

Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica e de Computação  
Departamento de Comunicações

**“ANÁLISE COMPARATIVA DE METODOLOGIAS PARA O  
GERENCIAMENTO DE PROJETOS DE DESENVOLVIMENTO DE  
SOFTWARE”**

AUTOR: ANDERSON LUIZ BARBOSA

ORIENTADOR: PROF. DR. LEONARDO DE SOUZA MENDES  
(DECOM – FEEC / UNICAMP)

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação (FEEC / UNICAMP) como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Banca Examinadora:

Prof. Dr. Márcio Luiz de Andrade Netto (DCA - FEEC / UNICAMP)

Prof. Dr. Waldomiro Pelágio Diniz de Carvalho Loyolla (UNIEMP)

CAMPINAS, SP, BRASIL

JUNHO/2006

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

B234a                      Barbosa, Anderson Luiz  
                                 Análise comparativa de metodologias para o  
                                 gerenciamento de projetos de desenvolvimento de software /  
                                 Anderson Luiz Barbosa. --Campinas, SP: [s.n.], 2006.

                                 Orientador: Leonardo de Souza Mendes  
                                 Dissertação (Mestrado) - Universidade Estadual de  
                                 Campinas, Faculdade de Engenharia Elétrica e de  
                                 Computação.

                                 1. Engenharia de software. 2. Administração de projetos.  
                                 3. Software - Produtividade. 4. Software - Desenvolvimento.  
                                 5. Programação (Computadores) – Gerência. 6. Controle de  
                                 qualidade. 7. Metodologia. I. Mendes, Leonardo de Souza.  
                                 II. Universidade Estadual de Campinas. Faculdade de  
                                 Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Comparative analysis of software development project management methodologies.

Palavras-chave em Inglês: Project management, Software engineering, Software quality.

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Márcio Luiz de Andrade Netto e Waldomiro Pelágio Diniz de Carvalho Loyolla.

Data da defesa: 30/06/2006

## Resumo

As empresas ao redor do mundo começam a adotar, de forma mais intensa, os padrões internacionais de gerenciamento de projetos (como o *PMBok Guide* e a ISO 10006); profissionais são capacitados para atuarem como gerentes de projetos nas mais diversas áreas de atividades. Mundialmente existe a preocupação com o gerenciamento adequado desses projetos a fim de garantir o atendimento às expectativas dos envolvidos e a otimização dos recursos alocados. Na área de software, as preocupações são cada vez maiores com relação a custos, prazos e qualidade em projetos de desenvolvimento de sistemas, forçando um repensar nas práticas até então utilizadas para o gerenciamento destes processos. O presente trabalho aborda as técnicas de engenharia de software que tradicionalmente são aplicadas ao desenvolvimento de projetos, constatando que, na maior parte das vezes, são enfocados apenas os aspectos técnicos; apresenta também uma descrição dos padrões de qualidade de uso genérico (como as normas ISO 9000) que passam a ser utilizados por empresas que desenvolvem software, e descreve a utilização de modelos de qualidade especificamente desenvolvidos para informática, como o CMM/CMMI e o PSP. A partir dos resultados da comparação entre os modelos de qualidade e os modelos de gerenciamento de projetos, é apresentada uma proposta para otimização do gerenciamento de processos de desenvolvimento de software, descrevendo a contribuição de cada um destes modelos, além das adaptações necessárias a fim de contemplar os aspectos específicos de software. Essa proposta é validada com um estudo de caso em um laboratório de desenvolvimento de projetos na área de telecomunicações com ênfase no emprego e desenvolvimento de software.

**Palavras-chave:** Gerenciamento de projetos, Engenharia de Software, Qualidade de Software.

## **Abstract**

International standards of project management (like PMBoK Guide and ISO 10006) have been used by companies around the world; professionals have been qualified to exercise the project management function in most areas of activities. A global concern exists for the appropriate administration of these several projects in order to achieve the expectations of those involved. On the software area, the concern is bigger when related to costs, time and quality in software development. This forces to rethink the practices used on the administration of these processes. This work addresses the techniques of software engineering that are being applied to the development of projects, verifying that most of the time spent is just focused on the technical aspects; it also presents a description of the patterns of quality for generic use (as the norms ISO 9000) that also have been used by companies that develop software, and it specifically describes the use of quality models developed for computer science, like CMM/CMMI and PSP. A proposal is presented for optimizing the management of software development processes, describing the contribution provided by each one of these models, together with the necessary adaptations in order to contemplate the specific aspects of software. The proposal is validate with a case study in a laboratory of telecommunications projects with a real software development project.

**Keywords:** Project Management, Software Engineering, Software Quality.

A Deus, o criador, pela dádiva da existência, o grande projeto.

A Débora, Andréia e Thiago, que dão sentido a este projeto,  
pelo apoio, carinho e paciência que tiveram durante o período  
deste trabalho.

## **Agradecimentos**

A meus pais, José Milton Barbosa e Margarida Maria Faria Barbosa, e a todos os familiares e amigos, pelo incentivo constante.

Ao professor Doutor Leonardo de Souza Mendes, pelo constante apoio. Sua dedicação, atenção dispensada aos seus orientados e sua determinação para o trabalho constituem um exemplo que certamente influenciarão por toda a minha vida.

Aos estimados professores e funcionários da Faculdade de Engenharia Elétrica e de Computação da Unicamp e do Departamento de Comunicações, pela convivência e troca de experiência e pela oportunidade de fazer o Mestrado.

Aos colegas professores, alunos, funcionários e direção do Centro Universitário Salesiano de São Paulo (UNISAL), pela oportunidade de crescermos juntos.

Agradeço aos Professores Márcio Andrade Netto, Max Henrique Machado Costa e Waldomiro Pelágio Diniz de Carvalho Loyolla pelas valiosas contribuições apresentadas para a conclusão deste trabalho.

A todos os colegas do LarCom e da IgnisCom, em especial Maurício e Meire, meus sinceros agradecimentos.

Por fim, agradeço à direção do Centro UNISAL pelo auxílio e apoio concedidos aos meus estudos.

# Sumário

<b>RESUMO .....</b>	<b>III</b>
<b>ABSTRACT .....</b>	<b>IV</b>
<b>DEDICATÓRIA .....</b>	<b>V</b>
<b>AGRADECIMENTOS.....</b>	<b>VII</b>
<b>SUMÁRIO .....</b>	<b>IX</b>
<b>LISTA DE FIGURAS.....</b>	<b>XIII</b>
<b>LISTA DE TABELAS .....</b>	<b>XV</b>
<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>XVII</b>
<b>CAPÍTULO 1 INTRODUÇÃO .....</b>	<b>1</b>
1.1    O PROBLEMA DO GERENCIAMENTO DE PROJETOS DE SOFTWARE .....	1
1.2    ESTRUTURA DA DISSERTAÇÃO.....	4
<b>CAPÍTULO 2 GERENCIAMENTO DE PROJETOS .....</b>	<b>7</b>
2.1    CONCEITUAÇÃO .....	7
2.1.1 <i>Conceito de projeto</i> .....	7
2.2    ORGANIZAÇÕES.....	8
2.2.1 <i>Project Management Institute (PMI)</i> .....	9
2.2.2 <i>Software Engineering Institute (SEI)</i> .....	11
2.2.3 <i>International Organization for Standardization (ISO)</i> .....	12
2.2.4 <i>Institute of Electrical and Electronics Engineers (IEEE)</i> .....	16
2.2.5 <i>Outras organizações</i> .....	16
<b>CAPÍTULO 3 DESENVOLVIMENTO DE SOFTWARE .....</b>	<b>19</b>
3.1    CONCEITUAÇÃO .....	19
3.2    HISTÓRICO .....	20

3.3	ENGENHARIA DE SOFTWARE .....	23
3.3.1	<i>Paradigmas da engenharia de software</i> .....	25
3.3.1.1	Modelo clássico .....	26
3.3.1.2	Modelo de prototipação.....	29
3.3.1.3	Modelo incremental.....	31
3.3.1.4	Modelo espiral .....	32
3.3.1.5	Técnicas de quarta geração.....	34
3.3.1.6	Outros modelos de desenvolvimento .....	37
3.4	PADRÕES DE QUALIDADE .....	38
3.4.1	<i>ISO 9000</i> .....	38
3.4.1.1	Histórico .....	38
3.4.1.2	Características .....	39
3.4.1.3	ISO 90003:2004 .....	45
3.4.2	<i>ISO 9126</i> .....	46
3.4.3	<i>ISO 12207</i> .....	46
3.4.4	<i>ISO 15504 - SPICE</i> .....	49
3.4.5	<i>ISO 19759 - SWEBoK</i> .....	53
3.4.6	<i>SEI – CMM e CMMI</i> .....	55
3.4.6.1	Histórico .....	55
3.4.6.2	Conceitos utilizados .....	56
3.4.6.3	Níveis de maturidade em software .....	58
3.4.6.4	Gerenciamento de software e o CMMI.....	60
3.4.6.5	Estrutura do CMMI .....	63
3.4.6.6	Adoção do CMMI pelas empresas .....	68
3.4.7	<i>SEI - PSP</i> .....	70
3.4.8	<i>SOFTEX – mps Br</i> .....	71
<b>CAPÍTULO 4 PADRÕES DE GERENCIAMENTO DE PROJETOS .....</b>		<b>75</b>
4.1	ISO 10006 .....	75
4.2	PMI - PMBoK.....	80
4.2.1	<i>Apresentação</i> .....	80



4.2.2	<i>Contexto do gerenciamento de projetos</i> .....	82
4.2.2.1	Ciclos de vida .....	82
4.2.2.2	Estruturas organizacionais.....	82
4.2.3	<i>Descrição do processo de gerenciamento de projetos</i> .....	86
4.2.3.1	Processos de Início .....	88
4.2.3.2	Processos de Planejamento.....	88
4.2.3.3	Processos de Execução.....	92
4.2.3.4	Processos de Monitoramento e Controle.....	92
4.2.3.5	Processos de Encerramento .....	94
4.2.4	<i>Áreas do gerenciamento de projetos</i> .....	94
4.2.4.1	Gerenciamento de integração do projeto.....	95
4.2.4.2	Gerenciamento do escopo do projeto .....	96
4.2.4.3	Gerenciamento de tempo do projeto.....	96
4.2.4.4	Gerenciamento de custos do projeto.....	96
4.2.4.5	Gerenciamento da qualidade do projeto .....	96
4.2.4.6	Gerenciamento de recursos humanos do projeto.....	97
4.2.4.7	Gerenciamento das comunicações do projeto .....	98
4.2.4.8	Gerenciamento de riscos do projeto .....	98
4.2.4.9	Gerenciamento de aquisições do projeto.....	98
<b>CAPÍTULO 5</b>	<b>ANÁLISE COMPARATIVA</b> .....	<b>99</b>
<b>CAPÍTULO 6</b>	<b>ESTUDO DE CASO</b> .....	<b>109</b>
<b>CAPÍTULO 7</b>	<b>CONCLUSÕES E PERSPECTIVAS FUTURAS</b> .....	<b>117</b>
7.1	CONCLUSÕES .....	117
7.2	TEMAS SUGERIDOS COMO PERSPECTIVAS FUTURAS DA PESQUISA.....	119
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	.....	<b>121</b>
<b>ÍNDICE DE AUTORES</b>	.....	<b>131</b>

## Lista de Figuras

Figura 1 - Modelo clássico .....	26
Figura 2 - Modelo de prototipação .....	29
Figura 3 - Modelo incremental.....	31
Figura 4 - Modelo espiral.....	32
Figura 5 - Técnicas de quarta geração .....	35
Figura 6 - Programação Extrema .....	38
Figura 7 - Elementos da norma ISO 9000 .....	40
Figura 8 - Processos e visões da ISO 12207:1995 .....	49
Figura 9 - Área chave gerenciamento de engenharia de software.....	55
Figura 10 - Níveis de Maturidade CMMI .....	59
Figura 11 - Visão gerencial e o processo de software .....	60
Figura 12 - Capacidade de estimar e medir resultados dos projetos .....	62
Figura 13 - Componentes do modelo CMMI .....	63
Figura 14 - Panorama mundial das empresas que adotam o modelo CMMI .....	68
Figura 15 - Panorama nacional das empresas que adotam o modelo CMMI.....	69
Figura 16 - Modelo de referência mps Br .....	72
Figura 17 – Disciplinas relacionadas ao gerenciamento de projetos .....	81
Figura 18 - Gerenciamento de projetos em uma organização funcional .....	83
Figura 19 - Gerenciamento de projetos em uma organização por projetos.....	84
Figura 20 - Gerenciamento de projetos em organização matricial .....	85
Figura 21 - Grupos de processos de gerenciamento de projetos .....	87
Figura 22 - Gerenciamento ao longo do ciclo de vida do projeto .....	87
Figura 23 - EAP para um projeto de software .....	90
Figura 24 - Processos de gerenciamento de projetos por área de conhecimento	95
Figura 25 - WBS do Projeto Conexão do Saber São José do Rio Preto.....	113
Figura 26 - Cronograma do Projeto Conexão do Saber São José do Rio Preto .	114

## **Lista de Tabelas**

Tabela 1 - Ciclo de vida das normas ISO .....	15
Tabela 2 - Níveis ISO 9000:1994 x elementos padrões.....	43
Tabela 3 - Principais requisitos e seções da ISO 9001:2000 .....	44
Tabela 4 - Empresas certificadas pela norma ISO 9001:2000.....	44
Tabela 5 - Processos e atividades da ISO 9000-3:1997 .....	45
Tabela 6 - Elementos da ISO 9126.....	47
Tabela 7 - Níveis e atividades do modelo SEI-PSP .....	70
Tabela 8 - Estruturas com diferentes níveis de orientação por projetos .....	85

## **Lista de Abreviaturas e Siglas**

ABNT - Associação Brasileira de Normas Técnicas  
AIEE - American Institute of Electrical Engineers  
ASP - Application Services Provider  
CAD - Computer Aided Design  
CASE - Computer Aided Software Engineering  
CMM - Capability Maturity Model  
CMMI - Capability Maturity Model Integration  
CMU - Carnegie Mellon University  
CNPQ - Conselho Nacional de Desenvolvimento Científico e Tecnológico  
CRM - Customer Relationship Management  
DFD - Diagrama de Fluxo de Dados  
EAP - Estrutura Analítica de Planejamento  
FGV - Fundação Getúlio Vargas  
IEEE - Institute of Electrical and Electronics Engineers  
IRE - Institute of Radio Engineers  
ISA - International Federation of the National Standardizing Associations  
IEC - International Engineering Consortium  
ISO - International Organization for Standardization  
ITU - International Telecommunication Union  
MPS Br - Programa de Melhoria de Processo do Software Brasileiro  
OMT - Object Modeling Technique  
PMBok - Project Management Body of Knowledge  
PMI - Project Management Institute  
PMP - Project Management Professional  
PSP - Personal Software Process  
SCM - Supply Chain Management  
SEI - Software Engineering Institute  
SPICE - Software Process Improvement and Capability dEtermination  
SOFTEX - Associação para Promoção da Excelência do Software Brasileiro  
SUCESU - Sociedade de Usuários de Informática e Telecomunicações

SWEBOK - Software Engineering Body of Knowledge

UML - Unified Modeling Language

USP - Universidade de São Paulo

XP - Extreme Programming

WBS - Work Breakdown Structure

WTO - World Trade Organization

# Capítulo 1

## Introdução

### 1.1 O problema do gerenciamento de projetos de software

A demanda pela produção de software tem crescido intensamente e sistematicamente desde os primórdios do processamento de dados e surgimento dos computadores, acerca de 60 anos. Desde então o nível de exigência e complexidade dos sistemas vem evoluindo a medida que novos ambientes, novas plataformas, novas metodologias e um número cada vez maior de usuários se envolvem com estes sistemas.

No início os sistemas eram operados/manipulados por um conjunto reduzido de pessoas – a complexidade operacional e a tecnologia disponível eram caras e de difícil acesso. Informações oriundas de sistemas de informação computadorizados eram normalmente acessadas pelos usuários por intermédio de mídia impressa (listagens), com uma defasagem de tempo (atualização) grande – muitas vezes a informação era referente ao mês anterior. Hoje a demanda exige informações atualizadas quase que instantaneamente, favorecida pelo desenvolvimento de diversas mídias de acesso (disquetes, CD-Rom, DVD, Internet) disponíveis a um custo bastante acessível.

Sistemas que no passado consumiam anos de desenvolvimento, com grandes equipes, hoje sofrem pressões cada vez maiores para redução de prazo e custo mas sem redução de qualidade e de escopo (pelo contrário, cada vez aumentam estas exigências). Infelizmente a evolução dos processos de desenvolvimento – em especial os de gerenciamento – não acompanhou a evolução das exigências dos usuários.

Brooks (1995) exemplifica, com uma cena pré-histórica, alguns desses problemas:

"Nenhuma cena pré-histórica é tão marcante quanto o esforço dos grandes animais atolados nos profundos lodaçais. Imaginamos ver os dinossauros, mamutes e megatérios, lutando para não afundar no pântano. E quanto mais se debatem mais afundam até desaparecerem, pois nenhum dos animais é tão

forte que, vencendo o lodaçal, possa escapar. A confecção de grandes sistemas tem sido o mortal pântano e grandes e poderosas forças afundaram inexoravelmente nele. Muitos conseguiram produzir sistemas que funcionaram, mas poucos atingiram os objetivos e orçamentos. Equipe após equipe, independente do tamanho ou recursos, afundaram no pântano. E não foi possível identificar a causa, descobrir qual o erro e evitar. É o somatório de pequenos fatos simultâneos ou sucessivos que conduziram até o fundo da armadilha. Todos reconhecem a gravidade do problema mas poucos discernem onde está a causa. A maior parte dos projetos não atingiu seus objetivos no tempo, mais por falta de calendários de eventos<sup>1</sup> do que pelas demais causas juntas".

Este trecho foi escrito para a primeira edição do referido livro e ainda hoje é válido. Vários trabalhos, dentre os quais Spectrum (1999) e Moraes (2004), mapearam a relação entre os fatores de sucesso e fracasso no desenvolvimento de projetos de software – boa parte dos projetos não é concluída e, dentre os projetos concluídos, apenas uma pequena parte é concluída dentro da meta original de escopo, custo, prazo e qualidade.

Alguns autores procuraram identificar possíveis razões para estes problemas relativos ao desenvolvimento de software. Pressman (2002) apresenta algumas destas possíveis razões:

Gerentes de nível médio e superior sem nenhum *background* em software recebem a responsabilidade pelo seu desenvolvimento. Há um antigo axioma da administração que afirma: "um bom gerente pode gerir qualquer projeto". Deveríamos acrescentar: "...se ele estiver disposto a aprender quais os marcos (*milestones*) que podem ser usados para medir o processo, aplicar métodos efetivos de controle, não levar em conta a mitologia e tornar-se fluente numa tecnologia que se modifica rapidamente". O gerente deve comunicar-se com todas as pessoas envolvidas no desenvolvimento de software - clientes, desenvolvedores, pessoal de apoio e outros. A comunicação pode interromper-se porque as características especiais do software e dos problemas associados ao seu desenvolvimento são mal compreendidas. Quando isso ocorre, os problemas associados à crise do software são exacerbados".

---

<sup>1</sup> Entende-se por calendário de eventos o planejamento e o acompanhamento do projeto de desenvolvimento de um software.

Ainda hoje, perguntas presentes há décadas continuam sem resposta ou com respostas insatisfatórias:

- Por que demora tanto tempo para que os softwares sejam concluídos?
- Por que os custos são tão elevados?
- Por que todos os erros não são descobertos e sanados antes da entrega para o usuário?
- Por que existe a dificuldade em medir o progresso do software enquanto este ainda está em desenvolvimento?

Fernandes e Alves (1992), apontam alguns dos fatores presentes na maior parte dos projetos de software que não tiveram pleno êxito:

- Absorção de inovações tecnológicas - introdução de novidades tecnológicas, novos ambientes, novas ferramentas - quase sempre causa impactos não previstos no desenvolvimento do projeto;
- Especificação de projeto incompleta - dificuldade em captar os requisitos dos usuários e transformá-los em uma especificação completa e viva (conhecimento agregado também ao longo do projeto);
- Metodologias inadequadas - não definição de qual metodologia será adotada;
- Subestimativa de riscos - ausência de um processo formal de avaliação e tratamento de riscos;
- Dificuldades de estimar prazos e recursos - método normalmente utilizado é a "achologia" ou "chutologia" uma vez que a cultura de estimativas baseadas em modelos matemáticos e registros históricos é adotada por poucas empresas;
- Dificuldade de aferir progresso - falta de marcos concretos e ponderados ao longo do projeto a fim de verificar o real avanço;
- Fraco acoplamento entre as estratégias empresariais e as de Tecnologia de Informação - priorização dos projetos de Tecnologia de Informação não está normalmente convergente com as diretrizes da empresa, o que se traduz em problemas;



- Cultura das organizações – Nolan (1977) define e caracteriza empresas de acordo com seu estágio na curva de aprendizado; para as empresas que estão no início desta curva (a grande maioria), o desafio do gerenciamento de projetos é extenuante, com freqüente omissão dos usuários em relação ao seu envolvimento requerido pelo projeto - eles pensam que isto é o negócio dos profissionais de Tecnologia de Informação e não deles;
- Ambiente típico de desenvolvimento de projetos - falta de instrumentos, excesso de pressão por parte dos usuários ("prazos para ontem"), falta de motivação para o pessoal envolvido no projeto e outros fatores relacionados.

A presente dissertação verifica o porque desse panorama tão pouco favorável à produção de software e propõe algumas medidas gerenciais, baseadas nos conceitos de gerenciamento de projetos tão bem utilizados em outras áreas da ciência e tecnologia, que poderiam tornar o desenvolvimento de software um processo controlável.

## **1.2 Estrutura da dissertação**

O presente trabalho está dividido em sete capítulos; o presente capítulo, Introdução, visa apresentar os problemas existentes no gerenciamento de projetos de desenvolvimento de software. Apresenta um panorama geral das origens do problema e demonstra o estágio atual nesta questão.

O capítulo 2, Gerenciamento de projetos, apresenta uma visão geral do universo do gerenciamento de projetos, as principais organizações mundiais e nacionais que desenvolvem pesquisas sobre esse tema, direta ou indiretamente. São apresentados os conceitos tradicionais de gerenciamento de projetos e é introduzido o grande questionamento: gerenciar projetos de desenvolvimento de software da forma como são gerenciados os projetos em outras áreas pode trazer benefícios significativos? A presente dissertação pretende contribuir para responder essa pergunta.

Dentre as organizações que tratam o tema gerenciamento de projetos foram selecionadas as seguintes:

- PMI - Project Management Institute - principal organização mundial sobre gerenciamento de projetos;
- SEI - Software Engineering Institute - um dos principais organismos mundiais que trata de forma profissional o tema desenvolvimento de software;
- ISO - International Organization for Standardization - um dos mais tradicionais organismos internacionais existentes, que tem a missão de padronizar os aspectos do desenvolvimento de produtos e garantia de sua qualidade.

Existem outras inúmeras organizações que poderiam ser mencionadas, incluindo as brasileiras Fundação Vanzolini, FGV, diversas universidades, dentre outras, que também têm preocupações constantes com os temas aqui abordados. Seria impossível mencionar todas sem esquecer de alguma. O objetivo maior é conhecer um pouco mais as principais organizações produtoras de material científico/tecnológico que podem auxiliar a melhorar a efetividade dos processos relacionados ao desenvolvimento de software.

O capítulo 3, Desenvolvimento de software, procura definir software enquanto produto e apresentar o histórico da evolução de seu desenvolvimento, desde seu surgimento como produto até os presentes dias, e as perspectivas futuras. Uma visão geral da disciplina engenharia de software também é apresentada, englobando os paradigmas de desenvolvimento de software e algumas abordagens tradicionalmente utilizadas.

Neste capítulo também são discutidos os modelos da ISO e do SEI (este desenvolvido especificamente para software) que visam garantir a qualidade, a padronização e o aperfeiçoamento dos processos produtivos.

O capítulo 4, Padrões de gerenciamento de projetos, apresenta o modelo proposto pela ISO e o modelo de gerenciamento de projetos proposto pelo PMI, suas principais características e preocupações. O modelo do PMI – o PMBoK Guide - é a base para a verificação da hipótese de que a aplicação dos conceitos de gerenciamento de projetos

utilizados com sucesso em empreendimentos de diferentes áreas pode ser um grande fator de evolução para a condução de projetos de desenvolvimento de software.

O Capítulo 5, Análise comparativa, tem por objetivo verificar os ganhos advindos da utilização do PMBoK como modelo de gerenciamento de projetos de desenvolvimento de software, bem como verificar possíveis adaptações em função das características peculiares dos projetos de software (a começar pelo próprio conceito de projeto discutido no capítulo 4). Também é discutida a utilização de documentos padronizados e adaptados, ora propostos pelo PMBoK, ora propostos pelas abordagens de desenvolvimento de software, ora propostos pelos modelos de qualidade, tanto da ISO quanto do SEI. Estes documentos são confrontados com a experiência acumulada do autor na coordenação de diversos projetos de softwares comerciais e tecnológicos, tanto para uso interno de uma organização quanto para implementações em outras organizações, e na área acadêmica, onde leciona disciplinas relacionadas a engenharia de software, gerenciamento de projetos e análise e projeto de sistemas de informação.

O capítulo 6, Estudo de caso, apresenta os resultados da aplicação das recomendações discutidas no Capítulo 5 em projetos reais de um laboratório de desenvolvimento de pesquisas e projetos na área de Redes de Comunicações, com forte ênfase no desenvolvimento de software.

Concluindo a dissertação, o capítulo 7, Conclusões e perspectivas futuras, apresenta uma síntese dos temas estudados e das conclusões oriundas das análises comparativas e do estudo de caso efetuados. Espera-se que estas conclusões possam servir para melhorar os problemas descritos no capítulo introdutório e que sirvam de inspiração para novos trabalhos. Certamente este tema não será esgotado tão cedo e diversos trabalhos poderão ser realizados visando a melhoria contínua dos processos relacionados a software.

## **Capítulo 2**

### **Gerenciamento de Projetos**

#### **2.1 Conceituação**

O PMI (2004) define gerenciamento de projetos como “a aplicação de conhecimentos, habilidades, ferramentas e técnicas no sentido de concluir atividades que atendam ou excedam às necessidades e expectativas dos envolvidos deste projeto”. A fim de atingir esses objetivos, algumas questões devem ser atentamente observadas:

- Escopo - tudo o que deve ser produzido no projeto;
- Prazos - ordem cronológica dos resultados/produtos do projeto;
- Custo - valor para a produção do projeto;
- Qualidade - tanto do processo de condução quanto dos produtos gerados pelo projeto;
- Diferentes necessidades e expectativas dos envolvidos no projeto;
- Separar o que é necessidade (requisito) de expectativa (requisito ainda não identificado).

##### **2.1.1 Conceito de projeto**

As organizações e as pessoas realizam trabalhos que devem ser gerenciados. Estes trabalhos podem ser categorizados em operações e projetos, embora existam diversas características em comum:

- São realizados por pessoas;
- Utilizam recursos;
- Devem ser planejados, executados e controlados.

Enquanto as operações são processos contínuos, os projetos se caracterizam por serem temporários e únicos. O PMI define projeto como: "um empreendimento temporário - com início e término bem definidos - cujo objetivo é criar um produto ou oferecer um serviço único, distinto de todos os produtos ou serviços produzidos anteriormente". (PMI, 2004)

Os projetos podem variar em tamanho e complexidade, podem envolver uma única organização ou diversas organizações (formando consórcios, parcerias ou outro tipo de associação). Como exemplos de projetos tem-se:

- Desenvolvimento de um novo produto ou serviço;
- Implementações de mudanças em uma organização;
- Desenvolvimento de um novo meio de transporte;
- Desenvolvimento de um software;
- Implementação de um novo processo de negócio.

O conceito de temporalidade em projetos é fundamental. Devem ser claramente determinados o início e o término no sentido de serem identificados os objetivos. Isso é vital a fim de que seja possível determinar se um projeto está concluído (atingiu seus objetivos) ou ainda se é impossível de se concluir. Não necessariamente este limite é pequeno - existem projetos com durações de vários anos (por exemplo, a implantação de uma usina nuclear).

Ainda utilizando o exemplo acima, o projeto para a implantação de uma usina nuclear termina assim que a mesma esteja com todas as condições de início de funcionamento. A partir deste ponto o projeto está concluído e entra na fase de operação.

## **2.2 Organizações**

Existem diversas organizações mundiais e nacionais que agregam profissionais preocupados com as questões envolvidas no gerenciamento de projetos e de operações. Podem ser destacadas as seguintes organizações:

### 2.2.1 Project Management Institute (PMI)

Fundado em 1969, o Project Management Institute (PMI)<sup>2</sup> é uma associação internacional, sem fins lucrativos, que tem como principais objetivos:

- Avançar o estado da arte na prática da gestão de projetos;
- Fomentar o profissionalismo na gestão de projetos;
- Promover a aceitação da gestão de projetos como profissão e disciplina.

Esta associação possui atualmente mais de 200.000 membros em 120 países, divididos em 247 capítulos (onde são discutidos temas mais específicos) em 37 países (o Brasil possui 13 capítulos – o Capítulo de São Paulo<sup>3</sup> foi o primeiro a ser criado, em 1998). Destes diversos capítulos são extraídas contribuições que são consolidadas no chamado *PMBok Guide* (Um guia do Conjunto de Conhecimentos em Gerenciamento de Projetos), guia com as melhores práticas nesta disciplina; este guia visa consolidar padrões (fases de um empreendimento e responsabilidades, entre outros), além de propor uma abordagem universal às diversas áreas de conhecimento no gerenciamento de projetos. O tópico 4.2, PMI – PMBoK, apresenta em detalhes este guia, que é um dos alicerces desta dissertação.

Outra preocupação do PMI é com o processo de profissionalização dos profissionais que lidam com o gerenciamento de projetos. Para tanto, desde 1984 são realizados nos diversos países exames de certificação onde são verificados os conhecimentos e a experiência dos profissionais, baseados nas práticas do PMBoK. O primeiro exame realizado no Brasil para a certificação de PMP (*Project Management Professional*) ocorreu no dia 13 de junho de 1998, contando com a participação de 33 pessoas (PLONSKI, 1998). Para se ter uma idéia do crescimento no número de PMP no mundo, em dezembro de 1998 existiam 10.125 PMPs (PLONSKI, 1998); ao final de 2005 existiam 181.281 PMPs, sendo considerada uma das 10 maiores certificações profissionais para os próximos anos<sup>4</sup>. Boa parte das empresas, em especial as de

---

<sup>2</sup> Maiores informações sobre o PMI em: [www.pmi.org](http://www.pmi.org).

<sup>3</sup> Maiores informações sobre o Capítulo São Paulo do PMI em: <http://www.pmis.org.br>.

<sup>4</sup> fonte: [http://www.pmi.org/info/AP\\_News-Certcities.asp](http://www.pmi.org/info/AP_News-Certcities.asp).

médio e grande portes, passaram a exigir de seus profissionais de gerenciamento de projetos esta certificação. Existem diversas publicações que abordam o processo de certificação, dentre os quais Heldman (2005), que apresenta um guia de preparação para o exame PMP.

O PMI também agrega diversos grupos de interesses para várias áreas de negócio. Para este trabalho dois grupos de interesse foram selecionados.

O primeiro é o grupo de interesse em sistemas de informação (PMI-ISSIG)<sup>5</sup>, fundado em 1990 e considerado como um capítulo pelo PMI em 1995, possui aproximadamente 14.000 membros distribuídos ao redor do mundo; tem como missão avançar o estado da arte no que diz respeito a técnicas, métodos, procedimentos e práticas de gerenciamento de projetos para os profissionais envolvidos com sistemas de informação, em especial para as indústrias. As discussões desse grupo abordam questões como:

- Compartilhar e disseminar informações sobre as melhores práticas de gerenciamento de projetos;
- Capacitar continuamente seus membros, através de programas de educação continuada e do compartilhamento e troca de informações;
- Demonstrar e promover os princípios profissionais de gerenciamento de projetos como a forma mais efetiva de planejar e gerenciar projetos de sistemas de informação;
- Compartilhar e disseminar informações com outros gerentes de projetos de sistemas de informação, ou seja, prover acesso a informações qualitativas e quantitativas a respeito dos profissionais de gerenciamento de projetos e das empresas.

O segundo é o grupo de interesse em tecnologia da informação e telecomunicações (PMI-ITTelecom)<sup>6</sup>, possui aproximadamente 3.000 membros distribuídos ao redor do mundo; tem como missão avançar o estado da arte no que diz respeito a técnicas, métodos, procedimentos e práticas de gerenciamento de projetos

---

<sup>5</sup> Maiores informações sobre o PMI-ISSIG em: <<http://www.pmi-issig.org/>>.

<sup>6</sup> Maiores informações sobre o PMI-ITTelecom em: <<http://www.pmi-ittelecom.org/>>.

para os profissionais envolvidos com tecnologia da informação, sendo um elo de ligação dos profissionais, da indústria e da academia.

### **2.2.2 Software Engineering Institute (SEI)**

O Software Engineering Institute (SEI)<sup>7</sup> foi estabelecido em 1984 pelo Congresso Americano como uma fundação federal de pesquisa e desenvolvimento com o propósito de verificar e acompanhar a transição entre as tecnologias de engenharia de software; é patrocinado pelo departamento de defesa americano (DoD) através do Office of the Under Secretary of Defense for Acquisition and Technology (OUSD - A&T). O SEI procura ser parceiro das organizações industriais e agências governamentais nos processos de desenvolvimento, aquisição e suporte aos sistemas de software.

Como um componente integral da Universidade Carnegie Mellon (CMU), o SEI mantém um corpo técnico altamente qualificado e conduz suas atividades de maneira sincronizada com a universidade. Pelo fato de ser patrocinado pelo DoD e também ser integrante da CMU, o SEI procura utilizar a tecnologia de ponta desenvolvida/pesquisada na universidade com os padrões e as exigências de mercado exigidos pela comunidade.

Em 1984, a comunidade de engenheiros de software desejava conseguir uma visão geral do estado da arte das práticas adotadas a fim de obter um conjunto de preceitos comuns a todos. Atividades de desenvolvimento e os resultados obtidos com os produtos eram imprevisíveis e o sucesso ou fracasso era totalmente dependente das pessoas envolvidas com as atividades. Enquanto várias tecnologias prometiam avanços, várias organizações não estavam em condições de adotar novas tecnologias - e ainda que fizessem não poderiam avaliar o seu real efeito.

A estratégia adotada pelo SEI foi desenvolver práticas na disciplina de engenharia de software e preparar indivíduos e organizações para usar e evoluir estas práticas. Como exemplo, tem-se o desenvolvimento do modelo de capacidade e maturidade de

---

<sup>7</sup> Maiores informações sobre o SEI em: <<http://www.sei.cmu.edu>>.



software (CMM/CMMI), discutido em detalhes no item 3.4.6, SEI – CMM e CMMI; o objetivo principal era auxiliar estas organizações a atingir um nível de maturidade suficiente para gerenciar a tecnologia. A estratégia habilitava o desenvolvimento de uma infra-estrutura na qual a tecnologia poderia ser instalada e os meios para estabelecimento de prioridades fossem disponíveis.

Com esta estratégia, o SEI implementou um modelo no qual a tecnologia poderia ser inserida e como a comunidade poderia ser preparada para adaptar esta tecnologia.

Os programas técnicos do SEI estão organizados em duas grandes áreas:

- Práticas de gerenciamento de engenharia de software;
- Práticas em técnicas de engenharia de software.

Estes programas são desenvolvidos visando habilitar a comunidade de software a lidar com o desenvolvimento de sistemas como linhas de produção, baseadas em regras claras, arquiteturas flexíveis de software com interfaces para padrões comerciais seguindo uma metodologia/disciplina.

### **2.2.3 International Organization for Standardization (ISO)**

A International Organization for Standardization (ISO)<sup>8</sup> é uma federação mundial composta por associações de padronização nacionais de 156 países. Tem, entre seus parceiros, a International Telecommunication Union (ITU)<sup>9</sup>, a World Trade Organization (WTO)<sup>10</sup> e o International Engineering Consortium (IEC)<sup>11</sup>. É uma organização não governamental, fundada em 1947; sua missão é promover o desenvolvimento da padronização e atividades correlatas no mundo com vistas a facilitar a troca internacional de bens e serviços e desenvolver, de forma cooperativa, as esferas intelectuais, científicas, tecnológicas e econômicas internacionais. É mantida pelas

---

<sup>8</sup> Maiores informações sobre a ISO em: <<http://www.iso.org>>.

<sup>9</sup> Maiores informações sobre a ITU em: <<http://www.itu.org>>.

<sup>10</sup> Maiores informações sobre a WTO em: <<http://www.wto.org>>.

<sup>11</sup> Maiores informações sobre o IEC em: <<http://www.iec.org>>.

taxas de inscrição de seus associados e pela renda obtida com a venda de cópias dos materiais (artigos, padrões e outros).

A preocupação com padronização mundial começou no campo da eletrotécnica: a IEC foi criada em 1906; trabalhadores pioneiros de outros campos foram congregados pela International Federation of the National Standardizing Associations (ISA), cujas atividades tiveram início em 1926; a ênfase maior das atividades da ISA estavam voltadas para atividades relacionadas a engenharia mecânica.

As atividades da ISA foram suspensas em 1942 em função da Segunda Guerra Mundial. Após um encontro ocorrido em 1946, em Londres, delegados de 25 países decidiram criar uma nova organização internacional cujo objetivo era "facilitar a coordenação mundial e unificação dos padrões industriais"; com isso, foi fundada a ISO, cuja data oficial de fundação é 23 fevereiro de 1947. O primeiro padrão publicado pela ISO foi feito em 1951, sob o título "Standard reference temperature for industrial length measurement".

Os trabalhos da ISO são resultados de acordos internacionais os quais são publicados como padrões internacionais.

O nome ISO não corresponde exatamente à sigla do nome da organização; isto se deve ao fato de ISO não ser um acrônimo e sim uma palavra, derivada do grego "isos" que significa "igual", o qual deu origem ao prefixo "iso-" que aparece em vários termos tais como isonomia (igualdade de leis ou de pessoas antes da lei) e isometria (de igual medida ou dimensão).

De "igual" para "padrão" - esta linha de pensamento levou à adoção do nome "ISO" como sendo o nome da organização mais facilmente reconhecido. Outro fator é que o nome ISO é usado ao redor do mundo para descrever a organização, embora os acrônimos resultados das diversas traduções de "International Organization for Standardization" sejam diversos entre as diversas nações e seus diferentes idiomas (por exemplo, IOS em inglês, OIN em Francês); com a adoção do termo ISO, em qualquer país a terminologia é a mesma.

A razão da existência da ISO tem relação com a existência de padrões não harmônicos para tecnologias similares em diferentes países e regiões, que contribuem para as chamadas "barreiras técnicas para comércio". Empresas de exportação tem

sentido a necessidade de padronização a fim de auxiliá-las na realização de negócios internacionais. As áreas de atuação da ISO são processamento de dados, comunicações, pacotes, distribuição de bens e serviços, produção e utilização de energia, construção naval, atividades bancárias e serviços financeiros, além de ter sempre em mente a questão da evolução das necessidades com o surgimento de novos tipos de negócio.

As razões principais, segundo a ISO, são:

- Progressos mundiais na liberação do comércio - necessidade de identificação e padronização entre as diversas nações;
- Inter-relações entre os diversos setores - todas as indústrias atuais atuam de forma integrada com componentes, produtos, regras de aplicação e outros. Isto faz com que o desenvolvimento de padrões seja realizado em todos os setores envolvidos;
- Sistemas de comunicação mundiais - especialmente impulsionado pela indústria de computadores, que necessitam de uma maior compatibilidade entre os diversos componentes;
- Padrões mundiais para tecnologias emergentes - preocupação em padronizar o uso de novas tecnologias (terminologia, informações relativas e outros), evitando problemas futuros;
- Desenvolvimento dos países - agências de desenvolvimento estão reconhecendo cada vez mais que padrões de infra-estrutura correspondem a uma das condições básicas para o sucesso das políticas econômicas que visam prover o desenvolvimento sustentável destes países em desenvolvimento.

Facilitando os processos de comércio, troca e transferência de tecnologia tem-se como ganhos:

- Aumento na qualidade e confiabilidade dos produtos por um preço adequado;
- Melhoria na saúde, segurança e proteção ambiental, com redução de lixo;
- Grande capacidade e interoperabilidade de produtos e serviços;
- Simplificação dos processos para aumento da usabilidade;

- Redução do número de modelos e conseqüente redução de custos;
- Aumento da eficiência de distribuição e facilidades adicionais de manutenção.

A ISO trabalha mundialmente no desenvolvimento e na padronização de normas nas mais diversas áreas. Toda norma segue um ciclo de vida padrão, partindo de um estágio preliminar (com o registro da necessidade de desenvolver uma determinada norma), passando pelos estágios de proposta, preparação, comitê, investigação, aprovação, publicação, revisão e retirada, quando a norma deixa de ser válida. A tabela 1 - Ciclo de vida das normas ISO – apresenta resumidamente o ciclo de vida das normas. Para este trabalho foram consideradas apenas as normas no estágio de publicação internacionalmente (60.60).

Tabela 1 - Ciclo de vida das normas ISO <sup>12</sup>

Estágio	Sub-Estágio							
	00	20	60	90				
				92	93	98	99	
	Registro	Inícios das ações principais	Conclusão das ações principais	Repetir uma etapa preliminar	Repetir fase atual	Abandonar	Prosseguir	
<b>00</b>	<b>Início</b>	00.00	00.20	00.60			00.98	00.99
<b>10</b>	<b>Proposta</b>	10.00	10.20	10.60	10.92		10.98	10.99
<b>20</b>	<b>Preparação</b>	20.00	20.20	20.60			20.98	20.99
<b>30</b>	<b>Comitê</b>	30.00	30.20	30.60	30.92		30.98	30.99
<b>40</b>	<b>Investigação</b>	40.00	40.20	40.60	40.92	40.93	40.98	40.99
<b>50</b>	<b>Aprovação</b>	50.00	50.20	50.60	50.92		50.98	50.99
<b>60</b>	<b>Publicação</b>	60.00		<b>60.60 Padrão Internacional Publicado</b>				
<b>90</b>	<b>Revisão</b>		90.20	90.60	90.92	90.93		90.99
<b>95</b>	<b>Retirada</b>		95.20	95.60	95.92			95.99

<sup>12</sup> Baseado em ISO (2006).

Dentre as principais contribuições advindas de padrões propostos pela ISO tem-se que milhares de empresas já implementaram (ou estão implementando) a ISO 9000, que provê um modelo para gerenciamento e garantia da qualidade. A série ISO 14000 provê um modelo similar para gerenciamento ambiental. Os padrões relacionados ao desenvolvimento de software e ao gerenciamento deste processo de desenvolvimento são discutidos no Capítulo 3 - Desenvolvimento de software.

#### **2.2.4 Institute of Electrical and Electronics Engineers (IEEE)**

O Institute of Electrical and Electronics Engineers (IEEE)<sup>13</sup>, fundado em 1963 com a junção do American Institute of Electrical Engineers (AIEE), de 1884, e com o Institute of Radio Engineers (IRE), de 1912, tem como principais objetivos:

- Avançar a teoria e a aplicação de tecnologias eletroeletrônicas e ciências correlatas;
- Apoiar e desenvolver novas tecnologias;
- Oferecer a seus membros uma variedade de programas e serviços.

Está organizado em 311 seções locais, em 10 regiões geográficas, com aproximadamente 1570 capítulos envolvendo membros com interesses tecnológicos semelhantes, 39 sociedades e 5 conselhos técnicos, compostos por 10 divisões técnicas, 1430 ramos estudantis em universidades de 80 países. Entre as sociedades está a IEEE Computer Society<sup>14</sup>, que em 2006 completa 60 anos de existência.

#### **2.2.5 Outras organizações**

Além das organizações internacionais apresentadas existem outras organizações que poderiam ser mencionadas, incluindo as brasileiras Fundação Carlos Alberto Vanzolini, Fundação Getúlio Vargas, diversas outras universidades e instituições que

---

<sup>13</sup> Maiores informações sobre o IEEE em: <<http://www.ieee.org>>.

<sup>14</sup> Maiores informações sobre a IEEE Computer Society em: <<http://www.computer.org>>.

tem preocupações constantes com os temas aqui abordados; mencionar todas seria extremamente difícil e fatalmente alguma instituição seria esquecida.

Dentre as contribuições destas instituições podem ser destacados os trabalhos dos capítulos brasileiros do PMI; para mencionar algumas iniciativas, tem-se os trabalhos do capítulo São Paulo de apoio à tradução do PMBoK para a língua portuguesa (desde a versão de 1996) e a criação de uma tradução padronizada para a terminologia de gerenciamento de projetos; tem-se o capítulo Rio de Janeiro<sup>15</sup>, que promove diversos encontros discutindo temas referentes ao gerenciamento de projetos, em especial de projetos que envolvam Tecnologia da Informação.

A Fundação Carlos Alberto Vanzolini<sup>16</sup>, fundada em 31 de março de 1967, é uma entidade sem fins lucrativos, criada, mantida e gerida pelos professores do Departamento de Engenharia de Produção da Escola Politécnica da Universidade de São Paulo para desenvolver atividades de caráter inovador na área de Engenharia de Produção e Administração de Operações, priorizando seus projetos por critérios de relevância econômica e social e pautando sua atuação por critérios de excelência acadêmicos, profissionais e éticos. Tem por objetivo a difusão da Engenharia de Produção e Administração Industrial. São várias as suas contribuições para a área de gerenciamento de projetos, com diversos cursos de capacitação e programas de pós-graduação, além de sua participação como auditora dos processos relativos às normas de qualidade.

A Sociedade de Usuários de Informática e Telecomunicações de São Paulo (SUCESU-SP)<sup>17</sup>, fundada em 1967, é uma sociedade civil representativa de setores das tecnologias da informação. Dentre seus diversos grupos de discussão existe um específico sobre o tema gerenciamento de projetos - (o chamado GU-GP<sup>18</sup>); promove diversos eventos na área de gerenciamento de projetos, como o Project Management Conference on Information Technology, evento este de grande impacto na comunidade

---

<sup>15</sup> Maiores informações sobre o Capítulo Rio de Janeiro do PMI em: <<http://www.pmirio.org.br>>.

<sup>16</sup> Maiores informações sobre a Fundação Carlos Alberto Vanzolini em: <<http://www.vanzolini.org.br>>.

<sup>17</sup> Maiores informações sobre a SUCESU-SP em: <<http://www.sucesusp.org.br>>.

<sup>18</sup> Maiores informações sobre o GU-GP em: <<http://www.sucesusp.org.br/html/grupos/ge.htm>>.

de software. A presente dissertação utiliza vários artigos e referências apresentados durante estes congressos, aos quais o autor esteve presente.

A Fundação Getúlio Vargas (FGV<sup>19</sup>) e a Universidade de São Paulo (USP<sup>20</sup>) - em especial a Faculdade de Economia, Administração e Contabilidade (FEA<sup>21</sup>) - também podem ser destacadas como instituições que tem preocupações com o tema de gerenciamento de projetos e participam das diversas iniciativas mencionadas (capítulo São Paulo do PMI, congressos da SUCESU-SP e grupo de usuários de gerenciamento de projetos, entre outras).

A Associação para Promoção da Excelência do Software Brasileiro (SOFTEX<sup>22</sup>), criada em 1993 pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ<sup>23</sup>), tem por missão “transformar o Brasil em um Centro de Excelência na produção e exportação de Software”; com sede em Campinas, provê diversos serviços dentre os quais capacitação, consultoria e certificação para empresas que desenvolvem software.

---

<sup>19</sup> Maiores informações sobre a FGV em: <<http://www.fgv.br>>.

<sup>20</sup> Maiores informações sobre a USP em: <<http://www.usp.br>>.

<sup>21</sup> Maiores informações sobre a FEA-USP em: <<http://www.fea.usp.br>>.

<sup>22</sup> Maiores informações sobre a SOFTEX em: <<http://www.softex.br>>.

<sup>23</sup> Maiores informações sobre o CNPQ em: <<http://www.cnpq.br>>.

## Capítulo 3

### Desenvolvimento de Software

#### 3.1 Conceituação

Pressman (2002) conceitua software como:

(1) instruções (programas de computador) que quando executadas, produzem a função e o desempenho desejados; (2) estruturas de dados que possibilitam que os programas manipulem adequadamente a informação; e (3) documentos que descrevem a operação e o uso dos programas.

Pode-se notar a preocupação em ampliar o entendimento comum de que software corresponde apenas ao programa - de fato, a complexidade atualmente existente requer tratamentos específicos ao que se chama de configuração de software, englobando, além dos aspectos já citados, o ambiente necessário ao software (bibliotecas dinâmicas, sistema operacional e tipo de equipamento, entre outros).

Diferentemente do hardware, o software é um elemento lógico e não físico, tendo como características diferenciadas:

- É desenvolvido ou projetado por engenharia (e não manufaturado no sentido literal);
- Enquanto o hardware se desgasta, o software se deteriora com o tempo. No caso do hardware, este desgaste pode ser solucionado com a substituição de componentes e, no caso do software, esta substituição (na realidade a liberação de uma nova versão) acarreta no surgimento de novos erros que prejudicam a operação normal;
- A maioria dos softwares é feita sob medida em vez de ser montada a partir de componentes já existentes - este item pretende ser otimizado com o maior uso das tecnologias orientadas a objetos, descritas ainda neste capítulo.



### 3.2 Histórico

O desenvolvimento de software passou por diversas etapas desde sua origem, na década de 50; a percepção de sua importância também se modificou muito ao longo do tempo: para Pressman (2002), enquanto a preocupação com o software era secundária durante as décadas de 50, 60 e 70 (uma vez que a grande meta era desenvolver um hardware que reduzisse o custo de processamento e armazenamento de dados), o que se vê é uma preocupação mais acentuada a partir da década de 80 (muito em função da grande evolução da microeletrônica durante essa década, que resultou em um maior poder de computação a um custo cada vez mais baixo).

A programação de computadores nasceu como uma arte (e secundária); os desenvolvimentos de software eram feitos virtualmente sem administração, até que os prazos comesçassem a se esgotar e os custos a subir abruptamente. As pessoas envolvidas com o software eram oriundas de diversas áreas (cientistas, administradores e outros), não especializados em software, dificultando ainda mais o entendimento dos problemas relacionados à computação.

Os primeiros sistemas eram especificados, desenvolvidos, operados e mantidos pelo mesmo grupo de pessoas, de forma bastante restrita - eram sistemas desenvolvidos com propósitos específicos, apenas para uma empresa; sua característica de processamento era notadamente em lote (orientação *batch*<sup>24</sup>). A documentação do software era, na maior parte das vezes, negligenciada, ora não existindo (e estando apenas "na cabeça" daquele que desenvolveu o software) ora existindo, porém incompleta, ou desatualizada com o passar das manutenções.

Um segundo momento na evolução de software surgiu com o aumento do número de usuários e solicitações por novos desenvolvimentos; neste momento começa a ênfase em sistemas multiusuários, com o surgimento de preocupações com a interface

---

<sup>24</sup> O processamento em lote (orientação *batch*) ainda hoje é utilizado, em especial nos processamentos de grandes volumes de dados (como em sistemas de folha de pagamento e inicialização de máquinas). Nota-se, contudo, que a ênfase de mercado atual não recai sobre este tipo de sistema.

homem-máquina (deve-se notar que até então isto não era necessário uma vez que o usuário era o próprio desenvolvedor). Com o aumento da demanda, várias empresas começaram a criar software para distribuição, visando atingir vários clientes; no início, estas empresas foram representadas pelos próprios fabricantes de hardware (o software ainda era considerado como secundário). Começa então o que vários autores – Maffeo (1992), Pfleeger (2004), Pressman (2002), Rezende (2005) e Sommerville (2003), entre outros - chamam de "crise de software": os profissionais não conseguiam atender toda a demanda por novos sistemas (normalmente chamado de *backlog*) em função de um número crescente de sistemas disponibilizados para os usuários estarem em constante manutenção, em sua maioria por motivos de correção. Isto ocorria por diversas razões:

- Falta de documentação adequada do software produzido;
- Falta de entendimento das necessidades dos usuários;
- Falta de capacitação e métodos aos profissionais envolvidos no processo de produção do software;
- Constantes atrasos na entrega do software, o que poderia defasá-lo antes mesmo do início de sua utilização.

Nota-se que vários destes itens ainda hoje constituem problemas no desenvolvimento de software.

O terceiro momento no desenvolvimento de software tem seu início em meados da década de 70 e é marcado pelo advento da microinformática, com a expansão do número de usuários e com exigências cada vez maiores por sistemas que apresentassem informações instantaneamente e em grandes volumes de dados. O processamento de dados, anteriormente restrito aos computadores, agora também é utilizado em outros equipamentos com microprocessadores: automóveis, microondas, robôs, equipamentos de diagnóstico médico e outros. Tem-se então um problema histórico: o profissionalismo já existente no desenvolvimento da tecnologia de hardware agora se vê a frente de novos desafios com o software para estes novos equipamentos, o que faz estes profissionais buscarem alternativas para esta capacitação.

Várias empresas de software foram criadas a partir do advento do microcomputador (Microsoft, por exemplo); com a expansão do uso deste novo

equipamento, os usuários potenciais passaram rapidamente das centenas ou milhares existentes na segunda geração para centenas de milhares agora na terceira geração, gerando um mercado consumidor cada vez mais cobiçado. Neste momento também inverte-se a curva de valor entre o hardware e o software: o hardware passa a ser considerado um produto primário, o software passa a ter um campo enorme de novas aplicações a serem desenvolvidas; assim é comum que o custo do software para um computador pessoal ultrapasse em várias vezes o custo do hardware deste mesmo equipamento.

A quarta geração de software é marcada pelo surgimento de novas tecnologias para o desenvolvimento do software - em especial a tecnologia de orientação a objetos. Novas aplicações de software levaram à evolução de sistemas especialistas, sistemas baseados em redes neurais e às novas arquiteturas (cliente servidor).

Nos últimos anos um novo desafio se iniciou e continua presente nos dias atuais: a evolução do hardware e do software em função da disseminação da Internet. Vários autores consideram que está surgindo uma quinta era de desenvolvimento de software, onde a distribuição das informações é radicalmente modificada: se existiam sistemas com centenas de milhares de potenciais usuários este número agora pode ultrapassar aos milhões de usuários em diferentes países. O tempo para o desenvolvimento de sistemas para a Internet é cada vez menor em função das exigências de agilidade das empresas que pretendem atuar neste segmento; surgem novas áreas de aplicação do software, dentre as quais:

- *E-commerce* (comércio eletrônico), voltado para empresas de varejo e de grande penetração com o público consumidor em geral;
- *ASP - Application Services Provider*, que são empresas que passam a oferecer serviços de *outsourcing* de aplicações (notadamente de sistemas de gestão empresarial e outros pacotes voltados a empresas de todos os portes); a grande vantagem deste tipo de solução é o rápido tempo de liberação da solução de tecnologia de informação para uso, o que realmente é muito interessante para empresas que estão se instalando em uma nova região ou aquelas que estão tecnologicamente (tanto em termos

de hardware quanto em termos de software) defasadas e não pretendem fazer grandes investimentos (ou seja, imobilizações) neste setor;

- CRM - *Customer Relationship Management*, que são sistemas voltados ao atendimento das necessidades dos clientes em todas as fases deste relacionamento, desde a prospecção de oportunidades, a produção de um produto/encomenda e o pós-venda, integrado aos sistemas de gestão e produção das empresas;
- SCM - *Supply Chain Management*, que são sistemas voltados ao atendimento da cadeia de suprimentos de uma dada organização, envolvendo todos os fornecedores, as cotações e compras e o relacionamento com os sistemas de gestão e produção.

O grande desafio atual é garantir o cumprimento dos prazos, custo e escopo dos softwares necessários neste novo ambiente com a qualidade adequada.

### **3.3 Engenharia de Software**

O desenvolvimento de software é algo recente, comparado a outras atividades: algo em torno de 50 anos. A preocupação com a padronização e qualidade no desenvolvimento de software é ainda mais recente, uma vez que o mesmo era considerado como uma parte secundária do conjunto hardware + software. Ainda hoje, onde notadamente o software é de vital importância, suplantando, em muitos casos, o hardware, processos de padronização encontram dificuldades entre a comunidade de profissionais, refletindo até mesmo no consenso quanto ao conceito de engenharia de software.

Várias definições podem ser obtidas a partir de diversos autores. Para Maffeo (1992) engenharia de software é

a 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 e segundo cronogramas e custos previamente definidos.

Esta definição é muito interessante uma vez que considera o fator gerencial como um dos mais importantes e críticos na construção do software; é justamente o estudo destes fatores (gerenciais) o escopo da presente dissertação.

Para Martin e McCluer (1991), a engenharia de software corresponde "ao estudo dos princípios e sua aplicação no desenvolvimento e manutenção de sistemas de software; corresponde assim, a uma metodologia para desenvolvimento de soluções em software". Neste conceito são apresentadas características voltadas à metodologia e suas ferramentas de apoio como fatores primordiais para a obtenção de um produto de software com qualidade.

Para Rezende (2005) a engenharia de software é a

metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características: a) adequação aos requisitos funcionais do negócio do cliente e seus respectivos procedimentos pertinentes; b) efetivação de padrões de qualidade e produtividade em suas atividades e produtos; c) fundamentação na Tecnologia da Informação disponível, viável e oportuna; d) planejamento e gestão de atividades, recursos, custos e datas.

Nesta definição o autor procura sintetizar as principais características do processo de desenvolvimento de software, englobando também as questões gerenciais.

Para Pressman (2002) a engenharia de software é

o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais. Abrange um conjunto de três elementos fundamentais (métodos, ferramentas e procedimentos), que possibilitam ao gerente o controle do processo de desenvolvimento do software e oferece ao profissional uma base para a construção de software de alta qualidade.

Esta definição é mais abrangente que as demais, detalhando os componentes que possibilitam atingir aos objetivos de um produto de software com qualidade, não deixando de lado as questões gerenciais; na verdade existe a correlação entre os fatores técnicos e os fatores gerenciais e ambos devem andar de forma alinhada a fim de garantir o sucesso do processo de obtenção do produto software. Os elementos mencionados têm as seguintes características:

- Métodos: proporcionam os detalhes de "como fazer" para construir o software, envolvendo um amplo conjunto de tarefas que incluem:

planejamento e estimativa de projeto, análise de requisitos de software e de sistemas, projeto da estrutura de dados, arquitetura de programa e algoritmo de processamento, codificação, testes e manutenção.

- Ferramentas: proporcionam um apoio automatizado ou semi-automatizado aos métodos, correspondendo a instrumentos que possibilitam os detalhes de "como fazer" para construir o software. Existem diversas técnicas para sustentar os métodos, como por exemplo, ferramentas CASE<sup>25</sup>, CAD<sup>26</sup>, Análise estruturada, Orientação a Objetos, Bancos de dados e linguagens de programação, entre outras.
- Procedimentos: constituem o elo de ligação entre os métodos e as ferramentas de engenharia de software, possibilitando o desenvolvimento racional e oportuno de software. Definem a seqüência na qual os métodos serão aplicados, os produtos a serem entregues, controles de qualidade e avaliação.

### **3.3.1 Paradigmas da engenharia de software**

Para Pressman (2002), "a engenharia de software compreende um conjunto de etapas que envolvem métodos, ferramentas e procedimentos". Essas etapas são citadas pela grande maioria dos autores como sendo os paradigmas da engenharia de software. Um paradigma da engenharia de software é escolhido tendo como base a natureza do projeto e da aplicação, os métodos e as ferramentas a serem usados, os controles e os produtos que precisam ser entregues. A seguir são detalhados os principais paradigmas descritos na literatura e utilizados pelo mercado.

---

<sup>25</sup> CASE - *Computer Aided Software Engineering* - tradicionalmente mantida na sua grafia original (em inglês), compreende ferramentas de apoio automatizado para as atividades de engenharia de software.

<sup>26</sup> CAD - *Computer Aided Design* - tradicionalmente mantida na sua grafia original (em inglês), compreende um conjunto de ferramentas de apoio automatizado para as atividades de desenho.

### 3.3.1.1 Modelo clássico

O modelo clássico, também conhecido como cascata ou *waterfall*, foi o primeiro modelo de desenvolvimento de software estudado pela engenharia de software e ainda hoje é um dos mais difundidos e utilizados. Requer uma abordagem sistemática e seqüencial no desenvolvimento do software. A figura 1 - Modelo clássico - ilustra as principais etapas.

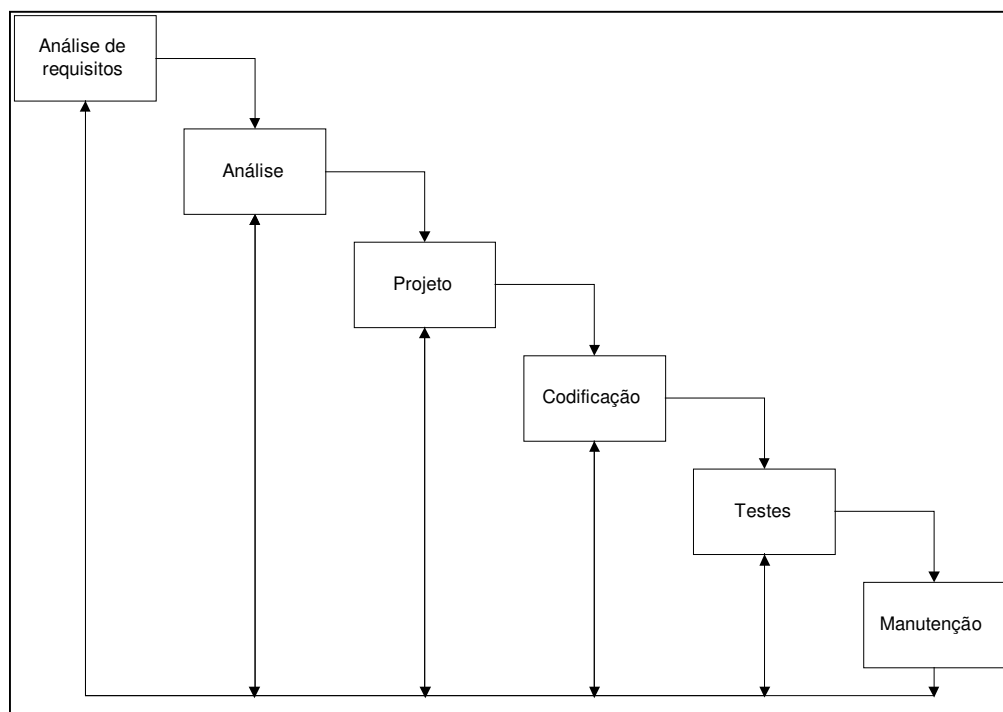


Figura 1 - Modelo clássico<sup>27</sup>

Este modelo funciona de maneira seqüencial, contemplando as seguintes etapas do processo de desenvolvimento de software:

- **Análise de requisitos:** consiste em levantar os requisitos dos usuários a fim de fazer uma primeira verificação de viabilidade; uma vez feita esta verificação, consiste em coletar, classificar e validar os requisitos de

<sup>27</sup> Baseado em Pressman (2002), Pfleeger (2004) e Jacobson et. al. (1998).

software necessários a fim de garantir o atendimento às necessidades dos usuários;

- Análise: consiste em detalhar os requisitos coletados na primeira etapa e garantir o perfeito entendimento das reais necessidades dos usuários. Técnicas como Análise Estruturada (DFD)<sup>28</sup> e Análise Orientada a Objetos (*Object Modeling Technique* (OMT) e *Unified Modeling Language* (UML), entre outras)<sup>29</sup> devem ser utilizadas nesta etapa;
- Projeto: consiste em projetar estruturas (projeto de banco de dados - relacional, orientado a objetos etc.)<sup>30</sup>, interfaces<sup>31</sup> e funções<sup>32</sup>;
- Codificação: atividade que consiste em transformar o projeto anteriormente elaborado em código, preferencialmente através do uso de ferramentas CASE;

---

<sup>28</sup> Para aprofundamento no tema recomenda-se as referências: Gane e Sarson (1984), com a abordagem de DFD - Diagrama de Fluxo de Dados; Martin e McCluer (1991), onde são comentadas as técnicas de análise estruturada e as ferramentas disponíveis no mercado; Yourdon (1990), onde são comentados os princípios e as técnicas de análise de sistemas de grande porte.

<sup>29</sup> Para aprofundamento no tema recomenda-se as referências: Jacobson et. al. (1998), que aborda a engenharia de software orientada a objetos e estudos de *use cases*; Medeiros (2004), Rational Software (1997) e Furlan (1998), onde são apresentados o ambiente da UML e seus diagramas, além de Jacobson (1998), onde é apresentado o Objectory, uma das bases da UML; Rumbaugh et. al. (1994), onde é apresentada a metodologia OMT, dentre outras referências; Yourdon (1999), onde são descritos os conceitos de orientação a objetos e, de uma forma bastante interessante, são apresentados exemplos práticos, correspondentes a estudos de casos completos; Martin e Odell (1995) e Ambler (1999) também abordam as questões de orientação a objetos, comparando-as com as tradicionais abordagens estruturadas.

<sup>30</sup> Para aprofundamento no tema recomenda-se as referências: Heuser (1999), onde são detalhados os princípios norteadores do bom projeto de banco de dados, com ênfase em banco de dados relacional; Korth e Silberschatz (1999), onde são apresentados os conceitos dos bancos de dados relacionais, bancos de dados orientados a objeto, além de aprofundar a discussão da melhor forma de se fazer bons projetos.

<sup>31</sup> Para aprofundamento no tema recomenda-se as referências: Microsoft (1995), onde são detalhados os princípios norteadores do projeto de interfaces em ambiente gráfico; Rocha e Baranauskas (2003), onde são detalhados aspectos técnicos e humanos que devem estar presentes na construção de interfaces, além de apresentar um guia para avaliação de interfaces.

<sup>32</sup> Para aprofundamento no tema recomenda-se as referências: Page Jones (1988), onde são detalhados os princípios norteadores do projeto estruturado de sistemas; Rational Software (1997), onde é apresentado o ambiente da UML e seus diagramas para projeto.



- Testes: etapa que consiste em testes internos (programas) e externos (sistema)<sup>33</sup>. Deve contar com a participação tanto do pessoal técnico quanto dos usuários;
- Manutenção: após a implantação do sistema inicia-se a etapa de manutenção; esta pode ser dividida em três tipos: evolutiva, caracterizada por necessidades adicionais no software originalmente especificado; corretiva, caracterizada pela necessidade de correção de um erro, quer seja de projeto, análise ou codificação; preventiva, caracterizada pela necessidade de adaptação do software a novos ambientes ou configurações.

Este modelo enfrenta uma série de questionamentos quanto a sua efetividade.

Dentre os principais podem ser destacados:

- Os projetos reais raramente seguem o fluxo seqüencial que este modelo propõe, o que causa grande dificuldade e ansiedade em todos os envolvidos;
- Muitas vezes é difícil para o cliente declarar todas as exigências de forma explícita, o que requer muita habilidade dos profissionais;
- O cliente deve ter paciência - o produto somente será disponibilizado no final do ciclo, podendo não contemplar as exigências dos usuários;
- Alto custo de correção de problemas quando identificados tardiamente (nas fases de manutenção, testes...);
- Manutenção: fase na qual o sistema passará a maior parte do tempo, muitas vezes em manutenções corretivas.

---

<sup>33</sup> Para aprofundamento no tema recomenda-se as referências: Pressman (2002) e Gustafson (2003), onde existem capítulos específicos sobre metodologias e procedimentos que auxiliam o teste de software, desde a identificação da abordagem a ser utilizada (testes de caixa branca, de caixa preta) até o que testar; Myers (1979), livro clássico sobre teste de software, apresenta as bases para esta etapa do desenvolvimento de sistemas; Barbosa (2002) apresenta um resumo sobre as estratégias de teste de software, além de uma descrição detalhada de como aplicá-las no caso de teste em banco de dados relacionais.

### 3.3.1.2 Modelo de prototipação

É um processo que capacita o desenvolvedor a criar um modelo do software que será implementado. A seqüência de eventos para o paradigma de prototipação é ilustrada na figura 2 - Modelo de prototipação.

Diferentemente do modelo clássico, a prototipação requer uma participação bastante ativa do usuário, uma vez que sempre dependerá de sua avaliação para avançar para a próxima fase. As etapas normalmente encontradas neste paradigma são:

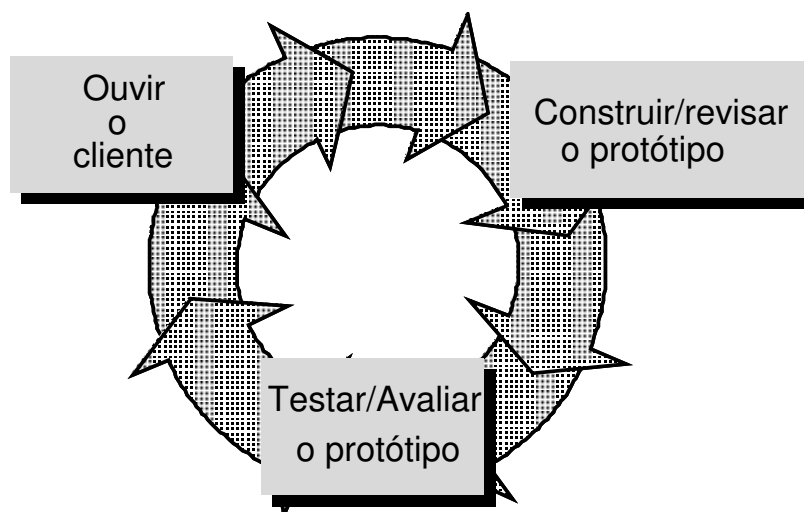


Figura 2 - Modelo de prototipação<sup>34</sup>

- Coleta e refinamento dos requisitos: desenvolvedores e usuários definem os objetivos globais para o software, identificam as exigências conhecidas e esboçam as áreas em que uma definição adicional é obrigatória;
- Projeto rápido: concentra-se na representação daqueles aspectos do software que serão visíveis aos usuários (basicamente entradas de dados, saídas de dados e navegação);

---

<sup>34</sup> Baseado em Pressman (2002).

- Construção do protótipo: construção utilizando ambientes de desenvolvimento rápido. Algumas vezes o protótipo consiste em telas e relatórios, outras em um aplicativo e outras ainda em esboço em papel;
- Avaliação do protótipo pelo cliente: avaliação do protótipo pelo usuário objetivando extrair críticas e sugestões;
- Refinamento do protótipo: etapa onde as críticas e sugestões do usuário são avaliadas e incorporadas ao protótipo; com isso, tanto usuários quanto os desenvolvedores passam a ter uma visão mais clara das reais necessidades. Repete-se então novo ciclo a partir do projeto rápido para a incorporação dessas novas necessidades;
- Engenharia do produto: uma vez definidos todos os produtos a serem gerados e dirimidas todas as dúvidas com o protótipo, o sistema propriamente dito pode ser desenvolvido utilizando conceitos de engenharia de software.

Idealmente o protótipo serve como um mecanismo para identificar os requisitos de software. Se um protótipo for construído, o desenvolvedor tentará usar fragmentos de programas já existentes ou aplicará ferramentas (por exemplo, geradores de relatórios e janelas, ferramentas gráficas) que possibilitem a geração rápida do protótipo. O protótipo não tem compromisso com a qualidade e com a forma de produção no tocante aos métodos; é como um cenário de um filme ou novela, onde a aparência é o aspecto mais importante.

Surge então a grande dificuldade deste método: não raro, ao final do protótipo, o cliente imagina que o produto está "quase" pronto e deseja, a todo custo, uma versão temporária e um produto final em prazos muito rápidos. Os desenvolvedores tem freqüentemente dificuldades em esclarecer que os objetivos do protótipo são relativos à eliminação das dúvidas existentes na coleta dos requisitos e que não é um produto - aliás o desenvolvimento do produto propriamente dito somente neste momento é iniciado, com a visão clara dos resultados a serem atingidos conseguida graças ao protótipo.

A fim de evitar esses problemas a recomendação é de que as regras do jogo sejam negociadas com o usuário antes do início do desenvolvimento do protótipo.

### 3.3.1.3 Modelo incremental

Desenvolvido a partir do modelo clássico, incluindo a filosofia iterativa da prototipagem, conforme ilustra a figura 3 - Modelo incremental. Parte do princípio que um determinado software não precisa ser concluído na sua totalidade a fim de ser entregue para os usuários – procura dividir o processo de desenvolvimento em partes, chamadas incrementos, funcionais, antecipando o contato dos usuários com o produto em elaboração. No primeiro incremento existe o planejamento do número total de incrementos a serem construídos, bem como a especificação das interfaces e interdependências de modo a garantir que um incremento necessite apenas de itens desenvolvidos em incrementos anteriores – e não posteriores - o que poderia causar inconvenientes nessa abordagem.

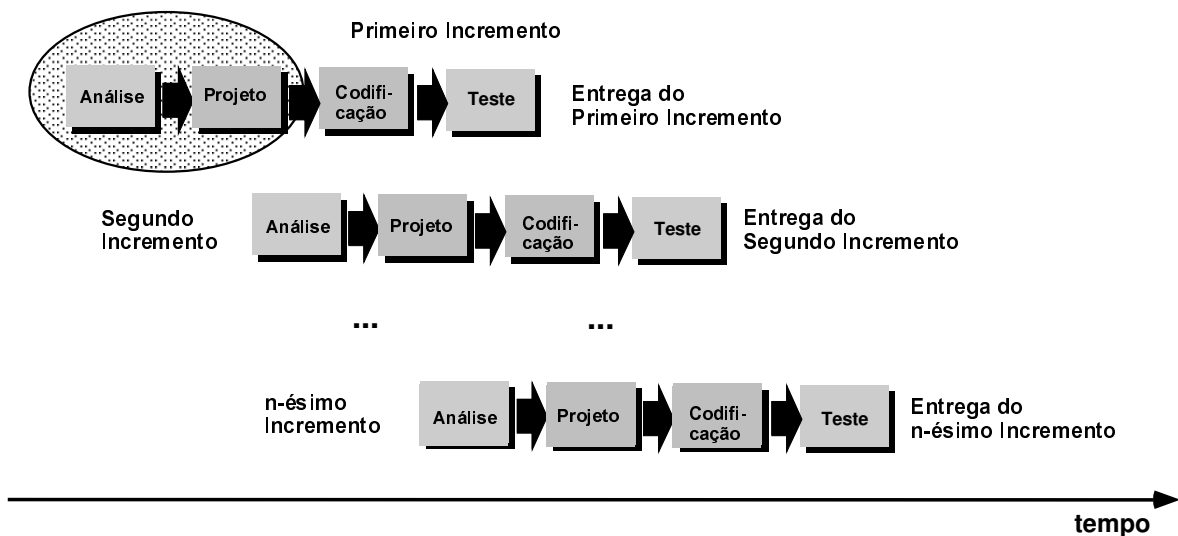


Figura 3 - Modelo incremental<sup>35</sup>

É uma abordagem utilizada por diversas empresas que produzem software, com equipes numerosas de profissionais. É importante garantir que os incrementos, em especial o primeiro, tenham boa aceitação por parte dos usuários a fim de garantir a

<sup>35</sup> Baseado em Pressman (2002).

continuidade dos demais; caso isso não seja possível, problemas de retrabalho podem acarretar em atrasos e custos elevados para a conclusão do software.

### 3.3.1.4 Modelo espiral

Este modelo foi desenvolvido a fim de conter as melhores características dos paradigmas clássico e prototipação, acrescentando um novo elemento - a análise de riscos; é definido em quatro quadrantes conforme ilustra a figura 4 - Modelo espiral.

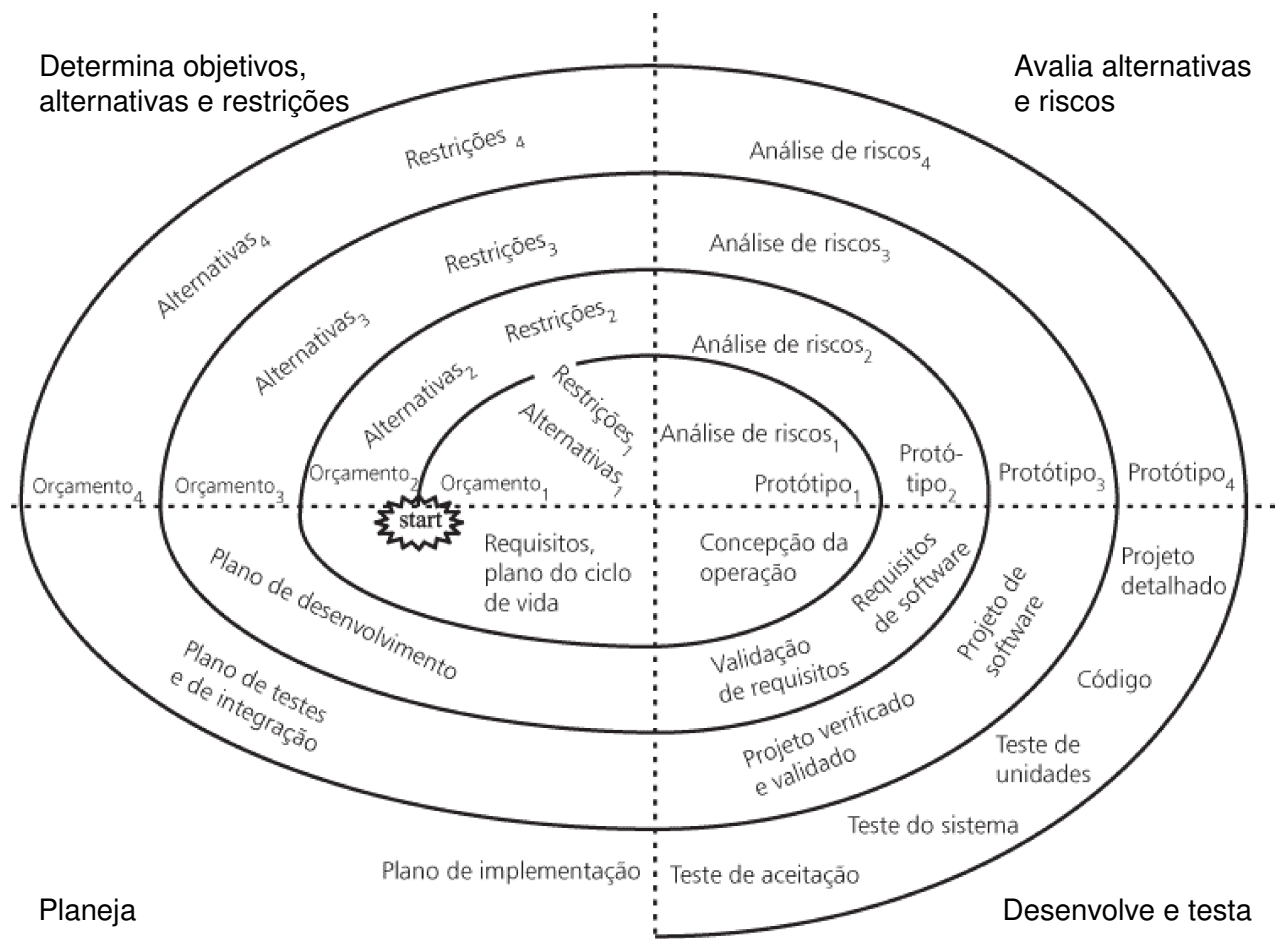


Figura 4 - Modelo espiral<sup>36</sup>

<sup>36</sup> Baseado em Pfleeger (2004).

Ao final de cada iteração ao redor da espiral (iniciando-se do centro e avançando para fora), versões progressivamente mais completas do software são geradas. Os quatro quadrantes representam as seguintes etapas:

- Planejamento: determinação dos objetivos, alternativas e restrições para o desenvolvimento da versão atual do software;
- Análise de riscos: análise das alternativas e identificação, priorização, acompanhamento e resolução dos riscos;
- Engenharia: desenvolvimento do produto, incorporando às versões já existentes as novas funcionalidades definidas nos quadrantes anteriores. Vale notar que a técnica de prototipação poderá ser utilizada caso na etapa de análise de riscos tenha sido identificado um volume de incertezas que justifique a construção de um protótipo; caso contrário, a abordagem clássica poderá ser utilizada para desenvolver as novas funcionalidades. Componentes previamente existentes podem ser utilizados – para alguns autores, como Pressman (2002) e Peters e Pedrycz (2001), essa utilização de componentes gera um novo paradigma – o modelo de componentes.
- Avaliação do cliente: avaliação, por parte do cliente, dos resultados da engenharia.

A incorporação da análise de riscos é a grande novidade e virtude deste modelo. Incorpora a necessidade de uma avaliação criteriosa dos riscos envolvidos com o projeto e a decisão de prosseguir, ou não, com o desenvolvimento do software. Esta decisão, que no modelo clássico muitas vezes não é visível até que seja tarde demais (muitas vezes apenas após a etapa de codificação, quando o sistema começa a ser testado visando a implantação), é tomada baseada nos levantamentos e requisitos da atual fase de desenvolvimento.

Uma preocupação constante é manter o entendimento de todos os envolvidos de que a abordagem deste modelo é evolutiva e requer a participação ativa e constante de todos. Um dos principais benefícios é a entrega de produtos (ou versões do software) intermediárias que já possuem características interessantes aos usuários, já podendo ser utilizada (inclusive é extremamente recomendável que isto ocorra a fim de fornecer subsídios para o planejamento do próximo ciclo de desenvolvimento).

### 3.3.1.5 Técnicas de quarta geração

O paradigma de técnicas de quarta geração da engenharia de software concentra-se na capacidade de se especificar software a uma máquina em um nível que esteja próximo à linguagem natural, ou de se usar uma notação que comunique uma função significativa. Entende-se por estas técnicas um amplo conjunto de ferramentas de software (CASE) que têm uma característica em comum: possibilitar ao desenvolvedor de software especificar alguma etapa do software em um nível elevado e automatizar a geração do código-fonte para a implementação desta característica; as ferramentas disponíveis atualmente para este paradigma incluem, entre outras, as seguintes características:

- Linguagens não procedimentais para consultas e operações em bancos de dados;
- Geradores de código-fonte;
- Geradores de interfaces, contemplando manipulação de dados, interação e definição de telas;
- Capacidade gráfica de alto nível;
- Planilhas eletrônicas.

A figura 5 - Técnicas de quarta geração - ilustra as etapas deste paradigma.

Este paradigma se inicia com a coleta de requisitos dos usuários; Pressman (2002) comenta que idealmente esta atividade, bem como a introdução destes dados em uma ferramenta CASE, deveria ficar a cargo dos usuários. Este tema é bastante polêmico, não havendo consenso entre os autores sobre a responsabilidade da coleta de requisitos e o apoio efetivo das ferramentas CASE disponíveis; de fato o que existe é a necessidade de se efetuar esta atividade da melhor maneira possível, uma vez que esta é parte vital do desenvolvimento do software. Deixar esta atividade apenas nas mãos dos usuários poderia fazer com que o software apenas informatize um processo tal qual ele é realizado hoje - e a atividade de análise consiste, antes de especificar um sistema, verificar o processo atual e melhorá-lo a fim de que o software seja implantado

em um ambiente otimizado. Outro ponto a ser considerado é o fato das ferramentas CASE, por mais avançadas e de alto nível que sejam, exigem preparação (e geralmente técnica) de quem as irá operar, o que poderá causar restrições quando da operação pelo usuário: detalhes vitais podem ser suprimidos assim como outros de importância relativa serem superestimados.

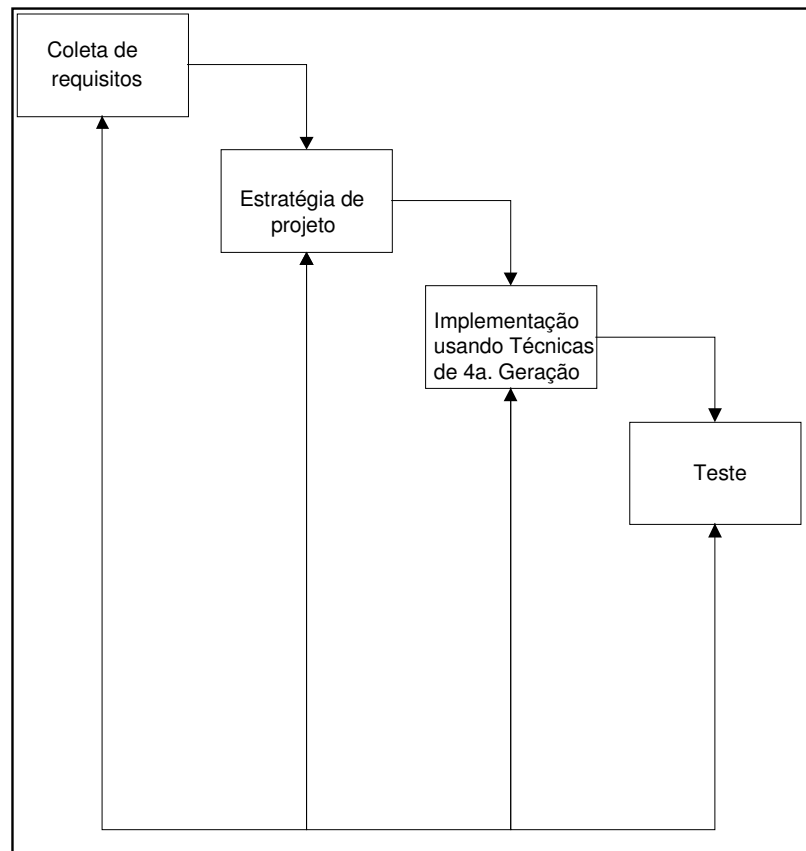


Figura 5 - Técnicas de quarta geração<sup>37</sup>

A etapa de projeto, subsequente à atividade de coleta de requisitos<sup>38</sup>, é obrigatória no desenvolvimento de sistemas de médio e grande porte utilizando este paradigma (alguns autores defendem que para desenvolvimentos de pequeno porte poder-se-ia ir diretamente para a etapa de implementação em uma ferramenta CASE). A tese do uso

---

<sup>37</sup> Baseado em Pressman (2002).

<sup>38</sup> Vale notar que a coleta de requisitos está presente em todos os paradigmas de engenharia de software; sem a identificação clara e precisa das necessidades dos usuários o resultado do produto (software) a ser gerado será sempre questionado: atingiu realmente aos objetivos iniciais?



do projeto é reforçada pelo fato desta atividade requerer e revisar conceitos de planejamento do desenvolvimento do software, sem os quais poder-se-iam surgir problemas de grande dificuldade de solução nas etapas posteriores de teste, implantação e aceitação pelos usuários.

A implementação então é efetivada utilizando ferramentas automatizadas. Um dos principais ganhos previstos é a não necessidade de tradução/codificação das recomendações do projeto em um código-fonte - esta atividade será realizada de forma automática pela ferramenta. Com isso, elimina-se um dos pontos de possíveis introduções de erros no software: a etapa de programação, onde problemas com relação ao entendimento da especificação e do projeto, além de características pessoais dos profissionais que fazem a codificação, introduzem erros – performance e características diferentes das projetadas, entre outros. Vários autores comentam que a atividade de programação deveria ser um processo de transformação de especificação/projeto em um código-fonte, sem a introdução da criatividade do programador (em outras palavras, estes autores defendem o uso de ferramentas automatizadas não só para acelerar o processo mas também para garantir que o código-fonte gerado realmente corresponda à especificação/projeto). Este é um tema extremamente polêmico e não faz parte do escopo do presente trabalho procurar elucidá-lo (até porque seria impraticável).

Após a obtenção do código-fonte são feitos exaustivos testes a fim de garantir a qualidade do produto gerado e o nível de atendimento aos requisitos anteriormente coletados. Também é essencial para a conclusão do desenvolvimento a verificação completa de toda a documentação produzida - em muitos casos gerada em parte pelas próprias ferramentas utilizadas para o desenvolvimento do software.

Algumas críticas têm sido feitas ao uso deste paradigma:

- Perda do conhecimento do processo de desenvolvimento como um todo, em função da automatização de algumas etapas;
- Geração de código não otimizado, o que pode causar problemas posteriores de performance e manutenção;

- Área de atuação das ferramentas de quarta geração ainda é restrita, notadamente circunscrita ao desenvolvimento de software comercial; várias áreas de aplicação de software não possuem ferramentas adequadas.

### **3.3.1.6 Outros modelos de desenvolvimento**

Além dos modelos já apresentados, existem diversos outros para o desenvolvimento de software. Ainda podem ser destacados os seguintes:

- Modelo de métodos formais: parte da especificação do software com a utilização de métodos matemáticos. Tem como principal objetivo a especificação de um software livre de ambigüidades, causa comum de problemas no desenvolvimento de software. É um processo de desenvolvimento mais lento e normalmente utilizado em partes de um determinado software, dificilmente no desenvolvimento de um sistema completo. Moura (2001) apresenta a concepção do modelo de métodos formais, com enfoque na fase de especificação, utilizando a notação Z. Pressman (2002) e Sommerville (2003) dedicam alguns capítulos aprofundando a utilização de métodos formais no desenvolvimento de software;
- Programação Extrema (XP) e Modelagem Ágil: Ambler (2004) descreve os princípios da Modelagem Ágil e sua relação com XP. Os principais objetivos são acelerar o processo de desenvolvimento de aplicações através da utilização de processos incrementais e iterativos; para tanto, procura dividir o produto a ser entregue em pequenas partes, de forma a serem especificadas, desenvolvidas, testadas e entregues muito rapidamente. A figura 6 - Programação Extrema - ilustra as etapas. Pelo fato de ser uma metodologia muito recente, os resultados práticos de sua adoção ainda são questionáveis, em especial por parte das empresas que produzem software – ainda é necessário desenvolvimento e acompanhamento para verificar a viabilidade dessa abordagem.

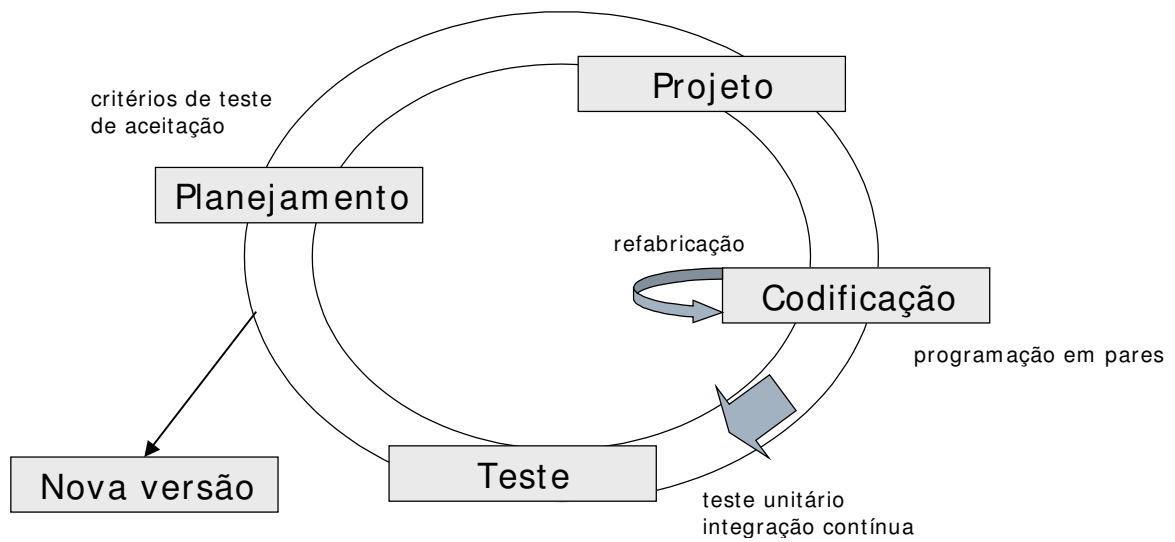


Figura 6 - Programação Extrema<sup>39</sup>

### 3.4 Padrões de qualidade

#### 3.4.1 ISO 9000

##### 3.4.1.1 Histórico

A série de padrões conhecida como ISO 9000 contempla uma coleção de padrões internacionais de qualidade, desenvolvidos pela ISO (descrita em detalhes em 2.2.3, International Organization for Standardization), aplicáveis aos sistemas de gerenciamento da qualidade e aos processos utilizados para a confecção de produtos. Esta série estabelece um conjunto básico de requisitos necessários ao sistema de qualidade a fim de garantir que o processo é capaz de produzir produtos de forma consistente, que atinjam as expectativas dos consumidores/usuários; em outras palavras, ela provê uma excelente base sobre a qual o processo produtivo das empresas pode ser aperfeiçoado, garantindo, como resultado final, a melhoria da qualidade dos produtos e serviços realizados. Foi desenvolvida inicialmente para

---

<sup>39</sup> Baseado em Pressman (2006).

contemplar relações contratuais entre empresas, notadamente para aquelas envolvidas com processos de manufatura; porém o mercado a vem adotando como padrão para diversas outras áreas, tais como indústria eletrônica, química, automotiva, de transportes, de saúde e bancos, entre outras.

A origem da série ISO 9000 data do ano de 1979, a partir da organização de um comitê técnico (*technical committee* 176 - TC176) que ficou encarregado de elaborar um padrão internacional para sistemas de qualidade (Schmauch, 1994). Em 1987 a ISO 9000 foi publicada e começou a ser adotada rapidamente por diversas nações como seu padrão de qualidade; vale notar que várias nações procuraram tomar por base a série ISO 9000 para desenvolverem seus padrões com alguma especificidade (por exemplo, a ANSI/ASQC Q91 corresponde a ISO 9001 (ISO, 1994) para os Estados Unidos). Uma das principais razões para o sucesso da ISO 9000 foi a globalização do mercado, onde a ISO passou a ser uma referência internacional com a qual os clientes poderiam estar assegurados quanto ao processo produtivo das empresas e, por conseguinte, com a qualidade dos produtos produzidos pelas diversas empresas ao redor do mundo. A última revisão nas normas da série ISO 9000 ocorreu em 2005 (ISO, 2005a).

#### **3.4.1.2 Características**

Segundo Schmauch (1994), a premissa norteadora da ISO 9000 é "se o processo produtivo e o sistema de gerenciamento estão funcionando adequadamente, certamente o produto a ser produzido também será adequado". Foi escrita de forma genérica e não para um caso específico, gerando assim como resultado um modelo e não uma especificação; descreve o que, no mínimo, deve ser acompanhado, não especificando como os processos devem ser feitos, deixando ao administrador/desenvolvedor a decisão de como se fazer os processos. A idéia da ISO 9000 não é a de que todos os sistemas de qualidade sejam iguais e sim de que todos contenham as mesmas características quanto aos requisitos; requerem que

as empresas saibam o que fazem, façam o que dizem e demonstrem que elas têm realmente feito isto - de maneira formal, através de documentos. Devem ser

escritos procedimentos que definem como cada atividade relevante no processo produtivo deve ser realizada, devendo ser suficientemente detalhados para garantir a continuidade do processo produtivo mesmo que venham a ocorrer mudança nos recursos para a confecção destes processos (SCHMAUCH, 1994).

A figura 7 - Elementos da norma ISO 9000 - ilustra as relações entre os elementos que constituem a norma.

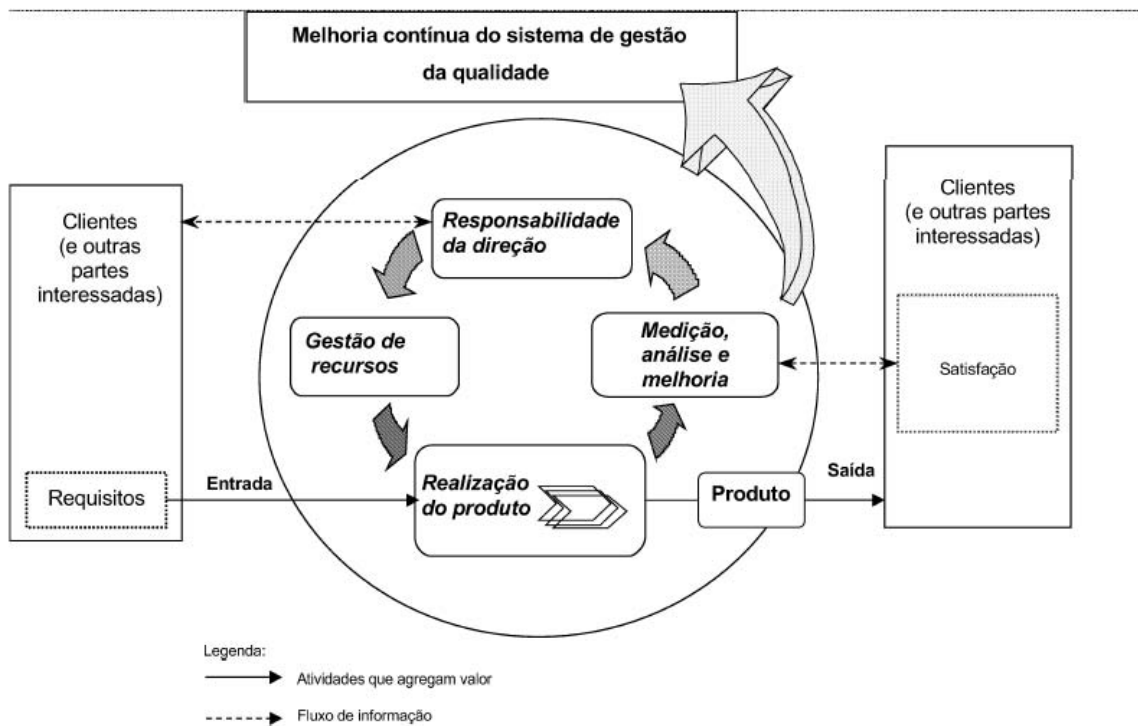


Figura 7 - Elementos da norma ISO 9000 <sup>40</sup>

A ISO 9000 enfatiza algumas áreas:

- Controle: garantir que a empresa está apta, em todas as etapas do processo produtivo, a identificar cada item, quem é responsável por eles, sua situação/progresso e onde está. Em outras palavras, "estar controlado ou sob controle significa que não se está fora de controle";
- Auditoria: a empresa deve, sempre que requisitada, poder exibir evidências de como está o processo produtivo;

<sup>40</sup> Baseado em ISO (2005a).

- Verificação/validação: a empresa deve possuir processos de verificação detalhados a fim de garantir que os produtos são produzidos de acordo com a especificação;
- Melhoria de processo: requisito principal da ISO 9000, que sempre aumenta as exigências das empresas no sentido de melhorar ainda mais o processo produtivo.

As empresas podem solicitar a auditoria dos seus processos a entidades devidamente registradas pela ISO a fim de garantirem sua aderência aos padrões. A empresa passará por auditorias e, caso aprovada, obterá o certificado de que está condizente com os padrões da ISO (a seguir serão detalhadas as diversas subdivisões da ISO); vale notar que este certificado deve ser freqüentemente renovado, com novas auditorias<sup>41</sup>.

A ISO 9000:1994 era composta por cinco partes:

- ISO 9000:1994 para sistema de qualidade - padrão para gerenciamento e garantia da qualidade: este padrão provê um guia para a seleção de quais dos demais padrões da série ISO 9000 é aplicável ao sistema de qualidade da empresa em questão. A ISO 9000-3:1994 - guia para a aplicação da ISO 9001:1994 para o desenvolvimento, fornecimento e manutenção de software - é uma adaptação que provê um guia para a aplicação de padrões aos processos de desenvolvimento de software (será detalhado ainda neste capítulo);
- ISO 9001:1994 para sistema de qualidade - modelo para garantia da qualidade no desenho/desenvolvimento, produção, instalação e fornecimento: este padrão provê um guia para a aplicação da ISO 9000 para as atividades envolvidas com desenho/desenvolvimento de produtos, produção, instalação e fornecimento;
- ISO 9002:1994 para sistema de qualidade - modelo para garantia da qualidade na produção e instalação: este padrão provê um guia para a

---

<sup>41</sup> Diversos autores trabalham a questão das auditorias da ISO, entre os quais Gil (1999).

aplicação da ISO 9000 para as atividades envolvidas com a produção e instalação de produtos.

- ISO 9003:1994 para sistema de qualidade - modelo para garantia da qualidade na inspeção final e teste: este padrão provê um guia para a aplicação da ISO 9000 para as atividades envolvidas com a inspeção final e os testes de produtos.
- ISO 9004:1994 para sistema de qualidade - guia - elementos de sistemas para gerenciamento e garantia da qualidade: este padrão provê um guia para a interpretação das normas ISO 9001:1994, ISO 9002:1994 e ISO 9003:1994.

A ISO 9000:1994 compreendia vinte elementos padrões, sendo que são todos obrigatórios para a ISO 9001:1994, dezoito são obrigatórios para a ISO 9002:1994 e doze para a ISO 9003:1994. A tabela 2 - Níveis ISO 9000:1994 x elementos padrões - ilustra a relação entre estes elementos e as partes constituintes da ISO 9000:1994. Com isso pode-se perceber que o nível de exigência da ISO 9001:1994 é o mais elevado - abrange todas as etapas do processo produtivo - enquanto que a ISO 9003:1994 é a que possui o menor número de elementos - abrange uma parte bem definida do processo produtivo.

A ISO 9000:2000<sup>42</sup>, que substituiu a ISO 9000:1994, efetuou uma série de modificações nas normas anteriores, a saber:

- Modificações consideráveis na norma ISO 9001:1994, gerando a ISO 9001:2000; está dividida em quatro requisitos e 32 seções;
- Elimina as normas ISO 9002:1994 e ISO 9003:1994;
- Atualiza a ISO 9004:1994, gerando a ISO 9004:2000.

A tabela 3 - Principais requisitos e seções da ISO 9001:2000 - ilustra a nova estrutura da norma, cujo foco do sistema de gestão de qualidade da ISO 9000 passou da garantia da qualidade para a satisfação do cliente, atendendo não somente aos requisitos explícitos, mas também aos requisitos implícitos.

---

<sup>42</sup> Vale ressaltar que em 2005 foi revisada a normal ISO 9000:2000, que passou a se chamar ISO 9000:2005; essa norma trabalha com os fundamentos e o vocabulário do sistema de gestão da qualidade.

Tabela 2 - Níveis ISO 9000:1994 x elementos padrões<sup>43</sup>

Elemento padrão ISO 9000:1994	ISO 9001	ISO 9002	ISO 9003
4.1. Responsabilidade gerencial	X	X	X
4.2. Sistema de qualidade	X	X	X
4.3. Revisão contratual	X	X	
4.4. Controle de desenvolvimento	X		
4.5. Controle de documentação	X	X	X
4.6. Compras	X	X	
4.7. Compras - produtos fornecidos	X	X	
4.8. Identificação e rastreabilidade de produtos	X	X	X
4.9. Controle de processo	X	X	
4.10. Inspeção e testes	X	X	X
4.11. Inspeção, medição e equipamentos de teste	X	X	X
4.12. Verificação de inspeção e teste	X	X	X
4.13. Controle de não conformidade	X	X	X
4.14. Ações corretivas	X	X	
4.15. Manipulação, armazenamento, embalagem e entrega	X	X	X
4.16. Registros de qualidade	X	X	X
4.17. Auditoria interna de qualidade	X	X	
4.18. Treinamento	X	X	X
4.19. Serviços	X		
4.20. Técnicas estatísticas	X	X	X

A ISO continuamente publica relatórios e pesquisas a respeito da evolução do número de empresas certificadas, bem como do grau de motivação para a certificação destas empresas. ISO (2003b) apresenta um sumário muito interessante relativo às principais motivações para as empresas adotarem a ISO 9001:2000 em 15 países – infelizmente o Brasil não foi incluído na lista.

A tabela 4 - Empresas certificadas pela norma ISO 9001:2000 – apresenta a evolução da certificação no período de 2000 a 2004. Em relação às empresas no Brasil o número de empresas certificadas passou de 182 no ano de 2001 para 6.120 no final de 2004; mundialmente existiam 44.388 empresas certificadas em 2001, passando para 670.399 no final de 2004, em 154 países. A China é o país com o maior número de empresas certificadas, com 132.926 até o ano de 2004.

<sup>43</sup> Baseado em Schmauch (1994), ISO (1994) e Rocha; Maldonado e Weber (2001).



Tabela 3 - Principais requisitos e seções da ISO 9001:2000<sup>44</sup>

<b>Requisitos ISO 9001:2000</b>	<b>Seções ISO 9001:2000</b>
4. Sistema de gestão da qualidade	4.1 Requisitos gerais 4.2 Requisitos de documentação
5. Responsabilidade da administração	5.1 Comprometimento da direção 5.2 Foco no cliente 5.3 Política de qualidade 5.4 Planejamento 5.5 Responsabilidade, autoridade e comunicação 5.6 Análise crítica pela administração
6. Gestão de recursos	6.1 Provisão de recursos 6.2 Recursos humanos 6.3 Infra-estrutura 6.4 Ambiente de trabalho
7. Realização do produto	7.1 Planejamento da realização do produto 7.2 Processos relacionados ao cliente 7.3 Projeto e desenvolvimento 7.4 Aquisição 7.5 Produção e fornecimento de serviço 7.6 Controle de dispositivos de medição e monitoramento
8. Medição, análise e melhoria	8.1 Planejamento 8.2 Medição e monitoramento 8.3 Controle de não-conformidade 8.4 Análise de dados 8.5 Melhoria

Tabela 4 - Empresas certificadas pela norma ISO 9001:2000<sup>45</sup>

<b>Resultados – ISO 9001:2000</b>	<b>2001</b>	<b>2002</b>	<b>2003</b>	<b>2004</b>
Empresas certificadas no Brasil	182	1.582	4.012	6.120
Empresas certificadas no Mundo	44.388	167.210	497.919	670.399
Nº. de países com empresas certificadas	98	134	149	154

<sup>44</sup> Baseado em Rocha, Maldonado e Weber (2001) e ISO (2000).

<sup>45</sup> Baseado em ISO (2005c).

### 3.4.1.3 ISO 90003:2004

Teve origem com a norma ISO 9000-3:1997 (ISO, 1997b), onde são apresentados os roteiros para aplicar a ISO 9001:1994 especificamente na área de desenvolvimento, fornecimento e manutenção de software. Todas as orientações giram em torno de uma "situação contratual", onde uma outra empresa contrata a empresa em questão para desenvolver um produto de software. A tabela 5 - Processos e atividades da ISO 9000-3:1997 - apresenta um resumo das principais características desta norma.

Tabela 5 - Processos e atividades da ISO 9000-3:1997<sup>46</sup>

<b>Processos</b>	<b>Atividades</b>
Estrutura do Sistema de Qualidade	Responsabilidade do fornecedor; Responsabilidade do comprador; Análise crítica conjunta
Atividades do Ciclo de Vida	Análise crítica do contrato Especificação dos requisitos do comprador Planejamento do desenvolvimento Projeto e implementação Testes e validação Aceitação Cópia, entrega e instalação Manutenção
Atividades de Apoio	Gerenciamento de configuração Controle de documentos Registros da qualidade Medição Regras, convenções Aquisição Produto de software incluído Treinamento

Em 2004 a norma foi atualizada para ISO 90003:2004 (ISO, 2004e), agora para a aplicação da norma a ISO 9001:2000 especificamente na área de desenvolvimento, fornecimento e manutenção de software.

---

<sup>46</sup> Baseado em Barreto Junior (1997).

### **3.4.2 ISO 9126**

Quando se pensa em qualidade de um "produto físico" é fácil imaginar padrões de comparação, provavelmente ligado às dimensões do produto ou a alguma outra característica física. No caso de software, como é possível definir exatamente o que é a qualidade? A norma ISO/IEC 9126:1991 (ISO, 1993) representa a proposta da ISO para padronização mundial dos aspectos relativos à qualidade de produtos de software.

Esta norma lista o conjunto de características que devem ser verificadas em um software para que ele seja considerado um "software de qualidade". São seis grandes grupos de características divididos em algumas subcaracterísticas. A tabela 6 - Elementos da ISO 9126 - contempla um resumo destas características.

Esta norma define minuciosamente o que se pretende avaliar em cada característica e subcaracterística. Em 2001 a norma foi revisada e subdividida em quatro outras normas, a saber:

- A ISO/IEC 9126-1:2001 (ISO, 2001a), que apresenta o modelo de qualidade para o produto software;
- A ISO/IEC 9126-2:2001 (ISO, 2001b), que apresenta as métricas externas para a avaliação do modelo de qualidade para o produto software;
- A ISO/IEC 9126-3:2001 (ISO, 2001c), que apresenta as métricas internas para a avaliação do modelo de qualidade para o produto software;
- A ISO/IEC 9126-4:2001 (ISO, 2001d), que descreve como aplicar as métricas ao modelo de qualidade para o produto software.

### **3.4.3 ISO 12207**

A ISO 12207:1995 (ISO, 1995b) formaliza a arquitetura do ciclo de vida do software, detalhando os diversos processos envolvidos nestes ciclos de vida. Estes processos estão divididos em três classes:

Tabela 6 - Elementos da ISO 9126<sup>47</sup>

<b>Característica</b>	<b>Subcaracterística</b>	<b>Pergunta chave para a subcaracterística</b>
<b>Funcionalidade</b> (satisfaz as necessidades?)	Adequação Acurácia Interoperabilidade Conformidade Segurança de acesso	Propõe-se a fazer o que é apropriado? Faz o que foi proposto de forma correta? Interage com os sistemas especificados? Está de acordo com as normas e leis? Evita acesso não autorizado aos dados?
<b>Confiabilidade</b> (é imune a falhas?)	Maturidade Tolerância a falhas Recuperabilidade	Com que frequência apresenta falhas? Ocorrendo falhas, como ele reage? É capaz de recuperar dados em caso de falha?
<b>Usabilidade</b> (é fácil de usar?)	Inteligibilidade Apreensibilidade Operacionalidade	É fácil entender o conceito e a aplicação? É fácil aprender a usar? É fácil de operar e controlar?
<b>Eficiência</b> (é rápido/"enxuto"?)	Tempo Recursos	Qual é o tempo de resposta e a velocidade de execução? Quanto recurso usa? Durante quanto tempo?
<b>Manutenibilidade</b> (é fácil de modificar?)	Analisabilidade Modificabilidade Estabilidade Testabilidade	É fácil de encontrar uma falha quando ocorre? É fácil modificar e adaptar? Há grande risco quando se faz alterações? É fácil testar quando se faz alterações?
<b>Portabilidade</b> (é fácil de usar em outro ambiente?)	Adaptabilidade Capacidade para ser instalado Conformidade Capacidade para substituir	É fácil adaptar a outros ambientes? É fácil instalar em outros ambientes? Está de acordo com padrões de portabilidade? É fácil usar para substituir outro?

- **Processos Fundamentais** - definem o início e a execução do desenvolvimento, operação ou manutenção do software durante o seu ciclo de vida. São eles: aquisição, fornecimento, desenvolvimento, operação e manutenção;
- **Processos Organizacionais** - definem como implementar uma estrutura constituída de processos de ciclo de vida e de pessoal, de forma a melhorar continuamente a estrutura e os processos. São eles: gerência, infraestrutura, melhoria e treinamento;

---

<sup>47</sup> Baseado em Barreto Junior (1997).

- Processos de Apoio - definem os processos que auxiliam aos demais processos. São eles: documentação, gerência de configuração, garantia da qualidade, verificação, validação, revisão conjunta, auditoria e resolução de problemas.

A norma detalha cada um dos processos descritos; ela define ainda como eles podem ser usados de diferentes maneiras por diferentes organizações (ou parte destas), representando diversos pontos de vista para esta utilização. Cada uma destas visões representa a forma como uma organização emprega estes processos, agrupando-os de acordo com suas necessidades e objetivos.

As visões têm o objetivo de organizar melhor a estrutura de uma empresa, para definir suas gerências e as atividades alocadas às suas equipes. Existem cinco visões diferentes: contrato, gerenciamento, operação, engenharia e apoio. A figura 8 - Processos e visões da ISO 12207:1995 - ilustra como essas visões se relacionam aos processos.

A ISO/IEC 12207:1995 é a primeira norma internacional que descreve em detalhes os processos, atividades e tarefas que envolvem o fornecimento, desenvolvimento, operação e manutenção de produtos de software. A principal finalidade desta norma é servir de referência para os demais padrões que venham a surgir. Lançada em agosto de 1995, ela é citada em quase todos os trabalhos relacionados à Engenharia de Software desde então.

À norma ISO 12207:1995 foram acrescentadas algumas normas:

- Duas normas complementares: a ISO/IEC 12207:1995/Amd 1:2002 e a ISO/IEC 12207:1995/Amd 2:2004. Esses complementos procuraram tornar a norma mais precisa no que diz respeito à terminologia, além de incluir novos requisitos aos processos;
- A ISO/IEC TR 15271:1998 (ISO, 1998a), que apresenta um guia para a aplicação da ISO 12207:1995.

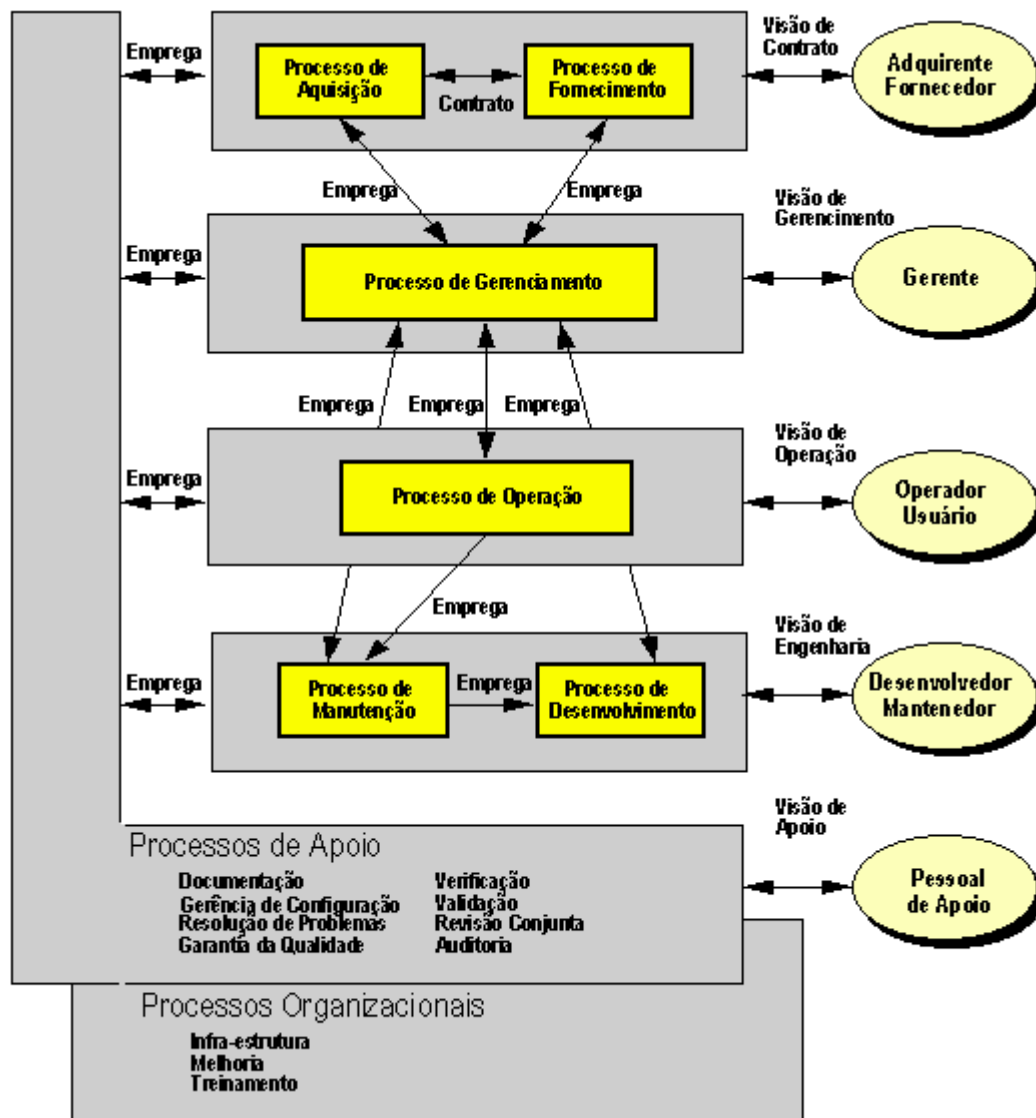


Figura 8 - Processos e visões da ISO 12207:1995<sup>48</sup>

### 3.4.4 ISO 15504 - SPICE

A norma ISO 15504, mais conhecida como SPICE - *Software Process Improvement and Capability dEtermination*, apresenta um padrão para a avaliação do processo de software, visando determinar a capacitação de uma organização. A norma visa ainda orientar a organização para uma melhoria contínua do processo. Ela cobre

<sup>48</sup> Baseado em Barreto Junior (1997).

todos os aspectos da qualidade do processo de software e foi elaborada em um esforço conjunto de cinco centros técnicos espalhados pelo mundo (EUA, Canadá/América Latina, Europa, Pacífico Norte e Pacífico Sul), contando também com a participação de uma comissão formada por profissionais vinculados à ABNT<sup>49</sup>.

O SPICE inclui um modelo de referência, que serve de base para o processo de avaliação. Este modelo é um conjunto padronizado de processos fundamentais, que orientam e garantem a boa engenharia de software; é dividido em cinco grandes categorias de processo:

- **CUS - Cliente-Fornecedor:** processos que impactam diretamente os produtos e serviços de software na relação entre o fornecedor e o cliente:
  - **CUS.1.** Adquirir software;
  - **CUS.2.** Gerenciar necessidades do cliente;
  - **CUS.3.** Fornecer software;
  - **CUS.4.** Operar software;
  - **CUS.5.** Prover serviço ao cliente.
- **ENG - Engenharia:** processos que especificam, implementam ou mantêm um sistema ou produto de software e sua documentação:
  - **ENG.1.** Desenvolver requisitos e o projeto do sistema;
  - **ENG.2.** Desenvolver requisitos de software;
  - **ENG.3.** Desenvolver o projeto do software;
  - **ENG.4.** Implementar o projeto do software;
  - **ENG.5.** Integrar e testar o software;
  - **ENG.6.** Integrar e testar o sistema;
  - **ENG.7.** Manter o sistema e o software.
- **SUP - Suporte:** processos que podem ser empregados por qualquer um dos outros processos:
  - **SUP.1.** Desenvolver a documentação;
  - **SUP.2.** Desempenhar a gerência de configuração;
  - **SUP.3.** Executar a garantia da qualidade;

---

<sup>49</sup> Maiores informações sobre a ABNT em: <<http://www.abnt.org.br>>.

- **SUP.4.** Executar a verificação dos produtos de trabalho;
- **SUP.5.** Executar a validação dos produtos de trabalho;
- **SUP.6.** Executar revisões conjuntas;
- **SUP.7.** Executar auditorias;
- **SUP.8.** Executar resolução de problemas.
- **MAN - Gerência:** processos que contém práticas de natureza genérica que podem ser usadas por quem gerencia projetos ou processos dentro de um ciclo de vida de software:
  - **MAN.1.** Gerenciar o projeto;
  - **MAN.2.** Gerenciar a qualidade;
  - **MAN.3.** Gerenciar riscos;
  - **MAN.4.** Gerenciar subcontratados.
- **ORG - Organização:** processos que estabelecem os objetivos de negócios da organização:
  - **ORG.1.** Construir o negócio;
  - **ORG.2.** Definir o processo;
  - **ORG.3.** Melhorar o processo;
  - **ORG.4.** Prover recursos de treinamento;
  - **ORG.5.** Prover infra-estrutura organizacional.

A norma define detalhes de cada um dos processos mencionados. Para cada um deles existe uma definição detalhada, uma lista dos resultados esperados da sua implementação bem sucedida e uma descrição detalhada de cada uma das práticas básicas. Vale ressaltar que esta norma também foi gerada a partir da ISO 12207 descrita anteriormente.

O SPICE, entretanto, não se limita a listar categorias e processos. Seu principal objetivo, na realidade, é avaliar a capacitação da organização em cada processo e permitir a sua melhoria. O modelo de referência do SPICE inclui seis níveis de capacitação:

- **Nível 0 - Incompleto:** tem como características falhas gerais ao se realizar o objetivo do processo. Não existem produtos de trabalho nem saídas do processo facilmente identificáveis;



- **Nível 1 - Realizado:** o objetivo do processo em geral é atingido, embora não necessariamente de forma planejada e controlada. Há um consenso na organização de que as ações devem ser realizadas e quando são necessárias. Existem produtos de trabalho para o processo e eles são utilizados para atestar o atendimento dos objetivos;
- **Nível 2 - Gerenciado:** o processo produz os produtos de trabalho com qualidade aceitável e dentro do prazo. Isto é feito de forma planejada e controlada. Os produtos de trabalho estão de acordo com padrões e requisitos;
- **Nível 3 - Estabelecido:** o processo é realizado e gerenciado usando um processo definido, baseado em princípios de engenharia de software. As pessoas que implementam o processo usam processos aprovados, que são versões adaptadas do processo padrão documentado;
- **Nível 4 - Predizível:** o processo é realizado de forma consistente, dentro dos limites de controle, para atingir os objetivos. Medidas da realização do processo são coletadas e analisadas. Isto leva a um entendimento quantitativo da capacitação do processo, produzindo a habilidade de prever a realização;
- **Nível 5 - Otimizado:** a realização do processo é otimizada para atender às necessidades atuais e futuras do negócio. O processo atinge seus objetivos de negócio e consegue ser repetido. São estabelecidos objetivos quantitativos de eficácia e eficiência para o processo, alinhados aos objetivos da organização. A monitoração consistente do processo segundo estes objetivos é conseguida obtendo *feedback*<sup>50</sup> quantitativo e o melhoramento é conseguido pela análise dos resultados. A otimização do processo envolve o uso piloto de idéias e tecnologias inovadoras, além da mudança de processos ineficientes para atingir os objetivos definidos.

---

<sup>50</sup> Neste trabalho optou-se por manter a palavra em seu original em inglês (*feedback*) em função desta ser utilizada pelo mercado em sua grafia original; no caso de tradução, entende-se pelos processos de retorno a uma dada ação ou processo, envolvendo críticas, sugestões e identificação de pontos positivos e pontos a melhorar.

Cada um dos processos mencionados anteriormente deve ser classificado nestes níveis; com isso, obtém-se uma matriz que contempla os níveis de cada processo executado pela organização, que servirá para avaliar a situação atual e promover melhorias nos pontos críticos.

A norma ISO 15504 é composta por várias partes, a saber:

- Parte 1 - Guia de Introdução e Conceitos, inicialmente publicada em 1998 (ISO 1998b) e posteriormente atualizada em 2004 (ISO, 2004a) ;
- Parte 2 - Modelo de referência para processos e capacidade de processos, inicialmente publicada em 1998 (ISO, 1998c) e posteriormente atualizada em 2004 (ISO, 2004b) ;
- Parte 3 - Realizando uma avaliação, inicialmente publicada em 1998 (ISO, 1998d) e posteriormente incorporada à parte 2, em 2004 (ISO, 2004b) ;
- Parte 4 - Guia para realização de uma avaliação, inicialmente publicada em 1998 (ISO, 1998e) e atualizada em 2004 (ISO, 2004c) ;
- Parte 5 - Um modelo de avaliação e guia de indicadores, inicialmente publicada em 1998 (ISO, 1998f) e atualizada em 2004 (ISO, 2004d) ;
- Parte 6 - Guia para qualificação de avaliadores, inicialmente publicada em 1998 (ISO 1998g) e atualmente em estágio de revisão;
- Parte 7 - Guia para uso no melhoramento de processos, inicialmente publicada em 1998 (ISO 1998h) e atualmente em estágio de revisão;
- Parte 8 - Guia para uso na determinação da capacidade do processo do fornecedor, publicada em 1998 (ISO 1998i) e atualmente em estágio de revisão;
- Parte 9 - Vocabulário, inicialmente publicada em (ISO 1998j) e posteriormente incorporada à parte 1, em 2004 (ISO, 2004a) .

### **3.4.5 ISO 19759 - SWEBoK**

A norma ISO 19759:2005 (ISO, 2005b), mais conhecida como SWEBoK - *Guide to the Software Engineering Body of Knowledge*, corresponde ao resultado de um projeto

iniciado em 1998 e considerado concluído em 2004, contando com a participação de diversas instituições, entre as quais IEEE Computer, Boing, Rational Software e SAP, especialistas (como Roger S. Pressman e Ian Sommerville, referências na área de engenharia de software com várias e importantes publicações) e com mais de uma centena de revisores. O propósito é congregiar um conjunto de conhecimentos aceitos internacionalmente como pertinentes à atividade de engenharia de software.

O SWEBoK (IEEE, 2004) apresenta uma introdução ao guia, com conceitos iniciais sobre software e engenharia de software, a área de atuação, a interface com outras disciplinas, além de categorizar e descrever as onze áreas chave da engenharia de software, a saber:

- Requisitos de software;
- Projeto de software;
- Construção de software;
- Teste de software;
- Manutenção de software;
- Gerenciamento de configuração de software;
- Gerenciamento de engenharia de software;
- Processo de engenharia de software;
- Métodos e ferramentas de engenharia de software;
- Qualidade de software.

Cada área chave é detalhada em processos e sub-processos. Para ilustrar a estrutura do SWEBoK, a figura 9 - Área chave gerenciamento de engenharia de software – apresenta os processos de uma das áreas chave previstas no modelo.

É uma iniciativa muito importante para a área de desenvolvimento de software, merecendo especial atenção por parte dos desenvolvedores. Como ainda é recente, devem ser avaliados os impactos de sua adoção pela comunidade.

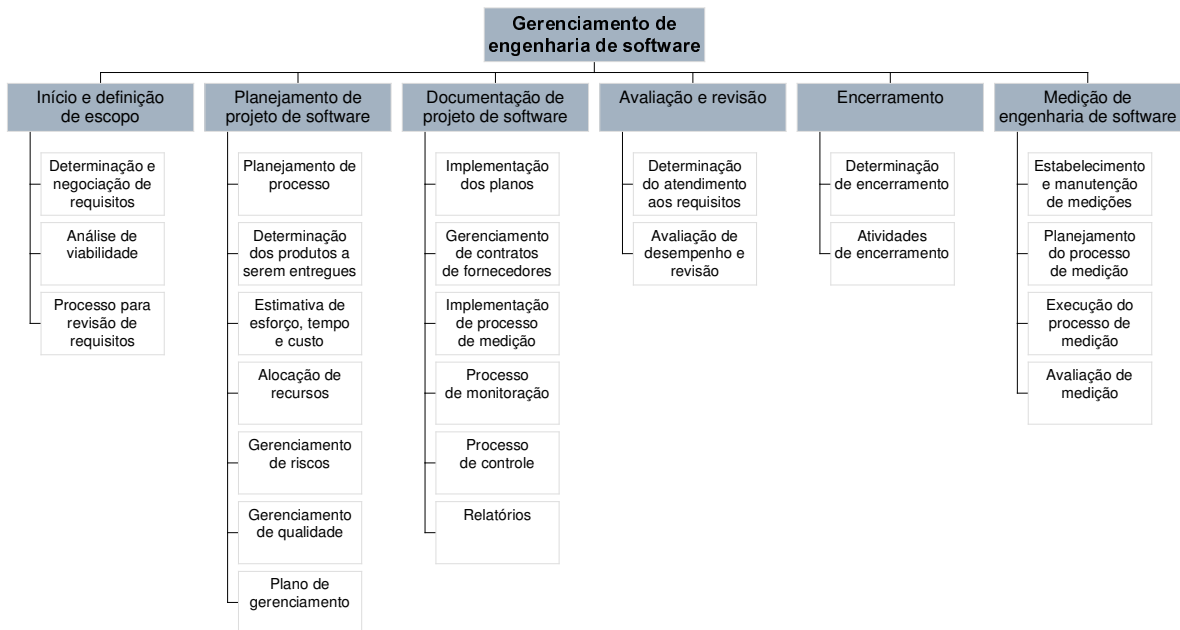


Figura 9 - Área chave gerenciamento de engenharia de software<sup>51</sup>

### 3.4.6 SEI – CMM e CMMI

#### 3.4.6.1 Histórico

Em novembro 1986, o SEI, em conjunto com a Mitre Corporation, iniciaram o desenvolvimento de um modelo (*framework*) de maturidade de processo que pudesse auxiliar as organizações a evoluírem seus processos de software. Este esforço foi iniciado a fim de atender a um pedido do governo federal norte-americano no sentido de prover um método que garantisse a capacitação de seus fornecedores de software. Em setembro 1987 o SEI publicou uma pequena descrição do modelo de maturidade de processo e um questionário de maturidade (HUMPHREY, 1995). O SEI pretendia utilizar este questionário como uma ferramenta para identificar quais áreas do processo de software das organizações pesquisadas necessitavam de melhorias. Infelizmente as empresas trataram o questionário como modelo - e não como veículo - para explorar os principais aspectos do processo de maturidade.

<sup>51</sup> Baseado em ISO (2005b).

Após quatro anos de experiência com o modelo de maturidade do processo de software e a versão preliminar do questionário de maturidade, o SEI gerou o Modelo de Capacidade e Maturidade para Software (CMM), em 1991 (PAULK et. al., 1993). O CMM é baseado no conhecimento adquirido e acumulado pelos processos de desenvolvimento de software e pelo extensivo *feedback* obtido tanto das indústrias quanto do governo (CMU; SEI, 1995).

A versão inicial do CMM (versão 1.0) foi revisada e utilizada pela comunidade de software durante os anos de 1991 e 1992. Um encontro foi realizado em abril 1992, do qual participaram cerca de 200 profissionais de software; como resultado deste encontro foi gerada a versão 1.1 (PAULK et. al., 1993).

Em 2000 o CMM foi atualizado para o CMMI (Integração); o novo modelo amplia os conceitos apresentados no CMM, focando em quatro grandes áreas: Engenharia de Sistemas, Engenharia de Software, Produção Integrada e Desenvolvimento do Processo (CMU; SEI, 2003). Esta versão procurou aproximar o CMM dos conceitos da norma ISO 15504, estudada anteriormente.

Outra mudança importante foi a criação de dois modelos para o CMMI:

- Modelo de representação do estágio atual da empresa, semelhante ao CMM, conforme (CMU; SEI, 2002b); este será o modelo de estudo deste trabalho;
- Modelo contínuo, que ilustra a constante evolução no estágio de maturidade da empresa, aproximando os conceitos do CMM com a norma ISO 15504, conforme (CMU; SEI, 2002a).

Está previsto para o segundo semestre de 2006 o lançamento do modelo CMMI versão 1.2.

#### **3.4.6.2 Conceitos utilizados**

O CMMI é uma base para a construção sistemática de um conjunto de ferramentas, incluindo o questionário de maturidade, o qual é útil no processo de melhoria da qualidade do software. O ponto essencial a ser sempre lembrado é que o

modelo - e não o questionário - forma a base para o aprimoramento do processo de software.

A estratégia é promover a evolução da engenharia de software de uma atividade desordenada e dispendiosa para uma atividade gerenciada, disciplinada e com qualidade de produtos controlada.

O CMMI, sendo um modelo para melhoria de processo de software, não define como implementar, e sim orienta na aplicação dos conceitos de gerência de projetos e de melhoria de qualidade para desenvolvimento e manutenção de software.

Entende-se por processo de software um conjunto de atividades, métodos, práticas e transformações que pessoas utilizam para desenvolver e manter software e seus produtos associados tais como planos de projeto, documentação, código, casos de teste e os manuais do usuário (PAULK et. al., 1993). Como resultado de uma organização madura, o processo de software torna-se melhor definido e mais consistentemente implementado através da organização. Nota-se que o CMMI avalia a empresa como um todo e não uma iniciativa isolada de apenas parte da empresa.

A capacidade do processo de software descreve o conjunto de resultados que podem ser atingidos seguindo-se um processo de software. A capacidade do processo de software de uma organização provê um meio de antecipar como serão os resultados com o próximo projeto de software que a organização irá conduzir.

O desempenho do processo de software representa os resultados reais atingidos seguindo-se o processo de software. Portanto o desempenho do processo de software tem seu foco nos resultados obtidos enquanto que a capacidade do processo de software foca os resultados esperados. Baseado nos atributos de um projeto específico e no contexto de como ele foi conduzido, o desempenho real de um projeto pode não refletir a capacidade total do processo de software da organização, isto é, a capacidade do projeto é restrita pelo ambiente. Por exemplo, mudanças radicais na aplicação e na tecnologia podem fazer com que os recursos alocados ao projeto tenham que passar por treinamentos que podem afetar a capacidade e o desempenho do projeto.

Maturidade do processo de software é uma extensão de como um processo específico é explicitamente definido, gerenciado, medido, controlado e produzido; implica em um potencial para crescimento na capacidade e indica que isto é aplicável a

toda organização (e não apenas a uma área restrita). O processo de software é melhor entendido pelas organizações maduras, usualmente através de documentação e treinamento, e o processo é continuamente monitorado e aperfeiçoado pelos seus usuários (MELLO FILHO, 1999a). A maturidade também implica que a produtividade e a qualidade resultantes do processo de software de uma dada organização podem ser sempre melhoradas através de análise do histórico dos processos já realizados utilizando este processo de software, revisando-o e mantendo este histórico organizado e atualizado. Isto é feito através da institucionalização de políticas, padrões e estruturas organizacionais que suportem, de forma corporativa, este processo (PAULK et. al., 1993).

#### **3.4.6.3 Níveis de maturidade em software**

O modelo foi organizado em cinco níveis, de acordo com a maturidade das organizações no tocante ao processo de software. Estes níveis definem uma escala para medir e classificar a maturidade do processo de software de uma dada organização e para avaliar a capacidade do processo de software; estes níveis também auxiliam as organizações a priorizarem seus esforços nas áreas carentes (PIRES, 2000). A figura 10 - Níveis de Maturidade CMMI - ilustra estes níveis e suas características principais.

Com a atualização para o CMMI, os níveis passam a ter a seguinte nomenclatura:

- Nível 1 – Inicial;
- Nível 2 – Gerenciado;
- Nível 3 – Definido;
- Nível 4 – Gerenciado quantitativamente;
- Nível 5 – Em otimização.

Cada nível de maturidade provê um conjunto de regras para a evolução contínua do processo de software; são compostos de um conjunto de objetivos de processo que, quando satisfeitos, tornam-se importantes componentes no processo de software. Para atingir cada nível de maturidade do modelo é necessário o estabelecimento de diversas

políticas e processos de software, resultando em aperfeiçoamento e capacitação da organização.

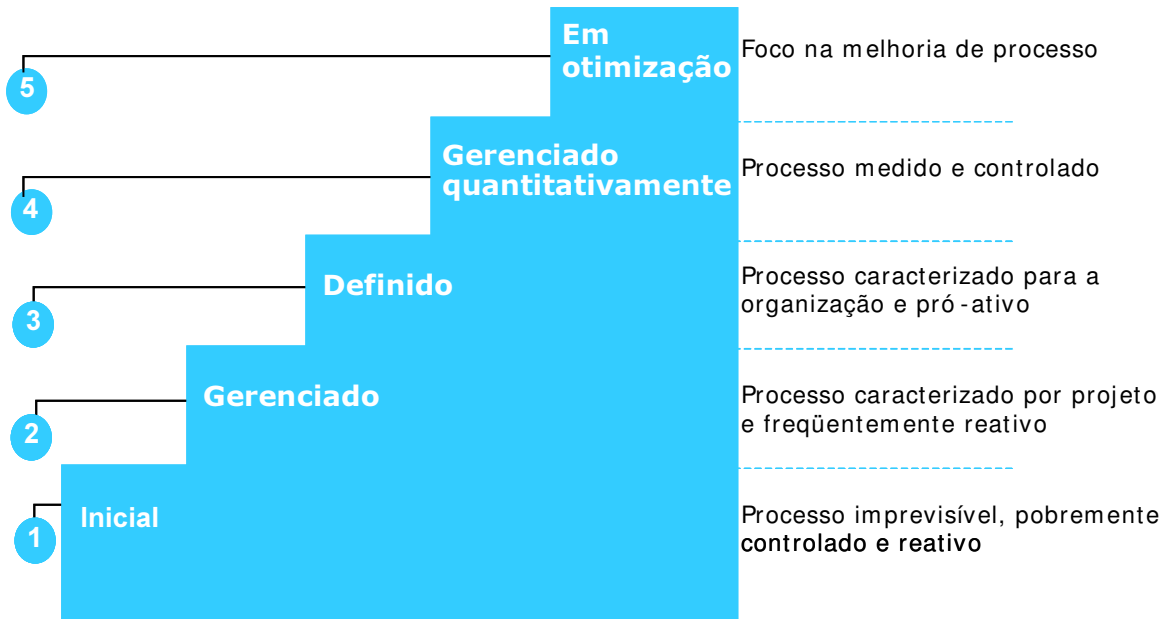


Figura 10 - Níveis de Maturidade CMMI <sup>52</sup>

Vale destacar que o nível 1 é o ponto de partida para a aplicação do CMMI, ou seja, é marcado pela ausência de controles satisfatórios.

O segundo nível é marcado pelo estabelecimento de processos básicos de gerenciamento de projetos a fim de verificar e acompanhar custos, prazos e funcionalidade. Um mínimo de disciplina no processo de armazenamento destas informações é requerido a fim de repetir os sucessos obtidos com projetos similares já realizados.

O terceiro nível é caracterizado pela documentação, padronização e integração dos processos de software, tanto gerenciais quanto técnicos, para uma organização como um todo. Todos os projetos passam por processos formais de aceitação na organização, tanto os projetos de desenvolvimento quanto os projetos de manutenção.

O quarto nível é marcado pela coleta de medidas detalhadas do processo de software e do produto final, com vistas à garantia da qualidade. Tanto o processo de

---

<sup>52</sup> Baseado em (CMU; SEI, 2002b).



software quanto o produto final produzido são quantitativamente compreendidos e controlados pela organização.

O quinto nível é caracterizado pelo contínuo processo de melhoria, habilitado por *feedback* quantitativo a partir dos processos e a partir da introdução, sempre acompanhada e avaliada, de novas idéias e tecnologias.

#### 3.4.6.4 Gerenciamento de software e o CMMI

Um dos benefícios a serem conseguidos com a evolução das organizações à luz do CMMI é a maior transparência e efetividade nos processos de gerenciamento. A figura 11 - Visão gerencial e o processo de software - ilustra a visibilidade gerencial em cada um dos cinco níveis.

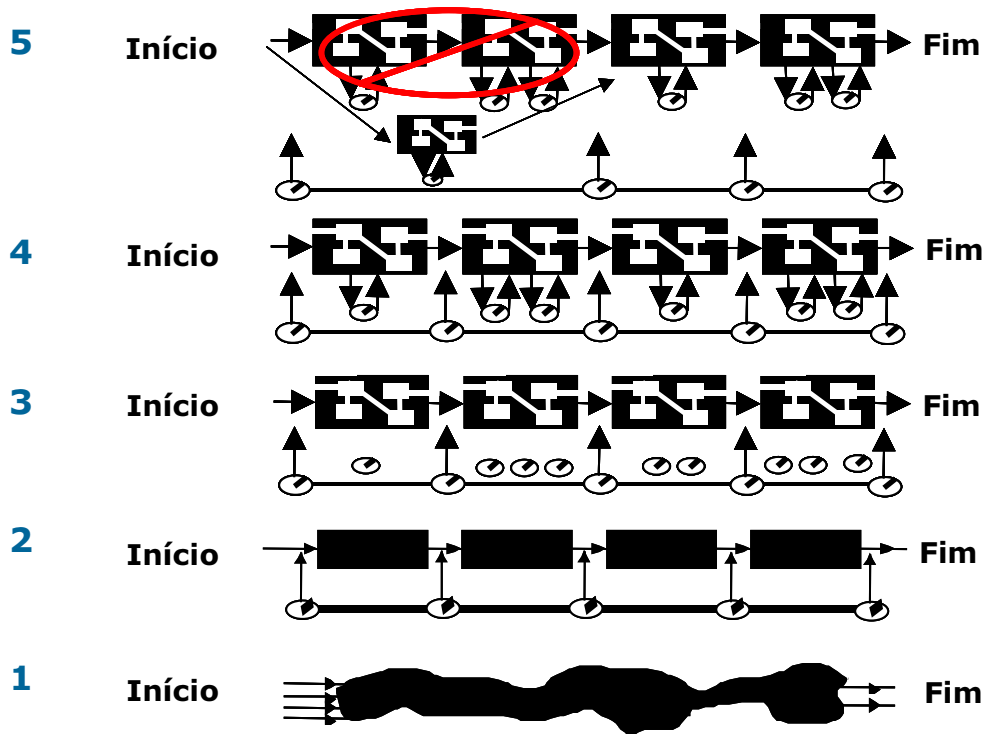


Figura 11 - Visão gerencial e o processo de software <sup>53</sup>

<sup>53</sup> Baseado em (CMU; SEI, 2003).

A visibilidade gerencial no nível 1 é nula, totalmente obscura; causando aos gerentes uma total incompreensão da real situação de cada projeto<sup>54</sup>. No nível 2 começam a existir marcos gerenciais - ainda que internamente a estes marcos continuem existindo "caixas pretas" - que possibilitam um mínimo de repetição e controle no processo; a postura do gerente ainda é reativa, ou seja, ele apenas reage aos problemas à medida que estes ocorram.

No nível 3 a estrutura interna das "caixas pretas" do nível 2 - as atividades envolvidas no processo de software - passa a estar visível. Tanto o pessoal técnico quanto o pessoal gerencial entendem as regras e responsabilidades sobre o processo e como as atividades interagem a cada nível de detalhe. Os gerentes passam a ter postura pró-ativa no que diz respeito aos riscos que possam vir a ocorrer.

No nível 4 o processo de software é documentado e controlado de forma quantitativa. Os gerentes estão habilitados a mensurar progressos e problemas dos projetos; também têm bases objetivas e quantitativas para a tomada de decisões.

No nível 5 novas formas de melhoria de construção de software são continuamente testadas, de forma controlada, a fim de melhorar a produtividade e a qualidade. A disciplina já implementada procura identificar formas ineficientes atualmente adotadas e substituí-las por processos otimizados. Os gerentes estão habilitados a estimar e a simular/avaliar os impactos quantitativos e a efetividade das mudanças.

Outro grande benefício a ser conseguido com a evolução das organizações à luz do CMMI é a melhoria do índice de previsão e de alcance dos reais objetivos dos projetos. A figura 12 - Capacidade de estimar e medir resultados dos projetos - ilustra esta melhoria em cada um dos cinco níveis.

---

<sup>54</sup> Paulk et. al. (1993) comentam que esta é a síndrome dos "noventa por cento", ou seja, 90% dos projetos em desenvolvimento estão com progresso em 90%. Infelizmente este percentual é sempre mantido...

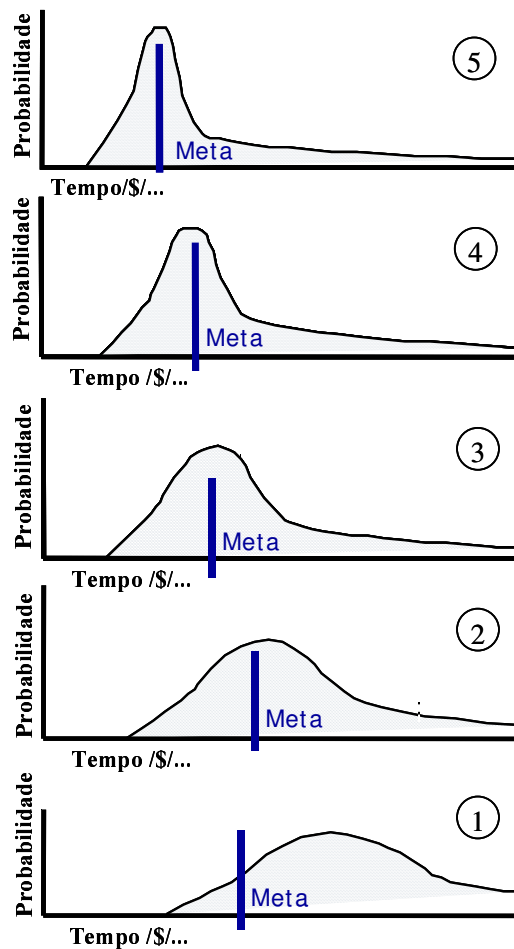


Figura 12 - Capacidade de estimar e medir resultados dos projetos <sup>55</sup>

Vale destacar que no nível 1 a meta inicial (*target*) é anterior ao do nível 2, muito mais em função da total falta de histórico e da ausência de mecanismos formais de estabelecimento de metas; de fato percebe-se que esta meta inicial será atrasada - mas certamente já é muito mais real que a observada no nível 1. A evolução é gradual, com as metas sendo antecipadas e o grau de incerteza sendo gradativamente diminuído.

Percebe-se, então, que, como resultados atingidos, tem-se uma maior aproximação entre a estimativa original e o resultado atingido após a realização do projeto; uma concentração maior de projetos nesta área da estimativa inicial e uma antecipação no processo de estimativas, provocada pela substituição das tradicionais

<sup>55</sup> Baseado em (CMU; SEI, 2003).

"folgas" - muitas vezes não baseada em fatos concretos - por estimativas a partir de históricos de projetos já efetuados.

### 3.4.6.5 Estrutura do CMMI

Com relação à estrutura do CMMI, cada nível pode ser decomposto em partes componentes, conforme ilustra a figura 13 - Componentes do modelo CMMI.

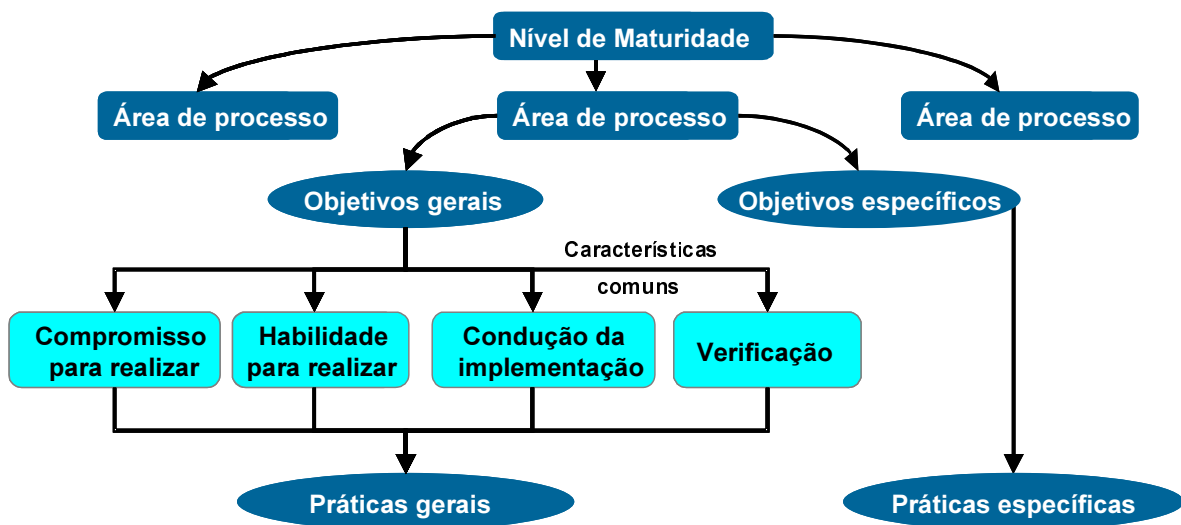


Figura 13 - Componentes do modelo CMMI<sup>56</sup>

Cada nível de maturidade é composto por várias áreas chaves de processo (*key process areas* - KPA); cada KPA é organizada em cinco seções chamadas de funções comuns; cada função comum especifica as práticas chaves que, quando coletivamente (toda a organização) endereçada, torna possível atingir os objetivos da área chave de processo.

<sup>56</sup> Baseado em (CMU; SEI, 2003).

#### 3.4.6.5.1 Áreas chaves de processos

Áreas chaves de processo indicam onde uma organização deve focalizar a melhoria dos processos de software. Cada área chave do processo identifica um ramo de atividades relativas que, quando realizadas coletivamente, terminam com êxito uma série de metas consideradas importantes para o aumento da capacidade dos processos. O caminho para realizar com êxito estas metas pode variar por projeto, dependendo do domínio da aplicação ou do ambiente. Todavia, todas as metas de uma área de processo devem ser realizada com êxito por toda a organização para satisfazê-la.

O uso da palavra "chave" implica que existem áreas de processo (e processos) que não são chave para resultar em um nível de maturidade. O CMMI não descreve em detalhes todas as áreas de processo envolvidas com o desenvolvimento e manutenção do software, apenas aquelas que devem ser identificadas como chaves definidas de capacidade dos processos.

Áreas chaves do processo podem ser vistas como requerimentos para soluções de níveis de maturidade. Para se atender com êxito um dado nível de maturidade, as áreas chaves de processo para o nível devem estar satisfeitas/realizadas - vale lembrar que o nível 1 não tem áreas chaves de processos.

Cada área chave é acompanhada por cinco características que definem as práticas chaves:

- compromisso para realizar os preceitos da área-chave;
- habilidade para realizar estes preceitos;
- forma como estas atividades são realizadas;
- medição e análise da efetividade;
- verificação da implementação dos preceitos.

A prática específica a ser executada em cada área chave de processo poderá envolver características de níveis mais elevados de maturidade. Por exemplo: para se atingir muitas das estimativas de projetos descritas como capacidade em planejamento de projetos, na área chave de processo para o nível 2, pode ser necessário desenvolver o processo de coleta dos dados, presente no nível 3.

#### 3.4.6.5.2 Áreas chave de processo para o nível 2 do CMMI

As áreas chave de processo no nível 2 focalizam-se no que se refere à determinação dos controles para os administradores do projeto. São sete áreas chave, a saber:

- Gerenciamento de requisitos: significa estabelecer um entendimento comum entre um cliente (requisitante) e uma equipe de projeto do produto requerido. Este consentimento é a base para o planejamento e o gerenciamento de um projeto;
- Planejamento do projeto de software: significa estabelecer planos razoáveis para a engenharia e administração do projeto;
- Monitoramento e controle do projeto de software: significa verificar o real andamento do projeto frente ao planejado, possibilitando adotar medidas preventivas e corretivas, quando necessário;
- Gerenciamento do subcontratados (terceiros e fornecedores) de software: significa selecionar subcontratados qualificados e gerir os contratos em função do projeto;
- Análise e medição: significa definir e acompanhar indicadores de andamento do projeto;
- Garantia de qualidade do produto e do processo: significa estabelecer gerenciamento, com visibilidade apropriada, ao longo dos processos e nos produtos criados;
- Gerenciamento de configuração do software: significa estabelecer e manter a integridade de um projeto, e seus produtos, por todo o ciclo de vida.

#### 3.4.6.5.3 Áreas chave de processo para o nível 3 do CMMI

As áreas chave de processo do nível 3 endereçam tanto projeto quanto organização empresarial, com o estabelecimento de uma infra-estrutura que institucionaliza a engenharia de software efetiva e a gerência de processos sobre todos os projetos. São 14 áreas chave, a saber:

- Desenvolvimento de requisitos;
- Soluções técnicas;
- Integração do produto;
- Verificação;
- Validação;
- Foco no processo organizacional;
- Definição do processo organizacional;
- Treinamento organizacional;
- Gerenciamento integrado do projeto;
- Gerenciamento de riscos;
- Integração do time do projeto;
- Gerenciamento integrado de fornecedores;
- Análise e tomada de decisão;
- Ambiente organizacional para integração.

#### 3.4.6.5.4 Áreas chave de processo para o nível 4 do CMMI

As áreas chave de processo para o nível 4 visam estabelecer um entendimento quantitativo do processo de desenvolvimento de software e do processo de trabalho criado pelo produto gerado com o software. São duas áreas chave, a saber:

- Gerenciamento quantitativo do projeto: significa controlar a execução, de forma quantitativa, de um projeto de software; inclui o resultado atualmente obtido através da comparação com projetos de software anteriormente

executados. O foco está na identificação das causas fundamentais da variação entre a medida do projeto atual e a prevista anteriormente.

- Desempenho do processo organizacional: significa coletar e prover indicadores quantitativos que possibilitem avaliar o desempenho do processo organizacional, compara-lo com outras organizações e buscar a melhoria contínua.

#### 3.4.6.5.5 Áreas chave de processo para o nível 5 do CMMI

As áreas chave de processo para o nível 5 focalizam-se nas modificações que ambos, organização e projeto, tem que implementar visando aprimoramento contínuo e baseado nas melhorias obtidas a partir das medições dos processos realizados. São duas áreas chave, a saber:

- Gerenciamento das mudanças de tecnologia: significa identificar os benefícios das novas tecnologias (como ferramentas, métodos e processos) e transferi-las para a organização de maneira ordenada; significa continuamente melhorar os processos de uma organização com o intuito de melhorar a qualidade, aumentar a produtividade e diminuir o tempo de desenvolvimento de software. O foco aqui é a eficiência interna que a inovação poderá trazer.
- Análise das causas e resolução: significa identificar as causas dos defeitos e outros problemas e tomar medidas preventivas para que não ocorram no futuro.

A preocupação do presente trabalho é conhecer os preceitos do CMMI e utilizá-lo, sempre em comparação com outros modelos, para procurar responder às perguntas chaves desta dissertação - o gerenciamento de projetos de desenvolvimento de software poderia ser otimizado com a introdução de práticas de gerenciamento de projetos tradicionalmente utilizadas em outras áreas? Quais as adaptações necessárias para o tratamento de software? Como relacionar com os modelos específicos para o



desenvolvimento de software (como o CMMI)? O capítulo 5 - Análise comparativa - apresenta estas considerações.

### 3.4.6.6 Adoção do CMMI pelas empresas

As empresas que desejam adotar o CMMI como padrão para seus processos de software podem se submeter à visita (avaliação) para verificação e enquadramento em qual nível do CMMI a mesma se encontra. Normalmente as empresas fazem um trabalho preliminar de preparação, contando com o apoio de empresas de consultoria, a fim de estabelecer as áreas chaves, verificar a uniformidade do processo ao longo de toda a organização, realizar avaliações prévias para então se submeter a um processo de avaliação. No Brasil o principal órgão avaliador é a Fundação Vanzolini.

A figura 14 - Panorama mundial das empresas que adotam o modelo CMMI - ilustra o panorama mundial da adoção do CMMI pelas empresas, no total de 696 empresas (até junho de 2006). Percebe-se uma concentração nos níveis iniciais do modelo (níveis 2 e 3).

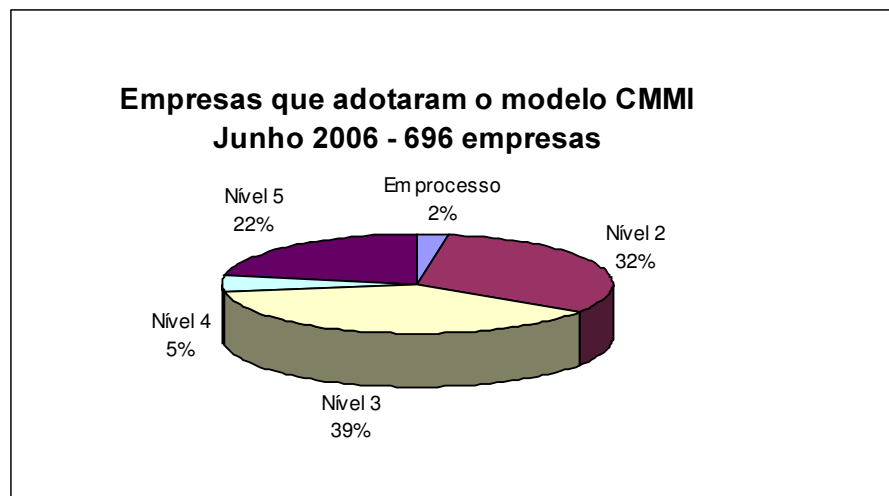


Figura 14 - Panorama mundial das empresas que adotam o modelo CMMI <sup>57</sup>

<sup>57</sup> Dados coletados e organizados a partir de (CMU; SEI, 2006).

Com relação à adoção do CMMI pelas empresas nacionais tem-se que poucas já adotaram e, das já adotaram, a maioria está nos níveis 2 e 3. A figura 15 - Panorama nacional das empresas que adotam o modelo CMMI - ilustra o panorama da adoção do CMMI pelas empresas no Brasil, no total de apenas 14 empresas (até dezembro de 2005), indicando um número pequeno em relação ao número mundial de empresas.

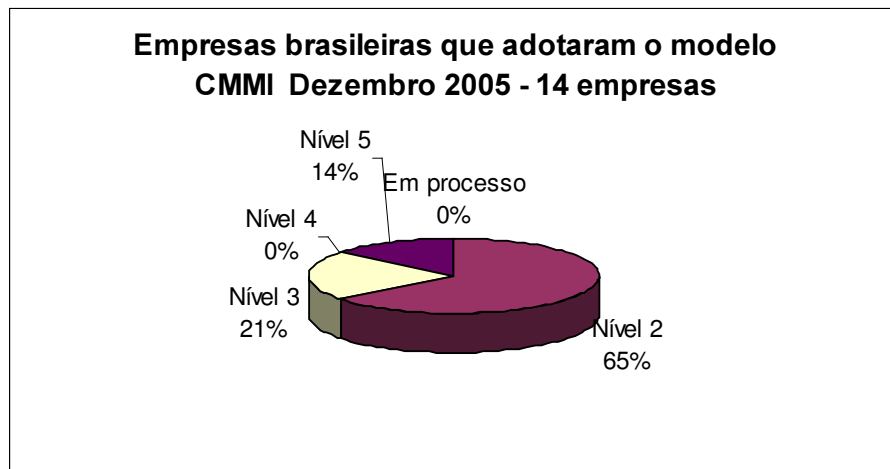


Figura 15 - Panorama nacional das empresas que adotam o modelo CMMI <sup>58</sup>

Em Volpe e Zabeu (1997) tem-se como a NEC Brasil iniciou a adoção do modelo CMM; outro exemplo pode ser obtido em Credicard (1999), onde a Credicard S/A apresenta os resultados obtidos com a implantação do modelo CMM e seu processo de enquadramento inicial - nível 2 - visando a implantação dos processos adequados para atingir os níveis mais elevados; em Stefanini (2005) tem-se a notícia da conquista do nível 5 do CMMI pela Stefanini e seu impacto, sendo a primeira empresa brasileira a atingir esse nível (as demais empresas no Brasil que possuem o nível 5 são multinacionais). Catho (2006) relata o processo de adoção do modelo CMMI pelo Instituto Eldorado, atualmente CMMI nível 3 e caminhando para CMMI nível 4 em 2007.

---

<sup>58</sup> Dados coletados e organizados a partir de (MCT, 2006b).

### 3.4.7 SEI - PSP

Os modelos CMM e CMMI são aplicados às empresas de software. O SEI percebeu a necessidade de definir um modelo voltado para o profissional de software e criou, em 1993, o PSP, que significa "Processo Pessoal de Software".

O PSP prevê um modelo de maturidade, um guia do conjunto de conhecimentos e uma certificação (SEI - *Certified PSP Developer certifications*), ainda em desenvolvimento.

O modelo de maturidade do PSP prevê a melhoria contínua na produção de software. A tabela 7 - Níveis e atividades do modelo SEI-PSP - ilustra esse modelo de maturidade.

Tabela 7 - Níveis e atividades do modelo SEI-PSP <sup>59</sup>

Nível	Atividades
<b>PSP0</b>	Processo atual; Registro de tempo de desenvolvimento; Registro de defeitos; Registro de tipos de defeito.
<b>PSP0.1</b>	Padronização de código; Medição de tamanho; Plano de melhoria de processo.
<b>PSP1</b>	Estimativa de tamanho; Relatório de teste.
<b>PSP1.1</b>	Planejamento das tarefas; Planejamento do projeto.
<b>PSP2</b>	Revisão de código; Revisão de projeto.
<b>PSP2.1</b>	Desenvolvimento de documentos modelo ( <i>templates</i> ).
<b>PSP3 (TSP)</b>	Desenvolvimento da equipe do projeto.

O guia do conjunto de conhecimentos do PSP (CMU; SEI, 2005b), agrupa os principais conceitos relativos ao desenvolvimento de software em sete áreas de competência:

---

<sup>59</sup> Baseado em (CMU; SEI, 2005a).

- Área 1: fundamentação dos conceitos;
- Área 2: conceitos básicos do Processo Pessoal de Software;
- Área 3: estimativa e mensuração de tamanho de software;
- Área 4: elaboração e acompanhamento de planejamento de software;
- Área 5: planejamento e controle de qualidade de software;
- Área 6: projeto de software;
- Área 7: extensões de processo.

O modelo prevê, para cada atividade descrita na tabela 7, padrões e procedimentos, permitindo a comparação com outros indivíduos que adotem o modelo. O treinamento do PSP, previsto em CMU e SEI (2005a), é realizado através da resolução de vários exercícios de desenvolvimento de programas. Além de servirem como exemplos de desenvolvimento, os exercícios propostos no treinamento do PSP são pequenos utilitários que ajudam aos indivíduos a aplicarem o PSP, pois permitem medir o número de linhas e objetos nos seus programas, calcular desvio padrão, prever intervalos etc.

#### **3.4.8 SOFTEX – mps Br**

O projeto mps Br - melhoria de processo do software brasileiro - é uma iniciativa sob a coordenação da SOFTEX e envolve três instituições de ensino, pesquisa e centros tecnológicos (Coppe/UFRJ e Universidade Católica de Brasília, Cesar, GenPRA); uma sociedade de economia mista, a Companhia de Informática do Paraná (Celepar) e dois Agentes SOFTEX, situados no Rio de Janeiro (Riosoft) e Campinas (SOFTEX Campinas).

Iniciado em 2003<sup>60</sup>, o projeto prevê um modelo de referência de processos, baseado nos conceitos de maturidade e capacidade de processo, para a avaliação e

---

<sup>60</sup> Weber e Cavalcanti (1999) apresentam alguns dados referentes às pesquisas efetuadas pela SEPIN/MCT (Secretaria de Política de Informática e Automação do Ministério da Ciência e Tecnologia), em conjunto com PBQP/SSQP-SW (Programa Brasileiro de Qualidade e Produtividade em Software), nos anos de 1993, 1995 e 1997, apresentando um panorama do uso dos padrões de qualidade pelas empresas nacionais. Esse panorama serviu de base para a criação do modelo mps Br.

melhoria da qualidade e produtividade de produtos e serviços de software, visando, por um lado, uma maior utilização pela indústria nacional e, por outro lado, uma capacitação para definição de ajustes e adaptações dos modelos internacionais para alinhamento com a realidade das decisões da política brasileira de software. Também prevê um modelo de negócio para melhoria de processo de software; esse modelo de negócio tem grande potencial de replicabilidade no Brasil e em outros países de características semelhantes no que se refere à indústria de software – a idéia é ter esse modelo adaptado para cada país, em especial para a América Latina, onde os contatos para potenciais parcerias já estão em andamento.

A figura 16 - Modelo de referência mps Br - ilustra os principais componentes do modelo, além de suas principais fontes (os modelos ISO 12207, ISO 15504 e SEI-CMMI).

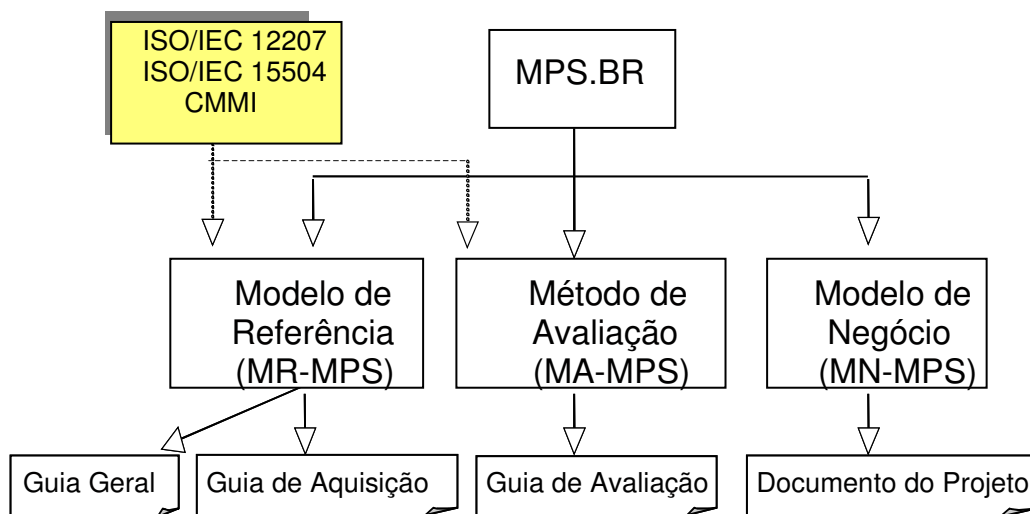


Figura 16 - Modelo de referência mps Br <sup>61</sup>

O mps Br é um projeto estruturante que vai promover a qualificação de um grupo amplo de empresas compatível com os padrões de qualidade aceitos internacionalmente pela comunidade de software, a custos acessíveis para a grande maioria das empresas brasileiras, sendo adequado ao perfil e cultura das empresas de software do país.

<sup>61</sup> Dados coletados e organizados a partir de SOFTEX (2005).

Os principais produtos do projeto mps Br serão os resultados de melhoria em um conjunto de empresas; um modelo de melhoria de processos de software (considerando as práticas da comunidade de software, nacional e internacional, com interpretação e aplicações com foco na estratégia competitiva das empresas brasileiras de software); um método de avaliação de processos; e orientações para a utilização destes resultados.

O modelo proposto prevê sete níveis de maturidade:

- Nível G - Parcialmente Gerenciado - primeiro nível do modelo, é composto pelos processos de gerência de projeto e gerência de requisitos;
- Nível F – Gerenciado - é composto pelo nível de maturidade G acrescido dos processos de gerência de configuração, garantia da qualidade, medição e aquisição;
- Nível E - Parcialmente Definido - é composto pelo nível de maturidade F, acrescido dos processos de treinamento, definição do processo organizacional, avaliação e melhoria do processo organizacional e adaptação do processo para gerência de projetos;
- Nível D - Largamente Definido - é composto pelo nível de maturidade E, acrescido dos processos de desenvolvimento de requisitos, solução técnica, validação, verificação, integração de produto, instalação de produto e liberação de produto;
- Nível C – Definido - é composto pelo nível de maturidade D, acrescido dos processos de gerência de riscos e análise de decisão e resolução;
- Nível B - Gerenciado Quantitativamente - é composto pelo nível de maturidade C, acrescido dos processos de desempenho do processo organizacional e gerência quantitativa de projeto;
- Nível A - Em Otimização – é o nível mais elevado do modelo mps Br; é composto pelo nível de maturidade B, acrescido dos processos de inovação e implantação na organização e análise e resolução de causas.

A adoção do modelo mps Br pelas empresas começou em 2005. Em MCT (2006a) estão as avaliações realizadas até o junho de 2006, com sete empresas, das quais duas estão no nível G, três no nível F, uma no nível E e uma no nível A.

## Capítulo 4

### Padrões de Gerenciamento de Projetos

#### 4.1 ISO 10006

A norma ISO 10006:1997 (ISO, 1997a) se refere às atividades de gerenciamento da qualidade no tocante ao gerenciamento de projetos; provê guias para os elementos do sistema de qualidade, conceitos e práticas para aprimorar e garantir a qualidade no processo de gerenciamento de projetos. Foi desenvolvida para ser aplicada ao gerenciamento de projetos de qualquer porte ou complexidade, servindo de base para que profissionais de gerenciamento de projetos e auditores de qualidade possam trocar experiências a respeito do projeto. Também teve por base a norma ISO 10005:1995 (ISO, 1995a), aplicável mais especificamente na elaboração de planos de qualidade. Foi atualizada em 2003, gerando a norma ISO 10006:2003 (ISO, 2003a).

Essa norma faz a distinção entre os dois aspectos básicos da aplicação dos conceitos de qualidade no gerenciamento de projetos: a qualidade do processo do projeto e a qualidade do produto do processo, considerando que ambos aspectos são vitais para o sucesso dos projetos, requerendo uma abordagem sistemática e contínua de forma a garantir a conformidade.

Também descreve, de forma resumida, os processos de gerenciamento de projetos que são considerados como aplicáveis à maioria dos projetos. Estão agrupados por afinidade em dez grupos:

- Grupo de processos estratégicos: primeiro grupo, relativo aos processos estratégicos necessários para legitimar e definir direções para o projeto;
- Grupo de processos de gerenciamento de interdependência: o segundo grupo que tem por finalidade gerenciar as interdependências entre os outros processos; tem como processos principais:

- 5.3.1. Iniciação do projeto e desenvolvimento do plano do projeto: consiste na análise dos requisitos dos clientes e demais envolvidos com o projeto, preparação de um plano de projeto e início dos demais processos;
- 5.3.2. Gerenciamento das interações: consiste na forma de gerenciar as interações ao longo do projeto;
- 5.3.3. Gerenciamento de mudanças: consiste em antecipar às mudanças e gerenciá-las ao longo do projeto;
- 5.3.4. Encerramento: consiste em encerrar os processos e obter *feedback*.
- Grupo de processos relacionados ao escopo: tem por finalidade gerenciar as relações envolvendo escopo do projeto, tendo como processos principais:
  - 5.4.1. Desenvolvimento conceitual: consiste na definição genérica daquilo que o produto do projeto deverá ser;
  - 5.4.2. Desenvolvimento e controle do escopo: consiste na documentação das características do produto a ser produzido pelo projeto em termos mensuráveis e controláveis;
  - 5.4.3. Definição das atividades: consiste na identificação e documentação das atividades e tarefas requeridas para que o projeto atinja seus objetivos;
  - 5.4.4. Controle das atividades: consiste em controlar o trabalho em desenvolvimento no projeto.
- Grupo de processos relacionados ao tempo: tem por finalidade gerenciar as relações envolvendo tempo/prazos no projeto, tendo como processos principais:
  - 5.5.1. Planejamento de dependências das atividades: consiste na identificação de relacionamentos e interações lógicas, além das dependências, entre as atividades do projeto;



- 5.5.2. Estimativa de duração: consiste na estimativa de duração de cada atividade associada com condições específicas e com os recursos necessários para realizá-la;
- 5.5.3. Desenvolvimento do cronograma: consiste na identificação das relações entre as metas de tempo para o projeto, as dependências entre as atividades e suas durações estimadas, visando a construção de um modelo (cronograma) para o desenvolvimento geral e posteriores detalhamentos do projeto;
- 5.5.4. Controle do cronograma: consiste no controle de realização das atividades do projeto em conformidade com o cronograma proposto, tomando as medidas adequadas para reconduzir o andamento ao inicialmente acordado.
- Grupo de processos relacionados ao custo: tem por finalidade gerenciar as relações envolvendo custo no projeto, tendo como processos principais:
  - 5.6.1. Estimativa de custo: consiste no desenvolvimento das estimativas de custo para o projeto;
  - 5.6.2. Orçamento: consiste na utilização dos resultados das estimativas de custo de forma a produzir um orçamento para o projeto;
  - 5.6.3. Controle de custo: consiste no controle dos custos e medidas a fim de evitar desvios em relação ao orçamento.
- Grupo de processos relacionados aos recursos: tem por finalidade gerenciar as relações envolvendo os recursos necessários ao projeto, tendo como processos principais:
  - 5.7.1. Planejamento de recursos: consiste em identificar, estimar, nivelar, distribuir e associar/alocar todos os recursos relevantes para o projeto;
  - 5.7.2. Controle de recursos: consiste na comparação do consumo atual de recursos frente ao previsto, tomando ações para evitar desvios.

- Grupo de processos relacionados ao pessoal: tem por finalidade gerenciar as relações envolvendo o pessoal necessário ao projeto, tendo como processos principais:
  - 5.8.1. Definição de estrutura organizacional: consiste na definição da estrutura organizacional necessária para possibilitar a execução adequada do projeto, incluindo regras para o projeto e definindo autoridades e responsabilidades;
  - 5.8.2. Alocação da equipe: consiste na seleção e na contratação (ou alocação) do pessoal necessário com as competências<sup>62</sup> apropriadas para as necessidades do projeto;
  - 5.8.3. Desenvolvimento da equipe: consiste no desenvolvimento das necessidades individuais e coletivas do grupo, objetivando melhorias nas competências e habilidades a fim de otimizar a performance do projeto.
- Grupo de processos relacionados à comunicação: tem por finalidade gerenciar as relações envolvendo as comunicações necessárias ao projeto, tendo como processos principais:
  - 5.9.1. Planejamento da comunicação: consiste em planejar as informações e sistemas de comunicações para o projeto;
  - 5.9.2. Gerenciamento das informações: consiste em tornar as informações necessárias disponíveis para a equipe e para os demais envolvidos no projeto;
  - 5.9.3. Controle da comunicação: consiste em controlar as comunicações de acordo com o plano de comunicações.
- Grupo de processos relacionados ao risco: tem por finalidade gerenciar as relações envolvendo os riscos existentes no projeto, tendo como processos principais:

---

<sup>62</sup> Utiliza-se o termo competência para definir, segundo Setzer (2006), "a capacidade de executar algo no "mundo real", diferenciando-se do conhecimento - que é sempre pragmático - por exigir mais do que informação - exige um conhecimento pessoal e a demonstração de uma habilidade".

- 5.10.1. Identificação dos riscos: consiste na determinação dos riscos envolvidos com o projeto;
  - 5.10.2. Estimativa de riscos: consiste em analisar a probabilidade da ocorrência de um evento de risco e o impacto deste evento de risco no projeto;
  - 5.10.3. Desenvolvimento de reação ao risco: consiste em desenvolver um plano de resposta aos riscos;
  - 5.10.4. Controle de riscos: consiste em implementar e atualizar o plano de resposta aos riscos.
- Grupo de processos relacionados a suprimentos: tem por finalidade gerenciar as relações envolvendo os suprimentos relativos ao projeto, tendo como processos principais:
    - 5.11.1. Planejamento e controle de suprimentos: consiste em identificar e controlar o que será comprado e quando;
    - 5.11.2. Documentação dos requisitos: consiste em compilar as condições comerciais e técnicas dos requisitos;
    - 5.11.3. Avaliação dos fornecedores: consiste em avaliar e determinar quais fornecedores podem ser convidados para o fornecimento dos produtos;
    - 5.11.4. Subcontratação: consiste em controlar as atividades relativas às propostas, avaliação das propostas, negociação, preparação e a contratação de atividades e produtos de fornecedores;
    - 5.11.5. Controle do contrato: consiste em garantir que a performance dos fornecedores condiz com os requisitos explicitados na contratação.

A norma contempla ainda um glossário com os principais termos utilizados e uma relação entre os processos aqui descritos com a série ISO 9000.

## 4.2 PMI - PMBoK

### 4.2.1 Apresentação

Segundo PMI (2004), o PMBoK Guide (Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos) é um termo que descreve um conjunto de conhecimentos referentes à atividade de gerenciamento de projetos; assim como em outras profissões (medicina, por exemplo), este conjunto de conhecimento é gerado e atualizado pelos acadêmicos e pelos profissionais, através de práticas tradicionalmente utilizadas, estudos e pesquisas em desenvolvimento, entre outras atividades.

O propósito original do documento escrito pelo *Standards Committee* do PMI (e adotado pelo PMI como seu documento base), tendo como seu diretor principal William R. Duncan, é o de identificar e descrever um subconjunto deste conjunto de conhecimentos referentes ao gerenciamento de projetos que tenha uma aceitação global, ou seja, reunir um subconjunto que seja aplicável à maioria dos projetos e que tenha seu valor reconhecidamente comprovado e aceito pela comunidade<sup>63</sup>.

A primeira iniciativa de consolidar este conjunto de conhecimento ocorreu em 1987 (atualmente conhecida como versão 0 do PMBoK Guide); em 1996 foi publicada a Edição 1996 do PMBoK Guide (conhecida atualmente como Primeira Edição), com diversas atualizações em relação à proposta anterior – é a base para o formato da versão atual; em 2000 foi publicada a Edição 2000 do PMBoK Guide (conhecida atualmente como Segunda Edição), com atualizações pontuais, principalmente no capítulo dedicado aos processos de gerenciamento de riscos de projeto; finalmente, em 2004, foi publicada a versão 2004 – Terceira Edição – do PMBoK Guide, com atualização em diversos capítulos – em especial nos capítulos dedicados aos processos de integração.

---

<sup>63</sup> Vale ressaltar que como a natureza de cada projeto é distinta, este subconjunto não tem a intenção de ser aplicado uniformemente a todos os projetos. Cada equipe deverá fazer uma análise preliminar e verificar quais os itens são aplicáveis ao projeto em questão.

A figura 17 – Disciplinas relacionadas ao gerenciamento de projetos - ilustra o relacionamento e a interpolação dos conhecimentos entre as diversas disciplinas de gerenciamento.

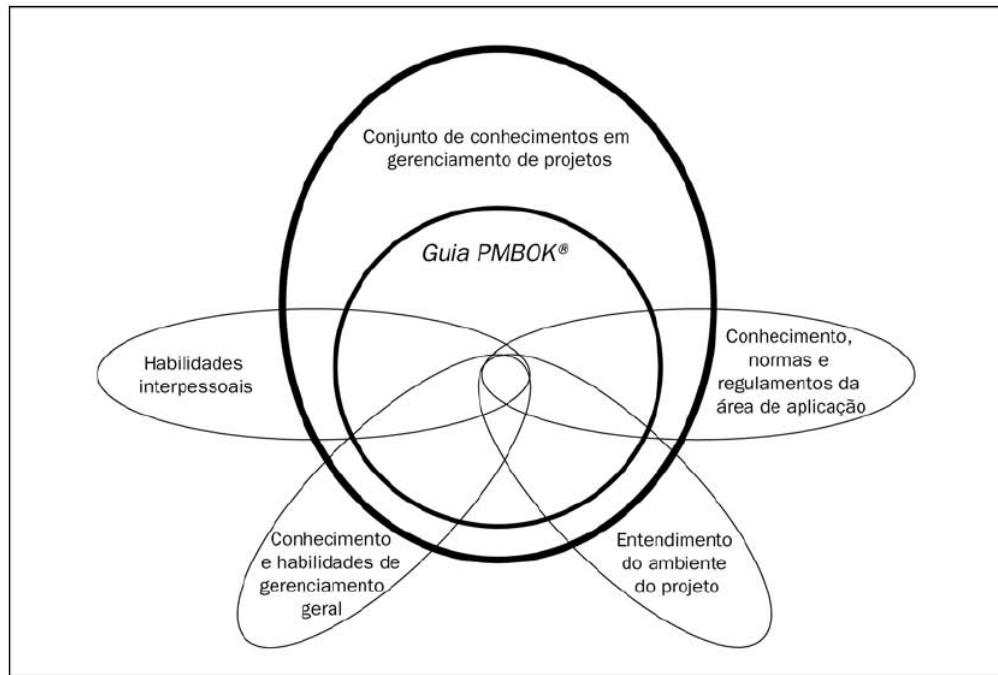


Figura 17 – Disciplinas relacionadas ao gerenciamento de projetos <sup>64</sup>

A área de gerenciamento geral se refere ao conjunto dos conhecimentos que englobam planejamento, organização, execução e controle das atividades das organizações.

A área de aplicações específicas engloba os aspectos específicos de algumas áreas de atuação. Este trabalho aborda uma destas áreas: a de desenvolvimento de software.

Outro objetivo do PMBoK é estabelecer um conjunto comum de termos sobre a disciplina gerenciamento de projetos, que ainda é recente e congrega profissionais de diversas origens acadêmicas. O processo de certificação de profissionais realizado pelo

---

<sup>64</sup> Conforme PMI (2004).

PMI, descrito em detalhes no item 2.2.1, Project Management Institute, se baseia, entre outros, neste subconjunto.

#### **4.2.2 Contexto do gerenciamento de projetos**

Todo projeto está inserido em um ambiente com características peculiares. Conhecer estas características é responsabilidade do time que conduzirá o projeto e é uma necessidade premente no sentido de garantir o sucesso desta empreita.

##### **4.2.2.1 Ciclos de vida**

Uma vez que todo projeto tem como característica básica ser único, estão implícitas em sua concepção e desenvolvimento algumas incertezas. Para diminuir essas incertezas, usualmente um projeto é dividido em fases, que auxiliam a determinar os objetivos e a gerenciar adequadamente o andamento das atividades. Estas diversas fases são conhecidas como "ciclo de vida do projeto", segundo PMI (2004).

O capítulo 3, Desenvolvimento de software, da presente dissertação apresenta os principais modelos para projetos de software.

Para cada fase devem ser identificados quais são os *deliverables*<sup>65</sup>, ou seja, qual o trabalho a ser executado na fase. Outro ponto importante é a identificação de quem deve ser envolvido em cada fase.

##### **4.2.2.2 Estruturas organizacionais**

---

<sup>65</sup>Termo utilizado na sua grafia em inglês, que significa os produtos/serviços a serem entregues/produzidos. Deve ser tangível e verificável em termos de sua conclusão. Como exemplos de *deliverables* tem-se a elaboração de um documento (projeto conceitual), a conclusão de um levantamento ou a construção de um protótipo.

A estrutura da organização que executa o projeto regula a disponibilidade e o grau de envolvimento dos recursos humanos da organização com o projeto. O PMI (2004) propõe um espectro contínuo de estruturas organizacionais que vai da estrutura puramente funcional àquela totalmente orientada para projetos.

Em estruturas puramente funcionais, o escopo percebido do projeto, sua coordenação e execução estão limitados a uma divisão funcional da empresa, conforme ilustra a figura 18 - Gerenciamento de projetos em uma organização funcional. Vale notar que as células destacadas representam especialistas envolvidos nos projetos.

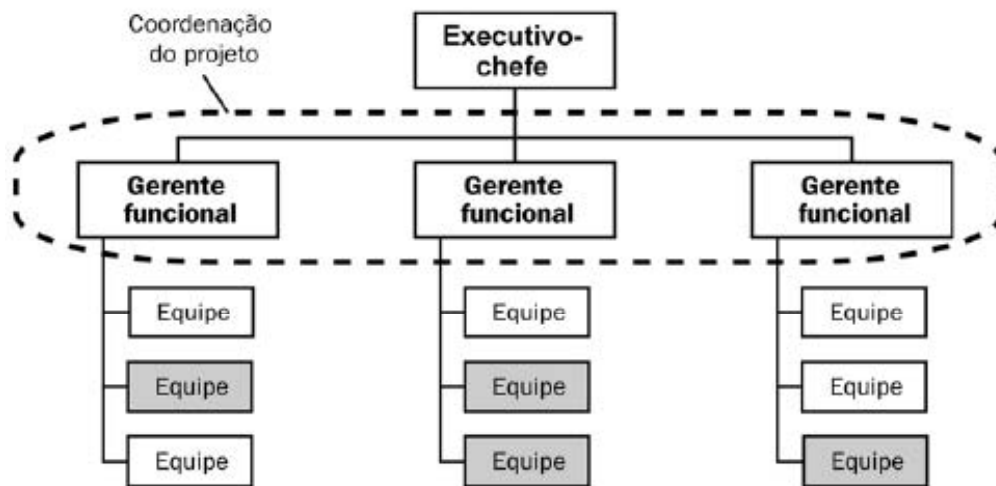


Figura 18 - Gerenciamento de projetos em uma organização funcional<sup>66</sup>

Na estrutura totalmente orientada para projetos a estrutura da organização é composta de times de projeto na empresa, conforme ilustra a figura 19 - Gerenciamento de projetos em uma organização por projetos. Ao término do projeto, o mesmo time pode ser alocado a um novo projeto ou os especialistas podem mudar de time, de acordo com a necessidade dos projetos que se iniciam.

A mudança de time também pode ocorrer em mudanças de fase do projeto. Por exemplo, na fase de planejamento pode haver uma maior necessidade de engenheiros enquanto que na fase de execução a necessidade de compradores pode ser maior; também pode ocorrer a necessidade de novos profissionais com características

---

<sup>66</sup> Conforme PMI (2004).

diferentes dos atualmente alocados ao time, quer seja por novas competências técnicas exigidas pelo projeto, quer seja por aspectos políticos, ou por outros fatores que possam vir a afetar os projetos gerenciados por uma dada organização.

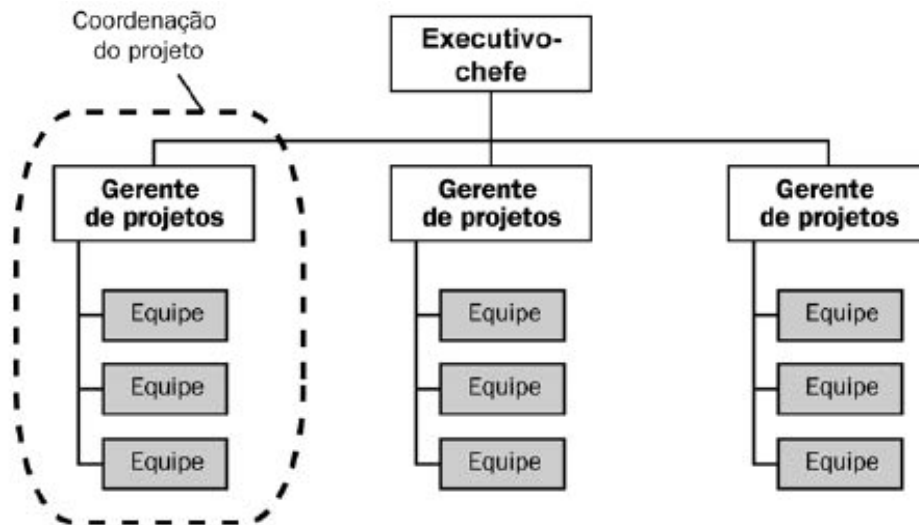


Figura 19 - Gerenciamento de projetos em uma organização por projetos<sup>67</sup>

É possível combinar características da organização funcional e da orientada para projetos através do uso de estruturas matriciais em que cada especialista se reporta a um gerente de projetos e a um gerente funcional, conforme ilustra a figura 20 - Gerenciamento de projetos em organização matricial.

A forma matricial é chamada de "forte" (do ponto de vista de orientação para projetos) se os gerentes de projeto se reportarem a uma hierarquia distinta da hierarquia funcional, "balanceada" se os gerentes de projeto se reportarem a gerentes ou diretores funcionais e "fraca" se a coordenação do projeto ficar a cargo de um especialista que não é gerente de projetos em tempo integral (segundo PMI (2004) e Dias (1999)).

Um resumo comparativo das características gerenciais nos diversos tipos de organizações descritos neste capítulo pode ser visualizado na tabela 8 - Estruturas com diferentes níveis de orientação por projetos.

---

<sup>67</sup> Conforme PMI (2004).



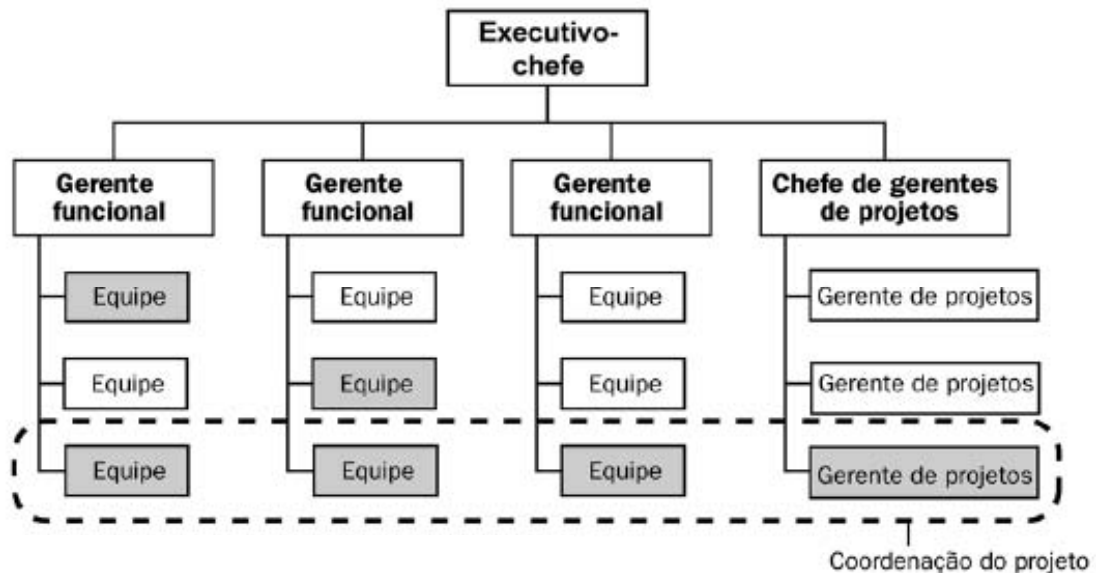


Figura 20 - Gerenciamento de projetos em organização matricial<sup>68</sup>

Tabela 8 - Estruturas com diferentes níveis de orientação por projetos<sup>69</sup>

Característica	Funcional	Matriz			Projetizada
		Fraca	Balanceda	Forte	
Autoridade do gerente de projetos	Pequena ou nenhuma	Limitada	Baixa a moderada	Moderada a alta	Alta a quase total
% do pessoal alocado totalmente a projetos	Praticamente 0	0 a 25%	15 a 60%	50 a 95%	85 a 100%
Dedicação do gerente de projetos a este papel	Em tempo parcial	Em tempo parcial	Em tempo integral	Em tempo integral	Em tempo integral
Título normalmente atribuído ao gerente de projetos	Coordenador ou Líder de projetos	Coordenador ou Líder de projetos	Gerente de projetos	Gerente de projetos ou de programas	Gerente de projetos ou de programas

<sup>68</sup> Conforme PMI (2004).

<sup>69</sup> Conforme PMI (2004).

Nota-se que a organização por projetos (ou projetizada) acentua a função de gerente de projetos, que tem alocação quase que exclusiva nesta função; já na organização funcional, a influência do gerente de projetos é muito limitada (a autoridade predominante é a do gerente funcional) e sua alocação a projetos é parcial (é compartilhada com outras funções); com isso a organização projetizada tem melhores condições de sucesso em seus projetos.

#### **4.2.3 Descrição do processo de gerenciamento de projetos**

Pode-se definir um processo como sendo uma série de ações conduzidas por pessoas objetivando um resultado específico. No sentido de facilitar a compreensão do processo de gerenciamento de projetos, o PMI adota um agrupamento dos diversos processos em cinco grupos, a saber:

- Processos de iniciação: reconhecem que um dado projeto ou fase pode ser iniciado e concluído;
- Processos de planejamento: estabelecem e mantêm um esquema de trabalho que objetiva atingir às necessidades que o projeto visa atender;
- Processos de execução: coordenam pessoas e demais recursos no sentido de realizar o planejamento estabelecido;
- Processos de monitoramento e controle: asseguram que os objetivos do projeto estão de acordo com o planejado, através de monitoramento e medições de progresso, tomando as medidas corretivas quando necessário;
- Processos de encerramento: garantem a aceitação formal do projeto ou da fase, finalizando-o.

Estes diversos grupos de processos estão ligados através dos produtos produzidos - o produto produzido por um grupo de processos será entrada para outro grupo. A figura 21 - Grupos de processos de gerenciamento de projetos - ilustra estas relações.

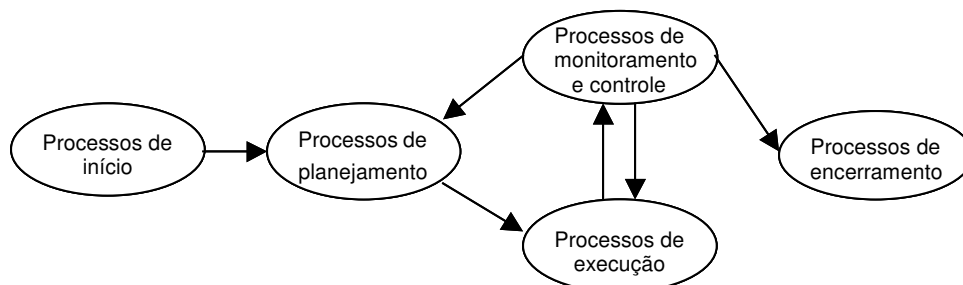


Figura 21 - Grupos de processos de gerenciamento de projetos <sup>70</sup>

Ao longo do ciclo de vida do projeto a influência de um determinado grupo de processos de gerenciamento de projetos é mais marcante, conforme ilustra a figura 22 - Gerenciamento ao longo do ciclo de vida do projeto.

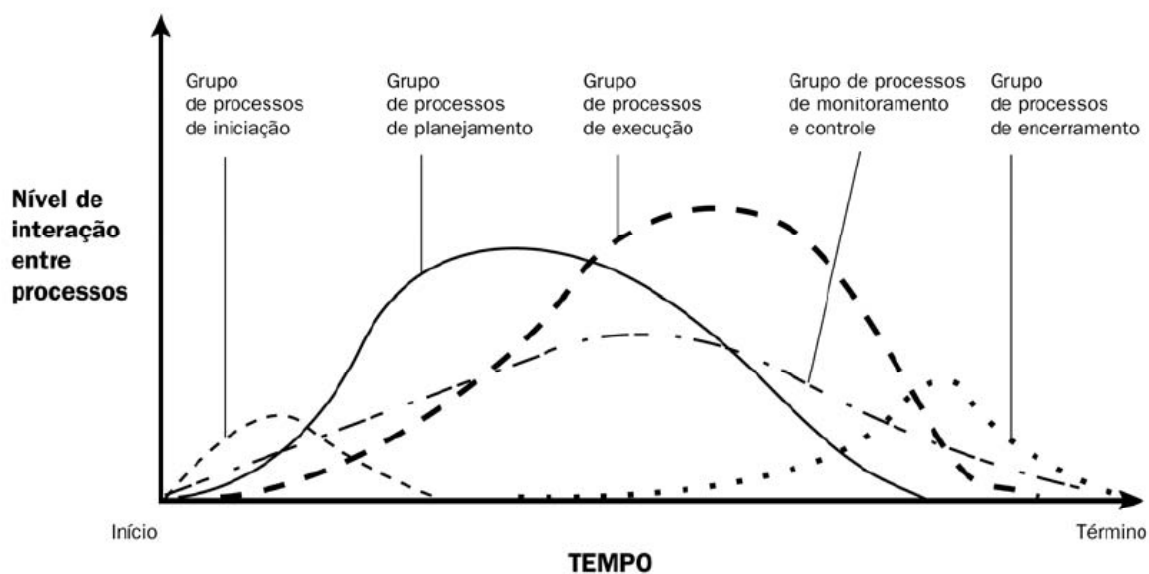


Figura 22 - Gerenciamento ao longo do ciclo de vida do projeto <sup>71</sup>

Cada processo de gerenciamento de projetos é descrito no PMBoK através de três componentes básicos:

- Entradas: Correspondem aos documentos ou itens documentáveis que servirão de base para a execução do processo;

<sup>70</sup> Conforme PMI (2004).

<sup>71</sup> Conforme PMI (2004).

- Técnicas e ferramentas: Mecanismos que podem ser aplicados às entradas para produzir os resultados;

Resultados e produtos: Correspondem aos documentos ou itens documentáveis gerados a partir da execução do processo. Estes três componentes possibilitam definir as ligações lógicas entre os processos (o resultado de um dos processos corresponde à entrada de outros processos) e os requisitos de cada processo. Na seqüência são apresentados os processos através dos grupos.

#### **4.2.3.1 Processos de Início**

De acordo com PMI (2004), os processos de início têm por finalidade garantir que todos os envolvidos no projeto (*stakeholders*) estejam cientes e comprometidos com o projeto a ser iniciado. É de vital importância e deve ser considerado como pré-requisito para os demais processos. São dois processos:

- 4.1<sup>72</sup> - Desenvolver o termo de abertura do projeto;
- 4.2 - Desenvolver a declaração do escopo preliminar do projeto.

#### **4.2.3.2 Processos de Planejamento**

De acordo com PMI (2004), os processos de planejamento são aqueles que têm a maior importância para o projeto, em função de envolverem aspectos e atividades que não podem ser desenvolvidos adiante. Existe um forte relacionamento deste grupo com os demais: têm como ponto de partida os resultados dos processos de início, são a base para os processos de execução do projeto e têm como retroalimentação os resultados dos processos de controle, onde pode haver a necessidade de novos planejamentos (ou replanejamentos).

---

<sup>72</sup>Vale notar que a numeração adotada para cada processo corresponde à numeração adotada por PMI (2004). Isto foi feito uma vez que é adotado desta forma pela comunidade internacional de gerenciamento de projetos e facilita a correlação com o *PMBok Guide*. Esta mesma notação é utilizada ao longo de toda a dissertação.

Ainda de acordo com PMI (2004), "planejamento não é uma ciência exata - diferentes times de projeto podem gerar diferentes planos de projeto para o mesmo projeto". Com isso, aspectos tais como experiência anterior, treinamento e qualificação dos participantes, entre outros, podem influir sobre o bom - e o mau - planejamento inicial.

Os processos de planejamento têm uma clara interdependência que faz com que eles sejam realizados essencialmente em uma mesma ordem na maioria dos projetos, podendo ser repetidos várias vezes durante o projeto; os principais processos de planejamento são:

- 4.3 - Desenvolver o plano de gerenciamento do projeto: consiste em agrupar todos os resultados dos processos anteriores de planejamento, organizá-los e armazená-los em um documento de forma coerente;
- 5.1 – Planejar escopo: corresponde ao desenvolvimento de um conjunto de regras de escopo do projeto que serão base para futuras decisões de projeto;
- 5.2 – Definir escopo: divisão do produto principal do projeto em subprodutos que sejam facilmente gerenciáveis;
- 5.3 – Criar Estrutura Analítica de Planejamento<sup>73</sup> (EAP): consiste no detalhamento do escopo, dos produtos e subprodutos através de uma estrutura hierárquica, visando maior facilidade para o gerenciamento do projeto; a figura 23 - EAP para um projeto de software – apresenta um exemplo de como poderia ser organizada a EAP para o desenvolvimento de uma versão de um software;
- 6.1 – Definir atividades: identificação das atividades específicas que devem ser realizadas para produzir os vários subprodutos do projeto;

---

<sup>73</sup> Também chamada de *Work Breakdown Structure* (WBS).

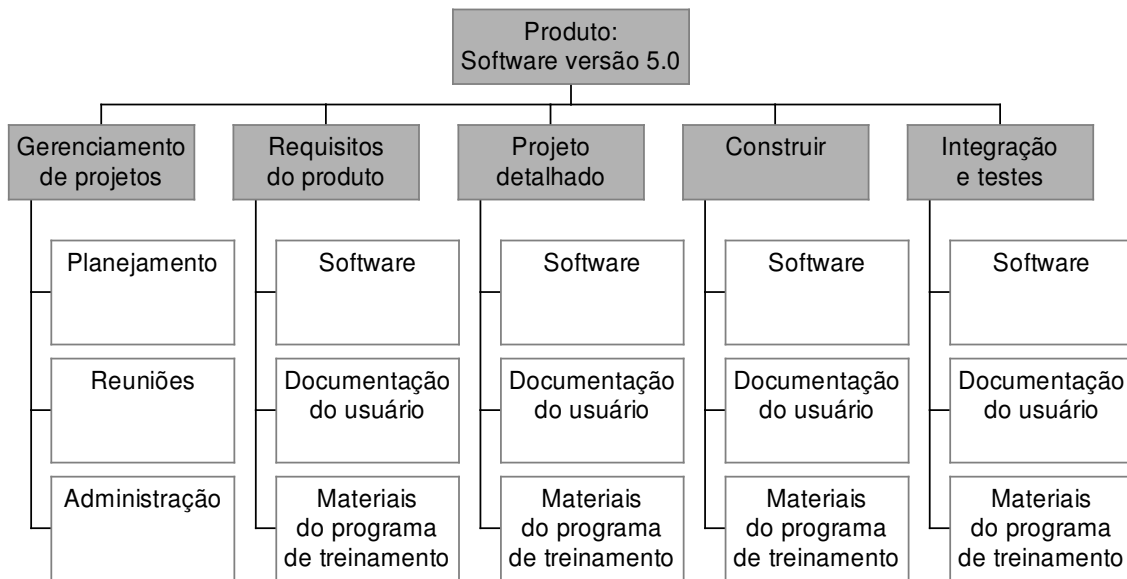


Figura 23 - EAP para um projeto de software <sup>74</sup>

- 6.2 – Seqüenciar atividades: Identificar e documentar as dependências<sup>75</sup> entre as atividades;
- 6.3 - Estimar recursos das atividades: consiste em estimar os recursos, a quantidade a ser utilizada e por quanto tempo para a realização de cada atividade em particular;
- 6.4 - Estimar duração das atividades: consiste em estimar o número de períodos de tempo que será necessário para a conclusão de cada atividade em particular;
- 6.5 – Desenvolver cronograma: consiste em determinar as datas de início e término para cada atividade do projeto;
- 7.1 - Estimar custos: desenvolver, de forma aproximada/estimada, a previsão de custos advindos dos recursos planejados;
- 7.2 - Orçar custos: alocar os custos estimados a cada item de trabalho;

<sup>74</sup> Conforme PMI (2004).

<sup>75</sup> Dependências entre atividades, também conhecidas como interdependência, correspondem às ligações de precedência entre as atividades, de forma a garantir um perfeito planejamento e controle de projetos. Tem origem na teoria *CPM (Critical Path Method)* - Método do Caminho Crítico - e é adotada pelos profissionais de planejamento como base para o controle de atividades. Maiores informações podem ser obtidas em Page Jones (1990), O'Connell (1994) e Brooks (1995).

- 8.1 - Planejar qualidade: identificar quais são os padrões de qualidade relevantes para o projeto e determinar como satisfazê-los;
- 9.1 - Planejar recursos humanos: identificar, documentar e determinar as regras de projeto, responsabilidades e relacionamento entre os participantes;
- 10.1 - Planejar comunicação: determinar que informações e que necessidades de comunicações são necessárias aos *stakeholders*: quem necessita de qual informação, quando estas serão necessárias e como elas serão enviadas para eles (meio físico);
- 11.1 – Planejar gerenciamento de riscos: consiste em definir como abordar, planejar e executar as atividades de gerenciamento de risco do projeto em questão;
- 11.2 - Identificar riscos: determinar quais são os riscos que podem vir a afetar o projeto e documentar as características de cada um;
- 11.3 - Analisar qualitativamente os riscos: priorizar os riscos para análise ou ação de contenção, com base na sua probabilidade de ocorrer e qual o impacto que ele pode vir a ter no projeto; avaliar quais são os riscos e quais podem vir a representar oportunidades;
- 11.4 - Analisar quantitativamente os riscos: analisar numericamente o efeito dos riscos nos objetivos do projeto;
- 11.5 - Planejar resposta aos riscos: definir um conjunto de passos para os riscos que foram transformados em oportunidades e um plano de ação para os demais (que continuaram como riscos);
- 12.1 - Planejar compras e aquisições: determinar o que, quando e como será contratado;
- 12.2 - Planejar contratações: documentar os requisitos de produtos, serviços e resultados, além de identificar possíveis fornecedores.

### **4.2.3.3 Processos de Execução**

Os processos de execução são caracterizados pelo desenvolvimento do projeto; são eles:

- 4.4 - Orientar e gerenciar a execução do projeto - processo principal que corresponde à execução do plano de projeto através da realização das atividades nele contidas;
- 8.2 - Realizar a garantia da qualidade: consiste em analisar o desempenho do desenvolvimento do projeto em períodos regulares de tempo e em atividades anteriormente planejadas de forma a garantir que o projeto está satisfazendo aos requisitos (padrões) de qualidade anteriormente estabelecidos;
- 9.2 - Contratar ou mobilizar a equipe do projeto - consiste em contratar ou alocar os componentes do time de projeto de forma a realizar o projeto;
- 9.3 - Desenvolver a equipe do projeto - consiste em desenvolver (capacitar, estimular) as capacidades dos componentes do time de projeto (individuais e coletivas) de forma a alcançar resultados de alto desempenho no projeto;
- 10.2 - Distribuir informações - consiste em tornar disponíveis, em tempo hábil e com qualidade, as informações na forma estipulada anteriormente para cada *stakeholder*;
- 12.3 - Solicitar respostas de fornecedores - consiste em obter, junto aos fornecedores, as cotações, as propostas, as ofertas e as requisições baseadas nos itens anteriormente planejados;
- 12.4 - Selecionar fornecedores - consiste em escolher uma proposta dentre as diversas apresentadas.

### **4.2.3.4 Processos de Monitoramento e Controle**

Os processos de monitoramento e controle estabelecem os procedimentos de acompanhamento necessários ao bom andamento do projeto; para tanto, existe a



necessidade de se verificar periodicamente o desempenho do projeto de forma a identificar possíveis variações nas diversas áreas (escopo, tempo, custo, qualidade etc.) e tomar as medidas cabíveis de forma a manter o bom andamento do projeto. Vale ressaltar que as medidas preventivas, identificadas antes mesmo de um desvio ocorrer, também são objeto deste grupo de processos. São processos deste grupo:

- 4.4 - Monitorar e controlar o trabalho do projeto – consiste em coletar, medir e disseminar informações sobre o desempenho, além de avaliar as medições e as tendências para efetuar melhorias no processo;
- 4.5 - Controle integrado de mudanças - corresponde à coordenação do tratamento das mudanças e modificações que porventura venham a ocorrer ao longo do projeto;
- 5.4 - Verificar escopo - consiste em aceitar formalmente os itens envolvidos com o escopo do projeto;
- 5.5 - Controlar escopo - corresponde ao controle das modificações de escopo;
- 6.6 - Controlar cronograma - consiste em controlar as mudanças feitas no cronograma do projeto;
- 7.3 - Controlar custos - consiste em controlar modificações no orçamento de custos do projeto;
- 8.3. Realizar o controle de qualidade - consiste em monitorar resultados específicos do projeto a fim de determinar se estão de acordo com os padrões de qualidade estabelecidos e identificar caminhos para eliminar possíveis problemas de desempenho;
- 9.4. Gerenciar a equipe do projeto - consiste em acompanhar o desempenho de membros da equipe, prestar contas aos envolvidos, resolver problemas e coordenar mudanças, visando a melhora contínua do desempenho do projeto;
- 10.3 - Reportar desempenho - consiste em coletar e disseminar informações de desempenho; incluem relatórios de progresso, medidas de desempenho e simulações/projeções;

- 10.4 – Gerenciar as partes interessadas - consiste em gerenciar a comunicação a fim de satisfazer os requisitos das partes interessadas no projeto e evitar – e resolver - problemas com elas;
- 11.6 - Monitorar e controlar os riscos - acompanhar os riscos identificados, monitorar os riscos residuais, identificar novos riscos, executar planos de respostas a riscos e avaliar sua eficiência durante todo o ciclo de vida do projeto;
- 12.5 - Administrar contratos – consiste no gerenciamento e na administração dos contratos com os fornecedores, tendo como base a proposta selecionada.

#### **4.2.3.5 Processos de Encerramento**

Os processos de encerramento têm por finalidade estabelecer um fim formal ao projeto desenvolvido. São dois os processos:

- 4.7 - Encerrar o projeto - corresponde à geração, coleta e disseminação das informações de forma a formalizar o encerramento de uma fase ou do projeto;
- 12.6 - Encerrar contratos - consiste em completar e encerrar os contratos com fornecedores, incluindo a solução de todas as pendências que porventura ainda possam existir.

#### **4.2.4 Áreas do gerenciamento de projetos**

A figura 24 - Processos de gerenciamento de projetos por área de conhecimento - ilustra os processos de gerenciamento de projeto nas diversas áreas de conhecimento. Vale notar que estes processos são os mesmos apresentados no tópico 4.2.3, Descrição do processo de gerenciamento de projetos, agora agrupados por área de conhecimento.

Na seqüência são detalhados os objetivos de cada área de conhecimento.

Á r e a s  d e  c o n h e c i m e n t o	<b>4. Gerenciamento de Integração</b>	<b>5. Gerenciamento de Escopo</b>	<b>6. Gerenciamento de Tempo</b>
	4.1 - Desenvolver termo de abertura 4.2 - Desenvolver a declaração de escopo preliminar 4.3 - Desenvolver o plano de gerenciamento 4.4 - Orientar e gerenciar a execução 4.5 - Monitorar e controlar o trabalho 4.6 - Controle integrado de mudanças 4.7 - Encerrar o projeto	5.1 - Planejar escopo 5.2 - Definir escopo 5.3 - Criar EAP 5.4 - Verificar escopo 5.5 - Controlar escopo	6.1 - Definir atividades 6.2 - Seqüenciar atividades 6.3 - Estimar recursos das atividades 6.4 - Estimar duração das atividades 6.5 - Desenvolver cronograma 6.6 - Controlar cronograma
	<b>7. Gerenciamento de Custos</b>	<b>8. Gerenciamento da Qualidade</b>	<b>9. Gerenciamento de Recursos humanos</b>
	7.1 - Estimar custos 7.2 - Orçar custos 7.3 - Controlar custos	8.1 - Planejar qualidade 8.2 - Realizar a garantia da qualidade 8.3 - Realizar o controlar da qualidade	9.1 - Planejar recursos humanos 9.2 - Contratar ou mobilizar a equipe do projeto 9.3 - Desenvolver a equipe 9.4 - Gerenciar a equipe
	<b>10. Gerenciamento de Comunicação</b>	<b>11. Gerenciamento de Riscos</b>	<b>12. Gerenciamento de Aquisições</b>
	10.1 - Planejar comunicação 10.2 - Distribuir informações 10.3 - Reportar desempenho 10.4 - Gerenciar as partes interessadas	11.1 - Planejar gerenciamento de riscos 11.2 - Identificar riscos 11.3 - Analisar qualitativamente os riscos 11.4 - Analisar quantitativamente os riscos 11.5 - Planejar resposta aos riscos 11.6 - Monitorar e controlar os riscos	12.1 - Planejar compras e aquisições 12.2 - Planejar contratações 12.3 - Solicitar respostas de fornecedores 12.4 - Selecionar fornecedores 12.5 - Administrar contratos 12.6 - Encerrar contratos

Figura 24 - Processos de gerenciamento de projetos por área de conhecimento <sup>76</sup>

#### 4.2.4.1 Gerenciamento de integração do projeto

Inclui os processos necessários a fim de garantir que os vários elementos componentes do projeto sejam adequadamente coordenados. Pode-se dizer que correspondem aos processos que correm paralelamente aos demais grupos de processos e visam garantir o correto funcionamento dos demais, a sincronização dos

<sup>76</sup> Conforme PMI (2004). A numeração de cada área de conhecimento corresponde ao respectivo capítulo do PMBoK Guide, ou seja, a área de conhecimento Gerenciamento de Integração corresponde ao capítulo 4 do PMBoK Guide, a área Gerenciamento de Escopo corresponde ao capítulo 5 e assim sucessivamente.

mesmos, a interface adequada, a atualização em tempo e o atendimento às necessidades dos diversos *stakeholders*.

#### **4.2.4.2 Gerenciamento do escopo do projeto**

Inclui os processos necessários a fim de garantir que o projeto inclua toda o trabalho necessário - e apenas o trabalho realmente necessário - para atender aos requisitos. Em uma primeira análise, consiste em definir e controlar tudo aquilo que deve, ou não deve, ser incluído no projeto.

#### **4.2.4.3 Gerenciamento de tempo do projeto**

Inclui os processos necessários a fim de garantir que o projeto em questão seja concluído dentro do tempo planejado. Consiste em uma série de processos necessários para detalhar as atividades e tarefas de forma a possibilitar o controle efetivo sobre o projeto.

#### **4.2.4.4 Gerenciamento de custos do projeto**

Inclui os processos necessários a fim de garantir que o projeto em questão seja concluído dentro dos padrões de custo (*budget*) aprovado. Primariamente são definidos tomando por base o custo dos recursos necessários para a conclusão das atividades do projeto.

#### **4.2.4.5 Gerenciamento da qualidade do projeto**

Inclui os processos necessários a fim de garantir que o projeto em questão satisfaça a todas as necessidades para as quais foi concebido. Segundo PMI (2004),

esta área de conhecimento mantém forte correlação com os preceitos das normas ISO e dos modelos de melhoria de processo, como o CMMI, devendo ter preocupações tanto com a qualidade do projeto quanto com a qualidade do produto resultado do projeto.

A preocupação é constante com a satisfação do cliente/usuário, através do entendimento, gerenciamento e garantia de que as necessidades e expectativas dos clientes sejam atendidas ou suplantadas (que requer que o projeto atenda às especificações - garantia de que o produto gerado pelo projeto realmente realize aquilo para que fora especificado - e que seja adequado ao uso - garantia de que o produto gerado pelo projeto satisfaça às necessidades reais dos clientes).

#### **4.2.4.6 Gerenciamento de recursos humanos do projeto**

Inclui os processos necessários a fim de garantir que o projeto faça uso da forma mais efetiva possível dos recursos humanos envolvidos com o projeto, incluindo todos os *stakeholders* (responsáveis, clientes, participante e demais envolvidos). Dentre os principais aspectos a serem abordados quando o tema é lidar com pessoas/indivíduos, tem-se negociação, comunicação, delegação, motivação, capacitação, mentoraç o, constru o do time de trabalho (relacionamento), lidar com conflitos, recrutamento, reten o de bons profissionais, rela o de trabalho, motiva o, sa de e seguran a, entre outros.

A natureza tempor ria do projeto faz com que o time do projeto tenha que definir quais s o os temas aplic veis e em que grau, diferentemente de uma empresa onde a tend ncia natural   de uma rela o mais duradoura; assim, temas como capacita o e resolu o de conflitos devem ser resolvidos de maneira  gil, uma vez que afetam diretamente o projeto em quest o.

#### **4.2.4.7 Gerenciamento das comunicações do projeto**

Inclui os processos necessários a fim de garantir que as informações acerca do projeto sejam geradas, coletadas, disseminadas, armazenadas e disponibilizadas no tempo adequado a todos os *stakeholders* do projeto, além de garantir que a informação seja entendida por todos que dela façam uso; envolve estudos sobre as melhores formas de divulgação (mídia), estilo da escrita, técnicas de apresentação e de reunião.

#### **4.2.4.8 Gerenciamento de riscos do projeto**

Inclui os processos necessários a fim de identificar, analisar e responder aos riscos do projeto, incluindo maximizar os resultados de eventos positivos e minimizar as conseqüências de eventos adversos.

#### **4.2.4.9 Gerenciamento de aquisições do projeto**

Inclui os processos necessários a fim de adquirir bens e serviços (chamados genericamente de produtos) a partir de empresas externas à organização que executa o projeto. O foco a ser dado nesses processos é sempre o da perspectiva do comprador em uma relação comprador-vendedor, incluindo aí aspectos contratuais, aceitação dos produtos e demais itens relacionados a essas operações.

## Capítulo 5

### Análise Comparativa

Grande parte da análise comparativa já foi realizada ao longo da descrição de cada modelo (qualidade genérico, específico para software ou de gerenciamento de projetos), abordando os pontos positivos e negativos. Este capítulo procura sumarizar as constatações efetuadas, ressaltando os principais aspectos envolvidos.

#### **Modelos de gerenciamento de projetos**

Conforme apresentado nos capítulos anteriores, existem várias metodologias e *frameworks* relativos aos processos de gerenciamento de projetos; apesar de tratarem do mesmo tema, um amplo consenso ainda é uma meta distante. Para ilustrar, a norma ISO 10006 (ISO, 2003a) é bastante criticada por Duncan, um dos membros mais atuantes do PMI; em seu artigo (Duncan, 1999), critica as empresas que pretendem adotar a ISO 10006 como seu padrão de gerenciamento de projetos por entender que esta norma omite diversos aspectos relevantes já abordados pelo PMBoK (PMI, 2004). Stanleigh (2005) apresenta um artigo bastante interessante onde afirma que as normas podem ser utilizadas de forma complementar, trazendo benefícios quando utilizadas em conjunto.

Uma referência interessante sobre o papel do gerenciamento de projetos pode ser obtida em Dinsmore (1998), onde o foco é demonstrar como o gerenciamento de projetos pode ser aplicado a todas as áreas de negócio, inclusive no maior projeto de cada indivíduo: sua própria existência; Dinsmore sugere que as práticas de gerenciamento de projeto podem (e devem) ser aplicadas à vida das pessoas, a fim de que as metas pessoais possam vir a ser atingidas. Seu discurso é baseado no PMBoK.

Martins (2005) parte do modelo de gerenciamento de projetos proposto pelo PMBoK e fez uma interessante abordagem para a utilização em desenvolvimentos de software orientados a objetos, com forte ligação com o padrão UML.

## **Padrões de qualidade**

No âmbito dos modelos de qualidade, tem-se que diversos autores procuraram relacionar o CMM/CMMI e a ISO 9000, considerando que a ISO 9000 pode corresponder a uma qualificação de CMM nível 3 (Paulk, 1994), podendo existir inclusive casos de empresas que ainda estejam no nível 1 do CMM (Pessoa, 1997); em outras palavras, estes autores consideram que o nível de exigência do CMM/CMMI é bem mais elevado que os preceitos da ISO 9000. Várias empresas que desenvolvem software inicialmente adotam a ISO 9001 e posteriormente aderem ao modelo CMM. Catho (2006) comenta que a ISO 9001 e o CMMI nível 3 possuem uma correspondência de 90% em seus requisitos.

Para Pessoa (1997), a ISO 9000 foca o sistema da qualidade de uma empresa na relação cliente / fornecedor enquanto que o modelo CMM/CMMI foca um sistema da qualidade para o desenvolvimento de software; a ISO 9000 tem um sistema de certificação de terceira parte, ou seja, órgãos credenciados que realizam auditorias em intervalos regulares de tempo para examinar o sistema da qualidade e garantir que o mesmo está funcionando adequadamente enquanto que o CMM/CMMI é avaliado por auditores credenciados pelo SEI, órgão privado que administra o modelo; o grau de abstração do modelo ISO é mais alto que o CMM/CMMI e se aplica a qualquer tipo de empresa enquanto que o modelo CMM/CMMI é muito mais profundo no tocante aos quesitos de desenvolvimento de software e, portanto, mais exigente. Complementa também que

os modelos ISO e CMM são compatíveis, sendo perfeitamente viável uma empresa que possui ISO 9000, implantar o modelo CMM, assim como também é perfeitamente possível uma empresa com ISO 9000 estar ainda no nível 1 do CMM pois são modelos com diferentes requisitos.

A área de gerenciamento da qualidade descrita pelo PMBoK faz referência aos preceitos de qualidade previstos pelas séries ISO 9000 e ISO 10000, estudadas em capítulos anteriores. Uma distinção se faz necessária no tocante à qualidade: ela é diferente de graduação, que pode ser entendida "por uma categorização ou



classificação para entidades que tenham a mesma funcionalidade mas diferentes requisitos de qualidade" (PMI, 2004).

Essa distinção é importante uma vez que um projeto com baixa qualidade é sempre um problema, enquanto que um projeto com baixa graduação nem sempre o é - por exemplo, um software com pequeno número de funcionalidade pode ser considerado de baixa graduação e isto não significa necessariamente um problema (ao contrário, pode ter sido um requisito), enquanto que um software com baixa qualidade (quer seja por problemas de performance, erros, comprometimento quanto a sua entrega) certamente indica problemas. Vale ressaltar que o nível de graduação do software deverá ser determinado em conjunto, envolvendo todos os *stakeholders* (equipe do projeto, clientes e fornecedores, entre outros).

O PMBoK aponta outro aspecto fundamental em relação ao gerenciamento da qualidade, que é totalmente aplicável ao software: a diferenciação de prevenção e inspeção, aparentemente semelhantes (não deixar passar erros), porém diferem drasticamente em suas formas: enquanto a prevenção consiste em um processo contínuo de melhoria (atualmente muito difundido pelos órgãos de qualidade), cujo foco é não ocorrerem erros, a inspeção é muito mais branda e consiste nos famosos testes apenas ao final do desenvolvimento a fim de evitar que o usuário final detecte o erro antes que os profissionais de informática - em outras palavras, perpetuando o erro. A busca constante no caso de software deve ser produzir software com qualidade, sem erros e correto da primeira vez<sup>77</sup>.

## **Escopo**

A área de escopo do projeto de desenvolvimento de software é normalmente negligenciada no tocante ao seu gerenciamento e controle. O levantamento e a definição do escopo normalmente são feitos utilizando técnicas de levantamento de

---

<sup>77</sup> Barbosa (2002) apresenta os principais conceitos relativos às estratégias de teste de software.

dados junto aos usuários e, após isto, utilizando modelos estruturados<sup>78</sup> (DFD), orientados a objeto (Casos de Uso), texto em linguagem natural (por exemplo, "português estruturado") ou outra forma de documentação, porém nem sempre são feitas as validações junto aos usuários e normalmente não existe um processo formal de atualização, autorizações de mudanças e controle deste escopo.

As disciplinas de gerenciamento contribuem com a formalização desses processos de controle e atualização, instituindo procedimentos e responsabilidades pelas ocorrências. Diversos autores incluem estes processos como sendo dos mais relevantes para o sucesso do desenvolvimento de sistemas. Para citar alguns, Connell (1999) procura se concentrar nos aspectos envolvidos com a habilidade de se dividir os ciclos de desenvolvimento de acordo com o escopo do projeto, entre outros fatores, utilizando uma abordagem evolutiva no desenvolvimento; Costa e Miranda (1999) também propõem que processos de controle de escopo sejam utilizados a fim de garantir coesão e profissionalismo na área de informática.

## **Prazos**

Tradicionalmente quando o termo gerenciamento é utilizado, a dimensão prazo é normalmente associada. Todas as abordagens metodológicas apresentadas são unânimes em atrelar o estabelecimento dos prazos a um efetivo entendimento do escopo, seu respectivo detalhamento (utilizando técnicas de decomposição e WBS), identificação dos produtos a serem gerados pelo projeto (*deliverables*) para então se chegar à identificação das atividades, as estimativas de prazos e a lógica entre estas atividades, sendo então gerado o cronograma do projeto.

Tão importante quanto a elaboração de um plano de projeto (cronograma) é o acompanhamento de sua execução a fim de detectar (e evitar) desvios e, caso não seja possível, ativar ações corretivas para que o projeto volte ao cronograma planejado. Replanejamento também deve ser uma preocupação constante, uma vez que o projeto pode sofrer modificações em função de um maior detalhamento dos aspectos

---

<sup>78</sup> Martin e McCluer (1991) apresentam uma descrição completa das técnicas estruturadas, bem como das ferramentas que podem suportar a sua aplicação.

envolvidos (em especial escopo); este replanejamento deve ser devidamente gerenciado e seu impacto, nas diversas áreas de gerenciamento de projeto, deve ser analisado e implementado.

Como referências para aprofundamento no tema de gerenciamento de prazos e seu impacto nas demais áreas de gerenciamento recomenda-se Launi (1999), que se dedica às preocupações para a criação de um planejamento de prazos (cronograma) efetivo; Martin e Tate (1998, 1999), que abordam a criação de cronogramas efetivos e na medida certa, discutindo aspectos como excesso de controle x escopo do trabalho a ser executado; Mello Filho (1999b), que aborda o processo de elaboração de cronogramas utilizando a abordagem evolutiva/incremental, Fabian-Isaacs e Robinson (1999), que enfocam o gerenciamento de prazos e seu impacto dentro do projeto e Vargas (2003), que apresenta a relação do gerenciamento de tempo/prazos com as demais áreas de gerenciamento de projetos, gerando um plano integrado de gerenciamento de projetos.

## **Custos**

Um dos aspectos mais importantes na hora da contratação do desenvolvimento de um software é o aspecto financeiro: quanto irá custar realmente o trabalho e qual o retorno (financeiro) oriundo da implantação do software. Ainda que todos os cuidados referentes a uma boa orçamentação inicial<sup>79</sup>, baseada nos requisitos do sistema (escopo), recursos necessários (tanto humanos quanto materiais), o risco de desvios de custo é grande, fazendo com que um processo de monitoração contínuo seja necessário.

As técnicas de acompanhamento de custos, baseadas em indicadores de desempenho atrelados ao trabalho produzido comparados com o custo esperado (o chamado *earned value*), introduzem ferramentas poderosas de detecção de desvios de custos, antecipando problemas. Para exemplificar, se o custo real em uma dada data está menor que o previsto, não necessariamente significa que algo bom (economia)

---

<sup>79</sup> Pillai e Nair (1997) apresentam um artigo interessante sobre os aspectos envolvidos com a estimativa de custo em projetos de software.

está ocorrendo - ao contrário, pode ser que o custo menor que o previsto indique que os esforços necessários não estão de acordo com o planejado e que um atraso nos prazos do projeto possa acontecer. Com o uso das técnicas mencionadas estas e outras questões podem ser avaliadas e medidas preventivas/corretivas podem ser tomadas.

## **Riscos**

O reconhecimento da necessidade de abordagem aos riscos presentes na maior parte dos modelos é de extrema importância. Um processo formal para identificação, quantificação, acompanhamento e controle destes riscos é vital para que o projeto tenha sucesso. Alguns autores, como Ambler (1998), argumentam que técnicas de orientação a objetos aplicadas ao desenvolvimento de software podem fazer com que alguns riscos de origem técnica, de qualidade e de prazos sejam eliminados através da construção e reutilização de componentes de software.

Os projetos de desenvolvimento de software estão intimamente ligados a riscos, motivados sobretudo pela natureza do software, a dificuldade em encará-lo como produto, a pressão normalmente feita aos desenvolvedores de software e a falta de foco nos aspectos gerenciais, entre outros fatores. Este conjunto de fatores faz com que o desenvolvimento de um software seja, por si só, um fator de risco a ser considerado.

Somente com o estabelecimento do processo formal de gerenciamento de riscos estes problemas podem ser solucionados. É preciso conscientizar sobre a necessidade de se identificar os riscos antes do projeto ser efetivamente iniciado, definir a melhor estratégia - muitas vezes o risco pode ser compartilhado com os clientes, trazendo benefícios a todos os envolvidos com o projeto - para abordá-lo. Vale notar também que os riscos podem surgir (e desaparecer) ao longo do projeto, sendo necessário um acompanhamento constante e atualização dos riscos anteriormente identificados.

Outro fator decorrente da análise de riscos consiste em detectar oportunidades. Por vezes, ao iniciar o desenvolvimento de um software o time do projeto percebe uma oportunidade, quer seja de novos negócios com o cliente em questão, quer seja de ampliação do escopo deste projeto, quer seja de expansão dos clientes atendidos pelo software a ser desenvolvido. É importante a institucionalização de processo formal para

tratar essas oportunidades identificadas, ainda que a decisão seja por não perseguir essas oportunidades. Várias empresas tiveram sucesso a partir desta abordagem, assim como outras não conseguiram se manter em função de não terem feito essa análise (ou identificação) de forma adequada.

### **Aspectos humanos**

Outra contribuição dos modelos estudados se refere ao relacionamento dos projetos com aspectos humanos, em especial algumas necessidades existentes, como a capacitação dos profissionais em disciplinas de gerenciamento de projetos<sup>80</sup> (além das disciplinas técnicas relativas a software), a necessidade do ser humano em ser mantido informado da real situação de um projeto (tratado no tópico gerenciamento de comunicações do PMBoK Guide, com procedimentos formais para garantir que as comunicações sejam completas, efetivas e claras, tanto dos pontos positivos quanto dos problemas relativos ao projeto), a habilidade de se trabalhar em times, a habilidade de resolução de conflitos, inevitável em projetos de desenvolvimento de software onde algumas necessidades e soluções para problemas são antagônicas.

Outro aspecto importante na área de desenvolvimento de software está relacionado ao perfil dos profissionais. A grande maioria tem vocação técnica, se preocupando apenas com aspectos envolvidos com a construção técnica do produto, não reservando tempo para as atividades gerenciais envolvidas neste processo. A recomendação de vários autores para a conscientização dos profissionais de que estes fatores são de vital importância para o sucesso dos projetos<sup>81</sup>, porém a implantação dessas recomendações é algo que caminha vagarosamente, uma vez que aspectos culturais estão amplamente disseminados. Nos cursos de formação de profissionais que lidam com o desenvolvimento de software nota-se que a grande maioria dos alunos

---

<sup>80</sup> Um excelente material pode ser obtido em ESI (1997), que relaciona os aspectos humanos no desenvolvimento de sistemas de informação. Vale comentar que a universidade George Washington é referência mundial quando o tema é gerenciamento de projetos, em especial quando envolve a capacitação de pessoas.

<sup>81</sup> Hayes (1999) fala em quebra de paradigmas para melhoria do processo de desenvolvimento; James (1997), explora as razões comuns para fracasso dos projetos de desenvolvimento de software.

(quer seja aqueles advindos de escolas técnicas em processamento de dados, quer seja aqueles que já atuam na área de informática, quer seja aqueles que estão iniciando na informática) tem necessidade e interesse especial pelos aspectos técnicos envolvidos, pelas linguagens de programação e ferramentas de desenvolvimento, porém, quando o assunto é ligado aos aspectos gerenciais envolvidos com o projeto, existe grande dificuldade. A cultura do "por a mão na massa" e "ligar o micro e começar a desenvolver o sistema" é muito forte, consistindo em uma barreira importante a ser ultrapassada na busca do profissionalismo e da qualidade.

### **Suprimentos e contratações**

Outro aspecto considerado pelos modelos apresentados neste trabalho é a possibilidade de contratação de serviços e produtos a partir de fornecedores externos à organização, prática cada vez mais comum no desenvolvimento de software. É importante ressaltar as preocupações advindas desta escolha, desde os aspectos de seleção de fornecedores, confiabilidade do fornecedor escolhido, regime de contratação desse fornecedor, cláusulas contratuais (jurídicas e técnicas), administração do contrato com este fornecedor, critérios de inspeção dos produtos da contratação, relacionamento entre o fornecedor e o time de projeto; com isso, a habilidade de negociação e trabalho em times deve ser fortalecida e as preocupações gerenciais redobradas em função desse novo panorama.

O porte atual dos projetos de desenvolvimento de software, associado às especialidades e especificidades existentes, faz com que dificilmente uma organização possua (e que estejam disponíveis) todos os recursos necessários; sendo o processo de contratação de pessoal especializado normalmente demorado, via de regra não atende às necessidades do projeto, agravado pelo custo elevado (uma vez que envolve pessoal especializado possivelmente valorizado pelo mercado) e pelo fato do projeto ter uma duração finita, findo o qual cessa a necessidade do recurso e passa a não ser interessante para a empresa a manutenção do mesmo - nesses casos é indicada a contratação de serviços junto a fornecedores. A relação pode ser do tipo parceria, que estabelece uma relação de cooperação mútua entre os envolvidos, pode ser específica

para um trabalho, pode ser alocação de mão de obra especializada para exercer alguma atividade ou ainda pode ser algo do estilo projeto global (chamado de *turn key*), onde um dado fornecedor se responsabiliza por toda a obra ou desenvolvimento, subcontratando outros fornecedores, caso necessário.

Com a contratação de serviços algumas preocupações adicionais devem ser levadas em consideração, como, por exemplo, porte do fornecedor, regime de contratação dos recursos por parte do fornecedor (isto tem grande impacto na legislação trabalhista brasileira, uma vez que pequenas empresas, via de regra, não contratam como empregados seus funcionários, contratando via empresa individual - a chamada "quarteirização" - com riscos trabalhistas tanto para a empresa que esta contratando estes indivíduos quanto para a empresa que originalmente contratou o serviço). Cláusulas contratuais adequadamente revisadas nos seus aspectos jurídicos, bem como a exigência das documentações referentes ao fornecedor contratado, são de vital importância para a contratação de serviços terceirizados; não é raro ocorrerem problemas trabalhistas em função de um dado projeto quando este já se encerrou, gerando ônus para os envolvidos.

### **Comunicações**

Aspecto vital de todo projeto, infelizmente nem sempre feito de forma satisfatória. Manter todos os *stakeholders* de um projeto com informações suficientes a respeito do andamento, com frequência adequada e de forma atualizada, é um desafio constante presente nos desenvolvimentos, desafio este agravado nas etapas mais técnicas do processo (por exemplo, durante a codificação) onde existem problemas de se medir efetivamente a situação do projeto.

Outro ponto importante é atentar para o fato de que as necessidades de informação são diferentes entre os *stakeholders*: enquanto o time do projeto provavelmente necessitará de informações detalhadas sobre o projeto, o gerente do projeto provavelmente necessitará de outro tipo de informação (normalmente tudo aquilo que está diferente do planejado, a fim de ser colocado em prática o gerenciamento por exceção, ou seja, se concentrar nos pontos onde pode, ou irão,

acontecer os problemas), o cliente necessitará de outro nível de detalhe (mais voltada ao andamento do projeto). No início do desenvolvimento deve ser feita esta distinção e os mecanismos para a geração destes dados devem ser providos e considerados como parte integrante do próprio desenvolvimento.

### **Integração das várias áreas**

Após a análise individual das áreas de gerenciamento, com seus impactos pormenorizados, cabe ao projeto uma avaliação conjunta de todas estas áreas; isto se aplica desde um planejamento do projeto integrado, tomando por base os requisitos individuais de cada área, até o procedimento global de controle de modificações, quer seja de escopo, prazo, custo e seu impacto por todo o projeto.

Deve ser constantemente revisada a fim de garantir um efetivo acompanhamento do projeto, sendo que seu uso e implantação são facilitados com a adoção de metodologias por parte da empresa responsável pelo projeto.

O grande enfoque deste grupo de atividades, que correspondem a atividades ao longo de todo o projeto, é padronizar os mecanismos a serem utilizados visando garantir o bom andamento do projeto. Aspectos como plano do projeto integrado e controle de modificações unificado são relevantes e primordiais para a efetiva conclusão do projeto com sucesso. Whiten (1995) e Roberts Jr. et. al. (1998) abordam os fatores que impactam a implantação de uma metodologia de desenvolvimento de sistemas, onde a integração das diversas atividades e etapas é parte fundamental. Vargas (2003) detalha como construir o plano integrado de projetos e os benefícios de sua utilização.



## Capítulo 6

### Estudo de Caso

O estudo de caso foi realizado na empresa IgnisCom Ltda., uma empresa desenvolvedora de soluções que envolvem o desenvolvimento de software, em um projeto real – o projeto Conexão do Saber; o propósito principal foi verificar como as recomendações apresentadas ao longo do presente trabalho poderiam efetivamente auxiliar no gerenciamento de projetos de desenvolvimento de software.

#### **A empresa IgnisCom**

A empresa IgnisCom (Ignis Comunicações S/C Ltda.) foi criada, segundo IgnisCom (2003), para pesquisar e desenvolver soluções para a área telecomunicações, em especial no desenvolvimento de softwares para análise e projeto de redes de telecomunicações; surgiu com o objetivo de desenvolver e comercializar novos produtos a partir da experiência obtida em pesquisas realizadas na UNICAMP com o apoio da FAPESP. Foi fundada em fevereiro de 2002, iniciando sua atuação como uma empresa integrante da Incubadora de Empresas de Base Tecnológica da UNICAMP (Incamp).

Tem por missão “pesquisar, desenvolver e comercializar produtos, serviços e soluções inovadoras em telecomunicações, tecnologia da informação e educação, que permitam aos nossos clientes aumentar a eficiência, produtividade e qualidade de suas atividades” IgnisCom (2003) e por visão “... seja reconhecida mundialmente como uma empresa inovadora, competente e eticamente correta, capaz de agregar novos valores, gerando vantagens competitivas para seus clientes e parceiros” IgnisCom (2003). A equipe de colaboradores é formada por profissionais altamente capacitados, mestres e doutores, com sólida competência técnica nas áreas de atuação da empresa.

Entre os serviços e soluções desenvolvidos pela empresa, destacam-se:

- pesquisa e desenvolvimento de software, em especial para gerenciamento, monitoramento, emulação, projeto e análise de redes de telecomunicações, bem como para predição e controle de tráfego;
- projeto e consultoria nas áreas de telecomunicações, de tecnologia da informação e de educação a distância, além de projeto, desenvolvimento e implementação de soluções (integração de redes e software) em telecomunicações e tecnologia da informação, com particular ênfase em Redes Corporativas e Redes Metropolitanas (Infovias Municipais);
- modernização da administração pública, com o desenvolvimento de soluções para o aumento da eficiência da administração pública, principalmente de municípios, com a construção de soluções em redes seguras para: modernizar a gestão de materiais e medicamentos; administração de centros de saúde e hospitais, cadastro de pacientes, cartão SUS, administração financeira, administração escolar e desenvolvimento de soluções para educação a distância, entre outros;
- educação e treinamento, com o desenvolvimento de cursos abertos ao público em geral, cursos fechados (*in-company*) e cursos de especialização, em parceria com instituições de ensino do Brasil e América Latina.

### **O projeto Conexão do Saber**

O projeto Conexão do Saber, escolhido para o presente estudo de caso, proporciona, segundo IgnisCom (2005), aos clientes um ambiente de comunicação coletiva voltado para a Educação, com foco especial no aluno; tem por finalidade abordar a inclusão digital de forma simples, ágil e objetiva, aplicando metodologias desenvolvidas e consolidadas através do programa *Partnership in Global Learning* (PGL)<sup>82</sup>; é uma solução completa para a área de educação, podendo ser aplicado no

---

<sup>82</sup> Projeto desenvolvido em parceria por Universidades (UNICAMP, FGV, PUC-Rio, Tecnológico de Monterrey e Universidade da Flórida) e Corporações (IBM, Lucent e Avaya); maiores informações em <<http://pgl.ufl.edu>>.

ensino infantil, fundamental e médio, permitindo que as comunidades que a integram, estados e municípios, possam compartilhar um ambiente digital de educação.

É composto por uma séria de sistemas e soluções, a saber:

- Sistema Infovia da Educação, que consiste em uma rede de comunicações que interliga as redes locais das escolas e outros órgãos ligados à área de educação;
- Sistema Intranet da Educação, que consiste em várias redes privadas virtuais;
- Sistema Quiosque da Educação, que possibilita aos pais e alunos acessarem as informações que a secretaria da educação deseja disponibilizar, como horários de aulas, disciplinas, notas, conceitos e sugestões, entre outros;
- Salas de Aprendizado, que possibilitam aos alunos conectados à Internet o acesso a exercícios e estudos dirigidos a partir de módulos;
- Módulos, que correspondem aos conteúdos disponibilizados no sistema para que os alunos possam realizar diversas atividades; já existem centenas de módulos desenvolvidos, além da possibilidade de desenvolvimento de novos módulos pelos professores;
- Capacitação de professores, que possibilita aos professores entrarem em contato com as possibilidades do mundo digital e do aprendizado a distância;
- Laboratórios Virtuais, que possibilitam aos alunos a realização de experiências em laboratório semelhantes aos laboratórios reais, muitas vezes não disponíveis pelo seu custo e complexidade;
- Portal da Educação, que possibilita o acesso, via Internet, para todos os sistemas do Conexão do Saber;
- Biblioteca Virtual, que possibilita a consulta e produção de documentos digitais por parte de professores e alunos;
- Estúdio Digital, que permite a professores e alunos a criação de programas que podem ser utilizados dentro da própria sala de aula, como material de estudo, ou então compartilhados, via Internet, com outras comunidades.

O projeto é personalizado para cada nova demanda. O estudo de caso se concentrou no projeto para a secretaria da educação da cidade de São José do Rio Preto.

## **ESTUDO DE CASO**

O estudo de caso foi realizado desde a conclusão da negociação do projeto com a secretaria de educação da cidade de São José do Rio Preto até meados da realização do projeto, envolvendo as etapas de:

- início do projeto propriamente dito;
- planejamento do projeto, com definição do escopo, definição da equipe, definição do cronograma e definição da forma de acompanhamento do projeto;
- acompanhamento da execução do projeto.

A interlocução durante o estudo de caso foi realizada com a equipe alocada ao projeto, em torno de 15 profissionais, e, principalmente, com o sócio proprietário da empresa IgnisCom, engenheiro Mestre Maurício Luis Bortoli, gerente do projeto em questão.

O processo de planejamento e acompanhamento dos projetos desenvolvidos pela empresa era feito através de planilhas de cálculo, utilizando a ferramenta MS-Excel como base para esses controles. Os conceitos de gestão de projetos estavam em processo inicial de implantação, com estudo das metodologias e aquisição de ferramentas de suporte à gerência de projetos, como o MS-Project e o WBS Chart Pro.

Durante a fase de planejamento foi introduzido o conceito de WBS. Após várias reuniões de trabalho, com apresentação dos conceitos trabalhados ao longo desta dissertação, foram detalhados o escopo do projeto e a respectiva WBS, conforme ilustra a figura 25 - WBS do Projeto Conexão do Saber São José do Rio Preto, trazendo maior clareza e visibilidade para os produtos esperados (*deliverables*) do projeto em questão.

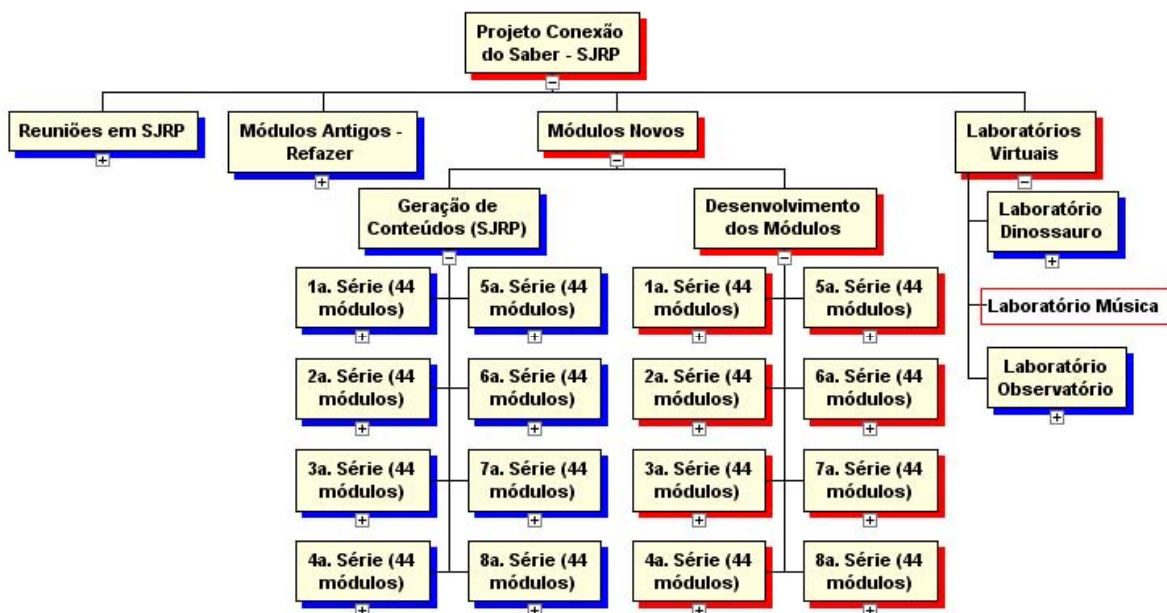


Figura 25 - WBS do Projeto Conexão do Saber São José do Rio Preto

Para a execução do projeto foram designados alguns responsáveis técnicos, visando um melhor acompanhamento do projeto. Esses profissionais ficaram responsáveis pelos grupos:

- desenvolvimento dos módulos;
- laboratório virtual Dinossauro;
- laboratório virtual Observatório;
- laboratório virtual Música;
- treinamentos.

Cada responsável técnico ficou a cargo de coordenar tecnicamente seu módulo e os profissionais envolvidos, reportando periodicamente ao gerente de projeto o andamento, as dificuldades e o progresso de seu módulo.

A partir da WBS e da definição dos responsáveis técnicos foram detalhadas, em conjunto com a equipe do projeto, as atividades necessárias para a conclusão de cada produto previsto. Foram consultados todos os envolvidos, as bases históricas (os projetos já desenvolvidos anteriormente) e gerado um cronograma contemplando as atividades, os recursos necessários e os prazos. A figura 26 - Cronograma do Projeto

Conexão do Saber São José do Rio Preto - apresenta o cronograma sumarizado do projeto.

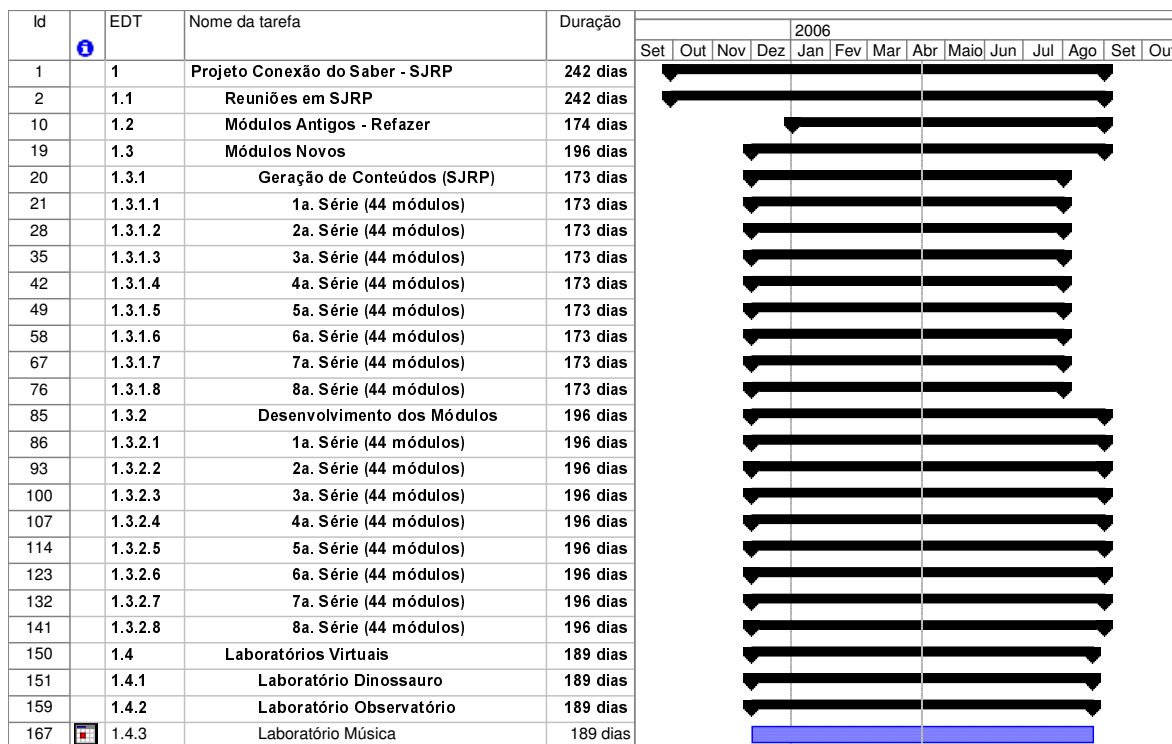


Figura 26 - Cronograma do Projeto Conexão do Saber São José do Rio Preto

Todos os envolvidos com o projeto (cliente e colaboradores) foram chamados para validar o cronograma. A gerência do projeto, após a validação, gerou a primeira linha base do projeto, que propiciará comparações sobre o real andamento do projeto frente ao inicialmente acordado. Foi estabelecido que o acompanhamento do projeto será realizado semanalmente, com nova distribuição do cronograma do projeto a todos os envolvidos, propiciando uma comunicação efetiva sobre o andamento do projeto, além de tornar possível a adoção de medidas preventivas (ou ainda corretivas) para garantir o bom andamento do projeto.

De acordo com o gerente do projeto, o principal benefício advindo da adoção das mediadas descritas neste estudo de caso foi a maior visibilidade a todos os envolvidos com o projeto. O sentimento de responsabilidade aumentou em todos, colaboradores e clientes, ficando transparente e propiciando identificação de pontos de melhorias. Espera-se que para novos projetos as medidas descritas neste estudo de caso possam

ser ampliadas e consolidadas, gerando uma cultura de gerenciamento de projetos e agregando novas competências em todos os envolvidos.

## Capítulo 7

### Conclusões e Perspectivas Futuras

#### 7.1 Conclusões

Pode-se concluir, pelo estudo dos modelos de qualidade e de gerenciamento de projetos, pela análise comparativa e pelo estudo de caso, que diversos aspectos podem ser introduzidos às práticas de gerenciamento de processos de desenvolvimento de software. O PMBoK Guide, através da visualização dos processos de gerenciamento de projetos agrupados em áreas de conhecimento e grupos – possibilita que o gerenciamento de projetos seja melhor definido e documentado, propiciando uma avaliação precisa dos impactos em todas as áreas e etapas do projeto, maximizando a possibilidade de sucesso desse projeto.

Vale ressaltar que o reconhecimento das áreas de gerenciamento como grupos de processos, com preocupações distintas, e de que existe um grupo geral que analisa os impactos nas demais áreas de gerenciamento é uma grande contribuição à área de desenvolvimento de software. Analisando separadamente estas áreas tem-se, em um primeiro momento, o impacto do andamento das atividades em cada segmento, com uma visão bastante detalhada, para, em um segundo momento, verificar o impacto global deste acompanhamento. O que normalmente ocorre no desenvolvimento dos software é a total falta de controle, quanto mais separado por área.

Outra valiosa contribuição é a definição formal das várias fases pelas quais um projeto passa, reforçando a importância da fase de planejamento, onde os diversos aspectos relativos a cada área de gerenciamento serão levantados e planejados. Outra característica importante é a necessidade do início formal do processo de desenvolvimento de um software, envolvendo todos os interessados (clientes, equipe e fornecedores, entre outros) em busca do comprometimento.



Também importante é a constatação da necessidade de um encerramento formal das atividades, com a resolução das pendências ainda existentes; sem isso, o projeto não pode ser considerado como efetivamente concluído - processo não muito difundido na área de software, com vários produtos sendo gerados e concluídos sem que todos os aspectos tenham sido cuidadosamente verificados, causando desgaste, má qualidade, custos e descrédito para com os profissionais de informática.

Os processos de controle são essenciais para garantir que os rumos planejados estão em execução e que medidas corretivas eficazes são adotadas, quando necessárias. Esses processos introduzem a necessidade de revisões periódicas nas bases do projeto (custo, prazo, qualidade, escopo) a fim de refletir as modificações, além de verificar globalmente o impacto, uma vez que as modificações ocorrem normalmente em mais de uma área.

Com a introdução destes conceitos nas organizações, o gerenciamento de processos de desenvolvimento de software certamente será otimizado. A implantação de uma base histórica contendo dados dos projetos já desenvolvidos, lições aprendidas ao longo destes projetos, principais modificações ocorridas nestes projetos (com respectivas análises e planos de ação), técnicas de engenharia de software utilizadas (incluindo técnicas de estimativa de software, como pontos de função<sup>83</sup>, pontos de funcionalidade<sup>84</sup> e linhas de código), premissas adotadas em cada projeto, restrições destes projetos, características do projeto e comparações entre o que foi planejado e o que foi efetivamente realizado, auxiliará e garantirá a evolução dos processos e dos profissionais envolvidos com o desenvolvimento do software.

Também é recomendável o intercâmbio de experiências com profissionais e empresas que utilizam técnicas para melhorar a efetividade de seus processos de desenvolvimento de software; várias empresas no Brasil estão empenhadas em adotar modelos como ISO e CMMI e, mais recentemente, mps-Br, e com isso a necessidade de aprofundamento nos requisitos, além da demonstração prática de que processos

---

<sup>83</sup> Para aprofundamento no tema pontos de função (*function points*) recomenda-se Moore (1998) e Vazquez (2003).

<sup>84</sup> Para aprofundamento no tema pontos de funcionalidade (*feature points*) recomenda-se Jones (1991) e Vazquez (2003).

claros e efetivos de gerenciamento de projetos estão sendo adotados, passa a ser vital. No mundo dos negócios isto passa a ser questão de sobrevivência para todos os envolvidos.

## **7.2 Temas sugeridos como perspectivas futuras da pesquisa**

A fim de continuar a abordagem aqui utilizada vários aspectos poderiam vir a serem trabalhados. Segue uma pequena lista com algumas sugestões de trabalhos futuros.

- Aprofundar a análise, através de pesquisas, de em que medida o mercado brasileiro (e mundial) vem adotando as práticas de gerenciamento de projetos para o desenvolvimento de software;
- Acompanhar a adoção do CMMI e do mps-Br pelas empresas brasileiras e mundiais, em que proporção isto está ocorrendo e quais são as perspectivas futuras;
- Acompanhar a aceitação e a evolução do SWEBoK, iniciativa importante para a profissionalização das práticas de engenharia de software;
- Verificar possível convergência entre os preceitos de gerenciamento do PMI e da ISO, ainda distantes e com divergências marcantes;
- Publicação efetiva de um adendo ao PMBoK Guide referente às peculiaridades com o gerenciamento de processos de desenvolvimento de software;
- Verificar a adoção dos conceitos de desenvolvimento ágil e outras metodologias de engenharia de software.

## Referências Bibliográficas

AMBLER, Scott W. Maturing the OO Software Process - reducing risks and aumenting quality. **Object Magazine**, Fevereiro 1998.

AMBLER, Scott W. UML Deployment Modeling and Beyound. **Software Development Magazine**, Abril 1999.

AMBLER, Scott W. **Modelagem Ágil: Práticas eficazes para a Programação eXtrema e o Processo Unificado**. Porto Alegre: Bookman, 2004.

BARBOSA, Anderson Luiz. Estratégias de Teste em Bancos de Dados Relacionais. **Revista Ciência e Tecnologia**: publicação do Centro UNISAL – Ano V, n. 7, Outubro de 2002. Campinas, São Paulo: UNISAL. ISSN 1677-9649.

BARRETO JUNIOR, José. **Qualidade de Software**. 1997. Disponível em <<http://www.geocities.com/josedearaujo/qualidade/index.htm>>. Acesso em: 01 mar.2006.

BROOKS, Frederick P. Jr. **The mythical man-month: essays on software engineering**. Addison-Wesley Publishing Company, 1995.

CATHO, Arthur. **Melhores práticas em Tecnologia – o caso Instituto Eldorado**. In Comitê de Tecnologia – Câmara Americana de Comércio – AMCHAM, Campinas, 28.abril.2006.

CMU; SEI. **The Capability maturity model: Guidelines for improving the software process** (SEI Series in Software Engineering). CMU - Carnegie Mellon University; SEI - Software Engineering Institute. Addison Wesley Longman, 1995.

CMU; SEI. **CMMISM for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Continuous Representation (CMMI-SE/SW/IPPD/SS, V1.1, Continuous)**. CMU - Carnegie Mellon University; SEI - Software Engineering Institute. Março 2002. Disponível em <<http://www.sei.cmu.edu/publications/documents/02.reports/02tr011.html>>. Acesso em: Acesso em: 02 mar.2006.

CMU; SEI. **CMMISM for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Staged Representation (CMMI-SE/SW/IPPD/SS, V1.1, Staged)**. CMU - Carnegie Mellon University; SEI - Software Engineering Institute. Março 2002. Disponível em <<http://www.sei.cmu.edu/publications/documents/02.reports/02tr012.html>>. Acesso em: Acesso em: 02 mar.2006.

CMU; SEI. **CMMI: Guidelines for Process Integration and Product Improvement** (SEI Series in Software Engineering). CMU - Carnegie Mellon University; SEI - Software Engineering Institute. Addison Wesley Longman, 2003.

CMU; SEI. **PSP for Engineers Student**. Version 1.0. CMU - Carnegie Mellon University; SEI - Software Engineering Institute. Março 2005. Disponível em <<http://www.sei.cmu.edu/tsp/psp/download/academic-form.html>>. Acesso em: 04 mar.2006.

CMU; SEI. **The Personal Software Process (PSPSM) Body of Knowledge**. Version 1.0. CMU - Carnegie Mellon University; SEI - Software Engineering Institute. Special Report CMU/SEI-2005-SR-003, Agosto 2005. Disponível em <<http://www.sei.cmu.edu/pub/documents/05.reports/pdf/05sr003.pdf>>. Acesso em: 04 mar.2006.

CMU; SEI. **List of Published SCAMPI Appraisal Results**. CMU - Carnegie Mellon University; SEI - Software Engineering Institute. Disponível em <[http://seir.sei.cmu.edu/pars/pars\\_list\\_iframe.asp](http://seir.sei.cmu.edu/pars/pars_list_iframe.asp)>. Acesso em: 03 mar.2006.

CONNELL, Charles. Improve your company's ability to estimate software-development cycles. **Datamation Magazine**, Abril 1999.

COSTA, Geraldo M.; MIRANDA, João Paulo Barbosa de. Implantando Processos de Desenvolvimento de Software. **Developers Magazine**, Setembro 1999.

CREDICARD. Utilizando o CMM para melhoria do processo de software. In.: **3<sup>rd</sup>. Project Management Conference**. São Paulo: SUCESU/SP, Dezembro 1999.

DIAS, Fernando Rodrigues Teixeira. **Sistemas Corporativos de Informação para o Gerenciamento de Projetos: Um Estudo Exploratório**. Dissertação de mestrado, Universidade de São Paulo, Faculdade de Economia, Administração e Contabilidade, Departamento de Administração, São Paulo, 1999.

DINSMORE, Paul Campbell. **Managing the Project of Your Life**. Newtown Square, PA: Project Management Institute, PMI Seminars & Symposium. Proceedings, 1998, p. 1176-1185.

DUNCAN, William R. **ISO 10006: Risky Business**. 1999. Disponível em <<http://www.pmpartners.com/resources/iso10006.html>>. Acesso em: 26 mar.2006.

ESI. **System Integration Project Management - Course Material**. ESI & The George Washington University, julho 1997.

FABIAN-ISAACS, Constance; ROBINSON, Ed.. The Project Management Puzzle. **Software Development Magazine**, Março 1999.

FERNANDES, Aguinaldo Aragon; ALVES, Murilo Maia. **Gerência Estratégica da Tecnologia da Informação**. Rio de Janeiro: LTC Editora, 1992.

FURLAN, José Davi. **Modelagem de Objetos através da UML**. São Paulo: Makron Books, 1998.

GANE, Chris; SARSON, Trish. **Análise Estruturada de Sistemas**. Rio de Janeiro, LTC Livros Técnicos e Científicos Ltda., 1984.

GIL, Antônio de Loureiro. **Auditoria de qualidade - ISO 9000/10000 - Visão 2000**. 3ª. Edição. São Paulo: Atlas, 1999.

GUSTAFSON, David A. **Teoria e problemas de engenharia de software**. Coleção Schaum. Porto Alegre: Bookman, 2003.

HAYES, Linda. To win at software development, change the game. **Software Development Magazine**, Julho 1999.

HELDMAN, Kin. **Gerenciamento de projeto: Guia para o exame oficial do PMI**. Rio de Janeiro: Elsevier, 2005.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. Porto Alegre: Editora Sagra Luzzatto, 1999.

HUMPHREY, Watts S. **A Discipline for Software Engineering** (SEI Series in Software Engineering). Addison Wesley Publishing Company, 1995.

IEEE. **Guide to the software engineering body of knowledge**. 2004 version. Los Alamitos, CA: IEEE - Institute of Electrical and Electronics Engineers, 2004.

IGNISCOM. **Perfil Empresarial**. Campinas: Ignis Comunicações S/C Ltda, 2003.

IGNISCOM. **Conexão do Saber – o próximo passo da Educação**. Campinas: Ignis Comunicações S/C Ltda, 2005.

ISO. **ISO/IEC 9126 Information technology -- Software product evaluation -- Quality characteristics and guidelines for their use**. Genebra, ISO - International Organization for Standardization, 1993.

ISO. **ISO 9000-1 - Quality management and quality assurance standards - Part 1: Guidelines for selection and use**. Genebra, ISO - International Organization for Standardization, 1994.

ISO. **ISO 10005:1995 - Quality management -- Guidelines for quality plans**. Genebra, ISO - International Organization for Standardization, 1995.

ISO. **ISO/IEC 12207 - Information technology -- Software life cycle processes**. Genebra, ISO - International Organization for Standardization, 1995.

ISO. **ISO 10006 - Quality management -- Guidelines to quality in project management**. Genebra, ISO - International Organization for Standardization, 1997.

ISO. **ISO 9000-3 - Quality management and quality assurance standards -- Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software**. Geneva, ISO - International Organization for Standardization, 1997.

ISO. **ISO/IEC TR 15271:1998 - Information technology -- Guide for ISO/IEC 12207 (Software Life Cycle Processes)**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-1:1998 - Information technology -- Software process assessment -- Part 1: Concepts and introductory guide**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-2:1998 - Information technology -- Software process assessment -- Part 2: A reference model for processes and process capability**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-3:1998 - Information technology -- Software process assessment -- Part 3: Performing an assessment**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-4:1998 - Information technology -- Software process assessment -- Part 4: Guide to performing assessments**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-5:1998 - Information technology -- Software process assessment -- Part 5: An assessment model and indicator guidance**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-6:1998 - Information technology -- Software process assessment -- Part 6: Guide to competency of assessors**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-7:1998 - Information technology -- Software process assessment -- Part 7: Guide for use in process improvement**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-8:1998 - Information technology -- Software process assessment -- Part 8: Guide for use in determining supplier process**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO/IEC TR 15504-9:1998 - Information technology -- Software process assessment -- Part 9: Vocabulary**. Geneva, ISO - International Organization for Standardization, 1998.

ISO. **ISO 9001:2000 - Quality management systems -- Requirements**. Geneva, ISO - International Organization for Standardization, 2000.

ISO. **ISO/IEC 9126-1:2001 - Software engineering -- Product quality -- Part 1: Quality model.** Genebra, ISO - International Organization for Standardization, 2001.

ISO. **ISO/IEC 9126-2:2001 - Software engineering -- Product quality -- Part 2: External metrics.** Genebra, ISO - International Organization for Standardization, 2001.

ISO. **ISO/IEC 9126-3:2001 - Software engineering -- Product quality -- Part 3: Internal metrics.** Genebra, ISO - International Organization for Standardization, 2001.

ISO. **ISO/IEC 9126-4:2001 - Software engineering -- Product quality -- Part 4: Quality in use metrics.** Genebra, ISO - International Organization for Standardization, 2001.

ISO. **ISO 10006:2003 - Quality management systems -- Guidelines for quality management in projects.** Genebra, ISO - International Organization for Standardization, 2003.

ISO. Special Report - Global perspectives on global standards a 15-economy survey of ISO 9000 and ISO 14000. **Revista ISO Management Systems.** Janeiro-Fevereiro 2003. Genebra, ISO - International Organization for Standardization, 2003. Disponível em [http://www.iso.org/iso/en/iso9000-14000/addresources/articles/pdf/survey\\_1-03.pdf](http://www.iso.org/iso/en/iso9000-14000/addresources/articles/pdf/survey_1-03.pdf). Acesso em: 01 mar.2006.

ISO. **ISO/IEC 15504-1:2004 - Information technology -- Process assessment -- Part 1: Concepts and vocabulary.** Genebra, ISO - International Organization for Standardization, 2004.

ISO. **ISO/IEC 15504-2:2003/Cor 1:2004 - Information technology -- Process assessment -- Part 2: Performing an assessment.** Genebra, ISO - International Organization for Standardization, 2004.

ISO. **ISO/IEC 15504-3:2004 - Information technology -- Process assessment -- Part 3: Guidance on performing an assessment.** Genebra, ISO - International Organization for Standardization, 2004.

ISO. **ISO/IEC 15504-4:2004 - Information technology -- Process assessment -- Part 4: Guidance on use for process improvement and process capability determination.** Genebra, ISO - International Organization for Standardization, 2004.

ISO. **ISO/IEC 90003:2004 Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software.** Genebra, ISO - International Organization for Standardization, 2004.

ISO. **ISO 9000:2005 - Quality management systems - fundamentals and vocabulary.** Genebra, ISO - International Organization for Standardization, 2005.

ISO. **ISO/IEC TR 19759:2005 - Software Engineering -- Guide to the Software Engineering Body of Knowledge (SWEBOK)**. Genebra, ISO - International Organization for Standardization, 2005.

ISO. **The ISO Survey of Certifications - 2004**. Genebra, ISO - International Organization for Standardization, 2005. Disponível em <<http://www.iso.org/iso/en/prods-services/otherpubs/pdf/survey2004.pdf>>. Acesso em: 01 mar.2006.

ISO. **International harmonized stage codes**. Genebra, ISO - International Organization for Standardization, 2006. Disponível em <<http://www.iso.org/iso/en/widepages/stagetable.html#60>>. Acesso em: 01 mar.2006.

JACOBSON, Ivar. Objectory is the Unified Process. **Object Magazine**, Abril 1998.

JACOBSON, Ivar et al. **Object-Oriented Software Engineering**. Addison-Wesley, 1998.

JAMES, Geoffrey. IT fiascos... and how to avoid them. **Datamation Magazine**, Novembro 1997.

JONES, Capers. **Produtividade no desenvolvimento de software**. São Paulo, Makron Books, 1991.

KORTH, Henry F.; SILBERSCHATZ, Abraham. **Sistemas de Bancos de Dados**. 3ª edição. São Paulo: Makron Books, 1999.

LAUNI, Joe. Create a Project Plan. **Software Development Magazine**, Maio 1999.

MAFFEO, Bruno. **Engenharia de Software e Especificação de Sistemas**. Rio de Janeiro: Editora Campus, 1992.

MARTIN, James; McCLUER, Carma. **Técnicas Estruturadas e Case**. São Paulo: Makron McGraw-Hill, 1991.

MARTIN, James; ODELL, James J. **Análise e projeto orientados a objeto**. São Paulo: Makron McGraw-Hill, 1995.

MARTIN, Paula K.; TATE, Karen. Three Secrets for Creating Successful Projects. **The Journal of Quality & Participation**, Novembro/Dezembro 1998.

MARTIN, Paula K.; TATE, Karen. How much planning is enough? **PM Network magazine**, Fevereiro 1999.

MARTINS, José Carlos Cordeiro. **Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML**. 2ª. Edição. Rio de Janeiro: Brasport, 2005.



MCT. **Avaliações MA-MPS - Lista de Avaliações MA-MPS publicadas**. MCT – Ministério da Ciência e Tecnologia, 2006. Disponível em <<http://www.softex.br/cgi/cgilua.exe/sys/start.htm?sid=222>>. Acesso em: 04 mar.2006.

MCT. **Tecnologia da Informação - Qualificação CMM no Brasil**. MCT – Ministério da Ciência e Tecnologia, 2006. Disponível em <<http://www.mct.gov.br/SEPIN/Dsi/qualidad/CMM.htm>>. Acesso em: 03 mar.2006.

MEDEIROS, Ernani Sales de. **Desenvolvendo software com UML 2.0: definitivo**. São Paulo: Pearson Makron Books, 2004.

MELLO FILHO, Moacyr Cardoso de. O CMM Como um Guia na Evolução dos Processos. **Revista Developers Magazine**, Setembro 1999.

MELLO FILHO, Moacyr Cardoso de. Planejamento de Projeto em Processo Iterativo e Incremental. **Revista Developers Magazine**, Setembro 1999.

MICROSOFT. **The Windows interface guidelines for software design**. Redmond, Washington: Microsoft Corporation, 1995.

MOORE, James W. Demystifying Software Engineering Standards. In **Conferência Anual da IFPUG - 1998**, Agosto 1998.

MORAES, Renato de Oliveira. **Condicionantes de desempenho dos projetos de software e a influência da maturidade em gestão de projetos**. 2004. 138 f. Tese (Doutorado em Administração de Empresas) – Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, São Paulo.

MOURA, Arnaldo Vieira. **Especificação em Z: uma introdução**. Campinas, São Paulo: Editora da UNICAMP, 2001.

MYERS, G. **The Art of Software Testing**. Wiley, 1979.

NOLAN, R. L. **Management Accounting and Control of Data Processing**. Nova York: National Association of Accountants, 1977.

O'CONNELL, Fergus. **How to run Successful Projects - the silver bullet**. Londres, Prentice-Hall, 1994.

PAGE-JONES, Meilir. **Projeto estruturado de sistemas**. São Paulo, McGraw-Hill, 1988.

PAGE-JONES, Meilir. **Gerenciamento de projetos: guia prático para restauração da qualidade em projetos e sistemas de processamento de dados**. São Paulo, McGraw-Hill, 1990.

PAULK, Mark C. et. al. **Capability Maturity Model for Software**, Version 1.1 Technical Report CMU/SEI-93-TR-024, 1993.

PAULK, Mark C. et. al. **A Comparison of ISO 9001 and the Capability Maturity Model for Software**. Technical Report CMU/SEI-94-TR-12, 1994.

PESSOA, Marcelo Schneck de Paula. Uma comparação entre os modelos ISO 9000 e CMM aplicados ao software. In.: **1<sup>st</sup>. Project Management Conference**. São Paulo: SUCESU/SP, Julho 1997.

PETERS, James F.; PEDRYCZ, Witold. **Engenharia de Software**. Rio de Janeiro: Campus, 2001.

PFLEEGER, Shari Lawrence. **Engenharia de Software – Teoria e Prática**. São Paulo: Prentice Hall, 2004.

PILLAI, Krishnakumar; NAIR, V. S. Sukumaram. A Model for Software Development Effort and Cost Estimation. **IEEE Transactions on Software Engineering**, vol. 23 nº 8, Agosto 1997.

PIRES, Fátima. **Modelo CMM de qualidade de software**. Informativo técnico nº 70 - revista de Informação e Tecnologia - Centro de Computação da UNICAMP, janeiro/fevereiro 2000.

PLONSKI, Guilherme Ary. PMI Brasil e o PMBoK. Artigo apresentado no **2nd. Project Management Conference** - SUCESU/SP, Julho 1998.

PMI. **PMBOK Guide** (Project Management Body of Knowledge). 3<sup>rd</sup>. Edition. Four Campus Boulevard, Newton Square, PA: PMI - Project Management Institute, 2004.

PRESSMAN, Roger S. **Engenharia de Software**. 5<sup>a</sup>. Edição. Rio de Janeiro: McGraw Hill, 2002.

PRESSMAN, Roger S. **Engenharia de Software**. 6<sup>a</sup>. Edição. São Paulo: McGraw Hill, 2006.

RATIONAL SOFTWARE. **UML - Quick reference for Rational Rose - version 1.1**. Rational Software, Setembro 1997.

REZENDE, Denis Alcides. **Engenharia de software e sistemas de informação**. 3<sup>a</sup>. Edição. Rio de Janeiro: Brasport, 2005.

ROBERTS Jr, Tom L. et. al. Factor that Impact Implementing a System Development Methodology. **IEEE Transactions on Software Engineering**, vol. 24 nº8 Agosto 1998.

ROCHA, Ana Regina Cavalcanti da; MALDONADO, José Carlos; WEBER, Kival Chaves. **Qualidade de Software – Teoria e Prática**. São Paulo: Prentice Hall, 2001.

ROCHA, Heloísa Vieira da; BARANAUSKAS, Maria Cecília Calani. **Design e avaliação de interfaces humano-computador**. Campinas, São Paulo: NIED/UNICAMP, 2003.

RUMBAUGH, James et al. **Modelagem e Projetos baseados em objetos**. Rio de Janeiro, Editora Campus, 1994.

SCHMAUCH, Charles H. **ISO 9000 for software developers**. Milwaukee: ASQC Quality Press, 1994.

SOFTEX. **MPS.BR - Melhoria de Processo do Software Brasileiro - Guia Geral**. Versão 1.0. SOFTEX, Abril 2005. Disponível em <<http://www.softex.br/media/guia.pdf>>. Acesso em: 28 out.2005.

SOMMERVILLE, Ian. **Engenharia de Software**. 6a. Edição. São Paulo: Addison-Wesley, 2003.

SPECTRUM Engenharia Ltda. Gerenciamento de Riscos. In.: **3<sup>rd</sup>. Project Management Conference**. São Paulo: SUCESU/SP, Dezembro 1999.

STANLEIGH, Michael. **Combining the ISO 10006 and PMBOK To Ensure Successful Projects**. Business Improvement Architects, 2005. Disponível em <<http://www.bia.ca/articles/pj-combining-iso-10006-pmbok-to-ensure-successful-projects.htm>>. Acesso em: 04 mar.2006.

STEFANINI. **Stefanini conquista nível máximo do CMMI**. 2005. Disponível em <<http://www.stefanini.com.br/news.aspx?id=5>>. Acesso em: 04 mar.2006.

SETZER, Valdemar W. **Data, Information, Knowledge and Competence**. Versão 3.1 – Fevereiro 2006. Disponível em <<http://www.ime.usp.br/~vwsetzer/data-info.html>>, 2006. Acesso em: 06 mar.2006.

VARGAS, Ricardo Viana. **Manual prático de plano de projeto**. Rio de Janeiro: Brasport, 2003.

VAZQUEZ, Carlos Eduardo et. al.. **Análise de pontos de função: medição, estimativas e gerenciamento de projetos de software**. São Paulo: Érica, 2003.

VOLPE, Renato Dela; ZABEU, Ana Cecilia P. A Experiência da NEC do Brasil no Planejamento e Controle de Projetos de Software. In.: **1<sup>st</sup>. Project Management Conference**. São Paulo: SUCESU/SP, Julho 1997.

WEBER, Kival Chaves; CAVALCANTI, Ana Regina. **Qualidade e Produtividade em Software**. São Paulo, Makron Books, 1999.

WHITEN, Neal. **Managing Software development projects: formula for success**. John Wiley & Sons, Inc, 1995.

YOURDON, Edward. **Análise Estruturada Moderna**. Rio de Janeiro, Editora Campus, 1990.

YOURDON, Edward. Análise e Projeto Orientados a Objetos. São Paulo: Makron Books, 1999.

## Índice de Autores

---

Ambler (1998).....	104
Ambler (1999).....	27
Ambler (2004).....	37
Barbosa (2002).....	28, 101
Barreto Junior (1997).....	45, 47, 49
Brooks (1995).....	1, 90
Catho (2006).....	69, 100
CMU; SEI (1995) .....	56
CMU; SEI (2002a) .....	56
CMU; SEI (2002b) .....	56, 59
CMU; SEI (2003) .....	56, 60, 62, 63
CMU; SEI (2005a) .....	70, 71
CMU; SEI (2005b) .....	70
CMU; SEI (2006) .....	68
Connell (1999) .....	102
Costa; Miranda (1999).....	102
Credicard (1999).....	69
Dias (1999) .....	84
Dinsmore (1998).....	99
Duncan (1999).....	99
ESI (1997) .....	105
Fabian-Isaacs; Robinson (1999).....	103
Fernandes; Alves (1992) .....	3
Furlan (1998) .....	27
Gane; Sarson (1984) .....	27
Gil (1999).....	41

Gustafson (2003).....	28
Hayes (1999) .....	105
Heldman (2005).....	10
Heuser (1999).....	27
Humphrey (1995).....	55
IEEE (2004) .....	54
IGNISCOM (2003) .....	109
IGNISCOM (2005) .....	110
ISO (1993) .....	46
ISO (1994).....	39, 43
ISO (1995a) .....	75
ISO (1995b) .....	46
ISO (1997a) .....	75
ISO (1997b) .....	45
ISO (1998a) .....	48
ISO (1998b) .....	53
ISO (1998c) .....	53
ISO (1998d) .....	53
ISO (1998e) .....	53
ISO (1998f) .....	53
ISO (1998g) .....	53
ISO (1998h) .....	53
ISO (1998i) .....	53
ISO (1998j) .....	53
ISO (2000).....	44
ISO (2001a).....	46
ISO (2001b).....	46
ISO (2001c) .....	46
ISO (2001d).....	46
ISO (2003a).....	75, 99

ISO (2003b) .....	43
ISO (2004a) .....	53
ISO (2004b) .....	53
ISO (2004c) .....	53
ISO (2004d) .....	53
ISO (2004e) .....	45
ISO (2005a) .....	39, 40
ISO (2005b) .....	53, 55
ISO (2005c) .....	44
ISO (2006) .....	15
Jacobson (1998) .....	27
Jacobson et. al. (1998) .....	26, 27
James (1997) .....	105
Jones (1991) .....	118
Korth; Silberschatz (1999) .....	27
Launi (1999) .....	103
Maffeo (1992) .....	21, 23
Martin; McCluer (1991) .....	24, 27, 102
Martin; Odell (1995) .....	27
Martin; Tate (1998) .....	103
Martin; Tate (1999) .....	103
Martins (2005) .....	99
MCT (2006a) .....	73
MCT (2006b) .....	69
Medeiros (2004) .....	27
Mello Filho (1999a) .....	58
Mello Filho (1999b) .....	103
Microsoft (1995) .....	27
Moore (1998) .....	118
Moraes (2004) .....	2

Moura (2001) .....	37
Myers (1979) .....	28
Nolan (1977).....	4
O´Connell (1994) .....	90
Page Jones (1988) .....	27
Page Jones (1990) .....	90
Paulk (1994) .....	100
Paulk et. al. (1993).....	56, 57, 58, 60, 61
Pessoa (1997) .....	100
Peters; Pedrycz (2001).....	33
Pfleeger (2004).....	21, 26, 32
Pillai; Nair (1997) .....	103
Pires (2000) .....	58
Plonski (1998).....	9
PMI (2004).....	7, 8, 80, 81, 82, 83, 84, 85, 87, 88, 89, 90, 95, 96, 99, 101
Pressman (2002) .....	2, 20, 21, 24, 25, 26, 28, 29, 31, 33, 34, 35, 37
Pressman (2006) .....	38
Rational Software (1997) .....	27
Rezende (2005).....	21, 24
Roberts Jr. et. al. (1998).....	108
Rocha; Baranauskas (2003) .....	27
Rocha; Maldonado; Weber (2001).....	43, 44
Rumbaugh et. al. (1994).....	27
Schmauch (1994) .....	39, 40, 43
Setzer (2006).....	78
SOFTEX (2005).....	72
Sommerville (2003).....	21, 37
Spectrum (1999).....	2
Stanleigh (2005) .....	99
Stefanini (2005) .....	69



Vargas (2003).....	103, 108
Vazquez (2003) .....	118
Volpe; Zabeu (1997).....	69
Weber; Cavalcanti (1999).....	71
Whiten (1995).....	108
Yourdon (1990).....	27
Yourdon (1999).....	27