenciamento de Projeto (GP7)* avalia quais ferramentas são utilizadas para melhorar a efetividade e eficiência do processo de desenvolvimento de software e facilitar a comunicação e informação entre os grupos envolvidos nos projetos. Num cenário de maior pontuação, as empresas se beneficiam da utilização plena de ferramentas com recursos avançados, como controle e monitoramento de custo, caminho crítico, recursos e probabilidade de durações. Além disso, as auxiliam efetivamente na tomada de decisões, geração de relatórios, comunicação e integração, dentro e fora da empresa.

Diante da importância do Gerenciamento do Nível de Serviço para o processo de desenvolvimento de software (ver 3.7.4), desenvolveu-se o indicador *Gerenciamento do Nível de Serviço (GP8)*. O projeto de desenvolvimento de um software nada mais é do que um fornecimento de serviço cujo resultado é um produto de software. Devido à importância da parcela “serviço” no resultado final do produto, é fundamental sua formalização com o cliente por meio de instrumentos contratuais para a gestão das expectativas e níveis de serviço. Esse instrumento, citado em metodologias como ITIL e COBIT, é chamado ANS – Acordo de Nível de Serviço (ou SLA – *Service Level Agreement*), que deve formalizar todo o plano de gerenciamento do projeto, compromissos (inclusive financeiros) e responsabilidades. Num nível de referência, de maior pontuação no indicador, os ANS’s devem ser acompanhados por métricas, que permitam avaliar o andamento do nível de serviço e a geração de planos de ações corretivas, gerando revisões nos ANS’s e melhoria na satisfação dos clientes.

Segundo o estudo de *benchmarking* em gerenciamento de projetos realizado pelo PMI no Brasil (PMI, 2006) citado anteriormente, 53% das empresas pesquisadas apresentaram problemas de prazo em no mínimo metade de seus projetos. Em 8% das empresas, mais de três quartos dos projetos tiveram problemas de prazo. Esse estudo mostra a dificuldade das empresas em alcançar os objetivos de prazo, servindo de base para a determinação de cenários de avaliação para o indicador *Cumprimento de Prazos (GP9)* desenvolvido neste trabalho, no qual as empresas com *performance* de projetos concluídos no prazo acima de 80% recebem nota máxima.

Segundo o mesmo estudo, 42% das empresas pesquisadas apresentaram problemas de custo em no mínimo metade de seus projetos. Em 6% das empresas, mais de $\frac{3}{4}$ dos projetos tiveram este tipo de problema. Esse resultado também mostra a dificuldade das empresas em alcançar os objetivos de custo, servindo de base para a determinação de cenários de avaliação para

outro indicador: *Cumprimento de Custos (GP10)*. Para atingir melhor cenário, acima de 75% dos projetos da empresa precisam alcançar os objetivos de custo neste indicador.

4.3.2 Adequação dos Indicadores Existentes

Nessa seção, serão feitas algumas observações sobre pequenas diferenças que deverão ser consideradas entre a realidade das empresas da indústria tradicional e das empresas de software, que resultarão em adaptações nos indicadores atualmente existentes no método utilizado pelo IEL/SC.

Serão analisadas somente as alterações nos cenários de pontuação dos indicadores, já que a maioria deles sofrerá alterações nas notas explicativas, com o objetivo de auxiliar as empresas respondentes a relacionar os indicadores à realidade do setor de software.

4.3.2.1 Fábrica e Equipamento

Uma consideração muito importante que tem reflexo em praticamente toda uma seção do questionário atual do Benchstar diz respeito à produção, relacionada nos setores industriais tradicionais à fabricação de bens tangíveis, diferente da realidade das empresas de software.

O “processo produtivo” de software não está vinculado aos indicadores do Benchstar da seção Fábrica e Equipamento, à exceção do indicador *Sistemas de informação (GO11)*. Como resultado, os indicadores que não farão parte do questionário do *benchmarking* para empresas de software são: *Organização no chão de fábrica, Automação, Produção puxada, Armazenagem e Manutenção dos equipamentos*.

Já o indicador *Sistemas de informação (GO11)* sofrerá alterações, baseando-se nos cenários do indicador com mesmo título do *Benchmarking Industrial*, uma vez que os cenários do Benchstar para esse indicador refletem empresas com pouca infra-estrutura tecnológica, diferentemente da realidade das empresas de software. E, como a seção Fábrica e Equipamento do questionário não existirá no *benchmarking* para empresas de software, o indicador fará parte da seção Gestão Organizacional.

4.3.2.2 Tempos de Ciclo

Assim como a anterior, e pelos mesmos motivos, praticamente toda a seção do questionário do Benchstar intitulada Tempos de Ciclo não se aplica às empresas da indústria de software,

com exceção de *Prazos de entrega dos subcontratados (AD5)* (que sofre adaptação em seus cenários, o que será abordado em 4.3.2.3). Isto retira do questionário os indicadores: *Percentual de entregas no prazo, Tempo de pedido à entrega, Tempo de produção, Tempo de introdução de um novo produto, Tempo de preparação dos equipamentos e Prazo de entrega dos fornecedores.*

4.3.2.3 Fornecedores

Outro ponto importante diz respeito à questão sobre fornecedores: na indústria de software (desconsiderando fabricantes de produtos com software embutido) o fornecimento traduz-se basicamente na utilização de empresas ou profissionais subcontratados, utilizados com o objetivo de adequar o quadro de desenvolvedores à flutuação de demanda de trabalho de diferentes projetos, bem como devido a variações de tecnologias exigidas nos mesmos.

Levando-se em conta essa realidade, alguns indicadores do Benchstar exigem uma adaptação de interpretação, a saber, os que se referem ou citam os fornecedores. Nos indicadores *Planejamento e análise das metas (GO2)* e *Sistemas de Informação (GO11)* deve-se considerar a integração com os fornecedores de serviços subcontratados, embora o texto do indicador não necessite alterações.

Já o indicador *Fornecedores e subcontratados (GQ4)* exige adaptação nos cenários, trocando o termo “compra”, que remete muito ao fornecimento de bens tangíveis, pelo termo “contratação”, e retirando o texto sobre parceria em programação da produção, do último cenário do indicador, que é restrito à realidade de processos produtivos tradicionais.

Também exige adaptação o indicador *Prazos de entrega dos subcontratados (AD5)*, no qual em seu terceiro cenário o texto “Entregas dos subcontratados sincronizadas com cronograma de produção, sem gerar estoques intermediários em excesso” deve ser substituído por “Atividades dos subcontratados integradas/sincronizadas com o processo de desenvolvimento.”

O indicador *Prazos de entrega dos fornecedores* deve ser retirado da ferramenta de *benchmarking* para as empresas de software, já que o indicador *Prazos de entrega dos subcontratados (AD5)* medirá melhor a *performance* relacionada, conforme parágrafo anterior.

Na seção do questionário referente a Desenvolvimento de Novos Produtos – DNP (incluída no Benchstar a partir do *Benchmarking Industrial*), outros indicadores mencionam os

fornecedores, exigindo que se dê a mesma interpretação, considerando-os subcontratados de serviços como, por exemplo, programação de código de software. São eles: *Geração de conceitos de produtos inovadores (IN1)* e *Envolvimento externo (DP2)*.

4.3.2.4 Gestão Organizacional

A maior parte dos indicadores do Benchstar que compõem a seção do questionário sobre Gestão Organizacional aplica-se diretamente à realidade das empresas de software. É o caso dos indicadores: *Declaração e compartilhamento da visão (GO1)*, *Planejamento e análise de metas (GO2)*, *Estilo de administração (GO3)*, *Participação dos empregados (GO4)*, *Política de incentivos (GO5)*, *Orientação para o cliente (GO7)*, *Rotatividade de mão-de-obra (GO8)*, *Conhecimento da concorrência (GO9)* e *Planejamento de marketing (GO10)*.

Já o indicador *Treinamento e educação* merece adaptações. A começar pelo título, que deve ser substituído por *Desenvolvimento de pessoal (GO6)*. Já no último cenário do indicador, o texto que cita a forte ênfase de treinamentos em qualidade deve ser acrescido de ênfase também nas novas tecnologias em software.

Os indicadores *Introdução de novos produtos*, *Desenvolvimento de produtos*, *Estimativa de demanda* e *Planejamento logístico* serão retirados do questionário, sendo que os dois primeiros já serão abordados pelos indicadores de DNP e inovação (seção 4.3.2.7), acrescentados no Benchstar a partir do *Benchmarking Industrial*.

4.3.2.5 Gestão da Qualidade

Apenas um indicador da seção de Gestão da Qualidade do questionário do Benchstar será mantido sem adaptação alguma, uma vez que se adequa bem às empresas de software: *Estratégia de tecnologia de Produto e Processo (IN7)*. No entanto, o indicador migrará para a seção chamada Inovação no novo questionário. Quanto aos outros indicadores, ou sofrem pequenos ajustes de termos, ou até mesmo são retirados do questionário.

Um dos indicadores que sofrem adaptações é o de *Visão da qualidade (GQ1)*, já que práticas como a inspeção do produto acabado, citada no cenário ruim do indicador, são necessárias na produção de software, ao contrário de processos produtivos tradicionais. O custo de refugo, por sua vez, não se adequa à realidade de empresas de desenvolvimento de software. Além disso, faz-se necessário acrescentar aos cenários a busca contínua pelo atendimento dos requisitos ao longo do processo.

Outro indicador que sofre adaptação é a *Qualidade dos produtos entregues pelos subcontratados (GQ2)*, cujo texto “produtos chegam a sua empresa em más condições” será retirado dos cenários.

O indicador *Reclamação de clientes (GQ3)* sofrerá pequena mudança no cenário de menor pontuação: a exclusão do texto “Devoluções e reparações freqüentes”, incomum para o setor.

O indicador *Pesquisa das necessidades dos clientes (GQ5)* deverá ser aplicado somente a empresas que desenvolvem software de prateleira e/ou customizáveis, e não será aplicado àquelas empresas que somente desenvolvem sob encomenda. Os cenários não mudam.

Por sua vez, quatro indicadores serão retirados e não farão parte do questionário: *Padronização dos processos* (seu equivalente será o novo indicador *Maturidade dos Processos*), *Qualidade das matérias-primas*, *Defeitos internos* (melhor abordado na nova seção Qualidade de software) e *Investimentos em pesquisa & desenvolvimento*.

A retirada deste último indicador deve-se ao fato das atividades de pesquisa e desenvolvimento fazerem parte do próprio desenvolvimento de software, mesmo estando, segundo alguns autores, restritas ao desenvolvimento de software caracterizados por resolução de incerteza científica e tecnológica. Dessa forma, a atividade principal do dia-a-dia da empresa se funde ao que se chama de P&D em setores tradicionais, dificultando a mensuração de orçamentos para P&D, conforme exigia o indicador do Benchstar. Mesmo com a exclusão do indicador, pode-se avaliar o P&D das empresas de software por outros indicadores que estarão incluídos na seção de inovação, medindo em que proporção as empresas realizam inovações radicais ou apenas incrementais.

4.3.2.6 Avaliação de Desempenho da Empresa

Na seção do questionário chamada Avaliação de Desempenho da Empresa, dois indicadores necessitam de adaptações. Um deles *Satisfação do cliente (AD1)*, já que o comportamento de clientes de empresas que desenvolvem software, especialmente customizados e sob encomenda, é diferente daqueles de empresas tradicionais. Os cenários do indicador do Benchstar, que mencionavam problemas como “clientes estão sempre mudando” e exigências como visitas regulares aos principais clientes e clientes fiéis com compras regulares, sofrerão melhorias. O cenário de pior pontuação mencionará uma empresa de software com reclamações freqüentes sobre problemas, que são resolvidos com dificuldade (seja técnica ou de ordem burocrática). O cenário mediano representará empresas com algumas reclamações,

tratadas como prioridade. O cenário de melhor *performance*, por sua vez, será de empresas com todos os clientes satisfeitos com a qualidade, prazo e custo, que recorreriam à empresa novamente em caso de necessidade de outros produtos.

O indicador referente a *Medidas de desempenho (AD3)* também sofrerá alteração em seu cenário intermediário, que cita o uso de indicadores operacionais, como controle de custos, volume de produção e produtividade. Para se adaptar melhor ao desenvolvimento de software, o cenário abordará o uso de indicadores operacionais, como controle de custos, prazos de projetos e métricas de software. Já no cenário de maior pontuação, o termo “moral dos empregados” será substituído por “satisfação dos colaboradores”.

Dois outros indicadores não sofrem mudança: *Variação nas vendas (AD2)* e *Lucratividade (AD4)*.

Os indicadores restantes serão retirados do questionário. *Rotatividade dos estoques* e *Avaliação do impacto ambiental* por não se aplicarem à realidade de empresas sem significativo valor em estoque de materiais e com atividade tipicamente sem grandes impactos ambientais. Já *Gestão de custos* e *Diminuição dos custos* sairão do questionário por já estarem sendo abordados pelos novos indicadores específicos para o gerenciamento de projetos.

4.3.2.7 Desenvolvimento de Novos Produtos

Os indicadores referentes à seção de Desenvolvimento de Novos Produtos, originários do método do *Benchmarking* Industrial, foram analisados e adaptados à realidade do processo de desenvolvimento de software, resultando em indicadores que irão compor duas diferentes seções do questionário de *benchmarking* para empresas de software: Inovação e Desenvolvimento de Produtos.

Inovação

A análise inicia pelos indicadores da seção Inovação, mais especificamente pelo indicador *Geração de conceitos de produtos inovadores (INI)*, que permanecerá sem alterações no novo método. O fato é que, tanto quanto ou até mais importante que em setores tradicionais, a participação de fontes externas na geração do conceito dos novos produtos é fundamental para a inovação dentro da empresa e para o sucesso dessa inovação. Conforme mostrou Duarte (2003), principalmente para o desenvolvimento de software para o segmento vertical (aplicações para determinados setores), os usuários clientes são fundamentais no fornecimento de informações para o desenvolvimento de soluções específicas e customizadas.

Também não sofrerá alterações nos cenários de avaliação o indicador *Relação com universidades e centros de pesquisa (IN5)*, já que essa prática tem mantida sua importância nas empresas de software, como por exemplo parcerias em universidades para o aperfeiçoamento do capital humano, ativo este estratégico para o setor. Também não é modificado o indicador *Estratégia de obtenção de novas tecnologias (IN6)*, cujos cenários mantêm-se válidos para as empresas de software.

Já o indicador *Capacidade de inovação (IN3)* sofre maiores adaptações, havendo substituição do termo “vendas” por “receita” (já que muitos projetos de software resultam em receitas ao longo de meses ou anos, através de serviços, provenientes de apenas uma venda), restrição de novos produtos como sendo aqueles desenvolvidos no último ano, e substituição das inovações de processo produtivo por processo de desenvolvimento de software, como por exemplo, adoção de novos modelos de processo e qualidade de software.

Da mesma forma, o indicador *Introdução de novos produtos (IN4)* passou a considerar apenas os produtos lançados no último ano. Outras adaptações foram: a caracterização de pequenas mudanças incrementais como correspondendo a melhorias de versão de software já existente, no pior cenário; alterações de softwares para diferentes nichos no cenário de pontuação intermediária; e produtos completamente novos como cenário de melhor *performance*.

O indicador *Planejamento do ciclo de vida do produto (IN2)* tem seu cenário intermediário alterado, onde o texto “estratégias de desenvolvimento do produto e processo interligadas” foi excluído por estar ligado à preocupação com o planejamento do processo produtivo no chão de fábrica, não aplicável às empresas de software.

O indicador do *Benchmarking Industrial Estratégia de tecnologia dos produtos* já tem equivalente no Benchstar (*Estratégia de tecnologia de produto e processo – IN7*), sendo, por esse motivo, não aproveitado para o novo questionário. Excluiu-se também o indicador *Infra-estrutura de P&D*, já que as empresas de software, como também foi explorado por Duarte (2003), caracterizam-se por P&D intrínseco ao processo de desenvolvimento de produto, embora restrito ao desenvolvimento de produtos realmente novos, dificilmente havendo necessidade de uma infra-estrutura específica para tal finalidade na empresa.

Acrescentou-se ainda na seção de Inovação do questionário, o indicador do *Benchmarking Industrial* chamado *Ambiente inovativo (IN8)*, que sofrerá uma pequena substituição do texto “mecanismos à prova de falhas” por “gerenciamento de riscos”, que é a forma pela qual se

minimizam os riscos em novos projetos implementados, adequado à realidade de empresas de software.

Desenvolvimento de Produtos

Os indicadores da nova seção Desenvolvimento de Produtos correspondem aos indicadores das subseções Prática e Operações de Engenharia e Processo de Desenvolvimento de Produtos do questionário de *benchmarking* para empresas tradicionais, consideradas as adaptações e exclusões a seguir apresentadas.

Os indicadores *Envolvimento externo (DP2)*, *Feedback de Campo (DP4)* e *Controle de Mudanças (DP3)* são completamente aplicáveis à realidade das empresas de software e não sofreram alterações em seus cenários.

Já os indicadores *Envolvimento multifuncional interno*, *Ferramentas de apoio para a introdução de novos produtos*, *Desenvolvimento automatizado de projeto* e *Performance funcional do produto* foram excluídos, não por não serem aplicáveis ao setor de software, mas sim por já terem sido abordados nos novos indicadores (ver seção 4.3.1): *Organização para Projetos (GP1)*, *Ferramenta de Apoio ao Gerenciamento de Projeto (GP7)*, *Ferramentas de Apoio à Engenharia de Software (CASE) (ES9)* e os indicadores de *performance* de qualidade de software.

Por fim, os indicadores *Simultaneidade no processo de engenharia (DP1)*, *Projeto para produção, uso e descarte* e dois indicadores de *Tempo de ciclo (do conceito do produto até a liberação para produção e da liberação para a produção até a disponibilidade no mercado)* sofreram adaptações. O primeiro deles, recebeu inclusão de modelos de processo de desenvolvimento de software sequenciais em seu pior cenário (modelo cascata), bem como retirado o termo “dentro da engenharia” do cenário intermediário, que remetia à situação de falta de integração entre o setor de engenharia e outros (como produção), característico nas empresas tradicionais.

O segundo, *Projeto para produção, uso e descarte*, teve primeiramente seu título alterado para *Projeto para produção e manutenção (DP5)*, mais condizente com os produtos de software. Já os seus cenários intermediário e superior foram adaptados para referenciar melhores práticas de desenvolvimento de software, como a modularização, reutilização de código e preocupação com manutenibilidade.

Os dois outros indicadores alterados, *Tempo de ciclo – Do conceito do produto até a liberação para produção* e *Tempo de ciclo – Da liberação para a produção até a disponibilidade no mercado*, foram transformados em apenas um indicador, nomeado *Tempo de ciclo de desenvolvimento (DP6)*, já que o processo produtivo das indústrias tradicionais de transformação, no seu caráter de reprodução física (geralmente em massa) do produto projetado não diz respeito às empresas de software, nas quais a produção (codificação) faz parte do próprio processo de desenvolvimento.

5 Aplicação-Piloto e Verificação do Método

5.1 Aplicações-Piloto

A avaliação do funcionamento do método do *benchmarking* ocorreu por meio de três aplicações-piloto em empresas de desenvolvimento de software. Esta avaliação consistiu em testar o método de aplicação e a ferramenta em si, verificando seu potencial e validando o processo de análise e comparação das empresas.

5.1.1 Caracterização das Empresas

As aplicações-piloto ocorreram em três empresas, aqui denominadas Empresas A, B e C (as letras não representam qualquer ordenação possível e foram aleatoriamente nomeadas), pertencentes ao setor de software e localizadas no estado de Santa Catarina.

Duas delas são micro-empresas, com menos de 15 colaboradores e faturamento anual inferior a um milhão de reais. A outra é uma pequena empresa, com 30 colaboradores e faturamento anual pouco acima de um milhão de reais. A maior parte do faturamento das empresas é proveniente do desenvolvimento de software, sejam eles de prateleira ou customizados, destinados ao setor de serviços e às indústrias.

5.1.2 Processo de Aplicação

O método de *benchmarking* para a indústria de software foi desenvolvido de modo a permitir o mesmo processo geral de aplicação, salvo questões particulares de duração, adotado nos métodos Benchstar e *Benchmarking Industrial*: envio do questionário à empresa a ser avaliada, que monta uma equipe interna de *benchmarking* composta por gerentes e colaboradores de diferentes áreas da empresa. A equipe responde previamente todo o questionário como forma de preparação para a etapa seguinte, a de reunião de consenso, na qual um facilitador visita a empresa e revisa, indicador a indicador, a avaliação e compreensão dos indicadores respondidos pela equipe interna. A pontuação consensada dos indicadores irá gerar o índice percentual de práticas e *performance* em cada área de avaliação, cujos resultados são apresentados à empresa avaliada por meio de um relatório de resultados de *benchmarking*.

O contato inicial para convite às empresas para participarem da aplicação-piloto para avaliação do método de *benchmarking* foi inicialmente feito, com o apoio do IEL/SC, por

meio de contato telefônico e correio eletrônico, sendo que foram enviados detalhes sobre o trabalho, bem como informações sobre os resultados que poderiam ser obtidos, tanto para a empresa quanto para a pesquisa.

O convite inicial solicitou a formação da equipe interna com todos os representantes das áreas da empresa, enfatizando a necessidade de que investissem quatro horas para a reunião de consenso, gerando discussão em busca do entendimento e consenso acerca dos cenários mais adequados para a empresa em cada indicador. No entanto, como houve dificuldade por parte das empresas em conseguir a presença de todos os representantes durante toda a reunião, foi solicitada pelas mesmas a divisão da reunião segundo as áreas do questionário, com o tempo necessário estimado, para que houvesse uma participação mais pontual de cada representante, resultando na definição de agenda apresentada na Tabela 4.

Tabela 4 – Agenda solicitada para reunião de consenso durante aplicação-piloto

Seções do questionário	Tempo necessário	Equipe (pelo menos 1 representante das áreas):
Gestão Organizacional	45 min	RH e Comercial
Gestão da Qualidade	20 min	Qualidade
Desempenho Geral da Empresa	20 min	Comercial
Inovação	30 min	Pesquisa e Desenvolvimento (P&D)
Desenvolvimento de Produtos	25 min	P&D / Produção de Software
Engenharia de Software	30 min	P&D / Produção de Software
Qualidade de Software	30 min	P&D e qualidade
Gerenciamento de Projetos	40 min	Gerente de projetos, se houver

A nova condição de participação dos representantes facilitou a aceitação das empresas, porém não garantiu a participação real dos representantes solicitados. Em duas das empresas, por exemplo, houve apenas a participação de um respondente na reunião, que devido aos cargos de liderança ocupados possuíam uma visão significativa de toda a empresa. No entanto, as respostas foram enviadas para revisão de outros membros da empresa para um aceite final. Na outra empresa, a reunião ocorreu parte com a participação de três colaboradores, parte com apenas um, sendo os resultados igualmente revisados por integrantes que não puderam estar presentes.

A avaliação com poucas pessoas durante a reunião permitiu a conclusão da mesma em menor tempo que as quatro horas previstas, embora a situação recomendada seria uma discussão

maior dos indicadores entre diversos colaboradores, explorando bem a avaliação da empresa durante o período programado.

Ao longo da reunião de consenso, os indicadores foram individualmente explicados pelo facilitador aos participantes com o objetivo do entendimento completo dos cenários e a conseqüente identificação daquele mais adequado à realidade de cada empresa. As dificuldades de entendimento também serviram como fonte de melhorias para o questionário, conforme será visto na seção 5.3.

Após a reunião de avaliação, os dados foram compilados e analisados, etapa para a qual houve a necessidade de consolidação de um padrão de análise, que permitisse a avaliação das empresas em cada uma das áreas de avaliação a partir da pontuação dos diferentes indicadores. Ressalta-se aqui o fato das seções do questionário (Quadro 5) não corresponderem diretamente às mesmas divisões das áreas de avaliação do modelo (Figura 9), sendo estas últimas compostas por indicadores que as impactam. Os resultados são apresentados na seção 5.1.3 e no Apêndice 1.

Os relatórios contendo os resultados comentados foram elaborados e entregues às empresas, juntamente com um questionário de avaliação de satisfação, por meio do qual as empresas puderam avaliar todo o trabalho: o questionário, o relatório de resultados, o processo de aplicação e uma avaliação geral final, cujos resultados também serão apresentados na próxima seção.

5.1.3 Análise de Resultados

Para a análise final das empresas, os indicadores das seções do questionário são distribuídos nas áreas de avaliação do modelo (Figura 9) às quais estão relacionados. O índice de cada área de avaliação é então obtido a partir da média da pontuação (1 a 5) dos indicadores que a compõem multiplicada por 20%, resultando nos Índices de Práticas (calculados somente por indicadores de práticas) e *Performance* (calculados somente por indicadores de *performance*) de cada área. A média simples dos índices de práticas das cinco áreas resulta então no Índice Geral de Prática da empresa, assim como a média simples dos índices de *performance* das cinco áreas resulta no Índice Geral de *Performance*, vistos nas últimas linhas da Tabela 5.

Convém ressaltar que cada um dos indicadores possui o mesmo peso dentro das áreas, assim como cada uma das cinco áreas de avaliação possui o mesmo peso, respeitando a mesma forma de pontuação adotada pelo método do Benchstar adotado pelo IEL/SC.

A definição de quais indicadores impactam cada uma das áreas de avaliação do método de *benchmarking* para o setor de software baseou-se na análise do escopo da área *versus* o quesito avaliado em cada indicador, observando-se as considerações feitas na seção 4.1. Os resultados dos índices obtidos nas aplicações-piloto são apresentados na Tabela 5.

A realização das aplicações-piloto e modelagem do relatório de apresentação dos resultados auxiliaram na verificação do potencial da ferramenta para um processo de *benchmarking*. A organização dos indicadores em áreas de avaliação permite uma análise tanto individual quanto coletiva das empresas, verificando quais as áreas mais deficientes, que demandam mais investimentos e quais as áreas mais desenvolvidas nas empresas.

Tabela 5 – Resultados das aplicações-piloto: índices de Práticas e Performance por área

Áreas de avaliação da gestão	Índice médio (%)	Índice das Empresas (%)		
		A	B	C
Gestão da Produção - Práticas	77,4	75,4	76,9	80,0
Gestão da Produção - Performance	76,3	92,0	60,0	86,7
Gestão da Qualidade - Práticas	73,6	82,2	68,9	70,0
Gestão da Qualidade - Performance	81,3	86,7	66,7	90,0
Gestão Organizacional - Práticas	63,1	71,1	48,9	68,9
Gestão Organizacional - Performance	83,0	75,0	80,0	95,0
Gestão da inovação - Práticas	59,1	68,6	51,4	57,1
Gestão da inovação - Performance	73,3	80,0	53,3	86,7
Gestão de Projetos - Práticas	77,5	82,5	72,5	77,5
Gestão de Projetos - Performance	74,8	76,0	56,0	92,0
Índice Geral de Práticas %	72,3	77,7	66,0	73,3
Índice Geral de Performance %	77,7	83,5	63,3	88,9

Os dados da Tabela 5 são plotados no gráfico radar, apresentado na Figura 12, para melhor visualização e análise. Os pontos nos eixos do gráfico mostram a pontuação, de 0 a 100%, em cada uma das áreas de avaliação, tanto em práticas quanto em *performance*. Quanto mais afastados do centro os pontos estiverem, melhor pontuada a empresa. Quanto mais uniforme se apresentar o polígono, melhor é o equilíbrio entre as práticas adotadas e *performance* obtida, seja entre práticas e *performance* de uma mesma área, seja entre as diferentes áreas.

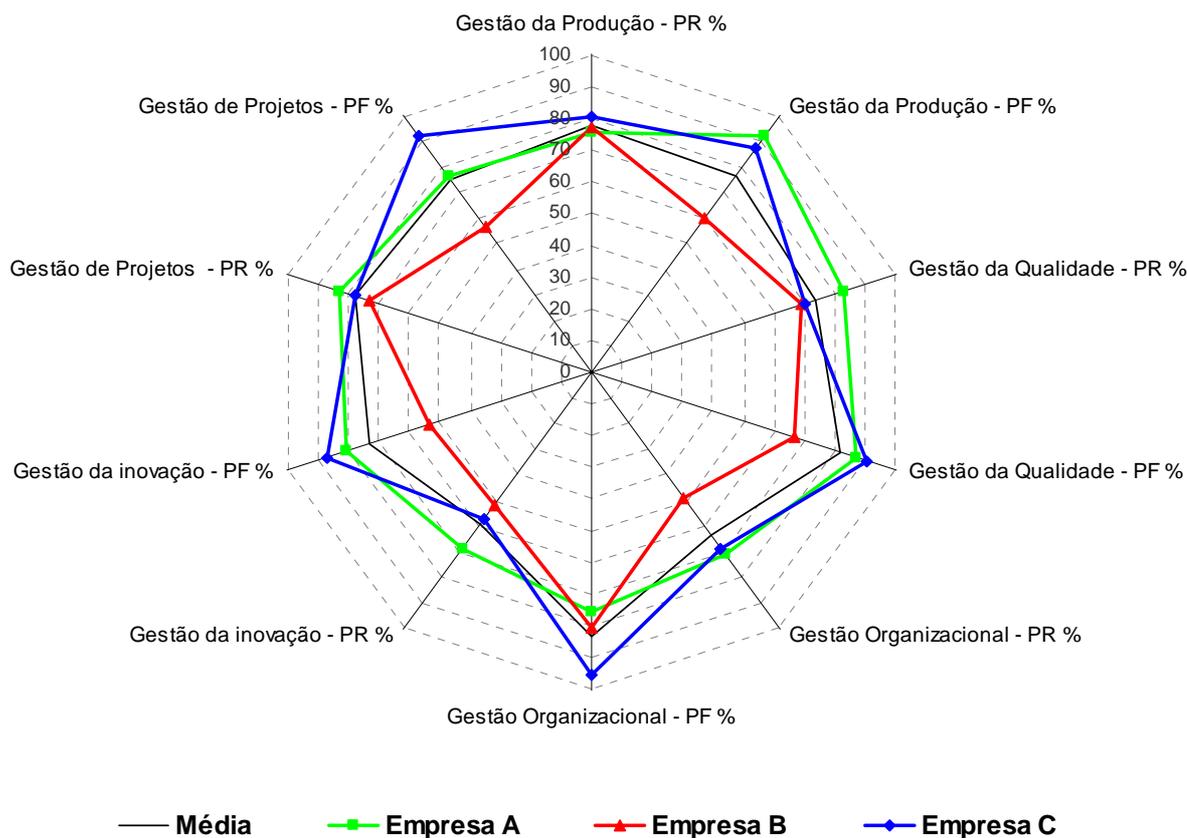


Figura 12 - Resultados das aplicações-piloto: índices de práticas e *performance* por área

O gráfico apresenta quatro linhas, três delas correspondentes a cada uma das três empresas que participaram da aplicação-piloto, e uma representando a pontuação média das empresas.

Analisando-se o gráfico com os resultados das aplicações-piloto, verifica-se de forma geral, nas áreas da empresa A, um grande equilíbrio dos índices de práticas em relação aos de *performance*, mostrando uma empresa bem estruturada nas diferentes áreas, com investimento eficaz em práticas, obtendo *performance* à altura. Já nas empresas B e C, o gráfico permite visualizar uma dispersão grande entre as duas empresas, que mostram situação similar somente na área de Gestão Organizacional, onde obtêm muita *performance*, apesar de poucas práticas. Nas outras áreas, a empresa B investe em práticas, mas tem dificuldades de obtenção de *performance*, com exceção da Gestão da Qualidade e Inovação. Já a empresa C apresenta um quadro vulnerável, onde obtém *performance* sem o embasamento em práticas sistemáticas estabelecidas (índices de práticas maior que os de *performance*).

O baixo índice médio de práticas em Gestão Organizacional (63,1%) reflete a deficiência de um planejamento pró-ativo do desenvolvimento de pessoal, sendo que os treinamentos, quando ocorrem, são para poucos e somente quando há oferta de cursos. A baixa pontuação

reflete, também, um estilo de administração que tende à centralização, com os gestores determinando atribuições aos funcionários, que possuem autonomia limitada. A gestão organizacional é prejudicada nas empresas pela falta de integração nos sistemas de informação, o que surpreende por se tratarem de empresas de tecnologia desenvolvedoras de sistemas de software. As práticas para medir desempenho, utilizadas pelas empresas, ainda ficam muito restritas a indicadores operacionais, possuindo poucos indicadores estratégicos usados na tomada de decisões.

Apesar do baixo índice de práticas, o índice médio de *performance* em Gestão Organizacional (83,0%) ficou muito bem pontuado nas empresas participantes da aplicação-piloto. O bom resultado é reflexo principalmente do forte crescimento das vendas e na atraente margem de lucro obtida. Além disso, as empresas apresentaram baixo índice de rotatividade da mão-de-obra (cerca de 10%), considerado pelos entrevistados baixo levando-se em conta o porte e o setor. O funcionamento da comunicação foi o indicador menor pontuado na performance dessa área, atingindo uma pontuação regular devido a uma grande variação no resultado das três empresas: desde falhas constantes de comunicação na empresa A até uma comunicação completa e em tempo na empresa C, com a empresa B numa situação intermediária.

O comportamento médio das empresas na área de Gestão Organizacional enquadra-se numa situação de vulnerabilidade, na qual as empresas atingem um bom índice de *performance*, porém, sem o mesmo nível de práticas adotadas. É um comportamento característico de empresas de pequeno porte, onde os gestores ainda costumam centralizar muitas decisões, a integração ainda não depende muito de um sistema instalado (já que o quadro é enxuto e a empresa pequena) e as taxas de crescimento costumam ser maiores que as empresas de grande porte. Porém, na medida em que a concorrência aumenta, e na medida em que a empresa começa a crescer e a ganhar porte e complexidade, as práticas vão se tornando cada vez mais importantes para a manutenção dos mesmos níveis de *performance*, de forma sustentada e sem vulnerabilidade. Segundo Seibel (2004), essa posição é muito instável e difícil de ser sustentada no longo prazo, se as condições de competição se acirram.

Na área de Gestão da Inovação, o desequilíbrio também ocorreu, embora em menor grau que na gestão organizacional, estando o índice médio de práticas igual a 59,1% frente a uma *performance* obtida de 73,3%. Pesaram no baixo índice de práticas o indicador de planejamento do ciclo de vida dos produtos futuros, uma vez que o planejamento de novas gerações de produtos não é sistemático e, quando ocorre (empresa A), tem um horizonte de curto prazo (3 anos). Também pesou negativamente a deficiência no planejamento do

desenvolvimento de pessoal, já mencionado como impactante na Gestão Organizacional, bem como a estratégia de tecnologia de produtos e processo, uma vez que a análise das necessidades de novas tecnologias é feita somente quando novos projetos a demandam, sem uma política explícita e contínua de investimento em novas tecnologias e sem monitoramento do nível tecnológico da concorrência. O relacionamento com universidades e centros de pesquisa mostrou-se intermediário, com contatos esporádicos resultando em projetos desenvolvidos em conjunto. A empresa com menor índice de práticas, a empresa B, diferenciou-se das demais principalmente na baixa pontuação em estratégia de obtenção de novas tecnologias, já que apresenta atitude reativa às inovações do mercado.

As empresas A, B e C apresentaram *performance* em Gestão da Inovação de 80,0%, 53,3% e 86,7%, respectivamente, com grande dispersão devido à empresa B. Contribuiu positivamente para o resultado o indicador de variação nas vendas, com pontuação máxima nas três empresas. Já o indicador de introdução de novos produtos, que verifica o nível de alterações nas inovações, apresentou grande variabilidade, tendo uma empresa pontuada em cada cenário: a empresa B com poucas mudanças incrementais nos produtos, a C com introdução de novas versões adaptadas para diferentes nichos e a empresa A com inovações consideradas significativas e radicais. A *performance* do indicador de capacidade de inovação, por fim, apresentou-se com pontuação baixa nas empresas A e B, nas quais não mais que metade da receita é proveniente de produtos com menos de um ano de vida e onde algumas inovações de processo foram obtidas; a empresa C atingiu pontuação máxima, com mais da metade de seu faturamento proveniente de novos produtos.

Pode-se concluir, da análise da Gestão da Inovação nas três empresas, que a empresa C apresenta uma ótima *performance* atual a partir de sua estratégia de tecnologia de produto e processo a curto prazo, que focadas aos produtos atuais da empresa, faz com que a maior parte de seu faturamento hoje seja proveniente de produtos novos, resultado de adaptações para diferentes nichos. Já a empresa A investe um pouco mais em suas práticas, com um horizonte de planejamento um pouco mais a médio prazo, porém ainda com a maior parte do faturamento proveniente de produtos com mais de um ano de vida. Porém suas inovações são mais radicais, o que pode lhe conferir vantagens competitivas maiores no futuro. A empresa B, ainda com estratégia reativa em relação às novas tecnologias, acompanha o crescimento de vendas das outras empresas, mas sem inovações significativas nos produtos.

O índice médio de 77,4% em práticas em Gestão da Produção foi reflexo de notas regulares na adoção de ferramentas computacionais de apoio à engenharia de software (CASE), devido

à pouca integração entre as ferramentas e à inexistência de ferramentas automatizadas de integração e testes, prototipação e simulação de sistemas. Foi reflexo também da necessidade de melhoria na utilização de informações dos clientes como *feedback* de campo para o desenvolvimento dos produtos. Porém, todas as outras práticas da área estão mais bem disseminadas, contribuindo positivamente para o índice.

Essas práticas permitiram uma *performance* em Gestão da Produção de 76,3% na média das três empresas, porém com a empresa B (60,0%) bem abaixo da A (92,0%) e C (86,7%). A principal dificuldade apresentada pelas três empresas ficou no tempo de ciclo de desenvolvimento, que por não ser medido, foi estimado pelas empresas como sendo estável, onde um resultado de referência seria uma redução do tempo a cada novo ciclo, redução esta monitorada e medida, mostrando maturidade no processo. O principal motivo da defasagem da empresa B em relação às outras consistiu em sua *performance* apenas regular nos indicadores de cumprimento de custos e prazos, e confiabilidade dos softwares. Esse último pode ser facilmente explicado pelo baixo investimento nas práticas de projetos voltados à produção e manutenção, no qual obteve nota 3 frente à nota 5 das demais empresas.

A Gestão da Qualidade, com índice médio de práticas de 73,6%, merece atenção especial por se tratar de item básico qualificador para qualquer empresa, principalmente quando se fala em empresas de base tecnológica. A qualidade de um processo de software está intimamente ligada à maturidade de seus processos, que tem apresentado oportunidades para melhoria nas empresas estudadas, uma vez que os processos, quando mapeados e monitorados quantitativamente (nem sempre o são), dificilmente são alvo de análise em busca de causas sistêmicas de problemas que levariam à melhoria contínua dos padrões de processo. Da mesma forma, o uso de métricas como prática de qualidade não é sistemático nessas empresas. Apesar de algumas métricas serem utilizadas durante os testes de desenvolvimento, auxiliando na correção de defeitos, os resultados são pouco utilizados para tomada de decisão ou melhoria de processos. A prática de destaque positivo fica por conta da orientação para o cliente, sendo que os contatos frequentes com os mesmos fazem com que todos os colaboradores conheçam e estejam orientados ao atendimento de suas necessidades.

Já a *performance* da Qualidade, com 81,3%, aponta o bom resultado das características da qualidade dos produtos de software produzidos nas empresas participantes da aplicação-piloto, como a eficiência, a funcionalidade e a usabilidade. Vale ressaltar que estes indicadores foram pontuados com base na percepção dos entrevistados, não se baseando em métricas adotadas pelas empresas ou na visão dos usuários. O indicador de reclamação de

clientes teve sua média prejudicada em função da empresa B, que apresenta taxa de reclamações maior que 1%, indicando que algo ainda precisa ser melhorado, o que é ratificado pela satisfação do cliente com relação à qualidade, prazo e custo, que não é completa, o que reforça a indicação de necessidade de investimento para o aprimoramento de práticas nessa área.

Em média, a área com índice de práticas de maior pontuação nas empresas correspondeu à Gestão de Projetos (77,5%), impulsionada principalmente pelas práticas bem implantadas de gerenciamento de escopo e utilização de ferramenta para gerenciamento de projetos. Porém, práticas como o gerenciamento do nível de serviços e o gerenciamento de riscos contribuíram para que o índice não fosse maior. Na primeira, houve grande variação no comportamento das empresas, cada uma posicionando-se em um cenário diferente: na mais crítica (empresa B), ausência de Acordos de Nível de Serviço (ANS); na melhor pontuada (empresa A), existência de acordos com definição quantitativa dos níveis de serviços, devidamente monitorados; e uma empresa no nível intermediário (empresa C). Na segunda prática, de gerenciamento de riscos, o comportamento das empresas foi mais homogêneo, nas quais há identificação e análise dos riscos, inclusive quantitativamente, porém com pouca atuação na resposta aos mesmos e nas quais as revisões nos planos de riscos são raras.

A *performance* de Gestão de Projetos ficou um pouco abaixo (74,8%) do índice das práticas, mas com grande variabilidade (empresa A, B e C com 76,0%, 56,0% e 92,0%, respectivamente). O cumprimento de custos e de prazos nos projetos mostraram-se os indicadores de *performance* mais bem pontuados da área, sendo que a melhor empresa atingiu pontuação máxima por apresentar elevado percentual de seus projetos cumprindo metas de custo e prazo, e a pior empresa num nível intermediário de pontuação, com cerca de metade dos projetos não atingindo as metas de prazo e custo. O funcionamento da comunicação pesou negativamente na área, mas também apresentando alta variabilidade: empresa A com falhas constantes de comunicação, empresa B com problemas esporádicos e a empresa C com uma comunicação completa, orientada por uma matriz de comunicação, contribuindo para a satisfação de todas as partes interessadas e envolvidas nos projetos. A baixa *performance* da empresa C (56%) na Gestão de Projetos foi resultado principalmente do indicador de reclamação de clientes, que pode ser reflexo de sua baixa prática de gerenciamento do nível de serviço (apontado no parágrafo anterior) ou de fragilidades apontadas em outras áreas de avaliação já comentadas (já que as áreas de avaliação não são isoladas e interagem entre si).

O Apêndice 1 – Tabela de indicadores por área: resultado da aplicação-piloto – traz uma tabela do relatório de resultados que mostra o detalhamento dos indicadores do questionário que compõem cada área de avaliação. Nela são apresentadas as médias de pontuação das três empresas em cada indicador, usadas para o cálculo dos índices médios de práticas e *performance* de cada área, apresentados na análise dos parágrafos anteriores.

Também são apresentados os valores da moda para cada indicador, uma vez que como se trata apenas de uma amostra de três empresas participantes da aplicação-piloto, nem sempre o valor da média representa bem o comportamento de cada indivíduo, sendo que a moda nos mostra as pontuações com maior ocorrência. Aqueles indicadores com moda inexistente nos servem como indicativo de que houve uma grande variabilidade na pontuação entre as três empresas.

Sendo assim, a composição final do relatório de resultados do *benchmarking* para empresas de software consiste no gráfico radar, na tabela dos índices das áreas avaliadas, na tabela de pontuação dos indicadores (apêndice 1) e nos comentários e análises escritos pelo facilitador.

5.1.4 Conclusões dos Resultados das Aplicações

O método desenvolvido permitiu a identificação das deficiências nas práticas adotadas, referente principalmente à gestão organizacional e gestão da inovação. A deficiência de investimentos no desenvolvimento de pessoal foi um dos principais problemas diagnosticados, causando impacto negativo nas áreas de gestão organizacional e de inovação. O estilo de administração nas empresas também apresentou oportunidades para melhoria, uma vez que as tomadas de decisão mostraram-se um tanto quanto centralizadas na alta administração.

Este quadro é condizente com a realidade da maioria das empresas de pequeno porte, que normalmente apresentam poucos recursos para o investimento no capital humano da empresa e nas quais o pequeno quadro de colaboradores leva à centralização de decisões nos diretores, que geralmente são os próprios donos da empresa.

Um ponto crítico diagnosticado pelas aplicações-piloto referiu-se à gestão da inovação, cujas práticas, que atingiram um índice médio de apenas 59,1%, revelam-se insuficientes para a manutenção da competitividade de empresas de base tecnológica no longo prazo. Pesou negativamente, além do pouco investimento em desenvolvimento de pessoal, o pouco planejamento dos produtos futuros, que quando ocorre objetiva apenas o curto prazo.

Constatou-se, também, o nível intermediário na maturidade dos processos das empresas, que junto com um limitado uso de métricas (mais restritas às fases de teste de software), resultaram num índice regular de práticas em gestão da qualidade. Do lado da *performance*, a taxa de reclamação de clientes indicou a existência de alguma oportunidade para melhoria na adoção de práticas com vistas a obter resultados.

No entanto, as aplicações-piloto revelaram ainda um bom investimento, por parte das três empresas, em práticas de gerenciamento de projetos, que contribuem muito para o processo de uma indústria como a de software, fundamentada principalmente nos projetos de desenvolvimento de produtos. O bom índice de adoção dessas práticas impulsionou bons resultados nas empresas A e C, vistos em indicadores de *performance* como os de cumprimento de custo e de prazo em seus últimos projetos.

As *performances* das diferentes áreas de avaliação ficaram muito bem pontuadas nas empresas A e C, com destaque para os índices na gestão da produção e gestão da qualidade (89,3 e 88,3%, respectivamente, considerando somente as duas empresas), devido ao bom cumprimento de custos dos desenvolvimentos e boa funcionalidade e eficiência dos softwares desenvolvidos.

5.2 Opinião das Empresas

Após a entrega do relatório de avaliação às empresas participantes da aplicação-piloto, procedeu-se uma avaliação, junto à equipe de cada empresa, sobre a satisfação quanto ao processo e seus resultados. Representantes das empresas avaliaram quatro aspectos solicitados: o questionário, o relatório de resultados, o processo de aplicação e uma avaliação geral final, onde críticas e sugestões puderam ser apresentadas para a melhoria do método.

O questionário foi avaliado pela empresa A como “diferenciado e objetivo”, com uma abordagem interessante capaz de identificar, de forma completa, as práticas e resultados da empresa por meio de indicadores coerentes com a área de tecnologia. Segundo um dos respondentes, o acompanhamento do facilitador e o uso de cenários facilitou o entendimento das questões, “deixando de ser um simples questionário como tantos outros”. Segundo a empresa B, “A técnica de pontuação por cenários é uma ferramenta muito boa, que permite retirar uma visão macro da empresa para consolidar os dados de cada área.”. Já um

representante da empresa C afirma que “o questionário é adequado, pois aborda os principais fatores relacionados a empresas de desenvolvimento de software”.

Quanto ao relatório de resultados entregue, segundo a empresa A, apresentou conteúdo coerente com a realidade da empresa, refletindo alguns pontos com deficiência já conhecidos e outros não. Apesar disso, a empresa sentiu falta de uma base de comparação com resultados de empresas do setor de software, o que configuraria melhor um processo de *benchmarking*. A solicitação é completamente coerente, o que não pôde ser atingida no processo inicial de aplicação-piloto por se tratarem justamente das aplicações iniciais de validação e consolidação de uma versão final da ferramenta, versão esta que será utilizada futuramente para a formação de um banco de dados com diversas empresas, o que permitirá o processo de *benchmarking* em si do setor.

O conteúdo do relatório, segundo a empresa B, foi “satisfatório ao permitir que em futuras avaliações possa haver comparações, em percentuais, para ver o índice de melhoria em cada área”. Já segundo a empresa C, “algumas questões deveriam ser alteradas para empresas de menor porte, para aumentar a precisão da avaliação”. Essa observação é importante e levanta duas questões que merecem ser comentadas. Primeiro, quanto à precisão da avaliação: no método proposto, no qual o tempo de avaliação relativamente curto frente ao espectro de áreas avaliadas na empresa é tido como um ponto positivo (empresas normalmente apresentam aversão a longos processos de diagnóstico), a precisão é fator muito dependente primeiro, da participação do facilitador do método, e segundo, da interação e consenso entre diversas pessoas de diferentes áreas da empresa. O fato dos cenários de um indicador exigirem práticas somente adotadas em empresas de maior porte não retira a precisão de uma pontuação regular ou baixa eventualmente obtida por uma pequena empresa. Além disso, a possibilidade ou não de uma empresa de menor porte atingir cenários mais exigentes somente poderá ser verificada futuramente (quando existir um banco de dados após diversas aplicações) se as empresas forem avaliadas numa mesma base de critérios, ou seja, mesmos indicadores. É justamente disso que trata a segunda questão, pois é importante a necessidade dos indicadores permitirem a avaliação de empresas de software de diferentes portes, o que enriquecerá o processo de *benchmarking* futuro, onde se poderá, através de análises sobre o banco de dados, comparar *performances* de empresas de diferentes portes, verificando inclusive se as mesmas práticas adotadas por grandes empresas líderes são também as adotadas pelas empresas líderes de menor porte, entre outras análises.

O resultado do *feedback* das empresas, quanto ao processo de avaliação utilizado, também foi positivo. A empresa B, no entanto, sugeriu que “o processo poderia ser um pouco mais extenso e envolver mais de uma pessoa dos níveis mais altos de gestão para equilibrar as visões”. A afirmação é verdadeira, porém, conforme apresentado no item 5.1.2, no contato inicial com a empresa foi solicitada a participação de diversas pessoas para o processo, sendo que infelizmente nem todas as empresas conseguiram seguir esta solicitação, como foi o caso da empresa B.

Por fim, as empresas foram solicitadas a avaliar o trabalho de uma forma geral. Como resultado, afirmaram se tratar de um bom trabalho, que “servirá como fonte de aperfeiçoamento”, segundo uma das empresas. “O trabalho é bastante apropriado e merece ser compartilhado com outras empresas desenvolvedoras” de software, citou a empresa C.

5.3 Contribuições

A aplicação prática do método desenvolvido permitiu o levantamento de diversos pontos passíveis de melhoria e melhor definição. O próprio método de aplicação utilizado, por exemplo, serviu para a verificação de vantagens e desvantagens atreladas à forma de avaliação das empresas. Também foram consolidados os indicadores, o questionário e o formato do relatório de apresentação dos resultados.

Quanto ao método de aplicação, visto no item 5.1.2, embora houvesse a sugestão de que os indicadores fossem respondidos por uma equipe representativa de pessoas das diferentes áreas da organização, percebeu-se que há uma dificuldade inerente às pequenas empresas de reunir tal grupo, seja pelo pequeno quadro de funcionários, gerando dificuldades de disponibilidade simultânea de agenda, seja pela própria inexistência das gerências.

Devido à importância para o resultado do processo, ficou como contribuição e sugestão a necessidade de se enfatizar a importância da formação da equipe já nos contatos iniciais com a empresa, que deve ter ciência dos benefícios do processo de *benchmarking* e ter o apoio da diretoria para que todos os passos de aplicação do método sejam realizados conforme suas etapas definidas. Para isto, é fundamental que se esclareça e se ressalte, nos contatos iniciais com a empresa, as etapas e principalmente os benefícios resultantes da rápida avaliação obtida pelo processo, que permite a radiografia completa da empresa em relação ao seu setor.

Como consolidação das etapas de aplicação do método, sugere-se a obrigatoriedade de realização de uma reunião de consenso dos indicadores na presença de um facilitador, conhecedor profundo dos indicadores e com perfil de moderação, em conjunto com no mínimo um representante de cada uma das diferentes áreas da empresa, levando em conta a realidade de seu porte.

Outra contribuição das aplicações-piloto para com o método de *benchmarking* para empresas de software consistiu na consolidação do modelo de avaliação e da composição final do relatório de resultados. Esse relatório, entregue às empresas alguns dias após as aplicações, apresentando o gráfico radar, tabela dos índices das áreas avaliadas e tabela de pontuação dos indicadores (sem identificação individual dos concorrentes), devidamente comentados e analisados pelo facilitador, irá incorporar futuramente a posição da média das melhores empresas já avaliadas pelo método, quando se dispuser da formação de um banco de dados significativo.

O processo de aplicação-piloto nas empresas também resultou na verificação de diversos pontos a serem melhorados no questionário, uma vez que o processo de aplicação permitiu verificar dificuldades de compreensão em cenários, redundância entre diferentes indicadores ou a existência de indicadores dispensáveis à avaliação pretendida das empresas.

Verificou-se que o indicador *Estratégia de Obtenção de Novas Tecnologias* apresentou dificuldades para ser compreendido nas aplicações-piloto, talvez por ser mais adequado à avaliação de empresas da indústria tradicional, foco do *Benchmarking Industrial*, método do qual foi retirado, sendo que o indicador é muito voltado à avaliação da incorporação do conhecimento da tecnologia nas compras de equipamentos ou sistemas, comum na indústria de transformação tradicional. Em empresas de software, acostumadas a desenvolver sua própria tecnologia, o indicador pode ser devidamente suprido pelo indicador chamado *Estratégia de Tecnologia de Produto e Processo*. A conclusão, portanto, é pela exclusão do indicador *Estratégia de Obtenção de Novas Tecnologias*.

Pôde-se constatar, ao serem analisados os resultados das aplicações-piloto, a necessidade de um indicador de *performance* para a avaliação específica da *Satisfação dos Colaboradores* das empresas, complementando a avaliação da *performance* da gestão organizacional juntamente com o indicador de *Rotatividade da Mão-de-obra*. Sendo assim, incorporar-se-á ao questionário os seguintes cenários para um indicador de *Satisfação dos Colaboradores*: “Problemas de satisfação interna moderados ou sérios” no cenário de menor pontuação; “Boa satisfação dos colaboradores. Pesquisa com média de 75%.” no cenário intermediário; e

“Colaboradores muito satisfeitos. Média de satisfação acima de 90%. Índice medido e avaliado sistematicamente.” no cenário de maior pontuação.

O indicador *Planejamento do Ciclo de Vida do Produto* sofrerá melhorias em seu título e cenários com o objetivo de torná-lo mais claro durante as futuras avaliações. No setor de software, diferentemente de outros setores tradicionais, a expressão “ciclo de vida” está freqüentemente associada a todas as etapas de desenvolvimento dos produtos de software. Percebeu-se, durante as aplicações-piloto, que esta associação do título não traduz adequadamente a questão que se deseja avaliar através dos cenários, que por sua vez também apresentaram oportunidade para melhoria. Com o objetivo de melhor adequar o indicador às empresas do setor de software, o título será alterado para *Planejamento de Produtos Futuros* e os cenários de pontuação três e cinco avaliarão apenas o horizonte de planejamento de famílias e gerações futuras de produtos, sendo inadequada a quantificação das novas famílias, uma vez que recai sobre especificidades do portfólio de cada empresa. Na prática do planejamento de produtos futuros, é mais importante saber se o planejamento é para longo prazo do que saber se são para duas, três ou mais famílias de produtos.

O último cenário do indicador *Simultaneidade no Processo de Engenharia* também sofreu melhorias após as aplicações-piloto. Para melhor refletir a questão que se pretende avaliar com o indicador, o texto do cenário de pontuação cinco foi mudado de “Total colaboração ao longo do ciclo de vida do produto, sistema de gerenciamento de projetos” para “Total colaboração entre as diferentes funções ao longo do ciclo de desenvolvimento do produto”.

O indicador *Envolvimento Externo* apresentou dificuldades de diferenciação entre os cenários da pontuação um e três, já que uma situação de limitada interação de planejamento com fornecedores e clientes poderia tanto resultar numa pontuação 1 quanto uma pontuação 3 para a empresa avaliada. Providenciou-se dessa forma uma melhoria da descrição do cenário de pior pontuação, que passa a corresponder às empresas com “cultura de empresa isolada; interação com grupos externos restringe-se às transações comerciais tradicionais”.

A necessidade de maior facilidade de compreensão do primeiro cenário do indicador *Controle de Mudanças* exige uma simplificação na sua descrição, que passará a ser: “Quando ocorre alguma mudança, a aprovação/liberação para produção é lenta; processo manual tradicional”.

O indicador *Engenharia de Requisitos (ES4)* teve seus cenários de pontuação três e cinco um pouco mais resumidos, também com o objetivo de torná-lo mais direto e fácil de compreender e avaliar. No cenário intermediário, os problemas com requisitos implícitos e com alterações

de requisitos foram incluídos. O cenário de referência, de maior pontuação, ficou mais resumido: o “modelo organizacional e de negócio do cliente são levados em conta. Baseiam-se nos requisitos de qualidade da ISO9126. Gerenciamento e rastreabilidade de requisitos durante todo o processo”.

Verificou-se ainda que o cenário de menor pontuação do indicador *Suporte e Manutenção (ES7)* não citava a ausência de planejamento do processo de suporte e manutenção para os produtos, situação que ficava subentendida no indicador e que poderia permitir variabilidade de pontuações. Sendo assim, o cenário mais deficiente passou a citar empresas sem planejamento de suporte e manutenção, além da ausência de recursos para tal.

Alguns indicadores de *performance* de qualidade de software, a saber *Funcionalidade dos Produtos (QS1)*, *Confiabilidade (QS2)*, *Usabilidade (QS3)*, *Eficiência (QS4)* e *Manutenibilidade (QS5)*, passaram a incluir, em seus piores cenários, as empresas que não conhecem as respectivas *performances* de seus produtos. Além disso, verificou-se, durante o processo de aplicações-piloto, a necessidade de se exigir a medição e avaliação formal de cada uma das características de cada indicador para que se possa pontuar na nota máxima cinco, sendo que o cenário de excelência passou a incorporar a frase: “característica é medida e avaliada”.

Para finalizar, além de outras pequenas correções realizadas em outros indicadores, que não necessitam ser comentadas neste trabalho, verificou-se a necessidade, no indicador de *Rotatividade da Mão-de-Obra* (referente ao pessoal permanente), de que futuramente seja agregada a informação de quais níveis de rotatividade correspondem a cada um dos cenários de pontuação, minimizando subjetividades na avaliação. O índice numérico, hoje coletado à parte no indicador, estará disponível para as devidas análises quando da formação do banco de dados futuro, o que permitirá maior objetividade na descrição dos cenários.

6 Conclusões e Recomendações

6.1 Conclusões

O presente trabalho permitiu o desenvolvimento de um método para *benchmarking* de práticas (métodos e ferramentas gerenciais adotados) e *performance* (resultados alcançados) em empresas de software, baseado num modelo geral para a avaliação e comparação entre empresas do setor, que reúne práticas de referência atualmente disseminadas pelos diferentes modelos e normas com relação a qualidade, inovação, processo de desenvolvimento e gestão organizacional, e baseando-se ainda nos modelos de *benchmarking* adotados pelo IEL/SC. O objetivo pretendido, de se dispor de uma ferramenta capaz de orientar as empresas do setor de software brasileiro, estratégica e operacionalmente, no seu processo de gestão, foi alcançado.

Com base em ampla pesquisa sobre práticas de referência, foi elaborado um questionário como ferramenta para a avaliação das empresas, contendo indicadores de práticas e resultados relacionados a: gestão organizacional, gestão da qualidade, avaliação de desempenho, inovação, desenvolvimento de produtos, engenharia de software, qualidade de software e gerenciamento de projetos.

Também com base nos métodos Benchstar e *Benchmarking* Industrial do IEL/SC, foram definidas as etapas do procedimento, métodos de cálculo e análise das pontuações dos indicadores e áreas de avaliação, trazendo as aprendizagens já validadas nos métodos acima citados, desde a coleta de informações nas empresas até a apresentação dos resultados, de maneira a orientar decisões de investimento nas mesmas.

O resultado das aplicações-piloto e avaliações de satisfação, conduzidas em três empresas desenvolvedoras de software localizadas em Santa Catarina, sendo duas micro-empresas e uma pequena empresa, levam a concluir que o método é bastante eficaz na avaliação e comparação de práticas e *performance* de gestão.

Através da aplicação do modelo, verificou-se a deficiência comum nas três empresas referente aos limitados investimentos em práticas de gestão organizacional e gestão da inovação. Pouco investimento no desenvolvimento das pessoas e pouco planejamento do ciclo de vida das famílias de produtos impactaram nessas áreas. As *performances* das diferentes áreas de avaliação ficaram bem pontuadas em duas das três empresas estudadas, com destaque para a

gestão da produção e gestão da qualidade, devido ao bom cumprimento de custos dos desenvolvimentos e boa funcionalidade e eficiência dos softwares desenvolvidos.

Além dos resultados obtidos e análises permitidas pelas aplicações-piloto, a pesquisa de avaliação junto às equipes de colaboradores das empresas avaliadas apontou, como resultado, que as empresas consideraram o método, juntamente com a participação do facilitador no processo, capazes de identificar de forma completa as principais deficiências e pontos fortes, servindo como orientação para o foco de investimentos e aperfeiçoamento. Enfatizaram, no entanto, a necessidade de se realizar um processo completo de *benchmarking*, devendo ser buscada a formação de um banco de dados com diversas empresas para comparação.

A grande contribuição do presente trabalho foi atingida, gerando um método que define um modelo de referência para a avaliação da gestão no nível individual das empresas de software, focando não somente os aspectos técnicos, mas também os de negócio, orientando as empresas em direção às melhores práticas e resultados, por meio de um processo rápido e simples de *benchmarking*.

6.2 Limitações e Recomendações

Apesar dos bons resultados obtidos com a aplicação-piloto do método em três empresas de software, aponta-se como limitação do presente trabalho a ausência de validação do modelo por especialistas e profissionais do setor de software, bem como a sua aplicação em uma amostra mais ampla e significativa de empresas.

Dessa forma, recomenda-se primeiramente como trabalho futuro a validação do modelo e método por especialistas da área, sejam profissionais ou acadêmicos, que o avaliem no que diz respeito ao seu potencial como avaliação de melhores práticas existentes e resultados alcançados.

Recomendam-se também pesquisas acadêmicas que investiguem a necessidade e possibilidade de definição de diferentes pesos e critérios para uma ponderação de importância e impacto entre os diferentes indicadores e áreas de avaliação existentes no método, a exemplo de trabalhos como o de Santos (2005), que adotou a técnica de *Analytic Hierarchy Process* (AHP) em um método de *benchmarking* para avaliação de cursos superiores.

Recomenda-se ainda, para a continuidade da presente pesquisa, a aplicação do método num amplo número de empresas do setor de software, inclusive de médio e grande porte, permitindo a validação do mesmo e a formação de um banco de dados representativo, viabilizando estudos futuros como:

- Comportamento de empresas de médio e grande porte;
- *Benchmarking* de abrangência regional e/ou nacional;
- Evolução de empresas no tempo, através de aplicações anuais do método;
- Verificação do nível de práticas e *performance* das empresas líderes e retardatárias;
- Análises estatísticas da relação práticas *vs.* *performance*, bem como outras relações;
- Futura aplicação em empresas estrangeiras, permitindo um *benchmarking* mundial.

Referências Bibliográficas

ABES - Associação Brasileira das Empresas de Software. Mercado Brasileiro de Software 2007. Disponível em <<http://www.abes.org.br>>. Acessado em: 22 julho 2007.

ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *ABNT Net*. Disponível em: <<http://www.abntnet.com.br>>. Acessado em: 05 junho 2007.

ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR ISO/IEC 12207*: Tecnologia da informação – Processos de ciclo de vida de software. Rio de Janeiro: ABNT, 1998. 35 p.

ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR ISO/IEC 9126-1*: Engenharia de Software – Modelo de qualidade. Rio de Janeiro: ABNT, 2003. 21 p.

AHMED, F.; CAPRETZ, L. F. Managing the business of software product line: An empirical investigation of key business factors. *Information and Software Technology*, v. 49, n. 2, p. 194-208, 2007.

AHMED, F.; CAPRETZ, L. F.; SHEIKH, S. A. Institutionalization of software product line: An empirical investigation of key organizational factors. *The Journal of Systems and Software*, v. 80, n. 6, p. 836-849, 2007.

AMBLER, S. W. *A Manager's Introduction to The Rational Unified Process (RUP)*. Toronto: Ambysoft, 2005. 24 p.

AMBLER, S. W. *Enterprise Unified Process (EUP) Home Page*. Disponível em: <<http://www.enterpriseunifiedprocess.com/>>. Acessado em: 01 julho 2007.

ANACLETO, A. *Método e Modelo de Avaliação para Melhoria de Processos de Software em Micro e Pequenas Empresas*. 2004. 190 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Catarina, Florianópolis, 2004.

ANPEI – Associação Nacional de Pesquisa, Desenvolvimento e Engenharia das Empresas Inovadoras. *Como alavancar a inovação tecnológica nas empresas*. São Paulo: [S.n.], 2004. 138 p.

ARAÚJO, E. E. R.; MEIRA, S. R. L. *Inserção Competitiva do Brasil no Mercado Internacional de Software*. Coletânea MDIC. Secretaria de Tecnologia Industrial, Comitê

Executivo de Comércio Eletrônico. 2004. 15 p. Disponível em:
<<http://ce.desenvolvimento.gov.br/software>>. Acessado em: 08 janeiro 2007.

ARVANITIS, S.; HOLLENSTEIN, H. Innovative Activity and Firm Characteristics – A cluster analysis with firm-level data of swiss manufacturing. In: EARIE 25th ANNUAL CONFERENCE, 25, 1998, Copenhagen. *Resumos*. Zurique: ETHZ, Agosto 1998. 31 p.

AZEVEDO, L. A. *Benchmarking para Instituições de Educação Tecnológica: ferramenta para a competitividade*. 2001. 275 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2001.

AZUMA, M. Software Products Evaluation System: quality models, metrics and processes – International Standards and Japanese Practice. *Information and Software Technology*, v. 38, p. 145-154, 1996.

BLASCHEK, J. R. S. Metodologia para Avaliação de Sistemas de Informação. In: ROCHA, A. R. C; WEBER, K. C. *Qualidade de Software*: seleção de textos. Curitiba: CITS, 1996. p. 109-124.

BONA, C. *Avaliação de Processos de Software: um estudo de caso em XP e ICONIX*. 2002. 122 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002.

BRASIL. Ministério da Ciência e Tecnologia. Secretaria de Política de Informática. *A qualidade no setor de software brasileiro*. In: ENCONTRO DA QUALIDADE E PRODUTIVIDADE EM SOFTWARE, 2005, São Paulo. Apresentação. São Paulo: agosto 2005a. 40 p.

BRASIL. Ministério da Ciência e Tecnologia. Secretaria de Política de Informática. *Qualificação CMM e CMMI no Brasil*. São Paulo: agosto 2006. 5 p.

BRASIL. Ministério do Planejamento, Orçamento e Gestão. Instituto Brasileiro de Geografia e Estatística. *Pesquisa Industrial – Inovação Tecnológica 2003*. Rio de Janeiro, 2005b. 148 p.

BURGELMAN, R. A.; CHRISTENSEN, C. M.; WHEELRIGHT, S. C. *Strategic management of technology and innovation*. New York: McGraw-Hill/Irwin, 2004. 1208 p.

CAMP, R. C. *Benchmarking: o caminho da qualidade total*. 3. ed. São Paulo: Pioneira, 1998. 250 p.

CARDOSO, D; SOUZA, A. A. Benchmarking & Competitividade Empresarial: O caso da área de laminação da companhia siderúrgica Belgo Mineira. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 19, 1999, Rio de Janeiro. *Anais do CD-ROM do XIX Encontro Nacional de Engenharia de Produção*. Rio de Janeiro: ENEGEP, 1999.

CARVALHO, A.; PIRES, C. G. S. Exportação de Serviços de Software: A Oportunidade do Ceará. *Science of Computing: Journal of Computer Science*, v. 1, n. 1, p. 5-19, 2007.

CASAROTTO, R. M. *Redes de Empresas na Indústria da Construção Civil: definição de funções e atividades de cooperação*. 2002. 226 f. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002.

CNI - Confederação Nacional da Indústria; SEBRAE - Serviço Brasileiro de Apoio às Micro e Pequenas Empresas. *Indicadores de Competitividade na indústria brasileira: micro e pequenas empresas*. Brasília: CNI, 2006. 134 p.

COLOMBO, R. M. T. *Processo de Avaliação da Qualidade de Pacotes de Software*. 2004. 183 f. Dissertação (Mestrado em Engenharia Mecânica) – Universidade Estadual de Campinas, Campinas, 2004.

CORNELSEN, S. G.; BUORO, F.; SBRAGIA, R.; LARUBIA, A. C. Inovação nas pequenas indústrias: resultados do Programa Mobilização Tecnológica. In: SBRAGIA, R.; STAL, E. *Tecnologia e Inovação: experiências de gestão na micro e pequena empresa*. São Paulo: PGT/USP, 2002. p. 237-262.

COSTA, C. A. A aplicação da linguagem de modelagem unificada (UML) para o suporte ao projeto de sistemas computacionais dentro de um modelo de referência. *Gestão e Produção*, v. 8, n. 1, p. 19-36, 2001.

DACORSO, A. L. R.; YU, A. S. O. Inovação e risco na pequena empresa. In: SBRAGIA, R.; STAL, E. *Tecnologia e Inovação: experiências de gestão na micro e pequena empresa*. São Paulo: PGT/USP, 2002. p. 217-236.

DEITOS, M. L. M. S. *A gestão da tecnologia em pequenas e médias empresas: fatores limitantes e formas de superação*. Cascavel: Edunioeste, 2002. 162 p.

DEY, P. K. Benchmarking project management practices of Caribbean organizations using analytic hierarchy process. *Benchmarking: An International Journal*, v. 9, n. 4, p. 326-356, 2002.

DOLCI, D. B.; BECKER, J. L. *Desenvolvimento de Sistemas de Informação com Objetos de Negócio: Proposta de um Guia para Análise da Reutilização de Software*. In: ENCONTRO

DA ASSOCIAÇÃO NACIONAL DE PÓS-GRADUAÇÃO E PESQUISA EM ADMINISTRAÇÃO, 23, 1999, Foz do Iguaçu. Apresentação. Foz do Iguaçu, 1999. 15p.

DUARTE, L. S. *Caracterização da Inovação Tecnológica no Setor de Software de Gestão Integrada: estudos de casos nas empresas de base tecnológica do estado de São Paulo*. 2003. 157 f. Dissertação (Mestrado em Desenvolvimento Econômico, Espaço e Meio Ambiente) – Instituto de Economia – Universidade Estadual de Campinas, Campinas, 2003.

FERREIRA, M. P. *Desenvolvimento de Software Alinhado aos Objetivos Estratégicos do Negócio: proposta de uma metodologia*. 2002. 194 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002.

GARCIA, L. F. F. *Introdução à Engenharia de Software*. Universidade Luterana do Brasil. Canoas: Ulbra, [2003?].

GLASS, R. L. Software maintenance, Y2K and other software non-crises. *The Journal of Systems and Software*, v. 54, n.2, p. 83-85, 2000a.

GLASS, R. L. Talk About a Software Crisis – Not! *The Journal of Systems and Software*, v. 55, n.1, p. 1-2, 2000b.

GOMES, S. B.; FALBO, R. A.; MENEZES, C. S. *Um Modelo para Acordo de Nível de Serviço em TI*. Disponível em: <<http://www.inf.ufes.br/~falbo/download/pub/2005-Sbqs.pdf>>. Acessado em: 22 novembro 2007.

GUTIERREZ, R. M. V.; ALEXANDRE, P. V. M. Complexo Eletrônico: introdução ao software. *BNDES Setorial*, Rio de Janeiro, n. 20, p. 3-76, set. 2004.

HAUSMAN, A. Innovativeness among small businesses: Theory and propositions for future research. *Industrial Marketing Management*, v. 34, n. 8, p. 773-782, 2005.

IEEE: INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. *SWEBOK: guide to the software engineering body of knowledge*. EUA: IEEE, 2004. 204 p.

IEL/SC: INSTITUTO EUVALDO LODI DE SANTA CATARINA. *Manual de Treinamento Teórico do Benchmarking Industrial*. Florianópolis: IEL/SC, 2005a.

IEL/SC: INSTITUTO EUVALDO LODI DE SANTA CATARINA. *Manual de Treinamento Teórico do Benchstar*. Florianópolis: IEL/SC, 2005b.

ISO: INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 14102*: Information technology - Guideline for the evaluation and selection of CASE tools. Genebra: ISO/IEC, 1995.

ISO: INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 15504-2*: Information technology - Process assessment - Part 2: Performing an assessment. Genebra: ISO/IEC, 2003. 26 p.

ISO: INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC 15504-5*: Information technology - Process Assessment - Part 5: An exemplar Process Assessment Model. Genebra: ISO/IEC, 2006. 170 p.

ISO: INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC Standards*. Disponível em: <<http://www.iso.org>>. Acessado em: 07 junho 2007.

ISO: INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *ISO/IEC TR 14471*: Information technology – Software engineering – Guidelines for the adoption of CASE tools. Genebra: ISO/IEC, 1999.

JONG, J.P.J.; MARSILI, O. The fruit flies of innovations: A taxonomy of innovative small firms. *Research Policy*, v. 35, n. 2, p. 213-229, 2006.

KAUSHIK, S. The software engineering model's categories of compliance. *Computer Standards & Interfaces*, doi:10.1016/j.csi.2006.11.011, 2008.

KIVIMÄKI, M.; LÄNSISALMI, H.; ELOVAINIO, M.; HEIKKILÄ, A.; LINDSTRÖM, K.; HARISALO, R.; SIPILÄ, K.; PUOLIMATKA, L. Communication as a determinant of organizational innovation. *R&D Management*, v. 30, n. 1, p. 33-42, 2000.

KRUCHTEN, P. *What is the Rational Unified Process?* Revista eletrônica da comunidade *The Rational Edge*. 2001. Disponível em <<http://www.therationaledge.com/content>>. Acessado em: 8 dezembro 2006.

KUBOTA, L. C. *O impacto da gestão nos resultados de empresas brasileiras de software*. 2007. 204 f. Tese (Doutorado em Administração) – Instituto Coppead de Administração, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

LEAL, F.; ALMEIDA, D. A. Uma análise da aplicação integrada de técnicas de mapeamento de processo com foco no cliente: estudo de caso do processo de atendimento de uma agência bancária. In: Simpósio de Engenharia de Produção, 10, 2003, Bauru. SIMPEP 2003: *Anais...* Bauru: FEB/UNESP, 2003. 11 p.

LEONARD-BARTON, D. *Nascentes do saber: criando e sustentando as fontes de inovação*. Rio de Janeiro: Fundação Getúlio Vargas, 1998. 368 p.

LUIZ, R. R. V. *Obtendo Qualidade de Software com o RUP*. Sistemas de Informação – Universidade de Uberaba – UNIUBE. Uberaba, 2006.

MACHADO, C. A. F.; BURNETT, R. C. *Gerência de projetos na engenharia de software em relação às práticas do PMBoK*. Companhia de Informática do Paraná – CELEPAR. 2002. 10p. Disponível em <<http://celepar7cta.pr.gov.br/portfolio.nsf>>. Acessado em: 10 junho 2008.

MARINI, M. J. *Uma ferramenta de suporte à avaliação da qualidade de software de aplicativos voltados à gestão empresarial*. 2002. 108 f. Dissertação (Mestrado em Ciências da Computação) – Universidade Federal de Santa Catarina, Florianópolis, 2002.

MARTINS, L. E. G. *Uma metodologia de elicitação de requisitos de software baseada na teoria da atividade*. 2001. 182 f. Tese (Doutorado em Engenharia Elétrica e de Computação) – Universidade Estadual de Campinas, Campinas, 2001.

MARTINS, L. E. G; JÚNIOR, V. B. *A Metodologia OMT*. Apostila. Piracicaba, 1999.

MASSA, S.; TESTA, S. Innovation or imitation? Benchmarking: a knowledge-management process to innovate services. *Benchmarking: An International Journal*, v. 11, n. 6, p. 610-620, 2004.

MAZO, E. M. *Benchstar – metodologia de benchmarking para análise da gestão da produção nas micro e pequenas empresas*. 2003. 174 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2003.

MEJRI, M.; KTARI, B.; FUJITA, H.; ERHIOUI, M. Static analysis of Lyee requirements. *Knowledge-Based Systems*, v. 16, n. 7, p. 361–382, 2003.

MELNIKOFF, S. S. *Engenharia de Software 1*. Apostila. Escola Politécnica da USP – Departamento de Engenharia de Computação e Sistemas Digitais, 1992.

NAKAJIMA, T. T. *Avaliação do Comportamento das Empresas em Relação à Qualidade no Processo de Desenvolvimento de Software*. 2004. 121 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2004.

NATO – North Atlantic Treaty Organization. Software Engineering. In: NATO SOFTWARE ENGINEERING CONFERENCE, 1969, Garmisch. *Report of NATO Science Committee*. Brussels: Scientific Affairs Division, NATO, 1969. 231 p.

NOGUEIRA, M. Qual a importância da adoção da norma ISO 12207 nas empresas de desenvolvimento de software? In: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 10, 2003, Bauru. *Artigo*. Bauru, 2003. 11p.

OECD: ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT. *OECD Information Technology Outlook: information and communications technologies*. Paris: OECD Publishing, 2006. 315 p.

OECD: ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT. *Oslo Manual: guidelines for collecting and interpreting innovation data*. 3. ed. Paris: OECD Publishing, 2005. 164 p.

OTERO, D. A. *Motivação dos teletrabalhadores em grupos de desenvolvimento de software*. 2002. 139 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002.

PEREIRA, J., et al. What do software practitioners really think about project success: a cross-cultural comparison. *The Journal of Systems and Software*, v. 81, n. 6, p. 897-907, 2008.

PMI: PROJECT MANAGEMENT INSTITUTE. *PMBOK: um guia do conjunto de conhecimentos em gerenciamento de projetos*. 3. ed. Pennsylvania: Project Management Institute, 2004. 388 p.

PMI: PROJECT MANAGEMENT INSTITUTE. *Relatório Final – Estudo de Benchmarking em Gerenciamento de Projetos - Brasil 2006*. Project Management Institute. 2006. 289 p. Disponível em: <<http://www.pmirio.org.br/estudo2006>>. Acessado em: 18 agosto 2007.

PORTER, M. *A Vantagem Competitiva das Nações*. Rio de Janeiro: Campus, 1993. 897p.

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

QUINTELLA, H. L. M. M; ROCHA, H. M. Avaliação da Maturidade do Processo de Desenvolvimento de Veículos Automotivos. *Gestão e Produção*. v.13, n.2, p.297-310, 2006.

REIS, C. *Caracterização de um Modelo de Processo para Projetos de Software Livre*. Monografia (Ciências da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, São Carlos, 2001.

REZENDE, D. A. *Engenharia de Software e Sistemas de Informação*. Rio de Janeiro: Brasport, 1999. 292p.

ROCHA, P. C.; BELCHIOR, A. D. Mapeamento do Gerenciamento de Riscos no PMBOK, CMMI-SW e RUP. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSOS DE SOFTWARE, 6., 2004, São Paulo. *Anais...* São Paulo: SIMPROS, novembro 2004. p. 279-290.

RODRIGUES, L. G. M. *Um modelo de avaliação dos requisitos no processo de desenvolvimento de software*. 2006. 74 f. Dissertação (Mestrado Profissional em Computação) – Instituto de Computação da Universidade Estadual de Campinas, Campinas, 2006.

ROQUE, R. F. *Estudo Comparativo de Metodologias de Desenvolvimento de Sistemas de Informação utilizando a Técnica Delphi*. 1998. 189 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1998.

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. *The Unified Modeling Language Reference Manual*. Boston: Addison-Wesley, 1999. 550p.

SANTOS, C. A. S. *Avaliação de Cursos Superiores de Tecnologia*. 2005. 188 f. Tese (Doutorado em Engenharia de Produção e Sistemas) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2005.

SANTOS, J. G. *Aspectos Sociais no Alinhamento entre TI e Negócio: estudo de caso de uma empresa de manutenção de turbinas aeronáuticas*. 2006. 179 f. Dissertação (Mestrado Profissionalizante em Administração) – Faculdade de Economia e Finanças IBMEC, Rio de Janeiro, 2006.

SAUR, R. A. C. *Perspectivas e projeções da indústria global de software e serviços*. Coletânea MDIC, Secretaria de Tecnologia Industrial, Comitê Executivo de Comércio Eletrônico. 2004. 21p. Disponível em <<http://ce.desenvolvimento.gov.br/SOFTWARE>>. Acessado em: 08 janeiro 2007.

SCHUMPETER, J. *The theory of economic development*. Cambridge Massachusetts: Harvard University Press, 1934. 170 p.

SEI: SOFTWARE ENGINEERING INSTITUTE. *Capability Maturity Model Integration (CMMISM) – CMMISM for Software Engineering – Versão 1.1*. Pittsburgh: Carnegie Mellon University, 2002. 639 p.

SEI: SOFTWARE ENGINEERING INSTITUTE. *Personal Software Process (PSPSM)*. Pittsburgh: Carnegie Mellon University, 2000. 55 p.

SEI: SOFTWARE ENGINEERING INSTITUTE. *The State of Software Measurement Practice: results of 2006 survey*. Pittsburgh: Carnegie Mellon University, 2006. 67 p.

SEIBEL, S. *Um modelo de benchmarking baseado no sistema produtivo classe mundial para avaliação de práticas e performances da indústria exportadora brasileira*. 2004. 217 f. Tese (Doutorado em Engenharia de Produção e Sistemas) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2004.

SILVA, E. L.; MENEZES, E. M. *Metodologia da Pesquisa e Elaboração de Dissertação*. 3.ed. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001. 121p.

SOARES, M. S. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. *Revista Eletrônica de Sistemas de Informação*. v.3, n.1, 2004.

SOFTEX: ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. *Pesquisa Censo SW 2001*. 2001. Disponível em <<http://www.mct.gov.br>>. Acessado em: 12 julho 2008.

SOFTEX: ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. *A indústria de software no Brasil 2002: fortalecendo a economia do conhecimento*. Campinas: SOFTEX, 2002. 80 p.

SOFTEX: ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. *Perfil das Empresas Brasileiras Exportadoras de Software*. Campinas: DPCT/UNICAMP e SOFTEX, 2005. 51 p.

SPALDING, M. *Sobrevivência de micro e pequenas empresas: sua relação com análise dos ambientes internos, externos e com a inovação*. 2006. 134 f. Dissertação (Mestrado em Gestão e Desenvolvimento Regional) – Departamento de Economia, Contabilidade e Administração, Universidade de Taubaté, Taubaté, 2006.

TORNATZKY, L. G.; FLEISCHER, M. *The processes of technological innovation*. Toronto: Lexington Books, 1990. 298p.

VASCO, C. G.; VITHOFT, M. H.; ESTANTE, P. R. C. *Comparação entre Metodologias RUP e XP*. Programa de Pós Graduação em Informática Aplicada. Curitiba: Pontifícia Universidade Católica do Paraná, 2005. 8 p.

WEBER, K. et al. *Melhoria de Processo do Software Brasileiro (MPS.BR): um Programa Mobilizador*. SOFTEX – Associação para Promoção da Excelência do Software Brasileiro. 2006. 10p. Disponível em <http://www.softex.br/mpsbr/_home/default.asp>. Acessado em: 03 março 2007.

ZANATTA, A. L. *xScrum: uma proposta de extensão de um Método Ágil para Gerência e Desenvolvimento de Requisitos visando adequação ao CMMI*. 2004. 180 f. Dissertação (Mestrado em Ciências da Computação) – Universidade Federal de Santa Catarina, Florianópolis, 2004.

ZAPELINI, W. B. *Um Modelo de Avaliação de Programas de Pós-Graduação Baseado no Benchmarking de Competências Organizacionais: estudo de caso nas engenharias da UFSC*. 2002. 254 f. Tese (Doutorado em Engenharia de Produção e Sistemas) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002.

Apêndices

Apêndice 1 – Tabela de indicadores por área: resultado da aplicação-piloto

Indicadores de Prática

Prática de Gestão da Produção		média	moda
ES9	Ferramentas Apoio Engenharia Software (CASE)	2,7	3,0
DP4	Feedback de campo	3,3	3,0
ES4	Engenharia de Requisitos	3,3	4,0
DP3	Controle de mudanças	3,7	4,0
QS6	Documentação dos Produtos	3,7	4,0
ES2	Iteratividade no Desenvolvimento	3,7	5,0
DP1	Simultaneidade no processo de engenharia	4,0	-
ES6	Implantação do Produto no Cliente	4,0	4,0
DP2	Envolvimento externo	4,0	-
ES1	Modelo de Processo de Desenvolvimento	4,3	5,0
DP5	Projeto para produção e manutenção	4,3	5,0
ES3	Análise e Modelagem de Arquitetura Software	4,3	5,0
ES7	Suporte e Manutenção	5,0	5,0

Prática de Gestão da Qualidade		média	moda
ES8	Maturidade dos Processos	2,7	3,0
QS7	Métricas de Software	2,7	3,0
GQ5	Pesquisa das necessidades dos clientes	3,0	-
AD3	Medidas de desempenho	3,7	3,0
GO4	Participação dos empregados	4,0	-
GQ1	Visão da qualidade	4,0	4,0
GQ4	Fornecedores e subcontratados	4,0	-
ES5	Testes de Software	4,3	4,0
GO7	Orientação para o cliente	4,7	5,0

Prática de Gestão Organizacional		média	moda
GO6	Desenvolvimento de pessoal	1,7	2,0
GO3	Estilo de administração	2,7	3,0
GO5	Política de incentivos	3,0	-
GO1	Declaração e compartilhamento da visão	3,0	4,0
GO11	Sistemas de informação	3,3	3,0
GO10	Planejamento de marketing	3,3	-
AD3	Medidas de desempenho	3,7	3,0
GO2	Planejamento e análise das metas	3,7	4,0
GO9	Conhecimento da concorrência	4,0	-

Prática de Gestão de Projetos		média	moda
GP8	Gerenciamento do Nível de Serviço	3,0	-
GP2	Gerenciamento de Riscos	3,3	3,0
GP3	Gerenciamento de Tempo	3,7	4,0
GP1	Organização para Projetos	4,0	4,0
GP4	Gerenciamento de Custos	4,0	4,0
GP6	Gerenciamento de Comunicação	4,0	4,0
GP7	Ferramenta Apoio ao Gerenciamento Projeto	4,3	5,0
GP5	Gerenciamento de Escopo	4,7	5,0

Prática de Gestão da Inovação		média	moda
GO6	Desenvolvimento de pessoal	1,7	2,0
IN2	Planejamento do ciclo de vida do produto	2,0	-
IN7	Estratégia de tecnologia de Produto e Processo	3,0	-
IN6	Estratégia de obtenção novas tecnologias	3,0	4,0
IN5	Relac. universidades e centros pesquisa	3,3	3,0
IN1	Geração de conceitos de produtos inovadores	3,7	4,0
IN8	Ambiente inovativo	4,0	4,0

Indicadores de Performance

Performance de Gestão da Produção		média	moda
DP6	Tempo de ciclo de desenvolvimento	2,0	2,0
QS2	Confiabilidade	4,0	-
GP9	Cumprimento de Prazos	4,0	-
AD5	Prazo de entrega dos subcontratados	4,3	5,0
QS5	Manutenibilidade	4,3	4,0
GP10	Cumprimento de Custos	4,3	5,0

Performance de Gestão da Qualidade		média	moda
AD1	Satisfação do cliente	3,7	4,0
GQ3	Reclamação de clientes	3,7	-
QS3	Usabilidade	4,0	4,0
GQ2	Qualidade produtos entregues subcontratados	4,0	-
QS1	Funcionalidade dos Produtos	4,3	4,0
QS4	Eficiência	4,7	5,0

Performance de Gestão Organizacional		média	moda
GP11	Funcionamento da Comunicação	3,0	-
GO8	Rotatividade da mão-de-obra	4,3	4,0
AD4	Lucratividade	4,3	5,0
AD2	Variação nas vendas	5,0	5,0

Performance de Gestão de Projetos		média	moda
GP11	Funcionamento da Comunicação	3,0	-
AD1	Satisfação do cliente	3,7	4,0
GQ3	Reclamação de clientes	3,7	-
GP9	Cumprimento de Prazos	4,0	-
GP10	Cumprimento de Custos	4,3	5,0

Performance de Gestão da Inovação		média	moda
IN4	Introdução de novos produtos	3,0	-
IN3	Capacidade de inovação	3,0	2,0
AD2	Variação nas vendas	5,0	5,0

Apêndice 2 – Questionário de benchmarking desenvolvido

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)