

Leonardo D. V. Ribeiro

UMA ANÁLISE DE REQUISITOS DE  
SOFTWARE DE GESTÃO DE PROJETOS  
PARA APOIO AO PROCESSO DE  
DESENVOLVIMENTO DE PRODUTOS

Dissertação apresentada à Escola de  
Engenharia de São Carlos da  
Universidade de São Paulo para a  
obtenção do Título de Mestre em  
Engenharia de Produção.

Orientador: Prof. Tit. Henrique Rozenfeld

SÃO CARLOS  
JULHO / 2006

# **Livros Grátis**

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

Milhares de livros grátis para download.



## **AGRADECIMENTOS**

Aos Professores Henrique Rozenfeld, Daniel Capaldo Amaral e Sergio Luis da Silva pelo apoio, motivação e amizade.

Aos meus pais, por toda a dedicação, carinho e motivação para realização deste trabalho.

A minha namorada Aninha, pela inesgotável paciência, amor e carinho.

A todos os amigos e colegas do Grupo de Engenharia Integrada do Núcleo de Manufatura Avançada (NUMA), em especial à Cristiane, Fernandinho e Francis, pelo carinho e pela ajuda incansável, além dos colegas de doutorado, mestrado e iniciação científica.

Aos amigos e parceiros de projeto Antonio Jucá, Rafael Netto e João Roriz, pelo enorme apoio na realização deste trabalho.

Aos colegas do projeto e-HUB: Regis K. Scalice, Godfried Augenbroe, Patrick Crehan, Mauro Mangini, Ren Zao, Nickos Floratos, Georgos Valossopoulos, Nicola Radic, e todo o pessoal de Delft e Aachen.

A Escola de Engenharia de São Carlos, com seus professores e funcionários que me acolheram estes anos todos.

A FAPESP, pelo suporte financeiro

A todos os demais que contribuíram de alguma forma para a realização deste trabalho.

## IV

### RESUMO

RIBEIRO, L. D. V, *Uma análise de requisitos de software de gestão de projetos para apoio ao processo de desenvolvimento de produtos*. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

A globalização força cada vez mais a união entre empresas através de acordos e parcerias, fazendo do desenvolvimento colaborativo de novos produtos um requisito básico para a competitividade e sobrevivência. A Gestão de Projetos (GP), essencial para o Desenvolvimento de Produtos (DP), é uma área complexa, e, portanto, é difícil estruturar todos os elementos necessários para garantir o sucesso de sua aplicação prática.

Um dos fatores mais importantes para a boa Gestão de Projetos é a gestão da comunicação e do conhecimento gerado durante o projeto.

Neste contexto, este trabalho visa o levantamento de requisitos de software para gestão de projetos, contendo métodos, práticas e ferramentas relevantes à indústria e focado no Processo de Desenvolvimento Colaborativo de Produtos. Estes requisitos devem apoiar conceitos de colaboração e gestão de conhecimento, baseando-se em conceitos retirados da literatura; resultados de trabalhos passados dentro do Grupo de Engenharia Integrada do Núcleo de Manufatura Avançada (NUMA) e requisitos provenientes das análises de *softwares* livres para gestão de projetos, complementando assim o desenvolvimento científico do grupo de pesquisa e as melhores práticas encontradas na indústria em geral.

Esta pesquisa tem caráter qualitativo e descritivo e, como métodos de procedimento de pesquisa, a pesquisa descritiva do tipo participante-observador, pesquisa bibliográfica e documental.

**Palavras-chave:** Gestão de Projetos, Desenvolvimento Colaborativo de Produtos, Gestão do Conhecimento, Engenharia de Software, Engenharia de Requisitos.

## **ABSTRACT**

Everyday, the globalization forces even more the joining of companies through business agreements and partnerships, making of the collaborative product development of new products a basic requirement to the competitiveness or even survival. The Project Management, essential to Product Development, is a complex area of knowledge and, therefore difficult to structuralize all the elements needed to ensure the success of its practical application inside different industries.

One of the most important factors necessary to a good Project Management is managing the communication and the knowledge created along the project.

Within this context, this work aims the develop a list of requirements for a project management software focused at the Product Development Process. These requirements must include concepts of geographically distributed collaboration and knowledge management. The main sources for this work are the general literature; work results from past projects inside the Engineering Integration Group of the Nucleus of Advanced Manufacturing of the University of São Paulo and requirements from analysis of open-source Project Management softwares, adding up to the scientific development of the research group and best practices found in the industry in general.

This research is qualitative and descriptive. It makes use of the research methods of participant-observer, bibliographical research and documental research.

**Key words:** Project Management, Collaborative and distributed Product Development, Knowledge Management, Software Engineering and Requirement Engineering.

## VI

### LISTA DE FIGURAS

Figura 1: processo de desenvolvimento de produtos .....	6
Figura 2: sub-processos do Desenvolvimento de Produto .....	7
Figura 3 - Atual Escopo do PDP .....	8
Figura 4: modelo de Referência da comunidade PDPNet.....	11
Figura 5: escopo dos principais sistemas corporativos .....	14
Figura 6: relações entre os processos de GP .....	16
Figura 7: áreas de conhecimento em GP.....	17
Figura 8: passos da gestão de conhecimento .....	27
Figura 9: modelo de Desenvolvimento de Software em Cascata.....	36
Figura 10: modelo evolucionário em espiral .....	37
Figura 11: processo de engenharia de requisitos .....	39
Figura 12: grandes Atividades em OOA / OOD.....	48
Figura 13: uma classe e sua instância .....	50
Figura 14: herança entre classes de objeto.....	51
Figura 15: blocos de Construção da UML.....	54
Figura 16: exemplo de um diagrama de caso de uso .....	56
Figura 17: exemplo de um diagrama de classes.....	57
Figura 18: abordagens para classificação de pesquisas científicas .....	59
Figura 19: método de trabalho .....	64
Figura 21: Estrutura dos principais componentes de uma parceria através do e-hub .....	74
Figura 22: aderência do Modelo de referência aos requisitos.....	77
Figura 23: tarefas suportadas por tipo de requisito .....	80
Figura 24: tarefas suportadas por tipo de requisito e fase do modelo .....	81
Figura 24: Diagrama de Classes: Usuários .....	137
Figura 25: diagrama de Casos de Uso: necessidades dos Usuários de Conhecimento .....	138
Figura 26: Diagrama de Casos de Uso: Necessidades de Estudantes e Profissionais.....	139

Figura 27: Diagrama de Casos de Uso: Necessidades da Equipe de Projetos DP .....	140
Figura 28: Diagrama de Casos de Uso: Necessidades dos Gerentes de Projeto .....	141

## VIII

### **LISTA DE TABELAS**

Tabela 1: números de tarefas detalhadas por fase do modelo.....	12
Tabela 2: áreas de conhecimento versus Processos de GP .....	18
Tabela 3: classificação das funcionalidades de softwares de gestão de projetos .....	23
Tabela 4: métodos baseados em processo para gestão de conhecimento em projetos.....	30
Tabela 5: métodos baseados em documentos para gestão de conhecimento em projetos .....	31
Tabela 6: atividades do Modelo de Referência atendidas pelos grupos original de requisitos .....	78
Tabela 7: classificação final dos requisitos após comparação com o modelo de referência.....	91
Tabela 8: requisitos levantados a partir de softwares de gestão de projetos.....	93
Tabela 9: requisitos baseados nas funcionalidades do MS-Project.....	97
Tabela 10: requisitos baseados no framework para sistematização de conhecimentos do grupo EI .....	98
Tabela 11: requisitos derivados do projeto e-HUBs .....	99
Tabela 12: referência cruzada entre requisitos de software e modelo de referência em DP .....	100
Tabela 13: lista final de requisitos .....	133

## **LISTA DAS PRINCIPAIS ABREVIATURAS**

DP = Desenvolvimento de Produtos

e-Cofactor = e-Collaboration Framework for Active Organizations. Programa de pesquisa para desenvolvimento de um framework para e-PLM. Direciona atualmente o corpo de pesquisadores do Grupo EI.

e-HUBs = E-engineering enabled by Holonomic and Universal Broker Services. Projeto internacional, envolvendo diversas universidades e empresas da Europa, Brasil e México.

FMEA = Failure Mode and Effect Analysis

GP = Gestão de Projetos

Grupo EI = Grupo de Engenharia Integrada do NUMA

OOA / OOD = Object Oriented Analysis / Object Oriented Design

PDP = Processo de Desenvolvimento de Produtos

PLM = Product Lifecycle Management (Gerenciamento do ciclo de vida do Produto)

NUMA = Núcleo de Manufatura Avançada

PERT/CPM = Program Evaluation and Review Technique / Critical Path Method

PMBOK = Project Management Body of Knowledge (Corpo de conhecimentos em Gestão de Projetos). Livro editado pelo PMI contendo as práticas mais aceitas em Gestão de Projetos.

PMI = Project Management Institute

QFD = Quality Function Deployment

RAD = Rapid Application Development

UML = Unified Modeling Language. Linguagem de modelagem popularmente usada dentro da filosofia de Orientação a Objetos

## SUMÁRIO

<b>Resumo</b>	<b>IV</b>
<b>Abstract</b>	<b>V</b>
<b>Lista de Figuras</b>	<b>VI</b>
<b>Lista de Tabelas</b>	<b>VIII</b>
<b>Lista das principais abreviaturas</b>	<b>IX</b>
<b>1 Introdução e justificativa</b>	<b>1</b>
1.1 Objetivos	2
1.2 Contexto deste trabalho e fontes de informação e conhecimento:	3
1.3 Estrutura do trabalho	3
<b>2 Desenvolvimento de Produtos</b>	<b>5</b>
2.1 O Desenvolvimento de Produtos como um processo	5
2.1.1 Sub-processos dentro do PDP	6
2.2 Ciclo de Vida de um Projeto de Desenvolvimento de Produtos	7
2.3 Modelo de referência em Desenvolvimentos de Produto	9
2.3.1 O Modelo de Referência desenvolvido pela comunidade PDPNet	10
2.4 Sistemas para Gestão do ciclo de vida de produtos (PLM)	13
<b>3 A Gestão de Projetos</b>	<b>15</b>
3.1 Processos dentro da Gestão de Projetos	15
3.2 Áreas de conhecimento relacionadas à gestão de projetos	16
3.3 Áreas de Conhecimentos versus Processos	18
3.4 Gestão colaborativa e distribuída de Projetos	20
3.4.1 Sistemas para Gestão de Projetos	20
<b>4 Gestão de Conhecimentos</b>	<b>25</b>
4.1 Conhecimento no Processo de Desenvolvimento de Produtos	26
4.2 Gestão do Conhecimento em Gestão de Projetos	28

	XI
<b>5 Desenvolvimento de Sistemas de Informação</b>	<b>34</b>
5.1 Metodologia de desenvolvimento de sistemas	34
5.1.1 Modelo clássico em cascata	35
5.1.2 Modelo evolucionário em espiral	36
5.2 Engenharia de requisitos	38
5.2.1 Concepção	40
5.2.2 Elicitação	42
5.2.3 Elaboração	44
5.2.4 Negociação	45
5.2.5 Especificação	45
5.2.6 Validação	46
5.3 Análise e design orientados a objeto	47
5.4 Unified Modeling Language	52
5.4.1 Apresentação geral dos blocos de Construção da UML	53
5.4.2 Diagramas em UML	55
<b>6 Metodologia</b>	<b>58</b>
6.1 Perguntas de Pesquisa	58
6.2 Referencial Teórico	58
6.2.1 Seleção dos procedimentos de pesquisa	60
6.2.2 Limitações do método de trabalho	62
6.3 Atividades do Trabalho	63
FASE 1: levantamento Inicial das necessidades / requisitos do sistema	64
FASE 2: desenvolvimento de referência cruzada os requisitos e um modelo de referência para o PDP	66
<b>7 Resultados do trabalho</b>	<b>69</b>
7.1 Classificação dos requisitos	69
7.2 Listas de requisitos de software de gestão de projetos	70

## XII

7.2.1	Requisitos levantados a partir de softwares de gestão de projetos	70
7.2.2	Requisitos levantados a partir de funcionalidades de software comercial de referência	71
7.2.3	Requisitos levantados a partir de projetos anteriores e discussões internas dentro do Grupo EI	71
Todas os requisitos levantados durante o projeto estão inseridos neste trabalho e podem ser vistos no apêndice B, Tabela 11, sendo agrupados de acordo com a classificação padrão apresentada no item 7.1 e na Apêndice A – Classificação final dos requisitos		
		76
7.3	Referencia cruzada entre modelo de referência para o PDP e lista inicial de requisitos	76
7.4	Lista final de requisitos de software, agrupados de acordo com a classificação final	82
<b>8</b>	<b>Discussões e aplicação prática na indústria</b>	<b>83</b>
<b>9</b>	<b>Referências</b>	<b>84</b>
<b>10</b>	<b>Apêndices</b>	<b>90</b>
	Apêndice A – Classificação final dos requisitos	91
	Apêndice B – Listas de requisitos de diversas fontes	93
	Apêndice C – Referência cruzada entre requisitos de software e modelo de referência em DP	100
	Apêndice D – Lista final dos requisitos	133
	Apêndice E - Necessidades específicas por ator (diagramas de caso de uso)	137

## 1 INTRODUÇÃO E JUSTIFICATIVA

“Problems worthy of attack prove their worth by hitting back”

Piet Hein

Empresas que conquistam mercado mais rápida e eficientemente criam uma significativa vantagem competitiva. Em um ambiente competitivo, o bom Desenvolvimento de Produtos (DP) se tornou um requisito para a sobrevivência; um extraordinário Desenvolvimento de Produtos se tornou uma vantagem competitiva (WHEELRIGHT & CLARK, 1992). No Processo de Desenvolvimento de Produtos (PDP), o trabalho colaborativo tem sido uma das principais alternativas à competição acirrada que as empresas enfrentam diretamente (LITTLER et al. 1995).

As dificuldades do Desenvolvimento Colaborativo de Produtos em uma escala global não são apenas técnicas. A complexidade gerencial necessária para gerir interações entre várias disciplinas, alocadas em localidades diferentes, impõe desafios adicionais ao processo cuja competição é baseada no tempo de desenvolvimento (CHEN et al. 2003). A Gestão de Projetos (GP) é um dos aspectos primordiais para ajudar a reduzir esses desafios (PMI, 2004). De fato, afirma-se que as iterações do Processo de Desenvolvimento de Produtos, por produzir algo único, tem características muito semelhantes às de um Projeto, na concepção clássica de Gestão de Projetos (VERZUH, 1999).

Para auxiliar os profissionais da área, diversos softwares para auxiliar a gestão de projetos surgem no mercado, com funcionalidades distintas (GARTNER, 2002). Essas funcionalidades distintas tornam difícil a comparação entre softwares, uma vez que são também distintos as necessidades da própria gestão de projetos dentro do processo maior onde está inserida.

Neste sentido, Badiru (1996), apresenta fatores para uma boa avaliação desses softwares

No entanto, é importante saber quais das necessidades em termos de software são genéricas e quais são específicas para apoiar o Processo de Desenvolvimento de Produtos. A partir dessa lista de requisitos, avaliada frente ao PDP, pode-se desenhar um modelo de software que contenha propostas de soluções para tais necessidades.

No entanto, não existe um Processo de Desenvolvimento único. Existem divergências nas visões que diferentes indústrias, profissionais e pesquisadores têm sobre o seu Processo de Desenvolvimento de Produtos (ROZENFELD et al. 2003).

Numa tentativa de integrar essas visões, grupos de pesquisadores de três universidades desenvolveram um modelo de referência único para o Processo de Desenvolvimento de Produto, integrado às práticas de gestão do conhecimento. Este processo foi posteriormente publicado na forma de livro (ROZENFELD et al. 2006)

Neste contexto, este presente trabalho visa fazer uma comparação de requisitos de software para gestão de projetos e compará-los com um processo genérico de desenvolvimento de produtos, identificando pontos de suporte. Este trabalho também inicia o desenvolvimento de um modelo conceitual de software que faça uso destes requisitos, para futuro desenvolvimento do software de GP para atender as necessidades específicas desse processo.

## **1.1 Objetivos**

Os objetivos gerais desta pesquisa podem ser descritos como:

1. Obter uma lista de requisitos para sistemas de apoio a gestão de projetos compatível com um modelo de referência de desenvolvimento de produtos

1.1. Listar os requisitos de software de diversas fontes.

1.2. Organizar os requisitos dentro de uma classificação coerente, tanto com a Gestão de Projetos como com o Processo de Desenvolvimento de Produtos.

1.3. Confrontar os requisitos com um processo genérico de Desenvolvimento de Produtos.

## **1.2 Contexto deste trabalho e fontes de informação e conhecimento:**

Essa pesquisa está inserida em um contexto mais amplo de programas de pesquisas desenvolvidos pelo Núcleo de Manufatura Avançada (NUMA), mais especificamente dentro do grupo de pesquisas em Engenharia Integrada (Grupo EI), que visa ao desenvolvimento de diversas pesquisas relacionadas com colaboração entre empresas, modelagem de empresas, gestão do conhecimento e o processo de negócios de Desenvolvimento de Produtos.

Outra fonte importante de experiência e conhecimento que será sistematizado neste trabalho é do projeto europeu e-HUBs (e-engineering enabled by Holonomic Universal Broker services). Um e-HUB deve ser o agente (empresa, com serviços e infra-estrutura) para a formação e coordenação de clusters de empresas para praticar a engenharia colaborativa e promover a venda de serviços de engenharia. (AMARAL et. al, 2004).

A equipe da qual o autor deste trabalho faz parte foi responsável pela definição de requisitos e da arquitetura da plataforma do e-HUBs.

Uma fonte importante dos requisitos de gestão de documentos e conhecimentos é a arquitetura para gerenciamento de conhecimentos explícitos desenvolvida anteriormente dentro do grupo de pesquisa (PDPNet).

Desta forma, esta pesquisa de mestrado pretende alavancar essas experiências e utilizar conceitos existentes sobre Gestão de Projetos e análises das ferramentas livres existentes para GP para gerar uma lista de requisitos de software que atenda a um modelo de referência para o Processo de Desenvolvimento de Produtos (ver item 2.3.1).

## **1.3 Estrutura do trabalho**

Neste capítulo serão apresentados as justificativas e os objetivos propostos deste trabalho.

No item 1.2 será apresentada caracterização do tema e contexto deste trabalho, além de seu relacionamento com pesquisas anteriores do grupo de pesquisa.

Nos capítulos 2, 3, 4 e 5 serão apresentados a revisão da literatura fundamental, incluindo definições importantes, estudos sobre o processo de desenvolvimento colaborativo de produtos, a gestão de projetos e a gestão de conhecimentos, assim como outros conceitos importantes e necessários para a realização do trabalho proposto.

No capítulo 6 será apresentada metodologia que se pretende adotar, contendo uma explicação sobre a seleção da abordagem metodológica e as etapas de desenvolvimento deste trabalho.

O capítulo 7 mostra os principais resultados obtidos neste trabalho.

O capítulo 8 apresenta discussões sobre aplicações práticas deste trabalho na indústria.

O capítulo 9 apresenta as referências bibliográficas desta dissertação.

O capítulo 10 apresenta os apêndices desta dissertação.

Os próximos capítulos mostram a revisão da literatura, apresentando os conceitos fundamentais para a elaboração deste trabalho, incluindo as áreas de conhecimento listadas a seguir:

- Desenvolvimento de produtos, apresentada no capítulo 2;
- Gestão de projetos, apresentada no capítulo 3;
- Gestão de conhecimento, apresentada no capítulo 4; e
- Desenvolvimento de sistemas de informação e gestão de requisitos, apresentada no capítulo 5

## **2 DESENVOLVIMENTO DE PRODUTOS**

De acordo com Clark e Fujimoto (1991), a estratégia de desenvolvimento de produto (DP) determinará como o produto sairá do mercado. A maneira como uma empresa realiza o DP, sua velocidade, eficiência e qualidade do trabalho irá determinar a competitividade do produto.

### **2.1 O Desenvolvimento de Produtos como um processo**

Uma definição clássica e simples de processo de negócio é a apresentada por Davenport (1994, p.7): “uma ordenação específica das atividades de trabalho no tempo e no espaço, com um começo, um fim, e *inputs* e *outputs* claramente identificados”.

Um processo de negócio é um fenômeno que ocorre dentro das empresas. Ele contém um conjunto de atividades, associadas às informações que manipula, utilizando os recursos e a organização da empresa, estando direcionado a um determinado mercado/cliente, com fornecedores bem definidos (ROZENFELD et al., 1996).

Clark e Fujimoto (1991) definem o Processo de Desenvolvimento de Produtos (PDP) como um processo pelo qual uma organização transforma dados sobre oportunidades de mercado, de acordo com as capacidades tecnológicas, em bens e informações para a fabricação de um produto comercial. Dessa forma, esse processo cobre diversas funções, como marketing, engenharia de produto, produção e quase todas as outras áreas de uma empresa (ver Figura 1).

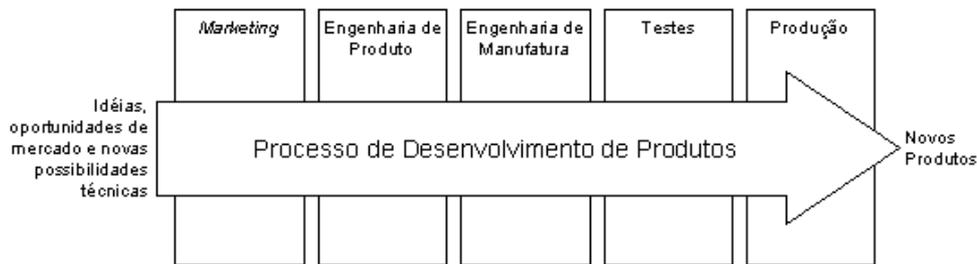


Figura 1: processo de desenvolvimento de produtos (adaptado de DAVENPORT, 1994, p.10)

Rozenfeld et al. (1996) destaca o caráter multifuncional do PDP, isso é, um processo de negócio que não se restringe a uma unidade funcional da empresa, mas que cruza diversas delas.

Segundo Davenport (1994), a visão por processo de negócio, ao contrário da visão tradicional por departamentos e áreas funcionais, facilita a medição dos resultados e a execução de projetos de melhoria.

Vale ainda destacar que o PDP encontra-se na interface entre a organização e o mercado, cabendo a ele: desenvolver um produto que atenda às expectativas de mercado, em termos de qualidade total do produto; e desenvolver o produto em tempo adequado, ou seja, mais rápido do que os concorrentes, e a um custo de projeto compatível. (FLORENZANO, 1999).

No entanto, Tatikonda & Rosenthal (2000) alertam para o excesso de formalidade no PDP. Para os autores, formalidade no PDP ocorre pela utilização de processos estruturados para gerenciar projetos, o que pode ser benéfico para projetos similares a outros já desenvolvidos pela empresa, mas que pode ser prejudicial para projetos únicos, para os quais valores de referência são mais difíceis de achar.

### 2.1.1 Sub-processos dentro do PDP

O caráter multifuncional do Processo de Desenvolvimento de Produtos permite que existam vários sub-processos inseridos dentro do processo maior.

Um Processo de Negócio pode conter diversos sub-processos, que representam uma visão separada do processo geral, relacionando atividades mais ligadas a uma área funcional ou simplesmente por afinidade, espalhando-se ao longo de todo o ciclo de vida do produto. Esta divisão do PDP em sub-processos permite um controle mais detalhado das atividades referentes ao processo, agrupando as atividades por função ou área de conhecimento (ROZENFELD et al. 2003). A Figura 2 mostra uma possível classificação de sub-processos:



Figura 2: sub-processos do Desenvolvimento de Produto

## 2.2 Ciclo de Vida de um Projeto de Desenvolvimento de Produtos

O processo de desenvolvimento de produto é tipicamente estruturado em várias fases. Vários autores apresentam diferentes estruturas, tanto em número de fases como no conteúdo de cada fase. Estas diferenças dependem, de maneira geral, do tipo do processo e das necessidades de cada caso.

Hameri e Nihtilä (1998) apud Guerrero (2002) propõe um ciclo de vida de produto para projetos “*one-of-a-kind*” resultado da análise de três ciclos compostos das seguintes fases: conceitual, projeto, manufatura e operação.

Já Wheelright e Clark (1992), apresentam uma seqüência clássica de quatro fases: desenvolvimento do conceito, planejamento do produto, engenharia do produto e processo, e produção piloto e crescimento da produção.

Uma tendência que tem prevalecido nos últimos anos mostra a expansão do escopo PDP para a inclusão de atividades ligadas ao planejamento estratégico da empresa, tradicionalmente desconectadas do PDP, e atividades de acompanhamento de todo o ciclo de vida do produto, da produção à retirada do produto do mercado (ROZENFELD et al. 2000) (ver Figura 3).



Figura 3 - Atual Escopo do PDP (ROZENFELD, 2000)

### 2.3 Modelo de referência em Desenvolvimentos de Produto

Um modelo é uma representação de algo. É uma abstração da realidade expressa de uma semântica definida previamente por um método de modelagem específico. (VERNADAT, 1996 *apud* AMARAL, XXXX).

Para representar um processo de negócio, um modelo deve descrever todos os elementos desse processo: atividades realizadas; informações de entrada e saída de cada atividade; os responsáveis pela sua execução, e os recursos utilizados. Todas as inter-relações entre esses elementos também precisam ser especificadas dentro do modelo. (ZANCUL, 2000).

Vernadat (1996) afirma que um modelo de empresa não é necessariamente um elemento único, podendo ser constituído de um conjunto organizado de modelos complementares e inter-relacionados, descrevendo diferentes aspectos da empresa.

Segundo Keller e Teufel (1998) *apud* Amaral, (2001), os modelos podem ser caracterizados entre modelo específico, quando descreve o processo de negócio de uma empresa específica, ou como modelo de referência, quando tem uma aplicação mais ampla e geral, podendo ser aplicado a mais de uma empresa, a um setor específico ou a toda a indústria, podendo-se classificar os modelos de referência em modelos de referência de setor, de sistemas de informação e de empresas.

Dentro da área de desenvolvimento de software, Avison e Fitzgerald (2002) *apud* Vidgen et al. (2002), pag. 3, definem metodologia como:

“Uma coleção de procedimentos, técnicas, ferramentas e documentação que irá ajudar os desenvolvedores de sistemas nos seus esforços para implementar um novo sistema. Uma metodologia consistirá de fases, contendo por sua vez subfases, que irão guiar o desenvolvedor na escolha de técnicas apropriadas para cada estágio do projeto e ajudá-lo a planejar, gerenciar e avaliar projetos de sistemas de informação (tradução nossa).”

### **2.3.1 O Modelo de Referência desenvolvido pela comunidade PDPNet**

Os subprocessos mostrados na Figura 2 (com exceção do sub-processo “Parcerias e Supply Chain”, que foi sugerido pelo autor deste trabalho), se encontram no modelo de referência desenvolvido por um grupo de pesquisadores, membros da comunidade PDPNet ([www.pdpnet.org.br](http://www.pdpnet.org.br)). Esta comunidade de prática, segundo a definição utilizada por Mosconi et al (2002), foi formada por pesquisadores da área de desenvolvimento de produtos da Universidade de São Paulo (USP), da Universidade Federal de São Carlos (UFSCar) e da Universidade Federal de Santa Catarina (UFSC) com o objetivo de integrar seus conhecimentos sobre o Processo de Desenvolvimento de Produtos, eliminando visões parciais que costumam existir no desenho de um processo complexo como o de DP (ROZENFELD et al. 2003).

O modelo do PDP foi uma das primeiras atividades desenvolvidas pela comunidade de prática formada pela comunidade PDPNet, sendo este um modelo de referência único sobre DP. O modelo integra as diversas visões sobre o tema, de acordo com as diferentes especialidades de cada um dos três grupos, sendo desenvolvido após anos de trabalho conjunto (ROZENFELD et al. 2003).

A primeira dificuldade dos pesquisadores foi a forma de representação do modelo. Esta deveria atender os seguintes requisitos:

- fácil compreensão por todos;
- facilidade de troca e manipulação dos arquivos do modelo;
- facilidade de leitura; e
- conter apenas as dimensões essenciais.

As dimensões iniciais foram definidas como sendo atividades, informações de entrada, informações de saída, métodos e ferramentas. Estas dimensões foram dispostas inicialmente em uma planilha, como mostra a Figura 4:

Macro-fases	Fases	Etapas	Atividades	Tarefas	Informações de Entrada	Informações de Saída	Métodos e Ferramentas
Pré-desenvolvimento							
Desenvolvimento							
Pós-desenvolvimento							

Explicação do conteúdo

Figura 4: modelo de Referência da comunidade PDPNet. (ROZENFELD et al. 2003).

Quanto ao conteúdo do modelo, ele foi dividido em três fases, como mostrado na Figura 4. Esta representação tem suas desvantagens, entre as quais a disposição das atividades em linha, que impede a sua representação na linha do tempo, dando uma falsa impressão de ordem cronológica e serial na ocorrência destas.

A representação final desse modelo se deu na forma de livro, onde se descrevem todas as macro-fases, fases, atividades e tarefas do processo, em ordem sequencial, incluindo explicações de conceitos, métodos ferramentas, além de comentários e *insights* vindos da experiência do grupo em projetos de parceria no desenvolvimento de produtos com empresas de ponta. (ROZENFELD et al. 2006)

Este modelo foi usado como referência base para a realização deste trabalho e vale, portanto, descrever brevemente o modelo nesta dissertação:

O modelo é voltado principalmente para desenvolvimento de produtos como bens de consumo duráveis ou de capital, embora seja genérico o suficiente para ser adaptável a outros cenários, em caso de necessidade. A facilidade desta adaptação varia ao longo do modelo de acordo com o nível de abstração da de cada fase (ROZENFELD et al. 2006).

O modelo trata de um escopo bastante amplo de desenvolvimento de produtos, acompanhado todo o seu ciclo de vida, se dividindo em 3 macro-fases: pré-desenvolvimento, desenvolvimento e pós-desenvolvimento. As primeira e a última são

mais genéricas (e portanto mais fáceis de adaptar). A fase de desenvolvimento é mais detalhada, descrevendo suas atividades considerando o foco do modelo (bens de consumo duráveis e bens de capital), além de aspectos tecnológicos, e processos de fabricação inerentes a esse.

A macro-fase de pré-desenvolvimento faz a ligação do processo de planejamento estratégico da empresa com o seu processo de desenvolvimento de produtos, identificando (ou se antecipando a) as necessidades do mercado e alinhando sua estratégia (e seu portfólio de produtos) a esta realidade. O *deliverable* principal desta macro-fase é, além do alinhamento do portfólio de produtos, a Declaração de escopo e o Plano de projeto inicial de um produto, que por sua vez será desenvolvido nas fases seguintes.

A macro-fase de desenvolvimento se inicia com o desenvolvimento do plano de projeto elaborado na fase anterior. É com base neste plano que se levantam as informações relevantes ao produto e seu mercado, se desenvolvem conceitos e soluções para os requisitos de produto e se realiza todo o desenvolvimento detalhado do produto e processo de fabricação. Esta fase termina com o lançamento do produto no mercado.

A macro-fase de pós-desenvolvimento acompanha o produto no mercado, avaliando sua performance e levantando informações que possam a levar a mudanças no próprio produto ou em outros produtos do portfólio da empresa. Esta fase termina com a retirada do produto do mercado. A Tabela 1 contém um resumo da estrutura do modelo:

Tabela 1: números de tarefas detalhadas por fase do modelo

<b>Macro fase</b>	<b>Fase</b>	<b>Tarefas detalhadas</b>
Pré-desenvolvimento	Planejamento do Projeto	36
	Planejamento Estratégico de Produtos	40
Pré-desenvolvimento Total		76
Desenvolvimento	Lançamento do Produto	81
	Preparação da Produção do Produto	58
	Projeto Conceitual	54
	Projeto Detalhado	109
	Projeto Informacional	57

Desenvolvimento Total		359
Pós-desenvolvimento	Acompanhar Produto e Processo	14
	Descontinuar Produto no Mercado	0
Pós-desenvolvimento Total		14
Processos de Apoio	Gerenciamento de Mudanças de Engenharia	19
	Melhoria (incremental) do PDP	17
Processos de Apoio Total		36
Transformação do PDP		16
Total Geral		501

## 2.4 Sistemas para Gestão do ciclo de vida de produtos (PLM)

O conceito de e-PLM foi proposto inicialmente pela AMR (Advanced Manufacturing Research) e adotado por diversos fornecedores de solução, como a SAP e a PTC. O CIMData apresentou posteriormente um conceito semelhante, com as mesmas características, denominado de Collaborative Product Definition Management (cPDM) (HORTA e ROZENFELD, 2001) (CIMDATA, 2001).

Mais recentemente, o CIMData (2002) adotou também o termo PLM, definindo-o como uma abordagem estratégica que utiliza um conjunto consistente de soluções de negócio para dar suporte à criação, gerenciamento, disseminação e uso colaborativo das informações do produto ao longo de seu ciclo de vida. Desta maneira esta abordagem integra pessoas, processos, sistemas e informações. É importante notar que PLM não é apenas uma tecnologia, mas uma abordagem de gerenciamento de toda a informação do produto ao longo do seu ciclo de vida, onde os processos de negócio são tão ou mais importantes quanto a informação gerenciada pelo sistema.

CIMData (2002) identifica três conceitos fundamentais dos sistemas PLM:

- “Acesso e uso das informações do produto de forma universal, segura, e gerenciável”;

- “Manutenção da integridade da definição e das informações dos produtos ao longo do seu ciclo de vida”;
- “Gerenciamento e manutenção dos processos de negócios usados para criar, gerenciar, disseminar e usar as informações do produto”. (tradução livre)

Os sistemas PLM coexistem com diversos outros sistemas corporativos e com estes se relacionam. No entanto, têm escopos distintos dentro do ciclo de vida do produto, como mostra a Figura 5:

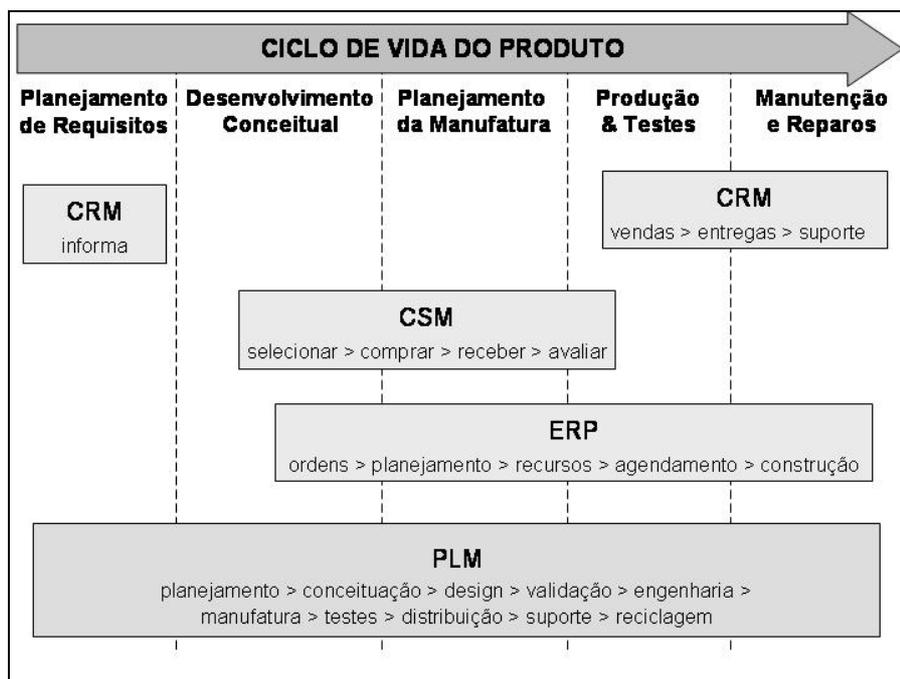


Figura 5: escopo dos principais sistemas corporativos (EDS, 2002)

### **3 A GESTÃO DE PROJETOS**

O Project Management Institute (PMI) é uma organização voltada para a melhoria da Gestão de Projetos, treinando profissionais e resumindo e divulgando melhores práticas em mais de 125 países, entre outras atividades. Segundo o PMI, um projeto pode ser definido como “um empreendimento temporário com o objetivo de criar um produto ou serviço único” (PMI, 2004).

A gestão de Projetos pode ser definida como a aplicação de conhecimento, habilidades, técnicas e ferramentas para projetar atividades que visem atingir ou exceder as necessidades e expectativas das partes envolvidas no projeto (PMI, 2004).

#### **3.1 Processos dentro da Gestão de Projetos**

Segundo PMI (2004), a Gestão de Projetos consiste basicamente em processos de iniciação, planejamento, execução, controle e encerramento do projeto (veja Figura 6). A iniciação considera os processos necessários para garantir que o projeto será executado. Os processos de planejamento visam planejar e manter um esquema de trabalho viável para se atingir os objetivos do projeto, envolvendo a determinação do escopo, a programação de atividades, o orçamento e os planos do projeto. A execução consiste em coordenar pessoas e recursos para executar o plano e envolve garantia da qualidade, distribuição de informações e seleção de fornecedores. Os processos de controle visam assegurar que os objetivos do projeto estão sendo atingidos, através da monitoração e avaliação do seu

progresso, controlando as mudanças, custos, qualidade e riscos. Por fim, o encerramento formaliza a aceitação do projeto ou fase, encerrando-o de forma organizada.

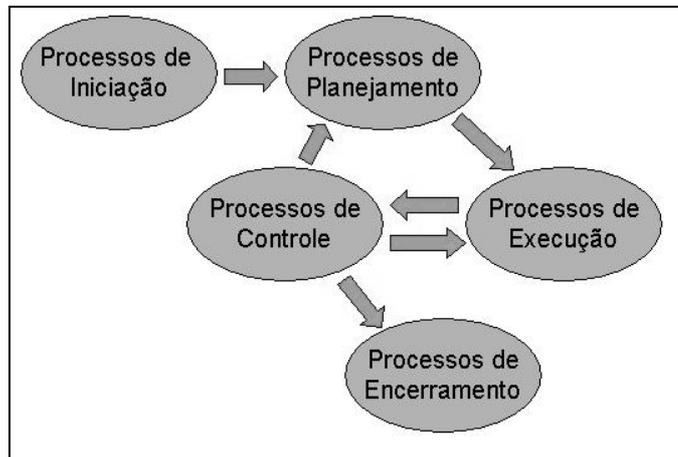


Figura 6: relações entre os processos de GP (PMI, 2004)

### 3.2 Áreas de conhecimento relacionadas à gestão de projetos

Diversos autores apresentam propostas diferentes para as áreas de conhecimento em GP, considerando diversas abordagens. Pretende-se utilizar neste trabalho a proposta do PMI, conforme apresentada no PMBOK – *Project Management Body of Knowledge*, que divide as áreas de conhecimentos em nove áreas distintas (vide Figura 7), as quais descrevem os conhecimentos de GP em termos dos processos que as compõe.

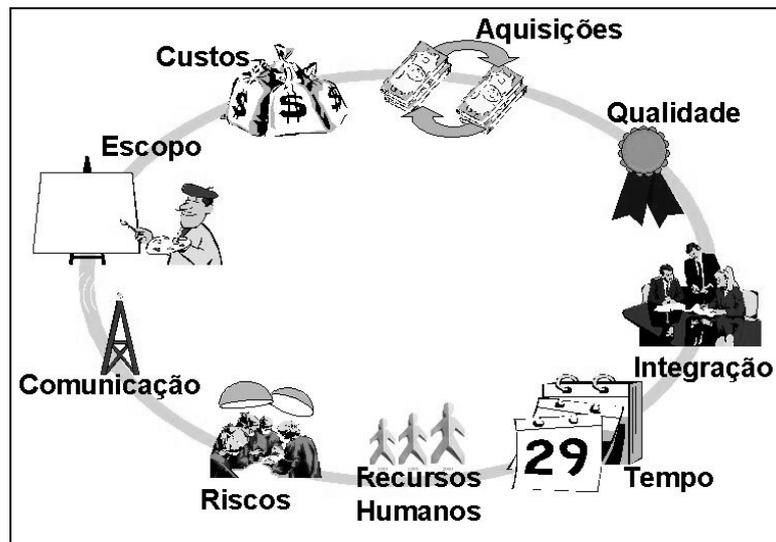


Figura 7: áreas de conhecimento em GP (adaptado de VALERI, 2000)

Outros autores apresentam divisões diferentes cujos aspectos mais interessantes serão incorporados neste trabalho dentro dessas nove áreas. Um exemplo é área de Gestão Ambiental, proposta por Valeriano (1998), que neste trabalho será incluída dentro da área de Gestão da Qualidade, tornando o trabalho mais genérico. Seguem abaixo as descrições das nove áreas:

- a) *Integração*: inclui elementos necessários para assegurar que os elementos de projeto estão coordenados apropriadamente. Esta área envolve fazer compensações entre objetivos e alternativas eventualmente concorrentes.
- b) *Escopo*: considera todos os processos necessários para garantir que o projeto contenha todo o trabalho necessário, e somente o trabalho necessário, para ser completado com sucesso.
- c) *Tempo*: assegura a conclusão dos trabalhos inerentes ao projeto no prazo planejado. Os principais processos que compõe esta área são a definição, o seqüenciamento e a estimativa de duração das atividades do projeto, bem como o controle da duração das atividades.
- d) *Custo*: assegura que o projeto será completado com as metas de custo e orçamento planejados, envolvendo o planejamento de recursos e orçamento.

e) Qualidade: assegura a satisfação das necessidades definidas no escopo. Compreende os processos de planejamento, controle e garantia da qualidade.

f) Recursos Humanos: assegura não só o uso ótimo do pessoal do projeto, mas também o relacionamento com clientes e fornecedores.

g) Comunicação: assegura a geração, armazenamento e disseminação das informações inerentes ao projeto.

h) Riscos: assegura a identificação e análise dos riscos do projeto, gerando respostas capazes de eliminá-los ou minimizar seus efeitos negativos dentro do projeto, se possível.

i) Aquisições (Procurement): esta área inclui os processos para obtenção de bens e serviços necessários ao projeto.

### 3.3 Áreas de Conhecimentos versus Processos

Todas as áreas de conhecimento citadas anteriormente (item 3.2) se relacionam com o processo de Gestão de Projetos (descrito no item 3.1), como é visto na Tabela 2. Em cada uma das intersecções mostradas na figura são apresentados um ou mais processos. O PMBOK agrupa 39 processos dentro dos cinco grandes grupos de processo (iniciação, planejamento, execução, controle e encerramento) e dentro das nove áreas de conhecimento.

Tabela 2: áreas de conhecimento versus Processos de GP (PMI, 2004)

Processos versus Áreas de conhecimento	Iniciação	Planejamento	Execução	Monitoramento e controle	Encerramento
<b>4. Integração</b>	4.1. Desenvolver project charter 4.2. Desenvolver declaração preliminar de escopo do projeto	4.3. Desenvolver plano de gerenciamento do projeto	4.4. Dirigir e gerenciar execução do projeto	4.5. Monitorar e controlar trabalho do projeto 4.6. Controle integrado de mudanças	4.7. Fechar projeto

<b>5. Escopo</b>		5.1. Planejamento do escopo 5.2. Definição do escopo 5.3. Criar WBS		5.4. Verificação do escopo 5.5. Controle do escopo	
<b>6. Tempo</b>		6.1. Activity definition 6.2. Activity sequencing 6.3. Estimar recursos da atividade 6.4. Estimar duração da atividade 6.5. Desenvolver cronograma		6.6. Controle do cronograma	
<b>7. Custos</b>		7.1. Estimar custos 7.2. Estimar orçamento		7.3. Controle do custo	
<b>8. Qualidade</b>		8.1. Planejar qualidade	8.2. Realizar garantia da qualidade	8.3. Controle da qualidade	
<b>9. Recursos Humanos</b>		9.1. Planejar recursos humanos	9.2. Adquirir time de projeto 9.3. Desenvolver time de projeto	9.4. Gerenciar time de projetos	
<b>10. Comunicações</b>		10.1. Planejar comunicação	10.2. Distribuir informação	10.3. Realizar relatórios 10.4. Gerenciar Interessados	
<b>11. Riscos</b>		11.1. Planejar gerenciamento do risco 11.2. Identificação de riscos 11.3. Análise qualitativa de riscos 11.4. Análise quantitativa de riscos 11.5. Planejar resposta a risco		11.6. Monitorar e controlar riscos	
<b>12. Aquisições</b>		12.1. Planejar compras e aquisições 12.2. Planejar contratos	12.3. Solicitar respostas dos vendedores 12.4. Selecionar vendedores	12.5. Administrar contratos	12.6. Fechar contrato

### **3.4 Gestão colaborativa e distribuída de Projetos**

Entende-se por Gestão colaborativa e distribuída de Projetos, o trabalho conjunto de equipes espalhadas geograficamente. Esta configuração de projetos representa riscos, mas também traz benefícios importantes para a organização. Para minimizar os riscos e maximizar os benefícios, pode-se fazer uso de diversas ferramentas e métodos. Estas questões são apresentadas em maiores detalhes a seguir.

O desenvolvimento colaborativo de produtos pode acontecer de várias formas, envolvendo diversas equipes de uma mesma organização ou de organizações distintas. De modo geral, é possível classificá-lo em horizontal, quando envolve potenciais competidores ou parceiros de tecnologia, e em vertical, quando integra clientes e fornecedores da mesma cadeia de valor (GUERRERO, 2002).

Desta forma, práticas que garantam a transparência do projeto, envolvendo clientes e fornecedores no desenvolvimento, tendem a melhorar o processo de desenvolvimento como um todo, fazendo de clientes e fornecedores uns dos principais parceiros e *stakeholders* de um projeto de Desenvolvimento de Produtos. Clark e Fujimoto (1991) analisam estas práticas e apresentam as suas vantagens e desvantagens. Os autores concluem que o envolvimento dos fornecedores diminui o tempo de desenvolvimento e melhora a produtividade, porém pode comprometer a qualidade do produto.

#### **3.4.1 Sistemas para Gestão de Projetos**

Segundo Badiru (1996), sistemas para gestão de projetos são ferramentas computacionais que auxiliam o gerente de projetos em tarefas operacionais ou na tomada de decisões.

Segundo esse autor, existem diversos fatores importantes na avaliação de softwares para gestão de projetos, sendo eles: custo, necessidade, planejamento do projeto,

gerenciamento de recursos, rastreamento do progresso do projeto, geração de relatórios, facilidade de uso, suporte de ferramentas para diferentes análises e requerimentos de hardware.

Um grande número de procedimentos e métodos tem sido desenvolvido para melhorar a aplicação das práticas gerenciais dentro da empresa. Este crescimento deve muito à disponibilidade de softwares que torna possível uma rápida implementação das técnicas de GP. No entanto, não se deve esperar que o computador substitua o tradicional processo decisório. O computador deve ser usado somente como uma ferramenta de apoio para facilitar a implementação de técnicas e métodos. A principal vantagem do computador é a sua velocidade de processamento de análises quantitativas necessárias a uma grande variedade de processos.

O mesmo autor define seis etapas para o processo evolucionário dos softwares de GP. Da primeira geração (anterior à Segunda Guerra Mundial), passando pelos sistemas dedicados PERT/CPM, e chegando à geração atual, com novos conceitos em inteligência artificial e comunicação *online*, aliados aos conceitos passados, unindo as gerações passadas com projetos gráficos, geração de relatórios, planilhas e análises de custo. Na primeira geração, os programas de GP eram complexos, caros e de difícil uso. Atualmente, a tecnologia desses softwares abriu novas possibilidades. Sua crescente popularidade, aceitação e uso são evidenciados pelo grande número de programas comerciais recentemente introduzidos no mercado.

Mesmo entre as grandes empresas de software proprietários, encontramos muita variedade de funcionalidades, dependendo da origem do software. Empresas de vulto, especialmente no ramo de construção, tinham necessidade de programas que auxiliassem o agendamento de tarefas como identificação do caminho-crítico, mas diversos fabricantes de software (por exemplo Ártemis International Solutions, Businees Engien, Plan View, Primavera Systems, etc.) têm ampliado seus produtos com outras funções, oferecendo um melhor gerenciamento integrado de recursos, *tracking*, gerenciamento de custos (baseados em custos de capital e hora trabalhadas), *budgeting*, e análise de valor agregado (GARTNER, 2002).

Outro ramo de sistemas, chamados PSA (Professional Service Administration) desenvolveram funcionalidades de calendários, gestão do escopo, eficiência do serviço por horas trabalhadas, etc., para atender ao mercado de empresas de serviço (GARTNER, 2002).

Esta diversidade de funcionalidades e funções dos softwares de gestão de projetos torna difícil uma comparação crítica sobre os diversos softwares existentes. Analisando os critérios de Badiru (1996), vemos que as necessidades variam de acordo com a aplicação, as funcionalidades variam de acordo com o ramo e geração do software e todos esses afetam o custo.

No entanto, pode se encontrar atualmente listas de comparação de softwares (GARTNER, 2002), com atualizações anuais.

Dentro do grupo de pesquisa, se realizaram pesquisas para desenvolver critérios de avaliação de software. Moretti *et al.* (2005) apresenta uma lista estruturada de critérios de avaliação de sistemas de apoio à gestão do conhecimento, que permitem avaliação de sistemas por meio de um sistema de pontuação.

Um trabalho com metodologia semelhante foi realizado dentro do mesmo grupo de pesquisa, o qual avaliou diversos softwares livres de gestão de projetos, desenvolvendo uma série de 38 critérios para sua avaliação, separados em oito grupos (Calendário e agenda; Gestão de atividades; Gestão de recursos; Gestão de custos; Gestão de documentos / relatórios e impressão; Ferramentas de monitoramento; Gestão de múltiplos projetos e Ferramentas de comunicação / integração), como pode ser visto na Tabela 3.

Tabela 3: classificação das funcionalidades de softwares de gestão de projetos

COD	GRUPO	SUB-COD	SUB-GRUPO
1	Agenda e reuniões	1.1	Personalização de calendário
		1.2	Integração da agenda entre projetos
		1.3	Integração da visão da agenda por atividade e por recurso
		1.4	Auxílio no agendamento de reuniões online
2	Gestão de atividades	2.1	Determinação de hierarquia e precedências das atividades
		2.2	Criação / visualização de gráficos (PERT, Gantt, etc.)
		2.3	Criação / visualização de WBS
		2.4	Gates e Milestones
		2.5	Criação, customização e <i>enactment</i> de workflows
3	Gestão de recursos	3.1	Configuração de recursos (cadastrar, definir características, etc.)
		3.2	Relacionamento entre pessoa / recurso vs. Atividade/ deliverable
		3.3	Visualização de disponibilidade de recursos / pessoas
		3.4	Nivelamento de recursos
		3.5	Padrões para alocação de recursos / pessoas
		3.6	Criação de times de projeto
		3.7	Criação de papéis para membros do projeto
		3.8	Aquisição de recursos
		3.9	Listagem de especialistas
4	Gestão de custos	4.1	Custo total do projeto e por atividade
		4.2	Custo por recurso
5	Gestão de informação	5.1	Metadados para documentos
		5.2	Sistema de vault
		5.3	Controle de acesso
		5.4	Controle de versão
		5.5	Controle de alterações
		5.6	Associação de documento (documento, best practice, etc.) a atividades / deliverables
6	Ferramentas de monitoramento	6.1	Uso de base-lines diversas
		6.2	Monitoração de progresso de atividades
		6.3	Ferramentas de análise de valor agregado
		6.4	Mostrar real vs. Baseline em gráficos, etc.

<b>7</b>	Gestão de múltiplos projetos	7.1	Compartilhamento de recurso entre projetos
		7.2	Listagem dos projetos (com filtros)
		7.3	Ferramentas de gestão de portfolio
<b>8</b>	Ferramentas de comunicação e integração	8.1	Ferramentas de comunicação síncrona
		8.2	Ferramentas de comunicação assíncrona
		8.3	Sistemas de alerta
		8.4	Ambiente online de projetos compartilhado
		8.5	Comunicação com agente externos (clientes, consultores, fornecedores)

Uma vez que os critérios de avaliação foram desenvolvidos para avaliar as principais funcionalidades dos sistemas em questão, este trabalho utilizou os critérios da Tabela 3 como classificação inicial para a lista dos requisitos deste trabalho.

## 4 GESTÃO DE CONHECIMENTOS

*"If you think education is expensive... try ignorance."*

W. Bok

Parafrazeando o Sr. Bok, pode-se dizer que se você acha que conhecimento na sua empresa é caro, experimente a ignorância. Este conceito não é novo, como constatado por Litter (1995), Ayas (1996), Nonaka e Takeuchi (1997), entre outros autores.

De acordo com Nonaka e Takeuchi (1997), conhecimento é específico ao contexto e é relacional, podendo ser dividido em dois tipos:

- Conhecimento tácito: conhecimento pessoal, sendo de difícil formulação. É não estruturado e subjetivo, relacionado à experiência pessoal de cada membro da organização.
- Conhecimento explícito: conhecimento codificado, expresso e transmitido em linguagem formal e sistemática. Organizado de maneira estruturada, pode ser armazenado em documentos e sistemas computacionais.

Nonaka (1991) explora este conceito com o seguinte exemplo:

“Conhecimento tácito é profundamente enraizado em ações e comprometido com um contexto específico de um indivíduo – uma perícia ou profissão, uma tecnologia, produto ou mercado em particular, ou em atividades de um grupo de trabalho. Conhecimento tácito consiste parcialmente de habilidades técnicas – do tipo de habilidade informal, difícil de capturar o seu “know-how”. Um mestre carpinteiro depois de anos de experiência desenvolve uma riqueza de conhecimento “nas pontas dos dedos”, mas é frequentemente incapaz de articular os princípios científicos ou técnicos por trás de sua habilidade.”

Por esta razão, o autor argumenta que, do ponto de vista organizacional, uma empresa geradora de conhecimento deveria tentar transformar seu conhecimento tácito em explícito (NONAKA, 1991).

Grandes firmas demonstram diversos exemplos de conhecimento explícito, indicado por sistemas administrativos complexos, com procedimentos e controles. Empresas menores, por sua vez, dependem muito mais de conhecimento tácito, sugerindo a ausência total ou parcial de sistemas, métodos e práticas de gestão de projetos (CURIE, 2003).

No entanto, Swan et al. (1999) apud Curie (2003) afirmam que existe um paradoxo, pois se ferramentas gerenciais tendem a aumentar a efetividade com que o conhecimento é aproveitado, podem simultaneamente reduzir o potencial de criação de conhecimento de uma organização, engessando os processos criativos pelas “amarras” das normas e procedimentos, de forma que não se pode descartar o conhecimento tácito.

Assim, pode-se afirmar que o conhecimento gera-se por meio da interação social, colocando conhecimentos tácito e explícitos em contato. Em uma organização, a inovação surge dessa interação ou conversão do conhecimento (NONAKA e TAKEUCHI, 1997).

#### **4.1 Conhecimento no Processo de Desenvolvimento de Produtos**

Litter (1995) observa que por se apresentar em ambientes dinâmicos, o conhecimento sobre o processo de desenvolvimento de produtos é incompleto no começo e gradualmente vai aumentando, através de vários processos de aprendizagem. Drucker *apud* Rozenfeld et al (2000) diz que quanto mais se avança na era da informação, mais o conhecimento está se tornando um componente crítico do sucesso competitivo das empresas.

De acordo com Mahchrzak *apud* Liebowitz et al (2003), a gestão de conhecimento pode também ser usada para inovação e criação de novas idéias para produtos e serviços, o que é reforçado por Huang & Newell (2003), quando afirmam que times de desenvolvimento de produto têm seu foco na inovação e criatividade, aproveitando

conhecimentos anteriores para gerar conhecimentos novos que não existem na organização. Este conhecimento deve ser disseminado por todo o time conforme seja criado, promovendo discussão e consenso, antes de ser transformado em conhecimento explícito.

Liebowitz et al (2003) descrevem este ciclo de gestão de conhecimentos em quatro passos, como mostra a Figura 8:

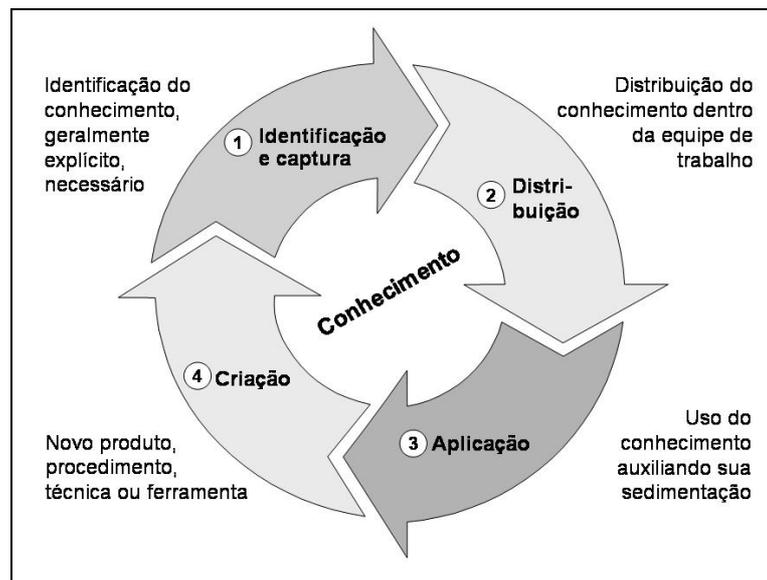


Figura 8: passos da gestão de conhecimento (LIEBOWITZ et al. 2003)

Uma vez que o conhecimento é identificado e capturado, é tipicamente distribuído com os demais membros do time de desenvolvimento de produto. Assim, este conhecimento é aplicado e internalizado dentro da nova situação que, por sua vez, cria um novo conhecimento, traduzido em uma nova técnica, ferramenta, procedimento ou produto.

Um aspecto importante encontrado na literatura é a importância do conhecimento qualitativo, muitas vezes “escondido” dentro das pessoas, de forma tácita (SHINDLER & EPLER, 2003), já que é mais difícil de explicitar. Uma razão para esta importância é que componentes numéricos não provêm respostas para questões cruciais do projeto de desenvolvimento. Respostas numéricas respondem a “o que”, “quando” e “quanto”, mas

raramente respondem ao “porque” ou ao “como”. Estas questões são normalmente respondidas por histórias, estudos de caso e relatórios descritivos de projetos.

Interessante notar que o conjunto de conhecimento sobre o próprio processo de DP pode ser registrado em um modelo de referência, uma vez que incorpora o conhecimento da organização sobre o processo, suas atividades, inter-dependências, restrições, métodos, ferramentas e padrões. A criação, sistematização, otimização, e reutilização de um modelo de referências para o processo constitui uma boa prática de Gestão de conhecimento dentro do PDP (ROZENFELD et al. 2006)

#### **4.2 Gestão do Conhecimento em Gestão de Projetos**

Se gerenciado corretamente, o conhecimento pode ser usado para reduzir o tempo de projeto e aumentar sua qualidade e a satisfação dos *stakeholders*. Considerar esta Gestão do Conhecimento, seja conhecimento explícito ou tácito, é um pré-requisito necessário para o sucesso de projetos nos ambientes dinâmicos de hoje. (LOVE et al. 2003).

De acordo com White e Fortune (2002), os três fatores críticos de sucesso em gestão de projetos mencionados por gerentes de projeto em suas pesquisas foram: objetivos / metas claros; suporte da alta gerência; fundos / recursos adequados. Iniciativas de gestão do conhecimento podem ajudar a melhorar a comunicação e prover mais informações e conhecimento para o gerente do projeto. Desta maneira, os objetivos e diretrizes do projeto devem ser melhor definidos e articulados, sendo repassados e seguidos pela equipe do projeto (LIEBOWITZ et al. 2003).

Segundo Shindler e Eppler (2003), o aprendizado sistemático de projetos permite às empresas, a longo prazo, desenvolver competências na realização de projeto que levam a uma vantagem competitiva sustentada. Os autores afirmam que a retenção sistemática de experiências dos projetos ajuda a empresa a comparar seus vários projetos e a documentar mais eficientemente mecanismos de solução de problemas.

Experiências, que são por definição parte das pessoas, não são geralmente parte da documentação do projeto e são raramente transferidas para outras pessoas da equipe

durante o desenvolvimento do projeto (ARGYRIS, 1999, *apud* SHINDLER & EPPLER, 2003). No entanto, essas experiências são acessíveis através de redes informais de comunicação, sem necessariamente passar pela informação codificada dentro da organização (BOWEN et al. 1994). É a famosa “conversa do cafezinho”...

Com o intuito de formalizar a captura de seu conhecimentos dentro do projeto, realizam-se workshops no término de um projeto; no entanto, essas sessões tendem a se concentrar em aspectos mais prementes no fim de um projeto de desenvolvimento de produto, como o desligamento de pessoas e recursos, transferência da responsabilidade do produto para a manufatura, etc. (MEREDITH & MANTEL, 1995). Esses encontros, quando acontecem, costumam não recuperar parte do conhecimento desenvolvido ao longo do projeto. Shindler & Epler (2003), através de pesquisas junto a diversas empresas interessadas em verificar métodos de aprendizado nos seus projetos e como esses afetavam a motivação dos membros do time na gestão de conhecimentos, identificaram várias razões para o que os autores chamaram de “amnésia” do projeto:

- Pressão de final de projeto (prazos, tarefas esperando os membros do time em suas respectivas áreas)
- Falta de vontade de aprender através do erros das pessoas envolvidas
- Relatos incompletos de experiências (modéstia em casos positivos, medo de sanções em casos negativos)
- Falta de conhecimento de métodos de relato de experiências
- Subestimação da importância da descrição de experiência para o próximo time
- Falta de procedimentos específicos para os relatos e metas para o cumprimento destas tarefas
- Esquecimento de experiências que não foram registradas anteriormente

Esses autores citam dois tipos de métodos para combater a amnésia: métodos baseados em processos, que se concentram em passos relevantes a serem realizados ao longo do projeto, e métodos baseados em documentos, que, por sua vez, focam em como

representar o conhecimento tácito de experiências do projeto de forma explícita, dentro dos mecanismos de armazenagem de conhecimento dentro da organização. As Tabela 4 e Tabela 5 descrevem os métodos levantados pelos autores.

Tabela 4: métodos baseados em processo para gestão de conhecimento em projetos (SHINDLER & EPLER, 2003)

	<b>Método</b>			
<b>Parâmetro</b>	<b>Revisão / auditoria</b>	<b>Pós-controle</b>	<b>Avaliação pós-projeto</b>	<b>Revisão pós-ação</b>
<b>Tempo da execução</b>	Depois do término do projeto ou no curso do projeto entre duas fases	Exclusivamente no término do projeto	Aproximadamente dois anos depois do fim do projeto	Durante o processo de trabalho
<b>Realizado por:</b>	Revisão: moderador Auditoria: Externo	Gerente do projeto	Unidade de avaliação pós-projeto (um gerente e assistentes)	Facilitador
<b>Participantes</b>	Time de projeto e terceiros envolvidos	Gerente de projetos (possível participação de outros membros do time)	Time de projeto e terceiros envolvidos no projeto	Time de projeto
<b>Objetivo</b>	Clarificar status, reconhecer previamente possíveis problemas e focar o time internamente	Servir como delimitação, em adição a uma finalização do projeto que foque no desenvolvimento isolado da conformidade do objetivo futuro do projeto	Aprender através dos erros. Transferir conhecimento para terceiros	Aprender através dos erros. Transferir conhecimento dentro do time de projeto
<b>Benefícios</b>	Melhoria da disciplina do time. Prevenção de pontos fracos e validação de estratégias	Resultado é um documento formal, que considera o escopo e objetivos do projeto, metas quantitativas e <i>milestones</i> e metas de orçamento. Também contém uma avaliação do resultado do projeto e recomendação de melhorias futuras	Geração de melhores práticas para projetos de larga escala, melhorias no processo de previsão e propostas de mudança	Reflexão imediata sobre as próprias atividades para melhoria de atividades futuras

<b>Modo de interação</b>	Reuniões presenciais	Modo “não cooperativo” de registro de experiências. Análise de relatórios de projeto existentes, documentos, “milestones” e metas de orçamento, identificando diferenças entre esforço estimado e real	Análise de documentos, reuniões presenciais	Reuniões do time de desenvolvimento
<b>Codificação</b>	Usualmente em relatórios sem circulação pré-definida com objetivo principal de transferir conhecimento	Usualmente em relatórios sem circulação pré-definida com objetivo principal de transferir conhecimento	Livretos	Flipcharts

Shindler & Epler (2003) ainda identificam métodos baseados em documentos para registro de conhecimentos ao longo do projeto, como mestrado na Tabela 5:

Tabela 5: métodos baseados em documentos para gestão de conhecimento em projetos

Parâmetro	Método		
	Micro-artigo	Histórias de aprendizado	RECALL
<b>Tamanho e detalhamento</b>	Até uma página	Entre 20 e 30 páginas	Várias telas
<b>Suporte de TI</b>	Possível, mas não necessário	Não necessário	Obrigatório (interface com base de dados)
<b>Participantes</b>	Não definido explicitamente, foco no autor do documento	Indivíduos e times	Usuário individual
<b>Suporte de pessoas dedicadas</b>	Autor, revisor	“Historiador do projeto” necessário ao longo do processo todo	Grupo de trabalho para revisão
<b>Frequência</b>	Regularmente, conforme necessidade	Uma única vez, no fim do projeto	Conforme necessidade
<b>Anonimato</b>	Não	Sim	Não
<b>Registro /</b>	Papel / banco de	Casos	Banco de dados /

<b>distribuição</b>	dados / intranet	acompanhados de workshops de apresentação	intranet
---------------------	------------------	---	----------

Os autores ainda afirmam que a coleta de experiências ao longo do projeto é normalmente requisitada, tanto na prática com na literatura, no fim do projeto, embora já se encontrassem afirmações na literatura que “[...] o aprendizado dever ser gerenciado juntamente com o projeto e deve ser integrado à gestão de projetos com a prática padrão” (AYAS, 1996).

Shindler e Epler (2003) reforçam a necessidade de um aprendizado contínuo ao longo do projeto, por meio de revisões regulares. Os autores propõem que critérios e indicadores de registro de experiências sejam incluídos nos Phase Gates entre as fases do projeto e listam as principais vantagens de revisões periódicas versus revisões no fim do projeto são:

- Eventos recentes são lembrados mais facilmente, especialmente em projetos longos
- Facilita o registro/ detalhamento das informações, versus uma revisão acumulada dos fatos
- Facilidade de reunir o time de projeto é maior durante o andamento do projeto, especialmente membros do time externos a organização

Ainda se poderiam citar outras vantagens do registro de lições aprendidas ao longo do projeto: o uso desse conhecimento dentro do mesmo projeto, facilitando correção / recorrência de erros recentes; Evita aspectos negativos do registro de experiências negativas, incentivando os membros do time a reportar erros o invés de escondê-los.

No modelo de referência em desenvolvimento de produto utilizado neste trabalho, a gestão de conhecimento está presente em todas as atividades, ainda que de maneira implícita (ROZENFELD et al. 2006). Este registro informal e implícito de conhecimentos ao longo do processo se identifica com o processo de revisão pós-ação, de Shindler e Epler (2003).

Adicionalmente, uma atividade genérica se repete explicitamente no final de cada fase do modelo: “documentar as decisões tomadas e registrar lições aprendidas”. Esta atividade é formaliza a necessidade de registro do conhecimento adquirido ao longo da fase, se identificando com o processo de revisão/auditoria da Tabela 4.

## 5 DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO

“Put three stakeholders in a room and ask them what kind of system they want.

You’re likely to get four or more different opinions”

Autor desconhecido

Vidgen et al (2002) definem sistemas de informações como um conjunto de componentes inter-relacionados – pessoas, procedimentos e tecnologias – que atuam em conjunto para coletar, processar, armazenar e distribuir informação para apoiar o controle, a tomada de decisões e o gerenciamento de organizações.

Vale notar que nesse capítulo trata-se de sistemas de informação *baseados no computador*. Embora a informação dentro de uma organização flua por diversos meios, formais e informais, trata-se aqui de como um software auxilia a organização a gerenciar seus conhecimentos explícitos, segundo a definição de Nonaka e Takeuchi (1997).

### 5.1 Metodologia de desenvolvimento de sistemas

Da mesma maneira que se encontram na literatura diversos modelos de processos de desenvolvimento genérico de produtos, a literatura de desenvolvimento de software apresenta diversas metodologias e modelos para ajudar desenvolvedores a realizar o seu

trabalho. Abaixo se apresentam dois desses modelos: o modelo clássico em cascata, que, apesar de não ser muito utilizado atualmente, apresenta uma divisão bastante didática das atividades envolvidas, e o modelo evolucionário ou em espiral, caracterizado pela iteratividade com o usuário, de modo que o sistema evolua gradualmente para versões mais complexas

### **5.1.1 Modelo clássico em cascata**

O ciclo de vida em cascata para desenvolvimento de sistemas subdivide o processo em estágios formais, onde as informações de saída de um estágio são as informações de entrada do estágio seguinte, passando pelos estágios de viabilidade, onde é avaliado o escopo do sistema, passando pela definição de requisitos. Estes requisitos, uma vez aprovados pelo usuário final, são especificados, codificados e testados (VIDGEN et al. 2002).

Este processo, também chamado de processo clássico, sugere uma abordagem sistemática e seqüencial e é usado de situações onde o problema é bem compreendido e o fluxo de trabalho é razoavelmente linear, típica de adaptações bem definidas ou melhorias incrementais em sistemas pré-existentes. No entanto, existem alguns problemas com esta abordagem: projetos raramente seguem o fluxo seqüencial proposto pelo modelo; usuários dificilmente conseguem determinar claramente todos os requisitos logo no início; uma versão do sistema só estará disponível para avaliação no fim do processo, quando correções custam muito caro (PRESSMAN, 2005).

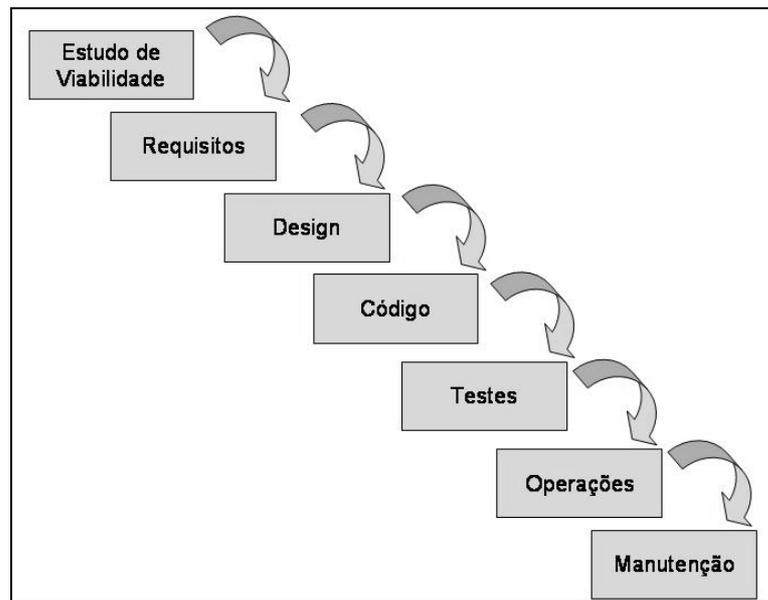


Figura 9: modelo de Desenvolvimento de Software em Cascata (adaptado de Vidgen et al. 2002 – tradução própria)

### 5.1.2 Modelo evolucionário em espiral

No modelo em espiral, mostrado na Figura 10, o sistema é desenvolvido a partir de protótipos que são refinados por meio de um sistema de feedback, alterando o próprio aplicativo, com modificações que se propagam como uma pedra arremessada na superfície de um lago. Todas as melhorias são, portanto, reativas ao feedback do usuário. O ponto focal desse modelo é que mudanças são esperadas desde o início do desenvolvimento e consideradas normais. Nesse processo, as constantes e pequenas mudanças, mesmo mais frequentes, têm um custo e tempo de implementação menores (VIDGEN et al. 2002).

Boehm (2001) *apud* Pressman (2005), descreve o modelo em espiral da seguinte forma:

“O modelo de desenvolvimento em espiral (...) que possui duas características diferenciadores principais: uma é a abordagem cíclica para crescer de modo incremental o nível de definição e implementação enquanto reduzem o nível de risco. A segunda é um conjunto de marcos de projeto para garantir aos interessados o comprometimento para uma solução viável e mutualmente satisfatória.

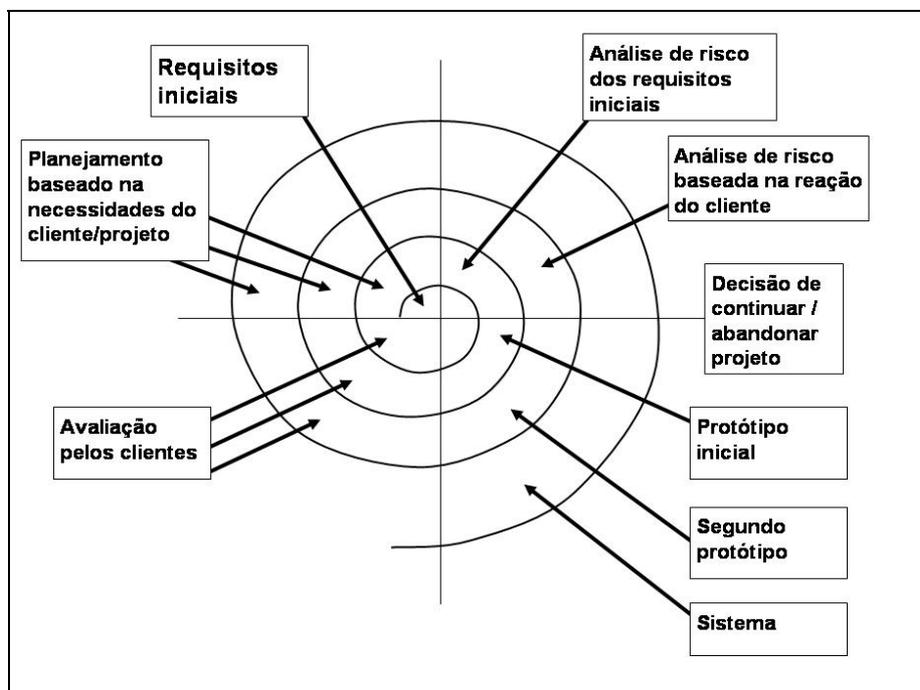


Figura 10: modelo evolucionário em espiral (adaptado de Vidgen et al. 2002)

## 5.2 Engenharia de requisitos

“Facts do not cease to exist because they are ignored”

Aldous Huxley

Requisitos definem os serviços que um sistema deveria proporcionar e delimitam o escopo e as restrições da operação do sistema. Os requisitos podem ser divididos da seguinte maneira (KOTONYA & SOMMERVILLE, 1998):

- Requisitos gerais definem de maneira geral o que o sistema deveria fazer;
- Requisitos funcionais definem parte da funcionalidade do sistema;
- Requisitos de implementação definem como o sistema deve ser implementado;
- Requisitos de performance especificam a performance minimamente aceitável para o sistema;
- Requisitos de usabilidade determinam o tempo máximo de espera para demonstração do uso do sistema.

De acordo com Presman (2005), engenharia de requisitos auxilia os engenheiros de software a compreenderem melhor o problema que tentam resolver, por meio de um conjunto de tarefas que levam ao entendimento do impacto do sistema no negócio, o que os interessados querem e como o usuário final interagirá com o sistema.

Requisitos são definidos durante os estágios iniciais dos desenvolvimento de sistemas e alterados / refinados ao longo do processo (veja Figura 9 e Figura 10).

Kotonya e Sommerville (1998) definem um processo de quatro etapas para a engenharia de requisitos, como pode ser visto na Figura 11.

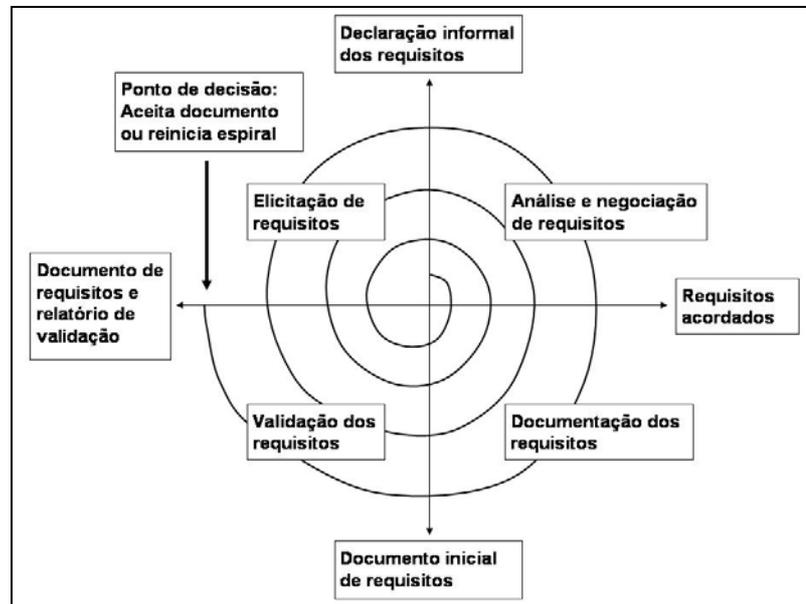


Figura 11: processo de engenharia de requisitos (Kotonya & Sommerville, 1998)

- Elicitação: “descoberta” dos requisitos, por meio de entrevistas com os interessados, consulta a documentação do sistema, conhecimento tácito e explícito sobre o domínio de conhecimento em questão;
- Análise e negociação: análise detalhada dos requisitos, envolvendo negociação com diferentes interessados, quando requisitos entram em conflito, informações são incompletas ou requisitos são incompatíveis com orçamento ou tempo de desenvolvimento do sistema;
- Documentação: documentação dos requisitos acordados, formalizando-os em um nível de detalhe que todos os interessados consigam entender, geralmente por meio de linguagem natural e diagramas; e

- **Validação:** verificação dos requisitos em termos de consistência e completude antes que os requisitos sejam utilizados no desenvolvimento do sistema.

Pressman (2005), distribui o processo de uma maneira mais granular, dividindo a fase de análise e negociação em duas e criando uma fase inicial de concepção antes da fase de elicitação. Essa fase inicial tem como objetivo ter um conhecimento mais geral do sistema, de negócio e da interação com os usuários do sistema. Esta distribuição mais granular tem se tornado uma tendência nas últimas publicações (PRESSMAN, 2005; SOMMERVILLE, 2006).

Abaixo seguem as principais fases do processo de engenharia de requisitos, segundo Pressman, 2005.

### **5.2.1 Concepção**

O objetivo desta fase é dar o “kick-off” no projeto de desenvolvimento de software, por meio das seguintes iniciativas:

- Identificação dos interessados: cada interessado tem sua visão própria do sistema, quais benefícios ele trará, qual a urgência do processo de desenvolvimento, quais os riscos envolvidos, entre outros. Identificar estas pessoas é fundamental para direcionar as perguntas necessárias para compreensão do ambiente de desenvolvimento
- Reconhecimento de múltiplos pontos de vista: diferentes interessados têm diferentes pontos de vista. Por exemplo, marketing pode estar interessado em funções que excitarão o mercado potencial do produto, enquanto a

controladoria estará preocupada com o orçamento do projeto. Usuários finais de diferentes áreas da empresa podem querer interfaces semelhantes (ou totalmente diferentes) às de sistemas pré-existentes. O trabalho desta fase é coletar todos esses requisitos de maneira organizada, independentemente de conflitos, que serão resolvidos posteriormente

- Colaboração: nesta fase deve-se identificar pontos de harmonia e conflito entre o conceito geral do sistema para diversas áreas, para garantir um conjunto de requisitos iniciais minimamente aceitável para todos os envolvidos. Pode-se realizar esse processo por votação, priorização, ou por apontamento de um campeão para um determinado grupo de requisitos, que teria a palavra final ou o voto de Minerva.
- As primeiras perguntas: busca-se nesta fase, por meio de perguntas livres de contexto, para cada interessado, levantar com pouco mais de detalhes quanto ao ambiente de desenvolvimento. Alguns exemplos:
  - Quem solicitou o desenvolvimento deste sistema?
  - Quem será o usuário final desta solução?
  - Qual o benefício econômico gerado pelo sistema?
  - Como você caracterizaria um “bom resultado” deste desenvolvimento?
  - Quais os problemas o novo sistema solucionará?
  - Você pode me descrever o ambiente em que a solução será usada? (número de usuários, uso diário / semanal, crítico, urgente...)

- Você poderia me conseguir material adicional sobre o ambiente / processo de negócio envolvido?
- Existe alguém mais que deveria estar envolvido neste processo, mas não está?

### 5.2.2 Elicitação

O método de questionamento, importante no processo de concepção não tem obtido muito sucesso no processo mais detalhado de elicitação, onde formatos que combinam elementos de solução de problemas, elaboração, negociação e especificação devem ser usados (PRESSMAN, 2005).

Kotonya e Sommerville (1998) apresentam uma lista de técnicas para serem realizadas em reuniões de elicitação de requisitos:

- Partição: organização do conhecimento como relações de agregação, onde o conhecimento do requisito é descrito em termos de suas partes. Por exemplo, em um sistema de viagem, uma reserva pode ser descrita como número do voo, data e dados pessoais do passageiro.
- Abstração: organização do conhecimento que relaciona entidades específicas com suas formas abstratas. Por exemplo, “Passageiro” pode ser homem, mulher, adulto ou criança.
- Projeção: organização do conhecimento de diversas perspectivas ou pontos de vista. Por exemplo: a reserva de um voo pode ser vista sob o ponto de vista do passageiro, do agente de viagens, da operadora no check-in, etc.

No final das reuniões diferentes técnicas podem ser usadas para priorização dos requisitos levantado. Pressman (2005), cita o QFD (Quality function deployment) como uma delas, como uma maneira de trazer de volta as necessidades do consumidor para o foco da discussão e priorizar os requisitos. O autor ainda classifica os requisitos oriundos de uma priorização via QFD de três maneiras:

- Requisitos normais: refletem objetivos e metas definidas para um produto ou sistema, com classificadores (para satisfazer o interessado, é necessária a presença deste requisito).
- Requisitos esperados: são requisitos implícitos ao produto ou sistema, sendo tão fundamentais que os clientes não necessariamente os explicitam.
- Requisitos impressionantes: são requisitos que refletem funcionalidades que vão além do esperado de um sistema “padrão”.

Kotonya e Sommerville (1998) citam o uso de duas outras ferramentas para o processo de análise dos requisitos elicitados:

- Checklists: lista de perguntas, com até 10 questões, para confrontar os requisitos e garantir que nenhum erro “óbvio” está “escapando”, padronizando-os em forma e conteúdo; e
- Matrizes de interação: matrizes cujo objetivo é descobrir interações entre requisitos, ressaltando conflitos e sobreposições.

Com a elicitação dos requisitos, um conjunto de funções e características do sistema começa a se materializar, mas é difícil mover para as atividades mais relacionadas ao desenvolvimento do software antes que o time de desenvolvimento entenda como essas

funções e características serão utilizadas em um série de casos, por diferentes usuários. Para fazer isso, deve-se gerar cenários (também chamados de caso de uso) que demonstram o uso do sistema a ser construído (PRESSMAN, 2005)

Kotonya e Sommerville (1998) lista quais atributos mínimos um caso de uso deve possuir:

- Uma descrição do estado do sistema antes do início do cenário;
- O fluxo normal dos eventos do cenário;
- Exceções ao fluxo normal de eventos;
- Uma descrição do estado do sistema após o término do cenário

### **5.2.3 Elaboração**

Casos de uso são cenários, que contam histórias estilizadas sobre como um usuário final (representando diferentes papéis possíveis) interage com o sistema em diferentes situações. A história pode ser em texto narrativo, uma descrição das interações ou representada em diagramas (PRESSMAN, 2005).

Além disso, cenários ajudam a revelar possíveis interações entre sistemas e revelam funcionalidades que antes não se julgavam necessárias (KOTONYA & SOMMERVILLE, 1998), sendo parte importante de alguns métodos de modelagem orientadas a objeto (FOWLER & SCOTT 1997; BOOCH, RUMBAUGH E JACOBSON, 1999).

O Primeiro passo em escrever um caso de uso é definir quem são os atores que estarão envolvidos na história. Uma vez definidos os atores, os casos de uso podem ser desenvolvidos (PRESSMAN, 2005).

#### **5.2.4 Negociação**

Infelizmente, uma vez listados e detalhados os requisitos, muitos se encontram em conflito. Inicia-se, portanto, um processo de negociação entre interessados, onde é necessário balancear funcionalidade, performance, tempo e características conflitantes originadas por desejos e necessidades de diferentes áreas da organização.

Boehm (1998) *apud* Pressman (2005) definem três atividades no processo de negociação:

- Identificação de dos principais interessados;
- Determinação das “situações vencedoras” de cada interessado
- Negociar com as “situações vencedoras” de cada interessado para reconciliá-las em uma situação “ganha-ganha”

Além dessas atividades, podemos citar também o intercâmbio de informações, para cada interessado entenda as razões dos demais

#### **5.2.5 Especificação**

Especificação envolve a documentação dos requisitos de forma que todos os principais interessados tenham condições de entendê-los, contendo uma descrição dos serviços proporcionados pelos sistema, restrições, propriedades, integração com sistema pré-existentes, interface com os usuários entre outros.

Kotonya e Sommerville (1998) listam os principais tópicos de um documento de requisitos: prefácio, glossário, requisitos gerais, arquitetura do sistema, especificações de hardware e requisitos de disponibilidade, confiabilidade e performance.

### **5.2.6 Validação**

Validação consiste em verificar o documento de requisitos quanto a consistência, completude e precisão, confrontando os requisitos entre si e contra o conhecimento e padrões da organização, procurando também identificar problemas de qualidade, requisitos mal escritos ou ambíguos, erros em modelos e cenários e problemas a serem resolvidos (KOTONYA & SOMMERVILLE, 1998).

Pressman (2005) apresenta um série de perguntas para examinar os requisitos no processo de validação:

- Os requisitos estão descritos claramente? Eles podem ser mal interpretados?
- A fonte dos requisitos é identificada? A descrição final do requisito foi confrontada com a fonte original?
- Os requisitos são relacionados de maneira quantitativa?
- Quais os outros requisitos que se relacionam com este requisito? Como? Essas relações estão claramente anotadas via alguma matriz de referência ou outro mecanismo?

### 5.3 Análise e design orientados a objeto

Este texto não pretende se estender muito na teoria por trás da análise e projeto orientado a objeto (Object Oriented Analysis and Design – OOA/OOD), mas apenas registrar uma breve noção sobre o assunto para que se possa acompanhar melhor o andamento do trabalho.

A essência da análise e do projeto orientados a objeto é enfatizar a consideração de um domínio de problema e uma solução lógica, segundo a perspectiva de objetos (coisas, conceitos e entidades) (LARMAN, 1999, p. 30).

O autor separa a análise e projeto orientado a objetos nas seguintes fases, como se vê na Figura 12.

- Análise orientada a objetos: ênfase na descoberta e na descrição dos objetos – ou conceitos – no domínio do problema.
- Projeto orientado a objeto: ênfase na definição dos elementos lógicos (em termos de software).
- Construção orientada a objeto: os elementos definidos no projeto são implementados em uma linguagem específica de programação orientada a objeto, como C++, Java, Smalltalk ou Visual Basic.

O autor ainda inclui no seu ciclo de desenvolvimento outras fases, como Refinar Plano, Sincronizar Artefatos e Testar, mas, por critérios de prioridade e importância, elas não foram descritas.

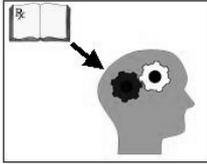
Grandes atividades da análise orientada a objetos			
FASE	1. Análise (Investigação do Problema)	2. Projeto (Solução Lógica)	3. Construção (Código)
REPRESENTAÇÃO		<b>Objeto: Livro</b> <b>Atributos:</b> - Autor - Ano publicação -Etc. <b>Procedimentos:</b> - Ler() - Editar() - Imprimir()	<pre>public class Livro {     Public void imprimir();     Private String titulo     Private integer Ano     Private integer Paginas }</pre>
PRINCIPAIS DELIVERABLES	<ul style="list-style-type: none"> <li>•Definição Raiz</li> <li>•Casos de uso essenciais</li> <li>•Modelo Conceitual</li> <li>•Diagramas de Seqüência</li> </ul>	<ul style="list-style-type: none"> <li>•Casos de uso reais</li> <li>•Modelo de classes de projeto</li> <li>•Esquema de banco de dados</li> <li>•Interfaces de usuários</li> </ul>	<ul style="list-style-type: none"> <li>•Código</li> </ul>

Figura 12: grandes Atividades em OOA / OOD, segundo Larman (1999)

Vale notar que esta divisão em fases tem intuito meramente didático, podendo e devendo haver sobreposição entre as fases.

Mais detalhes sobre casos de uso, modelo conceitual, modelo de classe, etc., podem ser vistos ao longo deste capítulo.

### Objetos

Vidgen et al (2002) apresentam uma definição bem simples de objeto: simplesmente “coisa”. *Qualquer* coisa que tenha interesse para o domínio do problema a ser estudado. No entanto, o autor atenta para as propriedades que um objeto deve ter:

- Identidade: um objeto é distinto e único, mesmo que se pareça idêntico a outro. Mesmo que os atributos mudem de valor, o objeto continua a ser o

mesmo (por exemplo, ao pintar uma cadeira, ela continua sendo a mesma cadeira, apesar do atributo “cor” ter sido modificado).

- Estado: atributo que define a transitoriedade de um objeto (por exemplo, um arquivo de projeto pode ter o status de temporário, bloqueado ou liberado).
- Comportamento: objetos têm comportamentos bem definidos, respondendo de uma determinada maneira a uma ação do usuário ou outro objeto externo.

### ***Classes e Instâncias***

Classe é a categoria que define as características, estrutura e comportamentos comuns de um grupo de objetos, como “carro”, “pessoa”, etc.

Um objeto é simplesmente uma instância de uma classe. Por exemplo: “USP” é uma instância da classe “Universidade”.

As instâncias de uma classe dividem uma série de atributos e operações comuns, como altura, peso, cor dos olhos, etc., mas cada instância tem seus próprios valores para esses atributos.

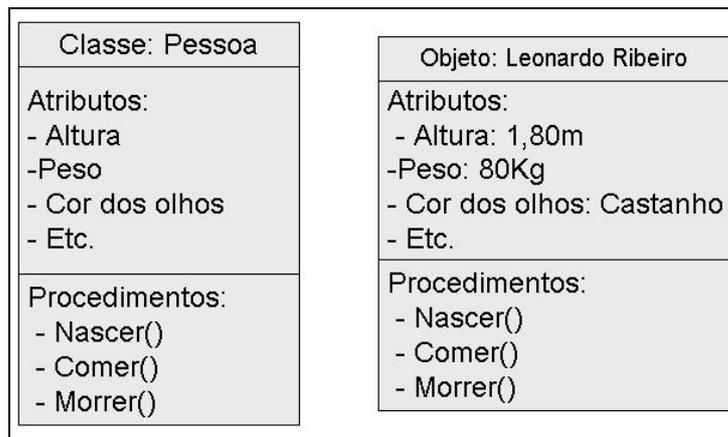


Figura 13: uma classe e sua instância

### ***Encapsulamento e comunicação por mensagens***

O princípio do encapsulamento dita que os atributos de um objeto dizem respeito somente a ele e somente um objeto pode alterar seus atributos.

Desta maneira, um objeto (ou instância de uma classe) é como uma caixa-preta e toda a comunicação com ele é feita por meio de uma interface previamente definida. Esta comunicação é feita por meio de mensagens. Como uma entidade externa não pode alterar os atributos de uma instância, ele envia uma mensagem invocando uma operação do objeto, solicitando que esta realize a mudança.

Mantido o bom encapsulamento dos objetos, um objeto não deve “se preocupar” com a forma com que outro objeto realizará uma operação. Tudo que ele precisa saber é o que deve ser feito e como enviar a mensagem necessária. Assim, se reduzem enormemente a complexidade da implementação do sistema e a dificuldade de realizar mudanças.

## *Herança*

Pode-se determinar uma hierarquia entre classes, havendo classes mães e filhas. As classes filhas herdam a estrutura e comportamento da classe mãe, embora possam ter atributos e comportamentos adicionais específicos.

Na Figura 15, as classes “carro de passeio” e “caminhão” herdam atributos e comportamento da superclasse “veículo”, embora cada uma possa ter características próprias.

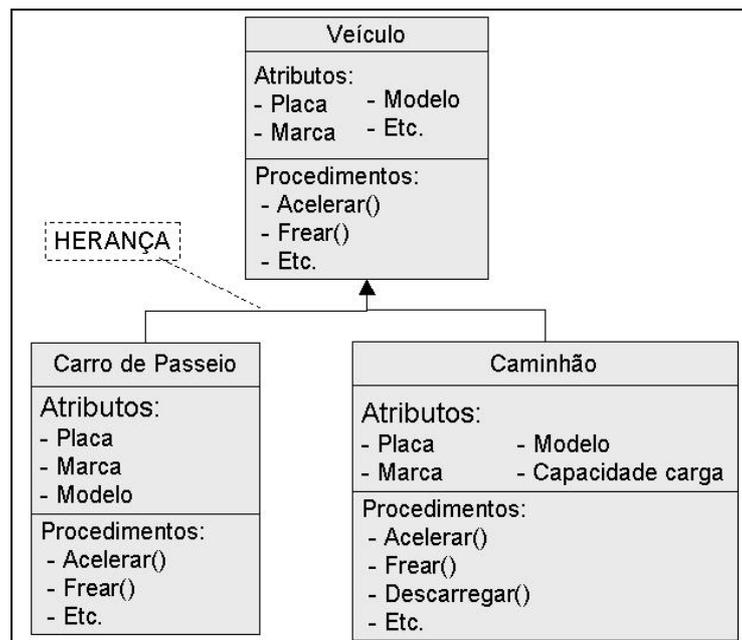


Figura 14: herança entre classes de objeto

Existe uma série de outros princípios e suas respectivas notações que mostram o relacionamento entre classes. Alguns serão mostrados dentro do item 5.4.

### ***Polimorfismo***

Como um objeto sabe a que classe pertence, ele sabe a quais mensagens ele responde. Desta maneira, a mesma operação pode ser aplicada a diversas classes com resultados diferentes, dependendo da classe.

Um bom exemplo de polimorfismo é mostrado por Vidgen et al. 2002. Imagine que a classe relógio tenha duas subclasses: analógico e digital. Ambas respondem à operação “mostrar hora()”, mas de maneiras diferentes. A entidade que envia a mensagem “mostrar hora()” não precisa saber que tipo de relógio está recebendo a mensagem, pois cada uma tem a sua própria forma (daí “polimorfismo”) de responder à mensagem. Novos relógios, como relógio de sol, atômico, etc., podem ser adicionados, desde que todos tenham como responder de alguma maneira à operação genérica “mostrar hora()”.

Encapsulamento, herança e polimorfismo são os fundamentos de orientação a objeto que ajudam a reduzir a complexidade de grandes sistemas de software, promovendo flexibilidade, reuso e extensibilidade (VIDGEN et al. 2002; LARMAN, 1999).

## **5.4 Unified Modeling Language**

A *Unified Modeling Language* (UML) é a sucessora de uma onda de métodos para análise e projeto orientados a objetos que surgiram nas décadas de 80 e 90. Ela unifica os

três principais métodos da época: os métodos de Grady Booch, James Rumbaugh e Ivar Jacobson (FOWLER E SCOTT, 1997).

A UML é chamada de linguagem e não de método universal de modelagem. A maior parte dos métodos de modelagem consiste de duas partes distintas, uma linguagem própria e um processo de modelagem bem definido. A linguagem é a representação que os métodos usam para expressar o modelo (FOWLER E SCOTT, 1997).

Segundo os próprios autores da UML, por ser uma linguagem ela é apresentada independente de um método de modelagem específico, cabendo a cada desenvolvedor usar os métodos mais condizentes com o sistema que estiver desenvolvendo e com o seu estilo. A UML é adequada para modelagem de sistemas, cuja abrangência inclui desde sistemas de informações corporativos até aplicativos Web (BOOCH, RUMBAUGH E JACOBSON, 1999).

A seguir fazemos uma apresentação geral dos blocos de construção da UML e em seguida mostramos com um pouco mais de detalhe os diagramas UML mais utilizados neste trabalho

#### **5.4.1 Apresentação geral dos blocos de Construção da UML**

Para compreender a UML, é necessário conhecer os seus blocos de construção básicos, como estes podem ser combinados e alguns mecanismos comuns aplicados a UML. Ninguém melhor para descrever estes elementos da UML do que os próprios autores da linguagem. Os principais destes elementos são descritos a seguir, conforme as definições de Booch, Rumbaugh e Jacobson (1999).

A Figura 15 lista os três blocos de construção da UML: itens, diagramas e relacionamentos.



Figura 15: blocos de Construção da UML

- Itens Estruturais: são as partes mais estáticas, os substantivos do modelo, representando elementos conceituais ou físicos.
- Itens Comportamentais: são as partes dinâmicas do modelo UML. São os verbos do modelo, alterando-se no tempo e no espaço.
- Itens de Agrupamento: são as partes organizacionais do modelo. Agrupam os demais elementos do modelo.
- Itens Notacionais: são as partes explicativas do modelo, usados pelos modeladores para explicar, comentar e esclarecer melhor as partes do modelo.

- Relacionamentos: elementos que modelam os diversos relacionamentos entre itens do modelo, podendo mostrar dependência, associação, generalização ou realização entre itens.
- Diagramas: um diagrama representa graficamente um conjunto de elementos de acordo com uma determinada perspectiva e com um determinado objetivo dentro da modelagem. Os diagramas, de acordo com a sua função, tendem a mostrar visões parciais dos elementos. A UML dispõe de nove tipos de diagrama, embora qualquer combinação de elementos dentro dos diagramas seja possível. 4 destes diagramas são apresentados com mais detalhes a seguir, no sub-capítulo 5.4.2

## **5.4.2 Diagramas em UML**

### ***Diagrama de Caso de Uso***

O diagrama de caso de uso representa atores e “casos de uso” (um tipo especial de classe) e é principalmente usado para modelagem de comportamento do sistema, dentro de uma visão estática.

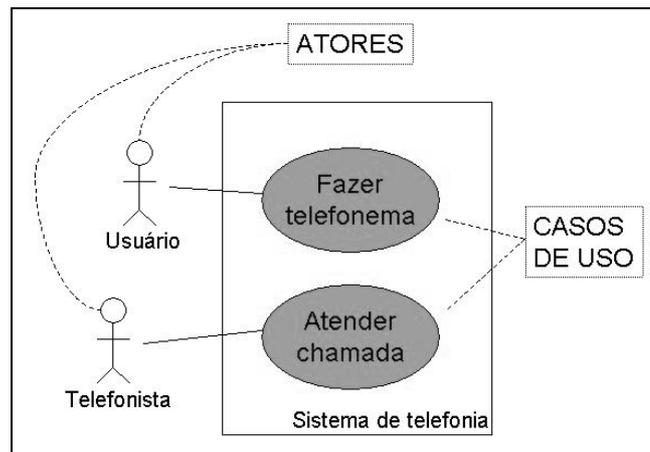


Figura 16: exemplo de um diagrama de caso de uso

### Diagrama de Classes

Um diagrama de classes apresenta um conjunto de classes, incluindo seus atributos, operações, interfaces, colaborações e relacionamentos. Tem uma visão estática e estrutural do sistema.

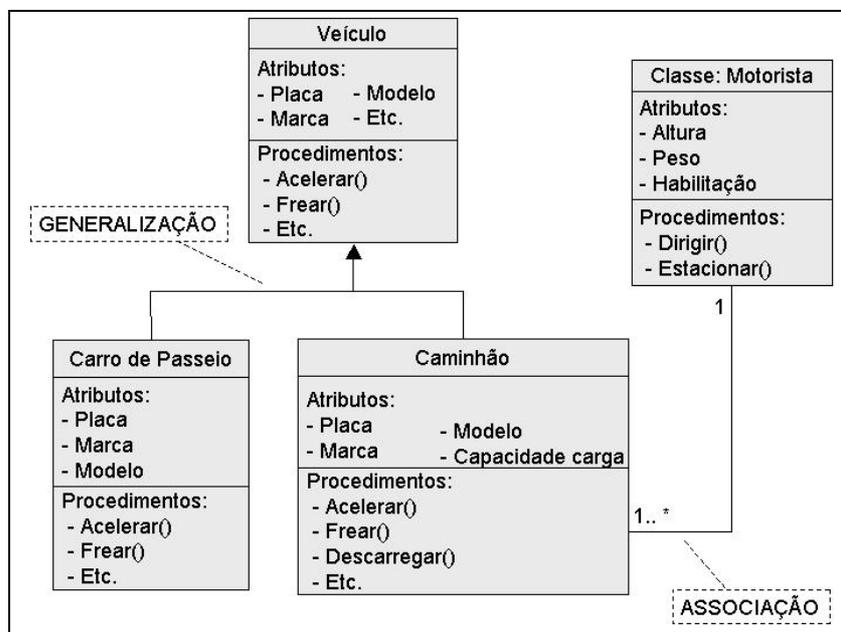


Figura 17: exemplo de um diagrama de classes

### ***Diagrama de Seqüência e Colaboração***

Os diagramas de seqüência e colaboração, chamados coletivamente de diagramas de interações, mostram as relações dinâmicas entre itens. Os diagramas de colaboração dão ênfase à organização estrutural dos objetos que enviam e recebem mensagens. Já os diagramas de seqüência dão ênfase à organização temporal das mensagens.

É importante notar que ambos são isomorfos, podendo ser convertidos de um para o outro sem perda de conteúdo.

## **6 METODOLOGIA**

Este capítulo apresenta uma discussão sobre a abordagem metodológica proposta para a realização deste trabalho. Pretende-se, com base no referencial teórico sobre pesquisa científica, e considerando os objetivos propostos e as perguntas de pesquisa, definir qual o melhor método para a realização do trabalho.

### **6.1 Perguntas de Pesquisa**

“It is better to know some of the questions than all the answers”

James Thurber

As perguntas de pesquisa, cujas respostas auxiliaram a atingir os objetivos da pesquisa (descritos no item 1.1), são:

**P 1.** Quais necessidades do processo de desenvolvimento de produtos podem ser reduzidas através de um software de gestão?

**P 1.1.** Como descrever essas necessidades em termos de requisitos de um software?

### **6.2 Referencial Teórico**

Gil (1999) define método científico como sendo “o conjunto de processos ou operações mentais que se deve empregar na investigação”.

Diversos autores classificam a pesquisa científica de acordo com diversos critérios distintos, como a natureza do problema (SILVA E MENEZES, 2000 *apud* HORTA, 2001); a abordagem do problema (MARTINS, 1994) e objetivos da pesquisas (DANE, 1990).

Outros autores, como Gil (1999), destacam os procedimentos de pesquisa como parte do ferramental necessário para o seu bom desenvolvimento.

O quadro-resumo da Figura 18 mostra os critério para classificação encontrados na bibliografia e seguidos neste trabalho. A classificação deste trabalho dentro do quadro-resumo aparece marcada em cinza escuro:

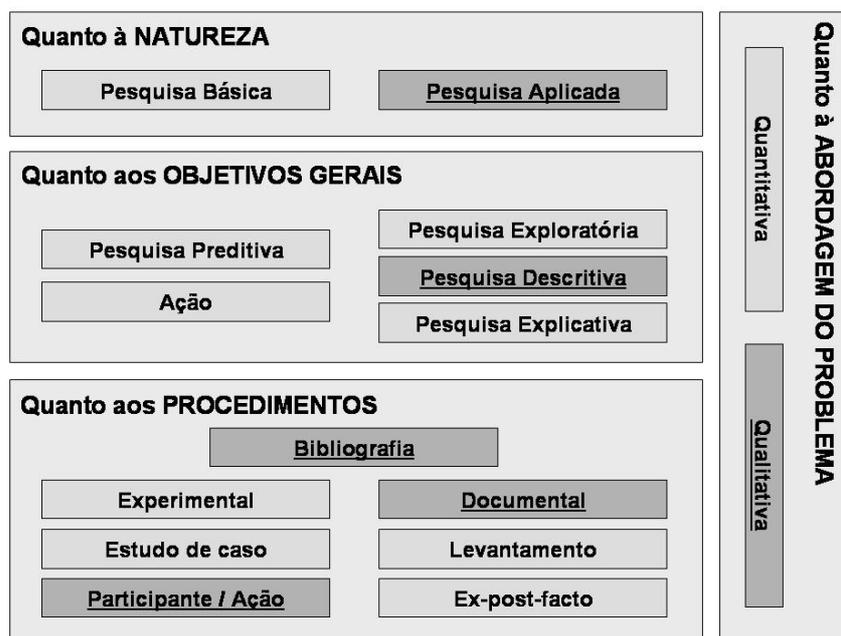


Figura 18: abordagens para classificação de pesquisas científicas (HORTA, 2001)

Utilizando a abordagem acima, pode-se classificar esta pesquisa quanto à natureza como pesquisa aplicada, por gerar conhecimentos para aplicação prática e relacionada a problemas específicos (SILVA E MENEZES, 2000 *apud* HORTA, 2001).

Quanto à abordagem do problema, classificou-se esta pesquisa como qualitativa, pois considera as relações dinâmicas entre os objetos de estudo, sem conseguir traduzi-las completamente em números. Embora este trabalho apresente algumas análises quantitativas, o caráter subjetivo das opiniões e informações prevalece.

Quanto aos objetivos gerais, classificou-se com pesquisa descritiva, por tentar descrever um determinado fenômeno, estabelecendo relações entre os requisitos de software de gestão de projetos e o processo de desenvolvimento de produtos.

Selecionou-se também os procedimentos de pesquisa mais adequados para a realização deste trabalho (ver item 6.2.1), adequando as suas atividades (ver item 6.3) a esses procedimentos.

### **6.2.1 Seleção dos procedimentos de pesquisa**

Foi feito um levantamento de requisito de software para gestão de projetos para apoiar o processo de desenvolvimento colaborativo de produtos, gerando uma lista integrada de requisitos de software. Esta lista parte de diferentes fontes, agrupadas dentro da classificação de Gil (1999):

- Pesquisa Bibliográfica: estudo da literatura sobre GP, Gestão do Conhecimento e sobre o PDP
- Pesquisa observador-participante: sistematização de experiências de projetos anteriores do Grupo EI; estudo de funcionalidades de softwares livres para GP
- Pesquisa documental: análise de Modelos de Referência já desenvolvidos (ver item 2.3.1), mas que não haviam sofrido o escrutínio de uma análise para os fins deste trabalho; documentos de projetos relacionados desenvolvidos pelo grupo de pesquisa com a participação do autor

O levantamento dos requisitos de software de gestão de projetos para apoiar o PDP colaborativo foi realizado inicialmente a partir de uma revisão bibliográfica sobre requisitos para softwares de gestão de projetos e estudo de softwares comerciais e livres. Estes requisitos devem ser classificados dentro de subgrupos de requisitos por uma questão didática, mas deve-se destacar que uma futura modelagem de um software que faça uso

desses requisitos deve tratá-los como um corpo único. A classificação adotada inicialmente foi a classificação apresentada na Tabela 3.

Em conjunto com o levantamento inicial, o autor participou de outros projetos de pesquisa do laboratório, de onde foram extraídos outros requisitos. Devido a descolamentos entre o foco dessas pesquisas e este trabalho, sendo que alguns projetos focam mais na colaboração entre membros do projeto ou mais no compartilhamento de conhecimento dentro da organização, exigiu-se que os requisitos retirados desses projetos fossem reestruturados de acordo com o foco deste trabalho. Ainda, essa diferença de foco entre as fontes faz com que alguns dos requisitos não possam ser enquadrados na classificação inicial, sendo necessária a criação de novos subgrupos de requisitos.

Independentemente da fonte (seja literatura, softwares ou projetos de pesquisa), não foi encontrada nenhuma análise que verificasse o alinhamento dos requisitos ou funcionalidades do software de gestão de projetos com as necessidades do Processo de Desenvolvimento de Produtos. Embora alguns autores destaquem o uso desse tipo de software dentro do processo (BRUCE et al. 1995; ROY & KODKANI, 2000; O'SULLIVAN, 2003, entre outros), não foi encontrada nenhuma referência em que ponto do PDP o software auxilia. Também não foi encontrada uma priorização dos requisitos de acordo com as necessidades específicas do PDP.

Como contribuição deste trabalho neste ponto da literatura, realizou-se uma comparação através de uma referência cruzada entre a lista estruturada de requisitos inicialmente levantada e um modelo de referência para o PDP. O método de comparação utilizado é uma adaptação do método de comparação de modelos de Kirchmer (KIRCHMER, 1998 *apud* ZANCUL, 2000). O método original consiste na comparação de um modelo de referência de um processo com um modelo de referência de um software, enquanto neste trabalho a comparação do modelo do PDP é feita diretamente com a lista de requisitos.

Esta comparação permite identificar tipos de requisitos que não atendem nenhuma necessidade específica do PDP e que, portanto, têm baixa prioridade. Também identifica

tipos de requisitos que atendem a necessidades importantes ou recorrentes dentro do PDP e logo têm alta prioridade.

A comparação ainda permite identificar necessidades do PDP que não são atendidas pelos requisitos levantados, devendo o autor levantar os requisitos necessários para atender a estas necessidades. Aqui vale ressaltar que o PDP apresenta necessidade de ferramentas de software diversos, mas que este trabalho focou apenas nas necessidades relacionadas à Gestão de Projetos.

Neste ponto é importante deixar claro que as ferramentas de gestão de projetos podem apresentar escopos diferentes, partindo desde ferramentas para gerenciar atividades, até ferramentas complexas, que permitem gestão das atividades, recursos, pessoas, conhecimentos, documentos, comunicação, custos, riscos entre outros aspectos da gestão de projeto, como detalhado no capítulo 3. Este trabalho tentou manter o escopo do software o mais amplo possível.

### **6.2.2 Limitações do método de trabalho**

A principal limitação deste trabalho é seu caráter teórico. Não foi verificado se os requisitos levantados neste trabalho atendem ao PDP de empresas específicas. Procurou-se reduzir essa limitação utilizando-se um modelo de referência de escopo amplo, que cobrisse todo o ciclo de vida do produto e que pudesse ser representativo do PDP de diversas indústrias e empresas.

Cabe lembrar que apesar de ser bastante genérico para permitir abstrações / adaptações para outras áreas e indústrias o modelo tem seu foco em empresas de manufatura de bens de consumo duráveis ou de capital, como já explicado no item 2.3.1

Um conjunto de pesquisadores do Grupo EI está neste momento levantando requisitos de software junto a empresas dentro da região de São Carlos – SP. Este levantamento poderá futuramente identificar características específicas do ambiente de negócio dessas empresas que não foram contemplados neste trabalho.

Outra limitação importante é a validade dos requisitos levantados diretamente a partir de funcionalidades de softwares comerciais e livres estudados. Nem sempre se tinha à disposição o software para ser instalado e verificado *in loco*. Logo, em alguns casos, teve-se que recorrer a manuais e descrições das funcionalidades feitas pela empresa desenvolvedora do software. Para reduzir esta limitação, procuraram-se avaliações isentas dos softwares estudados, por meio de análises comparativas.

Ainda sobre a validade das fontes, vale ressaltar o uso de fontes indiretas na literatura, onde os autores apresentam os requisitos já sintetizados. Para reduzir esta limitação procurou-se usar mais de uma fonte, procurando-se validar os resultados de um autor confrontando-os com outros estudos. Em alguns casos, foram procurados estudos específicos sobre softwares relacionados, como EDM/PDM, CSCW, entre outros. Em todas as situações, verificou-se se tal requisito se relacionava às melhores práticas de Gestão de Projetos incorporadas no PMBOK pelo PMI.

### 6.3 Atividades do Trabalho

Este trabalho foi dividido nas seguintes fases:

- **Fase 1:** levantamento inicial das necessidades / requisitos do sistema

Nesta fase, faz-se um levantamento inicial dos requisitos de um sistema genérico de gestão de projetos, a partir de fontes diversas (literatura, softwares comerciais, software livres e projetos de pesquisa desenvolvidos pelo Grupo EI).

- **Fase 2:** desenvolvimento de referência cruzada entre a lista inicial de requisitos e um modelo de referência em Desenvolvimento de Produtos e redação dos resultados

Nesta fase, verificam-se dois critérios: se os requisitos levantados atendem às necessidades das atividades descritas em um modelo de referência em DP, e se existe alguma necessidade dentro modelo de referência que poderia ser atendida por um

requisito ainda não listado. Esta referência cruzada tem como objetivo garantir a consistência entre a lista de requisitos levantados e o PDP.

As atividades do trabalho foram distribuídas dentro dessas duas fases, de acordo com a Figura 19, e descritas a seguir:

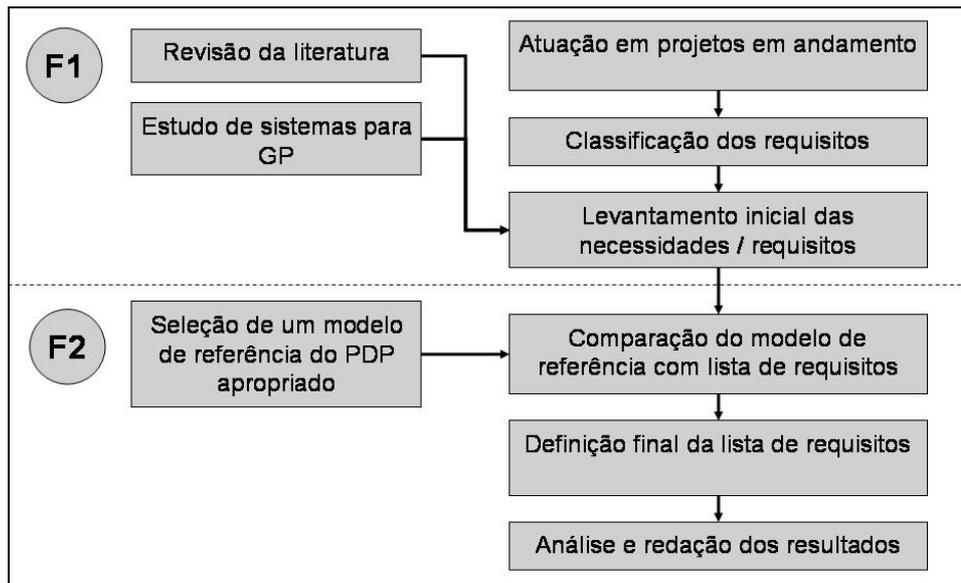


Figura 19: método de trabalho

### FASE 1: levantamento Inicial das necessidades / requisitos do sistema

- a) Revisão da Literatura: o estudo da literatura teve como objetivo embasar todo o trabalho. Inicialmente, foram estudados conhecimentos estruturais necessários ao desenvolvimento do modelo, de forma a conhecer assuntos como modelagem e orientação a objetos de maneira mais aprofundada. Posteriormente, se baseou em um “tripé de conhecimentos”, sendo a primeira perna deste tripé o estudo sobre Desenvolvimento de Produtos, sua visão como um processo e seu ciclo de vida.

Outra perna do tripé em que se baseia este trabalho é a Gestão de Projetos, de forma que efetuou-se um estudo da literatura fundamental no assunto,

envolvendo seus processos, áreas de conhecimento e um estudo sobre sistemas de software para auxiliar a gestão.

A terceira perna do tripé faz a ligação sobre estes dois primeiros conhecimentos. Para tanto, ela se divide em gestão de projetos colaborativos em ambientes geograficamente distribuídos, e na Gestão de Conhecimentos aplicada ao Desenvolvimento de Produtos e à Gestão de Projetos.

- b) Atuação em projetos em andamento: os projetos realizados pelo grupo EI, principalmente o Framework para sistematização de conhecimentos e o e-HUB (brevemente descritos nos itens 7.2.3.1 e 7.2.3.2, respectivamente), estruturaram as necessidades dos usuários e requisitos do software. Nesta atividade, adicionaram-se os requisitos levantados por essas pesquisas à lista inicial levantada a partir da literatura.

Por meio da atuação nessas pesquisas, pretendeu-se também fornecer uma visão mais prática da Gestão de Projetos, sistematizando essa experiência em benefício deste trabalho.

- c) Estudo de sistemas para gestão de projetos: consiste no estudo de funcionalidades de softwares analisados diretamente pelo autor e material encontrado como síntese literária, incluindo pesquisas anteriores dentro do grupo de pesquisa e pesquisa documental em manuais e tutoriais dos softwares analisados.
- d) Classificação da lista de requisitos: as listas de requisitos são muitas vezes encontradas na literatura de forma desestruturada. É necessário realizar uma classificação desses requisitos dentro de um padrão (vide item 4.1), de forma a facilitar a comparação com o modelo de referência em desenvolvimento de produtos.
- e) Levantamento e estruturação inicial da lista de requisitos: consiste no levantamento de requisitos de todas as fontes, agrupando-os dentro de uma classificação padrão (ver item 7.1), eliminando os requisitos redundantes e alinhando a terminologia e granularidade dos requisitos. O resultado desta etapa

do trabalho é uma lista de requisitos de software, estruturados e mutuamente exclusivos, para gestão de projetos. No entanto, esta lista ainda não foi confrontada com o processo de desenvolvimento de produtos, sendo esse o objetivo da fase dois deste trabalho.

## **FASE 2: desenvolvimento de referência cruzada os requisitos e um modelo de referência para o PDP**

f) Seleção de um modelo de referência de Desenvolvimento de Produtos (ver seção 2.3.1): destacou-se a importância de se selecionar um modelo apropriado, sendo os principais critérios:

- Especificidade / nível de detalhe: o modelo deve ser genérico o bastante para permitir que os resultados sejam válidos em uma variedade de domínios / indústrias.
- Disponibilidade: o pesquisador deve ter acesso a níveis de detalhe do modelo de referência a ser estudado. Muitos modelos de indústrias são completos e detalhados, mas são de acesso restrito ao pesquisador. Desta maneira, optou-se por utilizar o modelo desenvolvido dentro do PDPNet (ROZENFELD et al. 2006) devido ao fácil acesso do pesquisador a todos os seus detalhes.
- Atualidade / fidelidade às melhores práticas: o modelo deve ser academicamente correto, sendo fiel à literatura e às práticas industriais correntes.

A descrição do modelo de referência para o PDP escolhido é descrito nos itens 2.3.1 e 7.3).

g) Comparação do modelo de referência em PDP com a lista preliminar de requisitos: esta comparação tem como objetivo validar os requisitos iniciais. Pretende-se com esta atividade responder duas perguntas:

- Os requisitos levantados atendem às necessidades de ferramental aparentes no modelo de referência? Quais requisitos são os fundamentais e quais são opcionais?
- Existe alguma necessidade de ferramenta de software encontrada no modelo que não é atendida pelos requisitos levantados?

O método adotado para realização desta atividade é uma referência cruzada, uma adaptação no método de Kirshmer (KIRSHMER, 1998, *apud* ZANCUL, 2003).

Este processo já foi utilizado com sucesso para comparar um modelo de referência de processos de negócio com um modelo de referência de software (ZANCUL, 2003).

- h) Definição final da lista de requisitos: após a validação da lista de requisitos descrita no item (F), surgiram novos requisitos que devem ser agrupados dentro de uma classificação prévia. Alguns novos requisitos, no entanto, não se encaixaram em nenhum critério de classificação original. Surgiu, portanto, a necessidade de se fazer uma adaptação nos critérios de classificação preliminarmente usados e priorizar os requisitos de acordo com a importância desses no modelo de referência.
- i) Análise e redação dos resultados: a fase final consiste em analisar os dados encontrados, compará-los com as informações obtidas na revisão bibliográfica, sistematizar o conhecimento e compilar a tese. Parte desta atividade já foi realizada, ao se escrever esta dissertação.

A realização das atividades não se encontra totalmente limitada a estas fases, podendo e devendo haver superposição entre estas. Algumas atividades, como o estudo da bibliografia, a atuação em outros projetos e correções / alterações no modelo, ocorrem ao longo de todo o trabalho.

Outras atividades, como atendimento às disciplinas e publicações de artigos em congressos e periódicos, ocorrem paralelamente ao desenvolvimento do projeto.



## **7 RESULTADOS DO TRABALHO**

Neste item são apresentados os resultados do trabalho realizado:

- **7.1** - Classificação dos requisitos
- **7.2** - Listas de requisitos de software de gestão de projetos, separados pela fonte
- **7.3** - Comparação da lista de requisitos levantada no item 4.2 com um modelo de referência para Desenvolvimento Colaborativo de Produtos
- **7.4** - Lista final de requisitos agrupados de acordo com a classificação final

### **7.1 Classificação dos requisitos**

A maioria das fontes na literatura apresenta requisitos de forma desestruturada ou agrupada segundo critérios de classificação bastante divergentes entre si. Para facilitar a integração de todas as fontes utilizadas neste trabalho, optou-se inicialmente por utilizar a classificação de requisitos em oito grupos, apresentada na Tabela 3.

É importante ressaltar que esta classificação tem fins meramente didáticos e que o desenvolvimento de um software baseado nos requisitos levantados neste trabalho deve considerar todos os requisitos de maneira integrada.

## **7.2 Listas de requisitos de software de gestão de projetos**

Os requisitos foram levantados a partir de diversas fontes e divididos de acordo com o tipo de fonte utilizado.

Uma vez que cada fonte analisada apresenta um foco diferente, estas também apresentam diferentes níveis de granularidade. Por exemplo, enquanto algumas fontes apresentam “Ter capacidade de Gestão de Documentos” como um requisito, outras detalham a gestão de documentos do projeto em grande detalhe. Neste trabalho, o autor apresenta a lista final de requisitos sempre da forma mais detalhada possível.

Os itens 7.2.1, 7.2.2, 7.2.3.1 e 7.2.3.2 apresentam as listas iniciais de requisitos utilizadas na primeira parte deste trabalho.

### **7.2.1 Requisitos levantados a partir de softwares de gestão de projetos**

Estes requisitos foram retirados da compilação de requisitos de diversos softwares livres e comerciais, realizada pelo grupo de pesquisa.

Um primeiro levantamento realizado pelo autor deste trabalho apresenta uma lista de 74 requisitos, derivados das funcionalidades apresentadas por 141 softwares diferentes. Doze destes softwares foram estudados em maior detalhe, enquanto 139 foram estudados de forma mais abrangente e parcial.

Colegas dentro do grupo de pesquisa continuam e aprofundam este trabalho, fazendo um levantamento de critérios para avaliação de sistemas para gestão de projetos, baseados no estudo detalhado de 25 softwares de gestão de projetos, todos de código aberto.

A Tabela 8, encontrada no apêndice B, apresenta a compilação dos requisitos levantados nestes trabalhos, utilizando a classificação final, apresentada no item 7.1. Esta lista de requisitos foi o material básico sobre o qual se desenvolveu o trabalho.

Cada requisito levantado apresentado na Tabela 8 possui um código, para facilitar sua localização. Códigos “RI” são códigos retirados das funcionalidades levantadas pelo

autor, enquanto “RO” são códigos levantados em trabalhos posteriores do grupo de pesquisa.

### **7.2.2 Requisitos levantados a partir de funcionalidades de software comercial de referência**

Estes requisitos foram levantados a partir da avaliação do software MS-Project, (versão 2000), considerado uma referência para os softwares comerciais de gestão de projetos. Este levantamento foi feito de forma direta, avaliando pessoalmente o software, e indireta, baseando-se no trabalho de Savi (2003), que levantou uma série de funcionalidades do MS-Project, bem como em pesquisas em livros, tutoriais e manuais sobre o sistema.

Os requisitos levantados a partir das funcionalidades do Microsoft Project Podem ser encontrados no apêndice B, na Tabela 9.

### **7.2.3 Requisitos levantados a partir de projetos anteriores e discussões internas dentro do Grupo EI**

Neste item são descritos os requisitos levantados a partir de projetos anteriores do Grupo EI nos quais o bolsista teve participação direta. Embora alguns desses requisitos tenham sido documentados de forma bastante estruturada e explícita, em alguns casos esses foram levantados a partir de discussões internas com membros participantes de cada projeto e registradas nas anotações do autor deste trabalho. Neste caso os requisitos foram confrontados com outras fontes e discutidos com o orientador deste trabalho. A seguir apresenta-se uma breve descrição dos projetos.

### **7.2.3.1 Framework para sistematização de conhecimentos e direcionamento do grupo de pesquisa.**

O framework compreende um grupo de projetos que pretende sistematizar conhecimentos de gestão do ciclo de vida de produtos, organizando-os em elementos de infra-estrutura e de conteúdo.

Esse framework tem como objetivo direcionar as ações do grupo de pesquisa para a criação e desenvolvimento de uma infra-estrutura capaz de apoiar a criação, manutenção e divulgação do conhecimento. Este projeto guiará os esforços do grupo na criação e manutenção do conhecimento, bem como na aplicação do próprio framework.

Os elementos estruturais organizam e permitem um acesso sistemático aos elementos de conteúdo. Já os elementos de conteúdo são conhecimentos explícitos sobre o processo de desenvolvimento de produtos, dentro domínio de aplicação do framework.

#### ***Lista de requisitos derivados de discussões internas durante a elaboração do framework***

Esta lista foi derivada a partir da listagem de necessidades genéricas e específicas dos usuários do framework sendo criado. A lista original destas necessidades estava agrupada em necessidades genéricas ou de acordo com o usuário específico (gerente de projetos, membro de equipe de desenvolvimento de produtos, profissional em geral e estudantes). No entanto, a lista de requisitos derivado destas discussões foi reorganizada de acordo com a classificação padrão desenvolvida neste trabalho, apresentada no item 7.1.

Uma vez que esse framework ainda não foi totalmente finalizado, estando ainda no início do seu desenvolvimento, a lista de requisitos apresentada neste trabalho é preliminar. Ainda assim serve para o propósito deste trabalho, dando uma ênfase maior no domínio de conhecimento de gestão de conhecimento à lista de requisitos sendo levantados. A lista completa pode ser vista no apêndice B, na Tabela 10.

Também foi desenvolvida uma seqüência de diagramas de caso de uso, representando os requisitos desenvolvidos nesta parte do trabalho. Os diagramas podem ser vistos no apêndice E desta dissertação.

### **7.2.3.2 e-HUBs**

Estes requisitos foram levantando a partir do projeto *e-Engineering Enabled by Holonomic and Universal Broker Services* (e-HUBs), realizado em conjunto com oito parceiros entre empresas e instituições acadêmicas européias. O projeto fez parte do programa Information Society Technology da Comunidade Européia e teve como objetivo desenvolver um dispositivo que permita um conjunto integrado de serviços de colaboração e suporte às atividade de e-engineering, focando principalmente no projeto de desenvolvimento colaborativo de produtos.

O consórcio foi dividido em grupos de trabalho, para levantamento de requisitos para o sistema (software + pessoas da organização), aplicações de *e-engineering*, casos de aplicação prática e desenvolvimento do software de suporte.

O grupo de trabalho responsável pelo desenvolvimento do software desenvolveu uma plataforma de colaboração (Figura 20), que inclui funcionalidades de Gestão de Projetos e um avançado sistema para criação, customização e *enactment* de workflows, que guiam os usuários na realização do projeto.

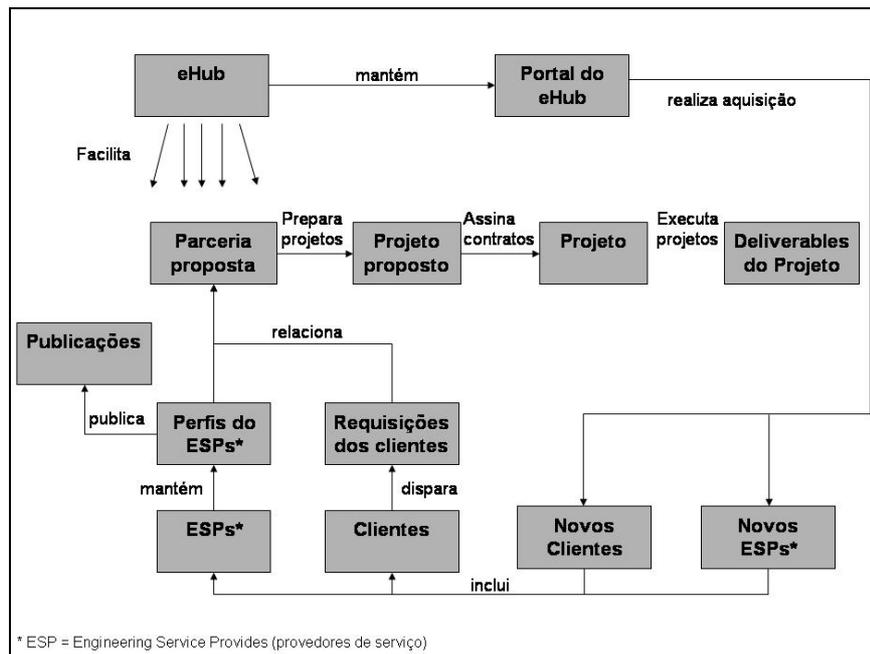


Figura 20: Estrutura dos principais componentes de uma parceria através do e-hub

(SHEVCHENKO, A. et al., 2003)

O e-HUB tem, de acordo com a definição do projeto, quatro funções principais:

- Criação e manutenção de uma comunidade, concentrando clientes e ESP (Engineering Service Providers). Este *pool* de clientes e provedores de serviços formam a base inicial do processo de parceria.
- Marketing ativo e passivo, por meio de publicações (vide Figura 20).
- Desenvolvimento de contratos, através de templates processos-padrão.
- Engenharia de Projetos, guiando membros das parcerias pelas suas diversas etapas. Esta função do e-HUB que será detalhada, na forma de requisitos, nesta dissertação.

Os requisitos do e-HUB foram levantados a partir de diversos estudos de caso de colaboração a distância em projetos de engenharia, em diversos ramos (PMEs, empresas globais), setores (equipamentos médicos, construção civil, projetos mecatrônicos, etc) e

países (Itália, México, Países Baixos, etc). (SHEVCHENKO, A. et al., 2003) (FLORATOS et al. 2002).

No desenvolvimento do protótipo do projeto e-HUB, foi criado um ambiente de colaboração via Internet para permitir uma análise destas parcerias com o uso da ferramenta desenvolvida no projeto, que permite entre outras coisas a criação e *enactment* de *workflows*, que designam as atividades, tarefas, deliverables e responsáveis de um projeto de colaboração, possuindo inclusive caminhos alternativos pré-definidos para cada tarefa (FLORATOS et al. 2002).

O ambiente ainda dispunha de uma biblioteca digital com sistema de gerenciamento eletrônico de documentos e funcionalidades como fórum para discussão, *whiteboards*, busca avançada, entre outros (FLORATOS et al. 2002).

### ***Requisitos levantados a partir do projeto e-HUBs***

Os requisitos levantados a partir do projeto e-HUB partem de praticamente de 3 fontes distintas: os requisitos para o sistema, desenvolvido pelo grupo de trabalho 1, liderado pela Delf University of Technology (SHEVCHENKO, A. et al. 2003), o design funcional e o protótipo final da plataforma de software, liderados pela European Dynamics (EUROPEAN DYNAMICS, 2003). (vide Figura 20).

Para garantir a integridade desses documentos, a equipe brasileira realizou uma comparação entre os requisitos e as funcionalidades da plataforma de software do protótipo, classificando as funcionalidades dentro dos seguintes grupos:

- Gerenciamento Eletrônico de documentos
- Facilidade de banco de dados
- Facilidades de comunicação
- Gestão de documentos do produto
- Ferramentas de segurança

- Taxonomia
- Ferramentas de engenharia
- Ferramentas de gestão (de projetos)
- Formulários e padrões

Esta comparação foi fundamental para o levantamento de requisitos do e-HUB realizado neste trabalho, pois já listava todos os requisitos iniciais e funcionalidades do protótipo de maneira estruturada.

Todos os requisitos levantados durante o projeto estão inseridos neste trabalho e podem ser vistos no apêndice B, Tabela 11, sendo agrupados de acordo com a classificação padrão apresentada no item 7.1.

Os códigos dados aos requisitos levantados a partir do e-HUB derivam de sua origem: códigos iniciados com “E1” derivam da lista inicial de requisitos, “E2” derivam do design funcional e “E3” derivam do protótipo do software.

### **7.3 Referencia cruzada entre modelo de referência para o PDP e lista inicial de requisitos**

A comparação do modelo de referência em PDP com os requisitos foi baseada no método de comparação de modelos proposto por Kirchmer (1998) *apud* Zancul (2003). No entanto, o método de Kirchmer é definido de forma abstrata, não especificando detalhadamente os procedimentos que devem ser executados durante a sua aplicação.

Determinou-se, então, uma adaptação da matriz de comparação realizada por Zancul (2003), na qual as linhas descrevem as etapas do modelo (macrofase, fase e atividades) do PDP e as colunas representam os requisitos de software, agrupados de acordo com a classificação detalhada no item 7.1.

A comparação dos modelos foi realizada relacionando-se as tarefas do modelo de referência do processo de desenvolvimento de produtos com os requisitos do software de gestão de projetos.

Para cada relação existente entre uma tarefa do modelo de desenvolvimento de produtos e um requisito foi feita a seguinte pergunta: “a tarefa apresenta esta necessidade / requisito?”. Em caso afirmativo, assinalou-se a intersecção entre a linha da tarefa e a coluna requisito (Figura 21). Esse procedimento foi repetido para todas as relações da matriz.

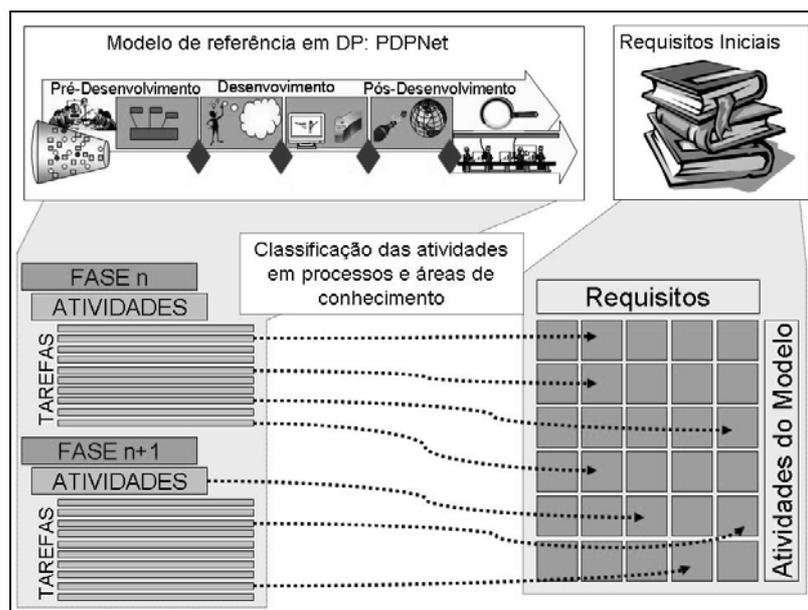


Figura 21: aderência do Modelo de referência aos requisitos

Os requisitos foram, em seguida, especificados no modelo de referência do processo de desenvolvimento de produtos. Para cada tarefa apoiada por um requisito, os nomes dos

requisitos foram completados em uma nova coluna (requisitos), incluindo os códigos correspondentes a sua classificação.

Na comparação, foi identificado que os requisitos previamente levantados na fase 1 deste trabalho apóiam de alguma maneira 81% das tarefas do processo de desenvolvimento de produtos.

As principais dificuldades enfrentadas na comparação foram decorrentes das diferenças existentes entre idiomas, a terminologia e o nível de detalhamento adotados em cada uma das fontes.

O estabelecimento de relações entre o modelo do processo e os requisitos de software também foi dificultado pela definição de diferentes níveis de detalhamento entre si.

Verificou-se que, na maioria das vezes, não seria possível estabelecer uma relação única entre as sub-atividades do modelo de referência e os requisitos de software, sendo que, em alguns casos, um requisito pode apoiar diversas atividades do PDP (logo é um requisito de alta prioridade) enquanto em outros casos apóia somente poucas atividades (sendo classificado portanto como um requisito de baixa prioridade ou “*Nice to have*”).

Para superar esse problema permitiu-se, seguindo as recomendações de Kirchmer (1998) *apud* Zancul (2003), que um ou mais elementos do modelo de referência fossem relacionados com um ou mais requisitos de software.

Deve-se ressaltar que a comparação com o modelo do PDP tem algumas limitações na identificação dos requisitos que podem ser utilizados para apoiar a gestão de projetos dentro do processo.

A Tabela 6 apresenta como os grupos de requisitos originais atendem as atividades do modelo de referência.

Tabela 6: atividades do Modelo de Referência atendidas pelos grupos original de requisitos

<b>Grupos de Requisitos (Fase 1)</b>	<b>Tarefas suportadas (#)</b>	<b>Tarefas suportadas (%)</b>
1 – Arquitetura	0	0%
2 - Agenda e Reuniões	11	2%
3 – Gestão de Atividades	35	7%

4 – Gestão de Recursos e Usuários	43	9%
5 – Gestão de Custos	72	14%
6 – Gestão da Informação	135	27%
7 - Ferramentas de Monitoração	168	34%
8 – Gestão de Múltiplos Projetos	80	16%
9 - Ferramentas de Comunicação	19	4%
10 - Templates e Modelos	44	9%
11 - Outras Ferramentas de Suporte	26	5%

A comparação da lista inicial de requisitos com o modelo de referência permitiu identificar algumas peculiaridades na classificação de requisitos utilizada na primeira fase:

- Os requisitos relacionados com a arquitetura do sistema não se relacionam com nenhuma tarefa do modelo e devem ser analisados de maneira independente. Desta maneira excluiu-se os requisitos de arquitetura de comparações futuras com o modelo de referência.
- Algumas tarefas se relacionam com a parte mais operacional dos requisitos de Gestão de Informação (na forma de Gestão de Documentos), enquanto outras são atendidas por requisitos mais relacionados com a gestão de conhecimentos. Decidiu-se, portanto, separar o grupo 6 em dois: 6-Gestão de documentos e 7-Gestão de conhecimentos
- Dada a sua importância, os requisitos relacionados com gestão da mudança foram separados do grupo Ferramentas de monitoração – e agrupados em outro grupo
- O grupo Outras ferramentas de suporte – deve ser melhor detalhado, incluindo subgrupos para cada tipo de ferramenta, como FMEA, QFD, etc.
- Dada a sua importância, requisitos relacionados com gestão de riscos foram separados do grupo “Outras ferramentas de suporte” – e agrupados dentro de um grupo próprio

Estas modificações foram realizadas na lista de requisitos, agrupando-os de acordo com esta nova classificação baseando-se no aprendizado decorrente da comparação com o modelo de referência. A nova classificação dos requisitos pode ser vista no Apêndice A, na Tabela 7.

A lista final de requisitos atende às necessidade das tarefas do modelo de referência com maior ou menor abrangência dependendo do tipo de requisito: sendo os tipos de requisitos ligado ao controle do projeto os que apresentaram a maior interação com o modelo de referência, com pode ser visto na Figura 22 a seguir.

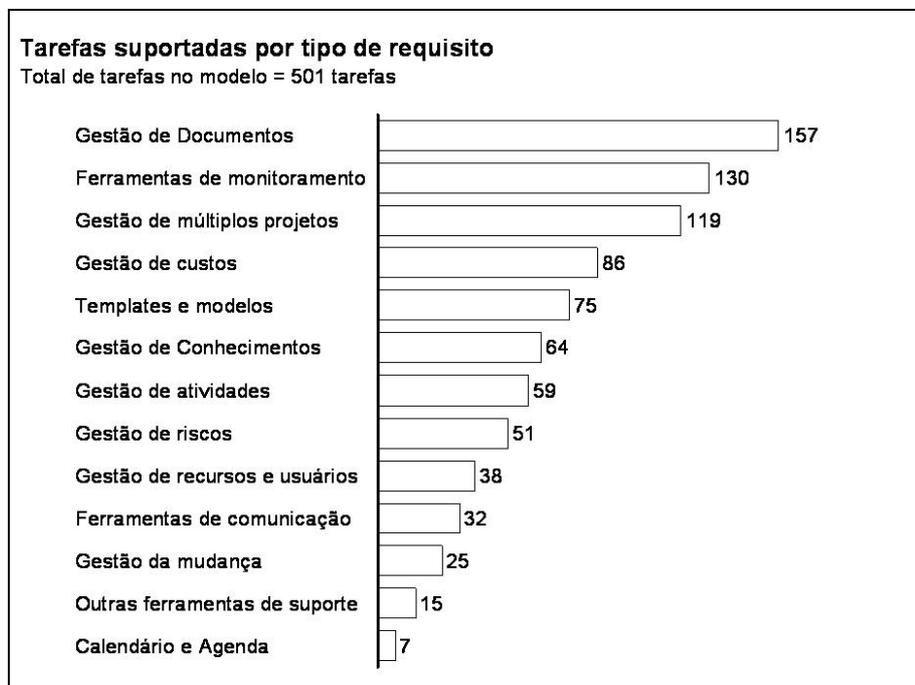


Figura 22: tarefas suportadas por tipo de requisito

Interessante notar também a distribuição do suporte ao modelo ao longo das fases do processo. Para praticamente todos os tipos de requisito, podemos notar uma maior incidência das interações na macro-fase de desenvolvimento, especialmente nas fases de

projeto detalhado, lançamento do produto e projeto conceitual, como pode ser observado na Figura 23, a seguir:

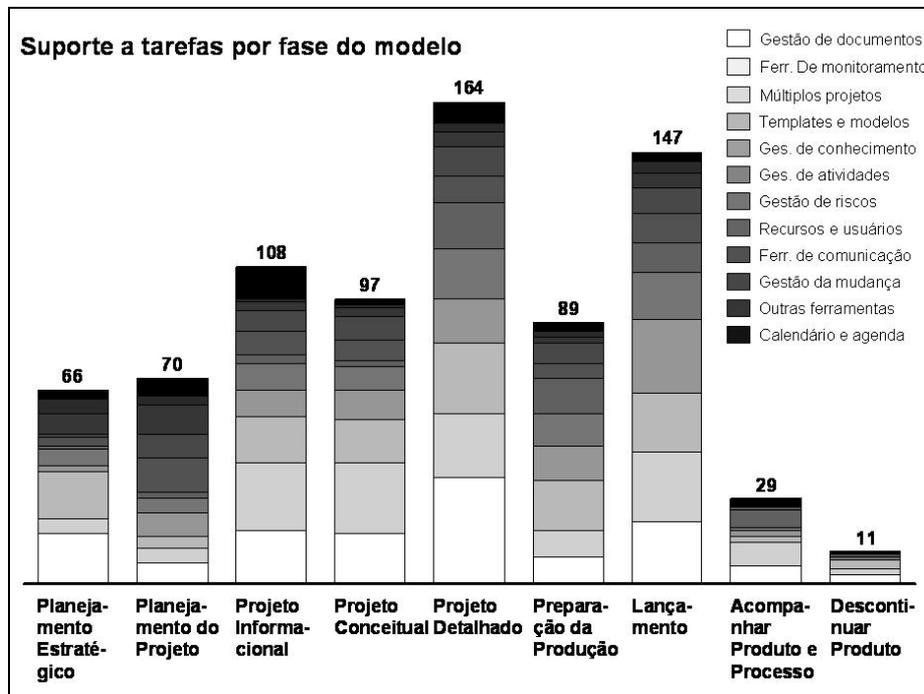


Figura 23: tarefas suportadas por tipo de requisito e fase do modelo

A Tabela 12, encontrada no apêndice C, contém uma representação estruturada do modelo de referência, com todas as suas macro-fases, fases, atividades e tarefas nas linhas e os requisitos, agrupados dentro da classificação final dos requisitos nas colunas. As tarefas (atividades de nível 4) foram comparadas, uma a uma, com os requisitos. Os relacionamentos positivos foram marcados com “1”, enquanto a ausência de suporte foi marcada com “0”. O número de suportes as tarefas foram somados nas agregações por atividade, fase e macro-fase, de acordo com as tarefas correspondentes.

O problema deste tipo de agregação é o diferente granularidade do modelo, sendo algumas fases descritas em bastante detalhe, com diversas tarefas, enquanto outras

atividades constam de poucas tarefas detalhadas. Assumiu-se, no entanto, que o detalhamento de uma atividade está diretamente relacionado à importância ou complexidade da atividade e que portanto estas discrepâncias não afetavam o resultado do trabalho.

#### **7.4 Lista final de requisitos de software, agrupados de acordo com a classificação**

##### **final**

Após avaliação dos requisitos contra o modelo de referência do PDP não só a classificação dos requisitos foi alterada, como alguns requisitos foram eliminados agrupados (eliminando redundâncias), enquanto outros foram adicionados a lista. A lista final de requisitos pode ser vista no apêndice D, na Tabela 13.

## **8 DISCUSSÕES E APLICAÇÃO PRÁTICA NA INDÚSTRIA**

A vantagem deste trabalho para a Indústria é a identificação das necessidades de software de organizações que desenvolvem produtos de acordo com boas práticas de Gestão de Projetos.

Este trabalho levanta os requisitos necessários para a especificação de elementos estruturais de um sistema de software, que pode e pode ser implementado em um portal na internet. A intenção inicial do programa Grupo EI do Núcleo de Manufatura Avançada é manter um portal como esse de livre acesso a todos os interessados em acumular elementos de conteúdo sobre o assunto, contribuindo para desenvolvimento deste.

No entanto, para que uma organização aproveite os frutos deste trabalho, não é imprescindível que estes requisitos sejam implementados em um modelo conceitual e este em software (no entanto, isso seria ideal), já que o modelo sendo desenvolvido já apresenta conceitos importantes que podem ser adicionados ao PDP específico da empresa e às práticas de Gestão de Projetos utilizadas.

Uma empresa que desenvolva seu próprio sistema, utilizando para isso o resultado deste trabalho, ou faça uso de portais externos, poderia gerenciar seus próprios projetos de Desenvolvimento de Produto dentro desta estrutura.

Uma conclusão importante deste trabalho é o fato dos requisitos mais ligado com o processo de controle de projetos (veja item 3.1.) são os que mostraram a maior importância dentro do modelo de referência do PDP (veja Figura 22). Isso mostra de forma clara que as necessidades de software dos times de desenvolvimento de produto está mais no controle acurado do projeto do na definição de suas atividades.

## 9 REFERÊNCIAS

AMARAL, D. C.; ROZENFELD, H. *Requisitos para a criação de modelos de referência para o processo de desenvolvimento de produto considerando a participação de fornecedores*. XV Congresso Brasileiro de Engenharia Mecânica. UNICAMP: Campinas-SP, 1999.

AMARAL, D. C. *Arquitetura para gerenciamento de conhecimentos explícitos sobre o processo de desenvolvimento de produto*. (Tese de Doutorado). Escola de Engenharia de São Carlos – USP, 2001.

AMARAL, D. C., AUGENBROE, G., SCALICE, R. K., RIBEIRO, L.D.V., NETTO, R., ROZENFELD, H. Using Enterprise Models for Defining a Web Based Collaborative e-Engineering Environment for Small and Medium Sized Enterprises. In: *e-Challenges*, Viena, Áustria, 2004.

AMERICAN HERITAGE. *The American Heritage College Dictionary*, 4 ed., Houghton Mifflin Co., 2002.

AYAS, K. Professional project management: a shift towards learning and a knowledge creating structure. *International Journal of Project Management*, 1996.

BADIRU, A. B. *Project management in manufacturing and high technology operations*, 2 ed., Editora John Wiley & Sons Inc., NY, 1996.

BOOCH, G, RUMBAUGH, J, JACOBSON, I, *The Unified Modeling Language User Guide*. Addison Wesley Longman Inc, 1999.

BOWEN, H. K., CLARK K.B., HOLLOWAY C.A., WHEELRIGHT SC. *The perceptual enterprise machine—seven keys to corporate renewal through successful product and process development*. Oxford University Press, NY, 1994.

CHEN, C. et al. Project scheduling for collaborative product development using DSM. *International Journal of Project Management*, 2003.

CIMData. Program Review of SAP's mySAP Product Lifecycle Management cPDM Program. ([www.cimdata.com](http://www.cimdata.com)), 10/2001.

- CIMData. PLM, empowering business the future of business, (www.cimdata.com), 2002.
- CLARK, K. B.; FUJIMOTO, T. *Product development performance: strategy, organization and management in the world auto industry*. Boston-Mass., Harvard Business School Press, 1991.
- CURRIE, W. L. *A knowledge-based risk assessment framework for evaluating web-enabled application outsourcing projects*. International Journal of Project Management 21 - pags 207–217, 2003
- DAVENPORT, T. H. *Reengenharia de processos*. Rio de Janeiro: Campus, 1994.
- DAVENPORT, T. H. & PRUSAK, L. *Conhecimento empresarial: como as organizações gerenciam seu capital intelectual*. Tradução de Lenke Peres. Rio de Janeiro: Campus, 1998.
- EUROPEAN DYNAMICS, *Functional Design of basic e-Hub platform & Alpha version of the eHUB prototype*, e-HUB project internal document, Information Society Technology, 2003
- EDS. *Collaborative Solutions for Product Lifecycle Management*. Leveraging your entire product lifecycle to meet the business imperatives of today's digital economy. 03/2002.
- FLORATOS, N. et al., *e-HUB prototype – D3.2*, e-HUB Project internal document. Information Society Technology (IST 2001-34031), 2002
- FLORENZANO, M. C. *Gestão de desenvolvimento de produtos: estudo de casos na indústria brasileira de autopeças sobre a divisão de tarefas, capacidade e integração interunidades*. São Carlos. 135 p. Dissertação (Mestrado) - Universidade Federal de São Carlos, 1999.
- FOWLER, M., SCOTT, K. *UML distilled: Applying the standard Object Modeling Language*. Addison Wesley Longman Inc. Massachusetts, 1997.
- GARTNER, 2002 *Project/Resource Management Magic Quadrant*. (www.gartner.com), 24 July 2002.
- GIL, A. C. *Métodos e técnicas da pesquisa social*. São Paulo: Atlas, 1999.
- GUERRERO, V. *Análise do Gerenciamento de Informação em um Ambiente Colaborativo e Distribuído de Desenvolvimento de Produto*. Tese (Mestrado), Escola de Engenharia de São Carlos – USP, 2002.

HORTA, L. *Estudo do Processo de Alteração de Engenharia*. Exemplar da Qualificação (Mestrado), Escola de Engenharia de São Carlos – USP, 2001.

HORTA, L, ROZENFELD, H. *Desenvolvimento de um Cenário de Aplicação de Sistemas PLM no Processo de Desenvolvimento de Produtos*. III CBGDP, Florianópolis, 2001.

HUANG, J. C., NEWELL, S. Knowledge integration processes and dynamics within the context of cross-functional projects. *International Journal of Project Management* 21 (2003)

JUCÁ JUNIOR, A.S. *Gestão de Projetos nas empresas de base tecnológica de São Carlos: um estudo sobre a aplicabilidade do escritório de projetos*. Qualificação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2004.

KNAPP, M., *Integrated Project Method*, Knapp & Moore Pty Limited, Austrália, [www.knappandmoore.com.au](http://www.knappandmoore.com.au), data de acesso: 2002.

KOTONYA, G. SOMMERVILLE, I. *Requirements Engineering: Processes and techniques*. John Wiley & Sons, 1998

LARMAN, C. *Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design*. Prentice-Hall, Inc., 1999.

LIEBOWITZ, J. MEGBOLUGBE, I., *A set of frameworks to aid the project manager in conceptualizing and implementing knowledge management initiatives*, *International Journal of Project Management*, 2003.

LIMA, K.K. (2004). *Aplicação de Sistemas de Apoio à Gestão do Conhecimento em Redes de Pesquisa: o Caso Instituto Fábrica do Milênio*. Qualificação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2004.

LITTLER, D.; LEVERICK, F.; BRUCE, M. *Factors affecting the process of collaborative product development: a study of UK manufacturers of information and communications technology products*. *Journal of Product Innovation Management*, 1995.

LOVE, P. E. D., EDUM-FOTWE, F., IRANI, Z., *Management of knowledge in project environments*, editorial, *International Journal of Project Management*, 2003.

MANER, R. *Rapid Application Development*, <http://csweb.cs.bgsu.edu/maner/domains/RAD.htm#2>, acessado em jan/ 2005.

MARTINS, G. A. *Epistemologia da pesquisa em administração*. São Paulo, Tese (Livre-Docência) – Departamento de Administração, Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo, 1994.

MEREDITH, J. R., MANTEL, S. J. *Project management—a managerial approach*. New York: John Wiley & Sons; 1995

MORETTI D. et al. *Critérios para avaliação de sistemas de apoio à gestão de conhecimento*. XXV ENEGEP, Porto Alegre, 2005

NONAKA, I. *The knowledge-creating company*. Harvard Business Review, pags:96–104, 1991

NONAKA, I. & TAKEUCHI, H. (1997). *Criação de Conhecimento na Empresa*. Rio de Janeiro: Campus.

OPM3 – *Organizational Project Management Maturity Model*. Knowledge Foundation. Project Management Institute – PMI. Pennsylvania, EUA, 2003.

OSI (Opens Source Initiative). *OSI*, 225. <http://www.opensource.org/> acessado em jan/2005)

PÁDUA, E. M. M. (1996). *Metodologia de pesquisa: abordagem teórico-prática*. Campinas, SP, Papirus.

PMI, *A Guide to the Project Management Body of Knowledge (PMBOK ed. 2004)*, Project Management Institute Inc., Pennsylvania, USA, 2004.

PRESSMAN, R. *Engenharia de Software*. 5ª edição. Rio de Janeiro, McGraw-Hill, 2005.

RIEHLE, D. *Framework Design. A Role Modeling Approach*, Tese (Doutorado) - Swiss Federal Institute of Technology Zurich, 2000.

ROZENFELD, H. Reflexões sobre a Manufatura Integrada por Computador. *Manufatura de Classe Mundial: mitos & realidade*, São Paulo: 1996.

ROZENFELD, H.; AMARAL, D.C.; TOLEDO, J.C.; CARVALHO, J. O Processo de Desenvolvimento de Produtos. In: *Fábrica do Futuro: entenda hoje como a sua indústria vai ser amanhã*. Cap. 6, ROZENFELD, H.; BREMER, C.F. São Paulo: Banas, 2000.

ROZENFELD et al. *Integrando os conhecimentos em um PDP de três grupos de pesquisa: proposta de um modelo de referência e suas aplicações*. IV CBGDP, Gramado, 2003.

ROZENFELD et al. *Gestão do desenvolvimento de Produtos – Uma referência para a melhoria do processo*. Editora Saraiva, São Paulo, 2006

SANTOS, A. R. et al. *Gestão do Conhecimento: uma experiência para o sucesso empresarial*. Curitiba: Champagnat, 2001.

SHEVCHENKO, A. et al. *Formal Requirements Specification for e-HUBs*. e-HUB project internal document, Information Society Technology, 2003

SHINDLER, M & EPPLER, M. *Harvesting project knowledge: a review of project learning methods and success factors*. International Journal of Project Management, 2003.

SOMMERVILLE, I. *Software Engineering*. Addison Wesley – 8<sup>th</sup> edition, 2006.

TATIKONDA, M.V., ROSENTHAL, S.R., *Successful execution of product development projects: Balancing firmness and flexibility in the innovation process*, Journal of Operations Management, vol. 18 Pag. 401-425, 2000

TERRA, J.C.C. *Gestão do conhecimento: o grande desafio empresarial*. São Paulo: Negócio Editora, 2000.

THIOLLENT, M. *Metodologia da Pesquisa-ação*. São Paulo: Cortez, 1998.

VALERI, S. *Estudo do Processo de Revisão de Fases no Processo de Desenvolvimento de Produtos em uma Indústria Automobilística*, (Tese de Mestrado). Escola de Engenharia de São Carlos – USP, 2000.

VALERIANO, D. L., *Gerência em Projetos*. São Paulo: Makron Books, 1998.

VERNADAT, F. B. *Enterprise Modelling and Integration: Principles and Applications*. London: Chapman & Hall, 1996.

VERZUH, E., *The fast forward MBA in Project Management*. John Wiley & Sons, Inc., 1999.

VIDGEN, R., AVISON, D., WOOD, B., WOOD-HARPER, T., *Developing Web Information Systems*. Butterworth-Heinemann, França, 2002.

WHEELWRIGHT, S.C.; CLARK, K.B. *Revolutionizing Product Development: quantum leaps in speed, efficiency, and quality*. New York: The Free Press. 1992.

WHITE, D., FORTUNE, J. *Current practice in project management: an empirical study*. International Journal of Project Management, 2002.

## 10 APÊNDICES

A seguir são mostrados Apêndices desta dissertação:

- APÊNDICE A: Classificação final dos requisitos
- APÊNDICE B: listas de requisitos de diversas fontes
- APÊNDICE C: referência cruzada entre requisitos de software e modelo de referência em DP
- APÊNDICE D: lista final de requisitos
- APÊNDICE E: necessidades específicas por ator (diagramas de caso de uso)

## Apêndice A – Classificação final dos requisitos

Tabela 7: classificação final dos requisitos após comparação com o modelo de referência

COD	GRUPO	SUB-COD	SUB-GRUPO
<b>1</b>	<b>Arquitetura</b>	1.1	Integração entre módulos e com outros softwares
		1.2	Integração com base de dados
		1.3	Compatibilidade de hardware e plataforma
		1.4	Forma de acesso (desktop, web based, etc)
		1.5	Interface com usuário
		1.6	Acesso / segurança
		1.7	Disponibilidade / segurança das informações
		1.8	Manutenção do sistema
<b>2</b>	<b>Calendário e Agenda</b>	2.1	Personalização de calendário
		2.2	Integração da agenda entre projetos
		2.3	Integração da visão da agenda por atividade e por recurso
		2.4	Auxílio no agendamento de reuniões online
<b>3</b>	<b>Gestão de atividades</b>	3.1	Suporte a configuração de duração de atividades
		3.2	Determinação de hierarquia e precedências das atividades
		3.3	Criação / visualização de gráficos (PERT, Gantt, etc)
		3.4	Criação / visualização de WBS
		3.5	Milestones
		3.6	Criação, customização e enactment de workflows
<b>4</b>	<b>Gestão de recursos e usuários</b>	4.1	Configuração de recursos (cadastrar, definir características, etc)
		4.2	Relacionamento entre pessoa/recurso vs. Atividade/deliverable
		4.3	Visualização de disponibilidade de recursos / pessoas
		4.4	Nivelamento de recursos
		4.5	Padrões para alocação de recursos / pessoas
		4.6	Criação de times de projeto
		4.7	Criação de papéis para membros do projeto
		4.8	Aquisição de recursos
<b>5</b>	<b>Gestão de custos</b>	5.1	Custo total do projeto e por atividade
		5.2	Custo por recurso

	5.3	Ferramentas para estudo de viabilidade econômica
<b>6 Gestão de Documentos</b>	6.1	Busca avançada
	6.2	Meta-dados para documentos (link com atividades, best practices, grupos, pessoas, projetos, etc)
	6.3	Sistema de vault
	6.4	Controle de acesso
	6.5	Controle de versão
	6.6	Controle de alterações
	6.7	Associação de informação (documento, best practice, etc) a atividade / deliverable
	6.8	Geração automática de relatórios / consolidação de indicadores
	6.9	Configuração de impressão
<b>7 Gestão de Conhecimentos</b>	7.1	Inserção / divulgação de best practices / conhecimentos
	7.2	Associação de best practices / conhecimentos as atividades
	7.3	Validação de best practices / conhecimentos
	7.4	Listagem de especialistas
	7.5	Criação/divulgação/ministério de cursos online
	7.6	Criação de micro-artigos e associação à tarefa, atividade, fase, projeto e/ou deliverable
	7.7	Sistema integrado de coleta e armazenagem de aprendizados junto ao banco de dados (RECALL)
	7.8	Avaliação dos conhecimentos dos usuários
<b>8 Ferramentas de monitoramento</b>	8.1	Uso de base-lines diversas
	8.2	Monitoração de progresso de atividades
	8.3	Monitoração de indicadores pré-definidos
	8.4	Ferramentas de análise de valor agregado
	8.5	Mostrar real vs. Baseline em gráficos, etc
	8.6	Suporte a phase Gates
<b>9 Gestão de múltiplos projetos</b>	9.1	Visualização integrada de tarefas e recursos
	9.2	Compartilhamento de recurso entre projetos
	9.3	Listagem dos projetos (com filtros)
	9.4	Ferramentas de gestão de portfolio
<b>10 Padrões, Templates e modelos</b>	10.1	Criação / disponibilização de templates
	10.2	Customização dos templates
<b>11 Ferramentas de comunicação</b>	11.1	Ferramentas de comunicação síncrona
	11.2	Ferramentas de comunicação assíncrona
	11.3	Sistemas de alerta
	11.4	Ambiente online de projetos compartilhado
	11.5	Comunicação com agente externos (clientes, consultores, fornecedores)

	11.6	Registro de comunicações
	11.7	Configurações regionais
	11.8	Comunicação com outros aplicativos
<b>12 Outras ferramentas de suporte</b>	12.1	Ferramentas de suporte específico ao PDP
	12.2	Ajuda Online
<b>13 Gestão de riscos</b>	13.1	Listar/classificar riscos (técnicos/mercadológicos)
	13.2	Associar risco a deliverabel/atividades/tarefas
	13.3	Agrupar estatisticamente riscos do projeto / portfolio
	13.4	Ferramentas de simulação
<b>14 Gestão da mudança</b>	14.1	Definir responsável por autorizar / realizar mudança em atividade / deliverable
	14.2	Associar solicitação de mudanças a documento, deliverable, atividade ou fase

## Apêndice B – Listas de requisitos de diversas fontes

Tabela 8: requisitos levantados a partir de softwares de gestão de projetos

CÓDIGO	REQUISITO	CLASSIFICAÇÃO
RO-1	Integrar com banco de dados	1
RO-2	Permitir Múltiplos usuários / permissões	1
RO-3	Permitir diversas formas de acesso	1
RO-4	Trabalhar em plataforma	1
RI-19	Integrar o Word com a base de dados, para trazer o texto da base de dados	1
RI-22	Integrar com o MS Project	1
RI-23	Importar dados	1
RI-24	Permitir upload e download de arquivos do MS Project	1
RI-62	Buscar na base	1
RI-63	Buscar em páginas HTML	1
RI-64	Buscar nos arquivos armazenados (Word, Powerpoint)	1
RI-93	Obter dados do MS Project	1
RI-94	Ter tela amigável para cadastro da cotação	1
RI-95	Ter tela amigável para comparação de diferentes cotações	1
RI-96	Ter funcionalidade de prestação de contas	1
RI-97	Permitir usuários responsáveis pela compra de	1

	registrar-la aproveitando os dados da cotação	
RI-98	Integrar com a lista de recursos permitindo o acompanhamento da aquisição	1
RI-104	Possuir espaço compartilhado do projeto	1
RI-106	Administrar dados do projeto	1
RI-107	Integrar com outros sistemas	1
RI-108	Permitir criação de portais corporativos	1
RI-109	Possuir estrutura descentralizada	1
RI-110	Permitir criação de intranets	1
RO-5	Permitir programação do calendário do projeto	2
RO-6	Permitir programação do calendário de recursos	2
RO-7	Permitir programação do calendário de tarefas	2
RO-8	Visualizar agenda de tarefas de um determinado recurso	2
RO-9	Visualizar agenda de determinado projeto	2
RO-10	Visualizar agenda de tarefas que engloba todos projetos da organização	2
RI-51	Agendar reuniões com integração de agenda	2
RI-52	Anexar material apresentado na reunião	2
RI-53	Registrar resultado com comunicação automática	2
RI-54	Armazenar valores dos indicadores de desempenho na data do gate	2
RI-101	Registrar datas intermediárias de entrega e verificação	2
RO-11	Visualizar e formatar gráfico PERT	3
RO-12	Visualizar e formatar gráfico Gantt	3
RO-13	Gerar WBS	3
RO-14	Definir e visualizar níveis para as atividades	3
RO-15	Definir restrições às tarefas	3
RO-16	Configurar tarefas	3
RO-17	Visualizar das tarefas envolvendo todos os projetos	3
RO-18	Visualizar das tarefas do projeto	3
RO-19	Visualizar das tarefas do recurso	3
RO-20	Determinar de duração das tarefas por diversos métodos	3
RI-25	Cadastrar e alterar de atividades, tempo e precedência	3
RI-26	Permitir visualização simples de um projeto	3
RI-27	Permitir Gantt	3
RI-28	Permitir Gantt com baseline	3
RI-39	Permitir que as ações do projeto sejam vistas como um todo	3
RI-49	Permitir hierarquização das tarefas do projeto	3
RI-50	Permitir definição dos gates pelos usuários	3
RI-56	Permitir definição de tarefas e deliverables pelo usuário	3
RI-57	Armazenar resultado atual contendo data da última alteração e responsável	3
RI-59	Armazenar resultado atual contendo data da última alteração e responsável	3
RI-60	Ler e descarregar projetos individuais	3

RI-61	Consolidar projetos para controle	3
RO-21	Configurar recursos	4
RO-22	Visualizar disponibilidade de uso dos recursos	4
RO-23	Possuir ferramenta de nivelamento de recursos	4
RO-24	Possuir padrões de alocação de recursos	4
RO-25	Listar recursos	4
RO-26	Configurar recursos	4
RO-27	Visualizar disponibilidade de uso dos recursos	4
RO-28	Permitir Nivelamento de recursos, inclusive entre projetos	4
RO-29	Permitir padrões de alocação de recursos	4
RO-30	Listar recursos	4
RI-6	Criar times de projeto /comunidades	4
RI-7	Definir integrantes	4
RI-8	Definir papéis	4
RI-9	Cadastrar missão do time	4
RI-58	Controlar responsável pela manutenção	4
RI-91	Definir responsável pela compra	4
RI-92	Relacionar recursos com as atividades	4
RI-103	Registrar recebimento gerando automaticamente informação para os usuários e finalização da aquisição	4
RO-31	Calcular pagamento dos recursos	5
RO-32	Calcular custo das atividades	5
RO-33	Calcular orçamento do projeto	5
RO-34	Listar custos por atividade	5
RI-90	Criar listas e acompanhamento da aquisição	5
RO-35	Gerar relatórios	6
RO-36	Permitir impressão	6
RO-37	Possuir ferramenta de Gerenciamento Eletrônico de Documentos	6
RI-1	Controlar autor	6
RI-2	Controlar código	6
RI-3	Controlar versão	6
RI-4	Controlar data	6
RI-5	Adicionar comentários	6
RI-15	Permitir diversos tópicos flexíveis com títulos configuráveis	6
RI-16	Controlar data	6
RI-17	Controlar versão	6
RI-18	Comunicar alterações	6
RI-20	Permitir links aos documentos referenciados	6
RI-21	Permitir a Responsável pelo projeto controlar os acessos	6
RI-78	Permitir classificação por categoria	6
RI-79	Permitir classificação por palavras-chaves	6
RI-80	Permitir classificação por acesso permitido	6
RI-81	Permitir anexar documentos a qualquer registro	6

RI-82	Armazenar documentos na biblioteca	6
	Permitir trazer para o micro local qualquer documento do sistema (menos do cofre)	6
RI-83		6
RI-84	Corresponde automaticamente a um download	6
RO-38	Gravar linha base	8
RO-39	Monitorar progresso das atividades	8
RO-40	Analisar valor agregado para orçamento e custos reais	5
	Análisar valor agregado para variação de custos e prazos	5
RO-41		5
RO-42	Analisar valor agregado para índices de desempenho	5
RO-43	Analisar valor agregado para orçamentos e projeções	5
	Permitir compartilhamento de recursos entre múltiplos projetos	5
RO-44		5
RO-45	Visualizar e formatar diagrama de bolhas	9
RO-46	Possuir Scoring models	9
RO-47	Analisar parâmetros ECV e IDP	9
RO-48	Permitir agrupamento de projetos	9
RI-29	Permitir controle multiprojeto	9
RI-30	Listar todos os projetos	9
	Permitir ferramentas para a avaliação do status de cada projeto	7,9
RI-31		7,9
RI-32	Permitir ferramentas para gerenciamento de portfolio	9
RI-41	Aplicar prioridade a um projeto	9
RI-42	Definir critérios para a descrição dos projetos	9
RI-43	Permitir classificação e hierarquização de projetos	9
RI-44	Visualizar portfolio de projetos	9
	Visualizar a lista de projetos sobre responsabilidade do usuário	3
RI-45		3
RI-46	Visualizar a lista de projetos da empresa	9
RI-47	Possuir ferramentas para avaliação dos projetos	9
RI-48	Possuir gráfico de bolhas	9
RO-49	Permitir notificações por e-mail	11
RO-50	Exportar projetos e tarefas	1
RO-51	Importar projetos e tarefas	1
RO-52	Exportar listas de contatos	1
RO-53	Importar contatos	1
RO-54	Possuir fórum de discussões interno	11
RO-55	Possuir fórum de discussões na Internet	11
	Permitir comunicação entre usuários de diversas formas	11
RO-56		11
	Enviar automaticamente mensagem com o código de ética e instruções básicas	11
RI-10		11
RI-73	Permitir Publicação / divulgação de documentos	11
RI-74	Possuir ferramenta de E-mail	11
RI-75	Possuir fórum de discussão	11
RI-76	Possuir ferramenta de Chat	11
	Permitir inserção de comentários em documentos / registros	11
RI-77		11

RI-11	Permitir Inserção / deleção de best practices	7
RI-12	Permitir Edição de best practices	7
RI-13	Permitir Recuperação de best practice	7
RI-14	Possuir Sistemática de validação de best practice	7
RI-34	Permitir listagens amigáveis para o cadastro e visualização dos riscos	13
RI-35	Possuir tabela padrão com critérios para avaliação sistemática dos riscos	13
RI-37	Cadastrar ações de risco	13
RI-38	Acompanhar ações de risco	13
RI-65	Armazenar cursos online	7
RI-67	Verificar andamento do treinamento	7
RI-68	Possuir link para os cursos dos modelos de referência	7
RI-69	Possuir link para os cursos do manual da qualidade	7
RI-70	Possuir páginas de introdução ao conhecimento (páginas do Conhecimento)	7
RI-72	Permitir criação de cursos online	7
RI-88	Permitir a definição de padrões como valores de diárias, etc.	10
RI-89	Permitir criação de contas com nome, tipo, descrição, fonte de recursos, etc.	5
RI-99	Permitir integração com a base de conhecimentos (estes especialistas devem fazer parte da base)	7
RI-100	Possuir padrões de contrato e de pedidos de compra	10
RI-102	Registrar ordens de pagamento	Inexistente
RI-105	Permitir controle do fluxo de produtos	Inexistente
RI-33	Definir e avaliar riscos	13
RI-36	Permitir geração de lista de ações para diminuição de riscos	13
RI-40	Cadastrar e organizar projetos	9
RI-55	Permitir criação e acompanhamento de indicadores de desempenho por recurso	8
RI-66	Possuir mecanismo para automatizar relacionamento entre usuários e cursos necessários	7
RI-85	Possuir repositório de documentos com segurança, permitindo só cópias	6

Tabela 9: requisitos baseados nas funcionalidades do MS-Project

CÓDIGO	REQUISITOS	CLASSIFICAÇÃO
MS-1	Gerenciar tarefas, recursos e times de projeto	3,4
MS-2	Permitir cálculo do caminho crítico	3
MS-3	Permitir cálculo dos custos	5
MS-4	Permitir importação e exportação de tarefas, recursos e times de projetos	3
MS-5	Armazenar melhores práticas (best practices)	7
MS-6	Comunicar com o time de projeto através do Microsoft	1, 11

	Outlook	
MS-7	Converter listas de recursos, tarefas, etc. do Microsoft Excel	1
MS-8	Configurar alertas	13
MS-9	Monitorar progresso das atividades	8
MS-10	Analisar projeto com múltiplas baselines	8
MS-11	Visualizar gráfico das tarefas	3
MS-12	Detalhar projeto através de relatórios em forma de tabelas	3,11
MS-13	Compartilhar totalmente com aplicações Microsoft Office	1
MS-14	Possuir Integração com MS Outlook	1, 11
MS-15	Possuir Integração com Web para a publicação de informações do projeto	1, 6
MS-16	Possuir um servidor central onde são armazenadas as informações através de um banco de dados	1
MS-17	Identificar grupos de tarefas com risco de não se completarem na data prevista	13
MS-18	Possuir ferramenta de comunicação assíncrona (fórum, e-mail, etc.)	11
MS-19	Possuir ferramenta de comunicação síncrona	11
MS-20	Armazenar informações documentos	6
MS-21	Possuir sistema de acesso controlado a documentos	6
MS-22	Possuir sistema de versão para documentos	6
MS-23	Permitir compartilhamento do espaço de trabalho (painéis)	6, 11

Tabela 10: requisitos desenvolvidos durante a elaboração do framework para sistematização de conhecimentos do grupo EI

<b>CODIGO</b>	<b>REQUISITOS</b>	<b>CLASSIFICAÇÃO</b>
C-34	Permitir controle de acesso dos usuários	1
C-35	Permitir controle de acesso dos documento por grupo / papel	1
C-36	Possuir sistema de busca	6
C-11	Definir plano de projeto (Gantt)	6
C-12	Comparar status das atividades com plano original	8
C-13	Identificar efeitos de possíveis atrasos no plano	8
C-16	Propor milestones (membro)	8, 9
C-17	Propor cronograma (membro)	3
C-18	Autorizar milestone (gerente)	3
C-19	Autorizar cronograma (gerente)	3
C-20	Informar status de cada atividade (membro)	8
C-21	Autorizar status de cada atividade (gerente)	8
C-22	Definir deliverables	3

C-23	Associar deliverable (documento) a atividade / fase / projeto	3
C-24	Criar projetos / subprojetos	9
C-29	Cancelar projeto	9
C-30	Congelar / descongelar projeto	9
C-2	Permitir acesso a lista de pessoas com mesmo interesse / especialistas	7
C-25	Alocar pessoas ao projeto	4
C-26	Verificar tempo livre das pessoas do projeto	4
C-15	Associar conhecimentos ao projeto	6
C-28	Possuir ferramenta para gestão de portfolio	9
C-3	Permitir comunicação entre usuários	11
C-4	Possuir ferramentas para assistir / ministrar curso online	7
C-14	Possuir ferramenta de reuniões virtuais de avaliação de atividade / fase / projeto	11
C-27	Criar grupo de interesse para projetos	11
C-31	Permitir chat síncrono entre usuários	11
C-32	Permitir avisos na página do projeto	11
C-33	Possuir software de alertas	11
C-1	Permitir acesso a soluções existentes na área de interesse	7
C-5	Possuir software de avaliação (ranking / rating) de especialistas / conhecimentos	7
C-6	Permitir registro / acesso de melhores práticas	7
C-7	Permitir referência cruzada, para sugerir melhores práticas	7
C-8	Possuir software para avaliação dos conhecimentos do usuários	7
C-9	Possuir sistema deve conter no seu conteúdo que suporte decisões	7
C-10	Permitir cadastro / acesso a currícula	7

Tabela 11: requisitos derivados do projeto e-HUBs

CÓDIGO	REQUISITOS	CLASSIFICAÇÃO
E1-4	Registrar comunicações feitas através do sistema	1
E1-16	Possuir capacidade de fazer backups regulares	1
E1-17	Permitir comunicação segura via Net	1
E1-18	Mecanismos de autenticação dos usuários / controle de acesso	1
E1-24	Conversão de documentos e formatos predefinidos	1
E1-25	Compatibilidade com sistema MS Office	1
E1-26	Sistema de segurança contra invasão	1
E1-27	Deve usar inglês como língua padrão	1

E1-28	Sistema deve permitir documentação em outras línguas	1
E1-29	Customização de menus em outras línguas	1
E2-1	Registro de usuários e controle de acesso à plataforma	1
E2-2	Registro de perfil do usuário	1
E2-11	Backups regulares	1
E2-12	EAI (Enterprise Application Integration)	1
E2-13	Uso de diferentes conjuntos de caracteres	1
E2-14	Suporte a diferentes formatos de data e fusos horários	1
E2-15	Segurança dos documentos por meio de criptografia	1
E2-16	Autenticação cruzada para login	1
E2-17	Compatibilidade com diferentes browsers	1
E2-18	Compatibilidade com diferentes PCs	1
E2-19	Tecnologia de busca avançada	1
E2-20	Suporte a mobile / wireless	1
E2-21	Habilidade de se integrar com outros softwares / fontes de informação	1
E2-24	Customização do perfil do grupo / projeto	1
E2-29	Interface gráfica	1
E2-30	Sistema de filtro e ordenamento das informações / documentos	1
E2-31	Alta disponibilidade	1
E2-32	Flexibilidade de customização do sistema	1
E2-33	Fácil manutenção	1
E2-34	Facilidade de operação	1
E2-35	Habilidade de incorporar informações em formato multimídia	1
E2-43	Diferentes perfil de acesso para gerentes	1
E3-14	Pesquisa avançada por palavras-chave, autor, data, tamanho do documento, tipo de serviço, projeto, etc.	6
E3-16	Administradores devem ter fácil acesso a informações estatísticas sobre o uso do software por projeto	1
E2-23	Criação de diferentes workgroups dentro de cada projeto	3
E3-15	Gerentes devem ter fácil acesso a informações estatísticas sobre o projeto	3
E1-12	Auxílio no recrutamento da equipe	4
E2-6	"Who is who" facilities	7
E2-28	Diretório de usuários	4
E1-9	Avaliação de custos do projeto	5
E1-5	Registro de toda documentação do projeto	6

E1-6	Campos predefinidos de metadados para permitir comparação	6
E1-21	Gestão de versão nos documentos	6
E1-22	Processamento / elaboração colaborativa de documentos	6
E2-8	Controle de acesso por documento	6
E2-9	Repositório de documentos (vault)	6
E2-10	Controle de versão	6
E2-36	Biblioteca	6
E2-39	Upload / download de arquivos no formato original	6
E2-40	Upload / download de arquivos no formato zip ou rar, etc.	6
E2-41	Visualização dos documentos informações na tela, sem necessidade de download	6
E2-45	"Bulk" downloading / uploading na biblioteca	6
E2-46	Acesso a documentos / newsgroups por e-mail	6
E2-22	Controle centralizado de diversos grupos / projetos	8
E1-19	Deve apoiar comunicação por voz e vídeo	11
E1-20	Mecanismo de notificação flexível para gerenciamento da equipe e processamento de documentos	11
E2-4	Comunicação assíncrona (e-mail)	11
E2-5	Grupos de discussão	11
E2-7	Suporte a calendário e web meetings	2
E2-25	Customização da área de trabalho	11
E2-26	Mecanismo de notificação	11
E2-27	Comunicação síncrona	11
E2-37	Painéis de publicação de notícias	11
E2-38	Facilitador para agendamento de "meetings"	2
E2-42	Ajuda online	1
E3-11	Conclusão de uma atividade deve disparar automaticamente alertas	3
E3-13	Permitir criação / uso / acesso a fórum de discussão	11
E1-3	Registro completo dos usuários	1,4
E2-44	Roteamento de documento para membros do grupo	6,3
E1-1	Templates de perfil de fornecedores	10
E1-2	Templates devem ser semi-automatizáveis	10
E1-7	Template para AD (Design da Arquitetura do produto)	10
E1-8	Template customizável para plano de projeto	10
E1-10	Fornecer forma de mediação para geração de contratos	6
E1-11	Fornecer ou indicar ferramentas para estudo de viabilidade de projeto	5
E1-13	Detalhamento do CRS (Customer Requirement Specification)	N

E1-14	Avaliação da performance do time e stakeholders	7
E1-15	Solicitar compartilhamento do conhecimento aos usuários no fim do projeto	7
E1-23	Modelos de documentos elaborados para cada fase / atividade padrão	10
E3-1	Desenhar graficamente os workflows de cada etapa do processo	3
E3-2	Alocar usuários do sistema a cada atividade do workflow	4
E3-3	Integração com base de dados e ferramentas do e-HUB	1
E3-4	Permitir desenho de workflow específico a partir de um workflow padrão	3
E3-5	Permitir atividades em paralelo ou em série	3
E3-6	Definir condições para realização / aprovação de cada atividade	3, 8, 9
E3-7	Permitir definição de prazos para cumprimento das atividades	3, 8
E3-8	Permitir uso de informações sobre atividades da WBS para desenho inicial do workflow	3
E3-9	Incluir templates dos principais documentos do projeto (Project Charter, etc.)	10
E3-10	Permitir visualização do status de cada atividade do workflow	3
E3-12	Conclusão de uma atividade deve disparar automaticamente ferramentas associadas a plataforma	3, 11
E2-3	Controle de acesso por grupos de usuários e papéis	1, 4





revisão/desenvolvimento do PEN														
Definir o prazo final	1	0	1	0	0	0	0	0	0	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Compilar a declaração de escopo do PEN	2	0	0	0	0	1	0	0	0	1	0	0	0	0
Planejar atividades para a revisão do PEN	9	2	1	1	0	1	0	0	0	1	2	0	1	0
Planejar atividades	1	0	1	0	0	0	0	0	0	0	0	0	0	0
Definir agendas de decisões e discussões	2	1	0	0	0	0	0	0	0	0	1	0	0	0
Planejar os recursos físicos necessários em cada reunião	3	1	0	1	0	0	0	0	0	0	1	0	0	0
Preparar plano de comunicação e de riscos	3	0	0	0	0	1	0	0	0	1	0	0	1	0
Consolidar Informações sobre tecnologia e mercado	3	0	0	0	0	2	0	0	0	1	0	0	0	0
Consolidar Informações de fontes de dados secundárias	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Identificar necessidade de dados primários	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Planejar coleta de dados primários	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Consolidar informações	2	0	0	0	0	1	0	0	0	1	0	0	0	0
Revisar o PEN	12	0	0	2	1	4	0	2	2	1	0	0	0	0
Revisar Missão	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Revisar segmentação do Mercado	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Revisar tendências tecnológicas	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Revisar posicionamento no Mercado	3	0	0	0	0	1	0	1	1	0	0	0	0	0
Revisar direcionamento da UM	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Revisar competências	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Revisar Recursos Necessários	3	0	0	1	1	0	0	1	0	0	0	0	0	0
Revisar Metas	1	0	0	0	0	0	0	0	1	0	0	0	0	0
Preparar Documento	2	0	0	0	0	1	0	0	0	1	0	0	0	0

<b>Analisar o Portfólio de Produtos da Empresa</b>	11	0	0	0	0	3	0	2	6	0	0	0	0	0
<b>Revisar/definir metodologia de avaliação de portfólio</b>	3	0	0	0	0	1	0	1	1	0	0	0	0	0
<b>Avaliar o posicionamento dos produtos</b>	3	0	0	0	0	1	0	1	1	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
<b>Avaliar o desempenho dos produtos</b>	1	0	0	0	0	0	0	0	1	0	0	0	0	0
<b>Avaliar tecnologias e plataformas utilizadas</b>	1	0	0	0	0	0	0	0	1	0	0	0	0	0
<b>Consolidar lista de idéias de novos produtos</b>	2	0	0	0	0	1	0	0	1	0	0	0	0	0
<b>Analisar projetos</b>	1	0	0	0	0	0	0	0	1	0	0	0	0	0
<b>Propor mudanças no portfólio de produtos</b>	9	0	0	0	0	2	0	0	5	2	0	0	0	0
<b>Identificar produtos a serem descontinuados</b>	1	0	0	0	0	0	0	0	1	0	0	0	0	0
<b>Identificar projetos a serem abandonados e congelados</b>	1	0	0	0	0	0	0	0	1	0	0	0	0	0
<b>Identificar novos projetos que deverão ser iniciados</b>	1	0	0	0	0	0	0	0	1	0	0	0	0	0
<b>Preparar minutas para cada um dos novos projetos</b>	3	0	0	0	0	1	0	0	1	1	0	0	0	0
<b>Consolidar o novo portfólio de produtos</b>	3	0	0	0	0	1	0	0	1	1	0	0	0	0
<b>Verificar viabilidade do portfólio de produtos</b>	7	1	0	2	1	1	0	0	1	0	1	0	0	0
<b>Avaliar viabilidade econômica do portfólio de projetos</b>	2	0	0	0	1	0	0	0	1	0	0	0	0	0
<b>Avaliar disponibilidade de recursos</b>	1	0	0	1	0	0	0	0	0	0	0	0	0	0
<b>Avaliar competências</b>	1	0	0	1	0	0	0	0	0	0	0	0	0	0
<b>Obter consenso sobre decisão final</b>	3	1	0	0	0	1	0	0	0	0	1	0	0	0

Decidir início do planejamento de um produto do portfólio	6	0	1	1	0	1	0	1	1	0	1	0	0	0
Monitorar o portfólio de produtos (avançado) e identificar a data dos novos projetos	2	0	0	0	0	0	0	1	1	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Revisar a minuta do projeto, definir o gerente de projeto, aprovar e realizar a comunicação	4	0	1	1	0	1	0	0	0	0	1	0	0	0
Planejamento do Projeto	70	4	12	10	8	7	2	5	4	5	3	2	8	0
Definir interessados do projeto	4	1	1	2	0	0	0	0	0	0	0	0	0	0
Planejamento organizacional dos interessados do projeto	3	1	1	1	0	0	0	0	0	0	0	0	0	0
Montagem da equipe com os interessados do projeto	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Desenvolvimento da equipe para a execução do projeto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir escopo do produto	6	1	0	0	0	1	0	0	1	1	1	1	0	0
Realizar reuniões para estudo da minuta de projeto e do portfólio de produtos	3	1	0	0	0	0	0	0	0	1	1	0	0	0
Definir diretrizes básicas que o produto deverá atender	3	0	0	0	0	1	0	0	1	0	0	1	0	0
Definir escopo do projeto	4	1	0	0	0	1	0	0	1	1	0	0	0	0
Realizar reuniões para preparação da declaração do escopo do produto e das restrições impostas pelo DP da empresa	4	1	0	0	0	1	0	0	1	1	0	0	0	0

Preparar declaração de escopo	3	0	1	1	0	1	0	0	0	0	0	0	0	0
Preparar EDT	2	0	1	1	0	0	0	0	0	0	0	0	0	0
Revisar a declaração do escopo do projeto	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Adaptar o modelo de referência	3	0	0	0	0	2	1	0	0	0	0	0	0	0
Classificar o projeto	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Identificar a versão adaptada do modelo	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Identificar necessidades de mudanças	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir atividades e seqüência	6	0	3	2	0	1	0	0	0	0	0	0	0	0
Identificar atividades	1	0	1	0	0	0	0	0	0	0	0	0	0	0
Definir relacionamentos entre as atividade	3	0	1	1	0	1	0	0	0	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Analisar a rede do projeto	2	0	1	1	0	0	0	0	0	0	0	0	0	0
Preparar cronograma	8	0	3	3	0	0	0	0	0	1	1	0	0	0
Estimar esforço necessário para a atividade	2	0	1	1	0	0	0	0	0	0	0	0	0	0
Alocar recursos necessários	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Otimizar a programação de atividades e recursos	2	0	1	1	0	0	0	0	0	0	0	0	0	0
Imprimir cronograma	3	0	1	0	0	0	0	0	0	1	1	0	0	0
Avaliar riscos	9	0	1	0	0	0	0	2	0	0	0	0	6	0
Planejar avaliação de risco do projeto de DP	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Identificar e caracterizar os riscos potenciais	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Analisar qualitativamente os riscos potenciais	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Analisar quantitativamente os riscos potenciais	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Planejar ações em resposta aos riscos potenciais	3	0	1	0	0	0	0	1	0	0	0	0	1	0

Planejar o controle e a monitoração de riscos	2	0	0	0	0	0	0	1	0	0	0	0	1	0
Preparar estimativa de orçamento do projeto com custos	4	0	1	1	2	0	0	0	0	0	0	0	0	0
Previsões dos custos relacionados às atividades e recursos	3	0	1	1	1	0	0	0	0	0	0	0	0	0
Alocação orçamentária dos custos estimados	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Analisar a viabilidade econômica do projeto	6	0	0	0	4	0	0	1	0	0	0	1	0	0
Definir custo-alvo	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Verificar manufacturabilidade do custo-alvo	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Definir volume de vendas	1	0	0	0	1	0	0	0	0	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Realizar avaliação econômica	3	0	0	0	1	0	0	1	0	0	0	1	0	0
Definir indicadores de desempenho	2	0	0	0	0	0	0	1	1	0	0	0	0	0
Selecionar indicadores de desempenho mais adequados para o presente projeto	2	0	0	0	0	0	0	1	1	0	0	0	0	0
Definir plano de comunicação	4	0	1	1	0	1	0	0	0	0	1	0	0	0
Determinar que, quando e como as partes envolvidas precisam de informações e comunicações	4	0	1	1	0	1	0	0	0	0	1	0	0	0
Planejar e preparar aquisições	4	0	1	0	1	0	0	0	0	1	0	0	1	0
Planejar o que será adquirido e quando	3	0	1	0	1	0	0	0	0	0	0	0	1	0
Preparar requerimentos de aquisição e identificar fornecedores potenciais	1	0	0	0	0	0	0	0	0	1	0	0	0	0
Planejar a gestão dos relacionamentos com os fornecedores	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Preparar Plano de Projeto	10	1	1	1	1	1	1	1	1	1	0	0	1	0



necessárias														
Pesquisar padrões / normas, patentes e legislação	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Pesquisar produtos concorrentes e similares	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Detalhar ciclo de vida do produto e definir seus clientes	4	0	1	0	0	1	0	0	2	0	0	0	0	0
Refinar o ciclo de vida do produto	3	0	1	0	0	1	0	0	1	0	0	0	0	0
Definir os clientes do projeto ao longo do ciclo de vida	1	0	0	0	0	0	0	0	1	0	0	0	0	0
Identificar os requisitos dos clientes do produto	4	0	0	0	0	1	0	0	0	0	0	3	0	0
Coletar as necessidades dos clientes de cada fase do ciclo de vida	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Agrupar e classificar as necessidades	2	0	0	0	0	1	0	0	0	0	0	1	0	0
Definir os requisitos dos clientes	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Valor dos requisitos dos clientes	1	0	0	0	0	0	0	0	0	0	0	1	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Definir requisitos do produto	3	0	0	0	0	0	0	0	0	0	0	3	0	0
Converter requisitos de clientes em expressões mensuráveis	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Analisar e classificar os requisitos do produto	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Hierarquizar requisitos de projeto do produto	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Definir especificações meta do produto	7	0	0	0	0	2	0	0	0	1	0	4	0	0
Valorar requisitos do produto	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Analisar perfil técnico e de mercado	1	0	0	0	0	0	0	0	0	0	0	1	0	0

Analisar restrições de projeto do produto (contrato, ambientais, legislação, normas, ...)	2	0	0	0	0	1	0	0	0	0	0	1	0	0
Elaborar o conjunto de especificações-meta do produto	3	0	0	0	0	1	0	0	0	1	0	1	0	0
Monitorar a viabilidade econômico-financeira do produto	15	0	2	0	4	0	0	7	2	0	0	0	0	0
Avaliar mudanças nas condições de mercado, características técnicas e indicadores	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Avaliar impacto das mudanças no projeto	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Atualizar as premissas financeiras do projeto	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Atualizar custo-alvo nos diversos níveis do produto	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Atualizar as receitas futuras	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Atualizar as necessidades de investimentos	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Calcular o novo fluxo de caixa	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Calcular novos indicadores financeiros	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Analisar novas premissas e indicadores	3	0	0	0	1	0	0	1	1	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Avaliar possíveis desvios e impactos no projeto	3	0	1	0	0	0	0	1	1	0	0	0	0	0
Avaliar fase	13	0	1	0	1	1	0	4	2	1	0	0	2	1
Avaliar o cumprimento das tarefas planejadas	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Avaliar os resultados obtidos segundo os critérios estabelecidos	2	0	0	0	0	0	0	1	1	0	0	0	0	0
Avaliar os demais critérios quantitativos e indicadores	3	0	0	0	0	0	0	1	1	0	0	0	1	0

Avaliar a viabilidade econômica	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Decidir implementar ações corretivas	2	0	0	0	0	0	0	1	0	0	0	0	0	1
Decidir se pode ser realizada a atividade de aprovação	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Preparar relatório para time de avaliação	2	0	0	0	0	1	0	0	0	1	0	0	0	0
Aprovar fase	25	0	1	0	1	3	1	2	8	6	1	0	2	0
Avaliar o relatório de auto-avaliação e discutí-lo com o time de desenvolvimento	3	0	0	0	0	1	0	0	1	0	1	0	0	0
Analisar as evidências dos resultados e término das atividades planejadas	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Analisar o portfólio de produtos e projetos	3	0	0	0	0	1	0	1	1	0	0	0	0	0
Analisar o estudo de viabilidade econômica	2	0	0	0	1	1	0	0	0	0	0	0	0	0
Aplicar os critérios de avaliação e tomar decisão	4	0	0	0	0	0	1	0	1	1	0	0	1	0
<aprovar> Preparar relatório	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<aprovar> Melhorar o processo do gate	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<aprovar> Ajustar critérios próxima fase	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<redirecionar> Definir ações corretivas	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<redirecionar> Analisar risco	3	0	0	0	0	0	0	0	1	1	0	0	1	0
Documentar as decisões tomadas e registrar lições aprendidas	4	0	0	0	0	1	1	0	0	1	0	0	1	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Projeto Conceitual / Conceptual Design	97	0	7	3	10	17	2	24	15	8	1	1	8	1
Atualizar o Plano do Projeto Conceitual	30	0	3	3	3	7	0	10	2	0	0	0	2	0
Analisar o plano de projeto atual	2	0	1	0	0	1	0	0	0	0	0	0	0	0
Analisar e sintetizar as novas condições para a realização do projeto	4	0	1	1	0	1	0	1	0	0	0	0	0	0



Definir portadores de efeito	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Desenvolver as alternativas de solução para o produto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir arquitetura para o produto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Identificar Sistemas, Subsistemas e Componentes (SSC)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir integração entre SSCs das alternativas de projeto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Analisar Sistemas, Subsistemas e Componentes (SSC)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Identificar e analisar aspectos críticos do produto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir parâmetros principais (forma, materiais, dimensões e capacidades)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir ergonomia e estética	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir fornecedores e parcerias de co-desenvolvimento	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Selecionar a concepção do produto	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Analisar as concepções alternativas	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Valorar as concepções alternativas	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Selecionar concepções mais adequadas	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Planejar o processo de manufatura macro/Definir plano macro de processo	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Atualizar estudo de viabilidade econômica	5	0	0	0	1	1	0	1	1	0	0	0	1	0
Monitorar a viabilidade econômico-financeira do produto	15	0	2	0	4	0	0	7	2	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>

Avaliar mudanças nas condições de mercado, características técnicas e indicadores	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Avaliar impacto das mudanças no projeto	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Atualizar as premissas financeiras do projeto	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Atualizar custo-alvo nos diversos níveis do produto	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Atualizar as receitas futuras	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Atualizar as necessidades de investimentos	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Calcular o novo fluxo de caixa	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Calcular novos indicadores financeiros	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Analisar novas premissas e indicadores	3	0	0	0	1	0	0	1	1	0	0	0	0	0
Avaliar possíveis desvios e impactos no projeto	3	0	1	0	0	0	0	1	1	0	0	0	0	0
Avaliar fase	13	0	1	0	1	1	0	4	2	1	0	0	2	1
Avaliar o cumprimento das tarefas planejadas	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Avaliar os resultados obtidos segundo os critérios estabelecidos	2	0	0	0	0	0	0	1	1	0	0	0	0	0
Avaliar os demais critérios quantitativos e indicadores	3	0	0	0	0	0	0	1	1	0	0	0	1	0
Avaliar a viabilidade econômica	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Decidir implementar ações corretivas	2	0	0	0	0	0	0	1	0	0	0	0	0	1
Decidir se pode ser realizada a atividade de aprovação	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Preparar relatório para time de avaliação	2	0	0	0	0	1	0	0	0	1	0	0	0	0
Aprovar fase	25	0	1	0	1	3	1	2	8	6	1	0	2	0
Avaliar o relatório de auto-avaliação e discuti-lo com o time de desenvolvimento	3	0	0	0	0	1	0	0	1	0	1	0	0	0

FASE / ATIVIDADE / TAREFA	SOMA	2	3	4	5	6	7	8	9	10	11	12	13	14
Analisar as evidências dos resultados e término das atividades planejadas	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Analisar o portfólio de produtos e projetos	3	0	0	0	0	1	0	1	1	0	0	0	0	0
Analisar o estudo de viabilidade econômica	2	0	0	0	1	1	0	0	0	0	0	0	0	0
Aplicar os critérios de avaliação e tomar decisão	4	0	0	0	0	0	1	0	1	1	0	0	1	0
<aprovar> Preparar relatório	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<aprovar> Melhorar o processo do gate	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<aprovar> Ajustar critérios próxima fase	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<redirecionar> Definir ações corretivas	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<redirecionar> Analisar risco	3	0	0	0	0	0	0	0	1	1	0	0	1	0
Documentar as decisões tomadas e registrar lições aprendidas	4	0	0	0	0	1	1	0	0	1	0	0	1	0
Projeto Detalhado / Detailed Design	164	0	9	5	15	36	16	22	24	17	3	2	10	5
Atualizar o Plano do Projeto Detalhado	30	0	3	3	3	7	0	10	2	0	0	0	2	0
Analisar o plano de projeto atual	2	0	1	0	0	1	0	0	0	0	0	0	0	0
Analisar e sintetizar as novas condições para a realização do projeto	4	0	1	1	0	1	0	1	0	0	0	0	0	0
Atualizar o escopo do produto	2	0	0	0	0	1	0	1	0	0	0	0	0	0
Atualizar e detalhar o escopo do projeto	2	0	0	0	0	1	0	1	0	0	0	0	0	0
Atualizar e detalhar as atividades, os responsáveis, os prazos e o cronograma	3	0	1	1	0	0	0	1	0	0	0	0	0	0
Atualizar e detalhar recursos necessários	2	0	0	1	0	0	0	1	0	0	0	0	0	0
Atualizar estimativa de orçamento do projeto	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Atualizar, monitorar, valorar e definir novos indicadores de desempenho	3	0	0	0	0	1	0	1	1	0	0	0	0	0

Analisar a viabilidade econômico-financeira do projeto	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Avaliar novos riscos	1	0	0	0	0	0	0	0	0	0	0	0	1	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Atualizar plano de comunicação	2	0	0	0	0	1	0	1	0	0	0	0	0	0
Planejar, atualizar e preparar novas aquisições	3	0	0	0	1	0	0	1	0	0	0	0	1	0
Definir/atualizar os critérios de passagem dos gates	3	0	0	0	0	1	0	1	1	0	0	0	0	0
Criar e detalhar SSCs, documentação e configuração	5	0	0	0	0	3	1	0	0	1	0	0	0	0
Criar, reutilizar, procurar e codificar SSCs	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Calcular e desenhar SSCs	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Especificar tolerancias	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Integrar os SSCs	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Finalizar desenhos e documentos	2	0	0	0	0	1	0	0	0	1	0	0	0	0
Completar BOM	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Decidir por fazer ou comprar SSC	6	0	1	1	2	1	1	0	0	0	0	0	0	0
Levantar informações de custos, tempo, capacidades e competências para o desenvolvimento/fornecimento dos SSCs	5	0	1	1	1	1	1	0	0	0	0	0	0	0
Orçar os SSCs dos fornecedores	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Decidir entre desenvolver e produzir ou comprar SSC	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Estimar os custos dos SSCs para a empresa	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Desenvolver fornecedores	5	0	0	0	0	1	2	1	0	0	1	0	0	0
Selecionar fornecedores	1	0	0	0	0	0	1	0	0	0	0	0	0	0

Enviar/atualizar especificações do produto	3	0	0	0	0	1	0	1	0	0	1	0	0	0
Avaliar amostras dos SSC recebidos	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Homologar fornecedores	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Planejar o processo de fabricação e montagem	16	0	0	0	0	5	6	1	0	3	0	1	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Planejar processo de fabricação macro	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Planejar processo de montagem macro	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Desdobrar parametros criticos dos componentes fabricados	2	0	0	0	0	0	1	0	0	0	0	1	0	0
Reutilizar planos de processo existentes	2	0	0	0	0	1	0	0	0	1	0	0	0	0
Definir / Avaliar componente em bruto	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir e Sequenciar operacoes	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Selecionar / Especificar máquinas e equipamentos	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Selecionar / Especificar pessoal e habilidades	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Especificar fixacao	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Especificar inspecao	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Selecionar / Especificar metodos	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Selecionar / Especificar ferramental	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Calcular sobremetal	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Calcular parametros de trabalho	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Descrever instruções de trabalho	2	0	0	0	0	1	0	0	0	1	0	0	0	0
Ilustrar operações	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Obter programa CNC	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Criar informações / documentos de apoio	2	0	0	0	0	1	0	0	0	1	0	0	0	0













Atualizar as receitas futuras	2	0	0	0	1	0	0	0	1	0	0	0	0	0
Atualizar as necessidades de investimentos	2	0	0	0	1	0	0	0	1	0	0	0	0	0
Calcular o novo fluxo de caixa	2	0	0	0	1	0	0	0	1	0	0	0	0	0
Calcular novos indicadores financeiros	2	0	0	0	1	0	0	0	1	0	0	0	0	0
Analisar novas premissas e indicadores	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Avaliar possíveis desvios e impactos no projeto	6	0	1	1	1	0	0	1	1	0	0	0	1	0
<b>Avaliar fase</b>	<b>13</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>2</b>	<b>1</b>
Avaliar o cumprimento das tarefas planejadas	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Avaliar os resultados obtidos segundo os critérios estabelecidos	2	0	0	0	0	0	0	1	1	0	0	0	0	0
Avaliar os demais critérios quantitativos e indicadores	3	0	0	0	0	0	0	1	1	0	0	0	1	0
Avaliar a viabilidade econômica	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Decidir implementar ações corretivas	2	0	0	0	0	0	0	1	0	0	0	0	0	1
Decidir se pode ser realizada a atividade de aprovação	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Preparar relatório para time de avaliação	2	0	0	0	0	1	0	0	0	1	0	0	0	0
<b>Aprovar fase- liberar producao</b>	<b>25</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>3</b>	<b>1</b>	<b>2</b>	<b>8</b>	<b>6</b>	<b>1</b>	<b>0</b>	<b>2</b>	<b>0</b>
Avaliar o relatório de auto-avaliação e discutí-lo com o time de desenvolvimento	3	0	0	0	0	1	0	0	1	0	1	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Analisar as evidências dos resultados e término das atividades planejadas	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Analisar o portfólio de produtos e projetos	3	0	0	0	0	1	0	1	1	0	0	0	0	0
Analisar o estudo de viabilidade econômica	2	0	0	0	1	1	0	0	0	0	0	0	0	0
Aplicar os critérios de avaliação e tomar decisão	4	0	0	0	0	0	1	0	1	1	0	0	1	0

<aprovar> Preparar relatório	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<aprovar> Melhorar o processo do gate	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<aprovar> Ajustar critérios próxima fase	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<redirecionar> Definir ações corretivas	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<redirecionar> Analisar risco	3	0	0	0	0	0	0	0	1	1	0	0	1	0
Documentar as decisões tomadas e registrar lições aprendidas	4	0	0	0	0	1	1	0	0	1	0	0	1	0
Lançamento do Produto	147	0	10	5	25	21	10	24	20	16	4	0	9	3
Planejar lançamento	30	0	3	3	3	7	0	10	2	0	0	0	2	0
Analisar o plano de projeto atual	2	0	1	0	0	1	0	0	0	0	0	0	0	0
Analisar e sintetizar as novas condições para a realização do projeto	4	0	1	1	0	1	0	1	0	0	0	0	0	0
Atualizar o escopo do produto	2	0	0	0	0	1	0	1	0	0	0	0	0	0
Atuaizar e detalhar o escopo do projeto	2	0	0	0	0	1	0	1	0	0	0	0	0	0
Atualizar e detalhar as atividades, os responsáveis, os prazos e o cronograma	3	0	1	1	0	0	0	1	0	0	0	0	0	0
Atualizar e detalhar recursos necessários	2	0	0	1	0	0	0	1	0	0	0	0	0	0
Atualizar estimativa de orçamento do projeto	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Atualizar, monitorar, valorar e definir novos indicadores de desempenho	3	0	0	0	0	1	0	1	1	0	0	0	0	0
Analisar a viabilidade econômico-financeira do projeto	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Avaliar novos riscos	1	0	0	0	0	0	0	0	0	0	0	0	1	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Atualizar plano de comunicação	2	0	0	0	0	1	0	1	0	0	0	0	0	0
Planejar, atualizar e preparar novas aquisições	3	0	0	0	1	0	0	1	0	0	0	0	1	0

Definir/atualizar os critérios de passagem dos gates	3	0	0	0	0	1	0	1	1	0	0	0	0	0
Desenvolver processo de vendas	9	0	0	0	2	2	2	0	0	3	0	0	0	0
desenhar processo de vendas	2	0	0	0	0	1	0	0	0	1	0	0	0	0
adquirir recursos	1	0	0	0	1	0	0	0	0	0	0	0	0	0
preparar documentacao comercial	2	0	0	0	0	1	0	0	0	1	0	0	0	0
desenvolver sistema de apoio a vendas	1	0	0	0	0	0	0	0	0	1	0	0	0	0
contratar/ alocar pessoal	1	0	0	0	1	0	0	0	0	0	0	0	0	0
treinar força de venda	1	0	0	0	0	0	1	0	0	0	0	0	0	0
treinar pessoal de apoio a venda	1	0	0	0	0	0	1	0	0	0	0	0	0	0
implantar processo de vendas	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Desenvolver processo de distribuicao	5	0	0	0	1	1	1	0	0	1	1	0	0	0
desenhar processo de distribuição	2	0	0	0	0	1	0	0	0	1	0	0	0	0
definir logistica do processo	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fechar acordos com distribuidores	1	0	0	0	0	0	0	0	0	0	1	0	0	0
adquirir recursos	1	0	0	0	1	0	0	0	0	0	0	0	0	0
desenvolver sistema de apoio a distribuição	0	0	0	0	0	0	0	0	0	0	0	0	0	0
treinar pessoal de apoio à distribuição	1	0	0	0	0	0	1	0	0	0	0	0	0	0
implantar/integrar o processo de distribuição	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Desenvolver processo de atendimento ao cliente	8	0	0	0	2	2	1	0	0	2	1	0	0	0
desenhar processo de atendimento ao cliente	2	0	0	0	0	1	0	0	0	1	0	0	0	0
comprar recursos	1	0	0	0	1	0	0	0	0	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
desenvolver documentacao de atendimento ao cliente	2	0	0	0	0	1	0	0	0	1	0	0	0	0



FASE / ATIVIDADE / TAREFA	SOMA	2	3	4	5	6	7	8	9	10	11	12	13	14
Promover evento de lançamento	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Gerenciar lançamento	6	0	0	0	0	0	3	3	0	0	0	0	0	0
Gerenciar resultados	2	0	0	0	0	0	1	1	0	0	0	0	0	0
Gerenciar aceitação inicial	2	0	0	0	0	0	1	1	0	0	0	0	0	0
Gerenciar satisfação do cliente	2	0	0	0	0	0	1	1	0	0	0	0	0	0
Atualizar plano de fim de vida	6	0	1	1	1	1	0	1	1	0	0	0	0	0
Monitorar a viabilidade econômico-financeira	23	0	1	1	8	0	0	2	7	0	0	0	2	2
Avaliar mudanças nas condições de mercado, características técnicas e indicadores	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Avaliar impacto das mudanças no projeto	2	0	0	0	0	0	0	0	1	0	0	0	0	1
Atualizar as premissas financeiras do projeto	3	0	0	0	1	0	0	0	1	0	0	0	1	0
Atualizar custo-alvo nos diversos níveis do produto	2	0	0	0	1	0	0	0	0	0	0	0	0	1
Atualizar as receitas futuras	2	0	0	0	1	0	0	0	1	0	0	0	0	0
Atualizar as necessidades de investimentos	2	0	0	0	1	0	0	0	1	0	0	0	0	0
Calcular o novo fluxo de caixa	2	0	0	0	1	0	0	0	1	0	0	0	0	0
Calcular novos indicadores financeiros	2	0	0	0	1	0	0	0	1	0	0	0	0	0
Analisar novas premissas e indicadores	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Avaliar possíveis desvios e impactos no projeto	6	0	1	1	1	0	0	1	1	0	0	0	1	0
Avaliar fase	13	0	1	0	1	1	0	4	2	1	0	0	2	1
Avaliar o cumprimento das tarefas planejadas	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Avaliar os resultados obtidos segundo os critérios estabelecidos	2	0	0	0	0	0	0	1	1	0	0	0	0	0

Avaliar os demais critérios quantitativos e indicadores	3	0	0	0	0	0	0	1	1	0	0	0	1	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Avaliar a viabilidade econômica	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Decidir implementar ações corretivas	2	0	0	0	0	0	0	1	0	0	0	0	0	1
Decidir se pode ser realizada a atividade de aprovação	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Preparar relatório para time de avaliação	2	0	0	0	0	1	0	0	0	1	0	0	0	0
Aprovar fase	25	0	1	0	1	3	1	2	8	6	1	0	2	0
Avaliar o relatório de auto-avaliação e discuti-lo com o time de desenvolvimento	3	0	0	0	0	1	0	0	1	0	1	0	0	0
Analisar as evidências dos resultados e término das atividades planejadas	2	0	1	0	0	0	0	1	0	0	0	0	0	0
Analisar o portfólio de produtos e projetos	3	0	0	0	0	1	0	1	1	0	0	0	0	0
Analisar o estudo de viabilidade econômica	2	0	0	0	1	1	0	0	0	0	0	0	0	0
Aplicar os critérios de avaliação e tomar decisão	4	0	0	0	0	0	1	0	1	1	0	0	1	0
<aprovar> Preparar relatório	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<aprovar> Melhorar o processo do gate	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<aprovar> Ajustar critérios próxima fase	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<redirecionar> Definir ações corretivas	2	0	0	0	0	0	0	0	1	1	0	0	0	0
<redirecionar> Analisar risco	3	0	0	0	0	0	0	0	1	1	0	0	1	0
Documentar as decisões tomadas e registrar lições aprendidas	4	0	0	0	0	1	1	0	0	1	0	0	1	0
<b>Pós-desenvolvimento</b>	<b>40</b>	<b>0</b>	<b>2</b>	<b>1</b>	<b>2</b>	<b>9</b>	<b>6</b>	<b>10</b>	<b>5</b>	<b>2</b>	<b>2</b>	<b>0</b>	<b>1</b>	<b>0</b>
Acompanhar Produto e Processo	29	0	1	0	2	6	6	8	2	1	2	0	1	0
Avaliar Satisfação do Cliente	6	0	1	0	0	2	1	1	0	0	1	0	0	0

Planejar a avaliação	4	0	1	0	0	1	0	1	0	0	1	0	0	0
Realizar a avaliação	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Analisar e consolidar a avaliação	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Monitorar desempenho do produto (técnico, econômico, de produção e de serviços)	13	0	0	0	2	1	1	7	2	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Monitorar o desempenho técnico do produto no mercado e nos serviços associados	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Monitorar o desempenho técnico do produto na produção	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Monitorar o desempenho do processo de produção	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Monitorar o desempenho em vendas	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Monitorar avaliação econômica do produto	3	0	0	0	1	0	0	1	1	0	0	0	0	0
Monitorar custo do produto	2	0	0	0	1	0	0	1	0	0	0	0	0	0
Monitorar aspectos relacionados ao meio-ambiente	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Consolidar informações sobre desempenho (técnico, econômico, ambiental, de produção e de serviços)	3	0	0	0	0	1	1	0	1	0	0	0	0	0
Realizar auditoria pós-projeto	6	0	0	0	0	2	3	0	0	0	1	0	0	0
Planejar os focos da auditoria	1	0	0	0	0	0	1	0	0	0	0	0	0	0
Conduzir a auditoria	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Relatar e comunicar a aprendizagem pós-projeto	3	0	0	0	0	1	1	0	0	0	1	0	0	0
Registrar lições aprendidas	4	0	0	0	0	1	1	0	0	1	0	0	1	0
Descontinuar Produto no Mercado	11	0	1	1	0	3	0	2	3	1	0	0	0	0
Analisar e aprovar descontinuidade do	3	0	0	0	0	1	0	1	1	0	0	0	0	0





Definir requisitos	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Desenhar solução	2	0	0	0	0	1	0	1	0	0	0	0	0	0
Executar melhoria	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Liberar solução	4	0	0	0	0	1	1	1	0	0	1	0	0	0
<b>Transformação do PDP</b>	<b>24</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>8</b>	<b>5</b>	<b>3</b>	<b>2</b>	<b>0</b>	<b>3</b>	<b>0</b>	<b>0</b>	<b>0</b>
Entender as motivações das melhorias	4	0	0	0	0	1	0	1	2	0	0	0	0	0
Avaliar últimos acontecimentos	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Avaliar estratégia da empresa	1	0	0	0	0	0	0	0	1	0	0	0	0	0
Consolidar estratégia da empresa	2	0	0	0	0	1	0	0	1	0	0	0	0	0
Analisar a situação	3	0	0	0	0	1	2	0	0	0	0	0	0	0
Diagnóstico e maturidade atual	1	0	0	0	0	0	1	0	0	0	0	0	0	0
<b>FASE / ATIVIDADE / TAREFA</b>	<b>SOMA</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
Criar visão estratégica	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Definir política de transformação	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir estratégias e objetivos das transformações	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Defnir ações	5	0	0	0	0	2	1	0	0	0	2	0	0	0
Selecionar e adotar o modelo de referência	2	0	0	0	0	1	1	0	0	0	0	0	0	0
Definir nível de maturidade	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Definir políticas para implantação dos processos	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Definir projetos de transformação	2	0	0	0	0	1	0	0	0	0	1	0	0	0
Implantar	12	0	1	1	1	4	2	2	0	0	1	0	0	0
Planejar	3	0	1	1	1	0	0	0	0	0	0	0	0	0
Definir requisitos	1	0	0	0	0	1	0	0	0	0	0	0	0	0
Desenhar solução	2	0	0	0	0	1	0	1	0	0	0	0	0	0

<b>Executar melhoria</b>	<b>2</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>						
<b>Liberar solução</b>	<b>4</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Apêndice D – Lista final dos requisitos**

Tabela 13: lista final de requisitos

GRUPO 1	GRUPO 2	REQUISITO
1	1,1	Integrar o documentos com a base de dados, para trazer o texto da base de dados
		Permitir integração com outros sistemas
		Compartilhar totalmente com aplicações Microsoft Office
		Permitir uso de EAI (Enterprise Application Integration)
		Ter compatibilidade com diferentes browsers
		Ter habilidade de incorporar informações em formato multimídia
		Permitir upload / download de arquivos no formato original ou formato .zip ou .rar
		Permitir Integração com ferramentas comerciais de e-mail, Lotus Notes, MS Outlook, etc
	1,3	Permitir compatibilidade com diferentes PCs
	1,4	Permitir criação de Portais corporativos
		Permitir descentralização do sistema
	1,5	Permitir criação Intranets independentes por projeto
		Possuir Interface gráfica
		Permitir facilidade de operação
	1,6	Permitir Visualização dos documentos informações na tela, sem necessidade de download
		Permitir Acesso a documentos / newsgroups por e-mail
		Permitir múltiplos usuários / permissões
		Possuir mecanismos de autenticação dos usuários / controle de acesso
1,7	Permitir Autenticação cruzada para login	
	Permitir diferentes perfil de acesso	
	Ter capacidade de fazer backups regulares	
	Possuir comunicação segura via Net	
1,8	Ter sistema de segurança contra invasão	
	Ter alta disponibilidade	
	Ter flexibilidade de customização do sistema	
2	2,1	Ser de fácil manutenção
	2,2	Permitir programação de um calendário único do projeto
	2,3	Visualizar agenda de tarefas que engloba todos projetos da organização
	2,4	Permitir programação do calendário de recursos
3	3,1	Agendar reuniões com integração de agenda
		Ter facilitador para agendamento de "meetings"
	3,2	Ter método de determinação de duração das tarefas
3,2	Permitir cadastro e alteração de atividades, tempo e precedência	
	Visualizar e formatar gráfico PERT	
		Visualizar e formatar gráfico Gantt

	<b>3,3</b>	Permitir geração de WBS
		Definir e visualizar níveis para as atividades
		Cálculo do caminho crítico
	<b>3,4</b>	Associar deliverable (documento) a atividade / fase / projeto
		Criar de diferentes workgroups dentro de cada projeto
	<b>3,6</b>	Desenhar graficamente os workflows de cada etapa do processo
Permitir desenho de workflow específico a partir de um workflow padrão		
Permitir uso de informações sobre atividades da WBS para desenho inicial do workflow		
<b>4</b>	<b>4,1</b>	Alocar usuários do sistema a cada atividade do workflow
		Permitir registro de perfis de usuários
	<b>4,2</b>	Configurar recursos
	<b>4,3</b>	Relacionar dos recursos com as atividades
	<b>4,4</b>	Visualizar graficamente disponibilidade de uso dos recursos
<b>5</b>	<b>5,1</b>	Ter ferramenta para Nivelamento de recursos
		Definir padrões de alocação de recursos
		Cálculo de custo atividades
		Cálculo do orçamento do projeto
	<b>5,3</b>	Ter ferramenta de análise do valor agregado para orçamento e custos reais
<b>6</b>	<b>6,1</b>	Análise do valor agregado para índices de desempenho
		Permitir orçamentos e projeções
		Fornecer ou indicar ferramentas para estudo de viabilidade de projeto
	<b>6,2</b>	Buscar na base de dados
		Buscar em páginas HTML
	<b>6,3</b>	Buscar nos arquivos armazenados (Word, Powerpoint)
		Ter tecnologia de busca avançada
	<b>6,4</b>	Permitir pesquisa avançada por palavras-chave, autor, data, tamanho do documento, tipo de serviço, projeto, etc.
		Ter sistema de filtro e ordenamento das informações / documentos
		Controlar data
<b>6,5</b>	Adicionar comentários	
	Permitir uso de campos predefinidos de metadados para permitir comparação	
<b>6,8</b>	Armazenar de documentos em sistema vault	
	Permitir "Bulk" downloading / uploading na biblioteca	
<b>6,9</b>	Controlar de acesso por documento	
	Rotear de documento para membros do grupo	
<b>7</b>	<b>7,1</b>	Controlar versão
		Detalhar projeto através de relatórios em forma de tabelas
	<b>7,1</b>	Gerar de relatórios
<b>7</b>	<b>7,1</b>	Permitir fácil acesso a informações estatísticas sobre o projeto
		Permitir fácil acesso a informações estatísticas sobre o uso do software por projeto
<b>7</b>	<b>7,1</b>	Ter facilidades de configuração para Impressão
		Permitir Inserção / deleção de best practices
<b>7</b>	<b>7,1</b>	Editar de best practices

		Associar conhecimentos ao projeto
	7,2	Recuperar de best practice Permitir referência cruzada, para sugerir melhores práticas
	7,3	Permitir validação de best practice, através de workflow pré-definido
	7,4	Acessar a lista de pessoas com mesmo interesse / especialistas Avaliar (ranking / rating) de especialistas / conhecimentos
	7,5	Armazenar de cursos online Verificar andamento do treinamento Ter links para os cursos Ter ferramentas para assistir / ministrar curso online
	7,8	Avaliar os conhecimentos do usuários Avaliar a performance do time e stakeholders
8	8,1	Gravar linha base
	8,2	Registrar de datas intermediárias de entrega e verificação Permitir visualização do status de cada atividade do workflow Monitorar progresso das atividades Identificar efeitos de atrasos no plano
	8,3	Permitir criação / acompanhamento de indicadores de desempenho Acompanhar ações de risco
	8,4	Permitir ferramentas para a avaliação do status de cada projeto
	8,5	Permitir definição dos gates pelos usuários Análisar do projeto com múltiplas baselines
9	9,1	Visualizar das tarefas envolvendo todos os projetos Consolidar projetos para controle Listar todos os projetos
	9,2	Permitir compartilhamento de recursos entre múltiplos projetos
	9,4	Definir, visualizar e formatar Scoring models, diagrama de bolhas, etc Aplicar prioridade a um projeto Criar hierarquização (projetos / subprojetos) Permitir aprovação, congelamento, descongelamento e cancelamento do projeto Definir condições para realização / aprovação de cada atividade
10	10,1	Cadastrar padrões para missão, ecope statement, etc do projeto Permitir a definição de padrões para cada atividade Permitir templates para AD (Design da Arquitetura do produto) Permitir criação de modelos de documentos elaborados para cada fase / atividade padrão
	10,2	Templates devem ser semi-automatizáveis
11	11,1	Ter Web meetings Ter Chat
	11,2	Ter E-mail Ter Fórum Ter ferramenta de comentários em documentos / informações / atividades
	11,3	Disparar alertas automáticos na conclusão / atraso nas atividades / deliverables Ter software de alertas
	11,4	Compartilhar o espaço de trabalho (painéis)

	<b>11,6</b>	Registrar as comunicações feitas através do sistema Registrar resultado com comunicação automática
	<b>11,7</b>	Permitir documentação em outras línguas Customizar menus em outras línguas Permitir uso de diferentes conjuntos de caracteres Apoiar diferentes formatos de data e fusos horários
	<b>11,8</b>	Apoiar ferramentas mobile / wireless Apoiar comunicação por voz e vídeo
<b>12</b>	<b>12,1</b>	Facilitar identificação de requisitos do produto / processo Facilitar identificação de riscos do produto / processo
	<b>12,3</b>	Ter ajuda online
<b>13</b>	<b>13,1</b>	Listar ações para diminuição de riscos Possuir tabela padrão com critérios para avaliação sistemática dos riscos Cadastrar ações de risco
	<b>13,2</b>	Identificar grupos de tarefas com risco de não se completarem na data prevista Agrupar estatisticamente riscos do projeto / portfólio
	<b>13,4</b>	Ter Ferramentas de simulação para acompanhamento de atividades / riscos
<b>14</b>	<b>14,1</b>	Definir prazo de validade de deliverables do projeto Definir processo de validação de alterações em deliverables
	<b>14,2</b>	Armazenar alteração atual contendo data da última alteração e responsável Encaminhar solicitação de mudanças para responsáveis

**Apêndice E - Necessidades específicas por ator (diagramas de caso de uso)**

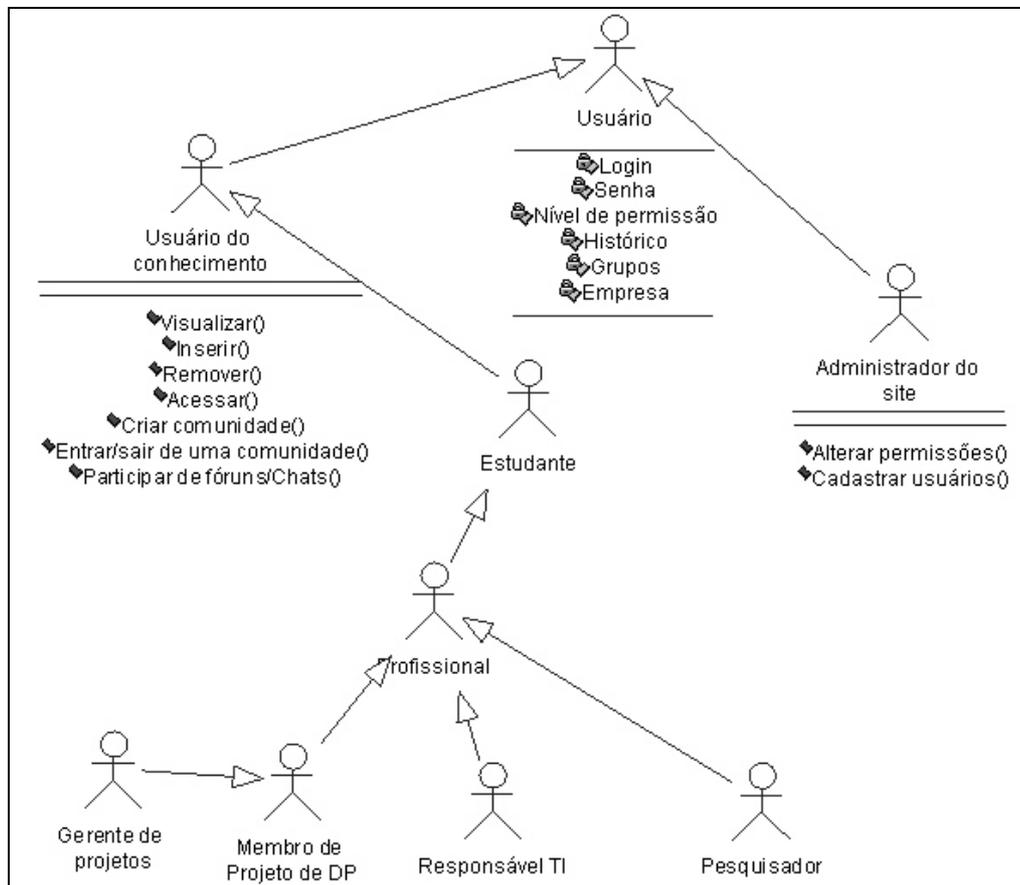


Figura 24: Diagrama de Classes: Usuários

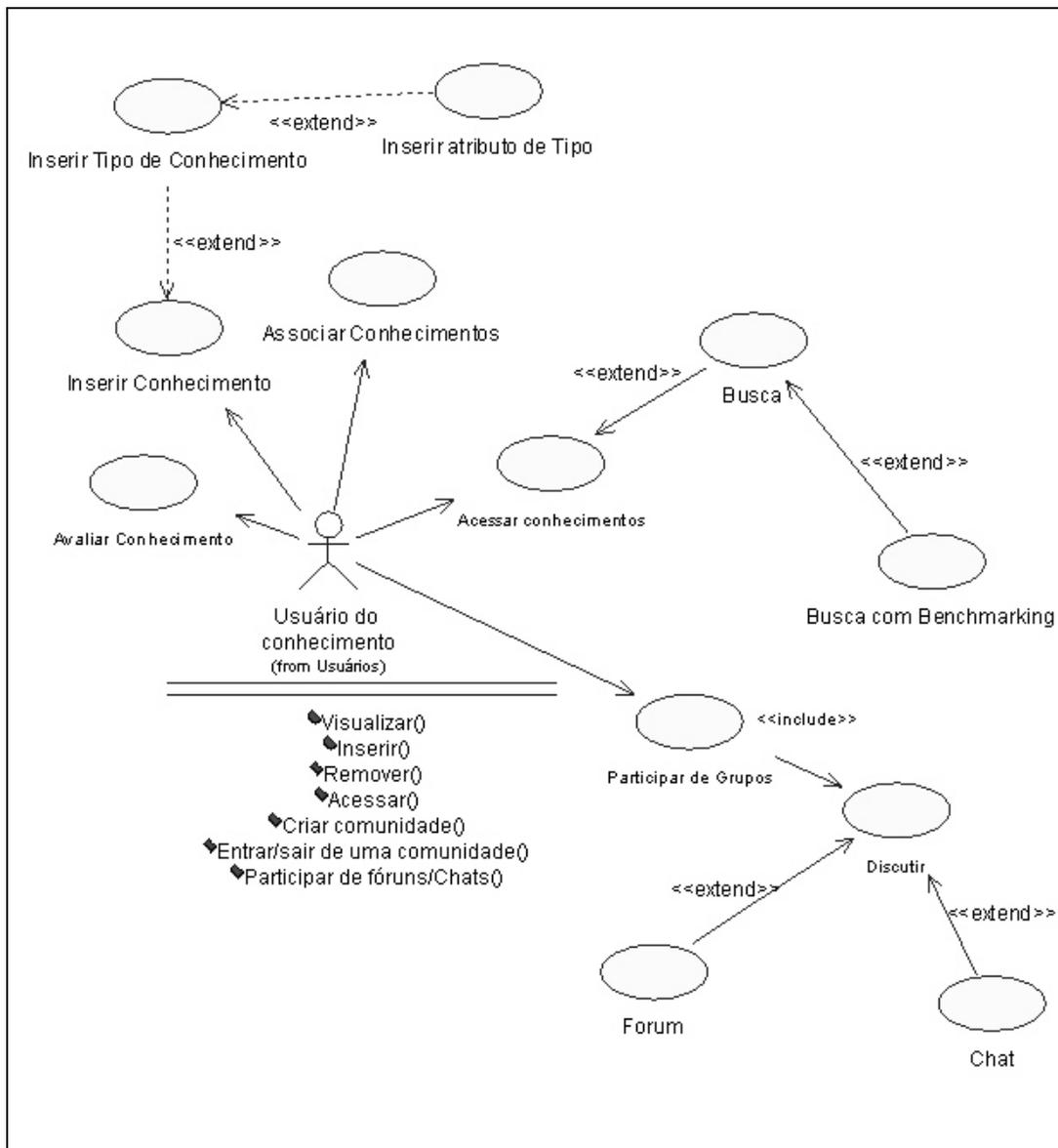


Figura 25: diagrama de Casos de Uso: necessidades dos Usuários de Conhecimento

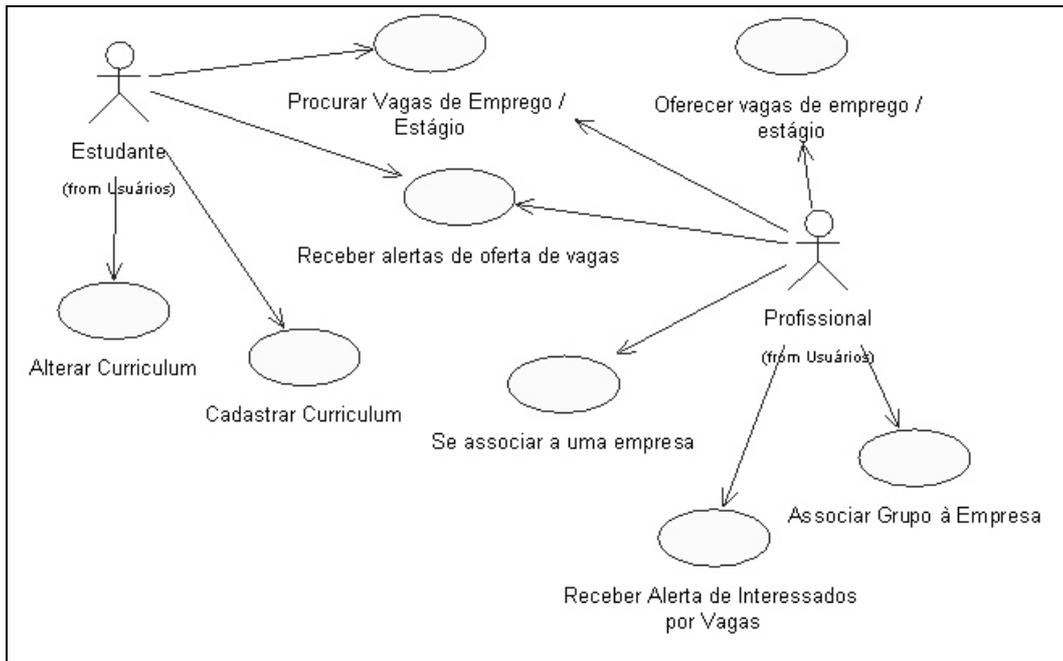


Figura 26: Diagrama de Casos de Uso: Necessidades de Estudantes e Profissionais

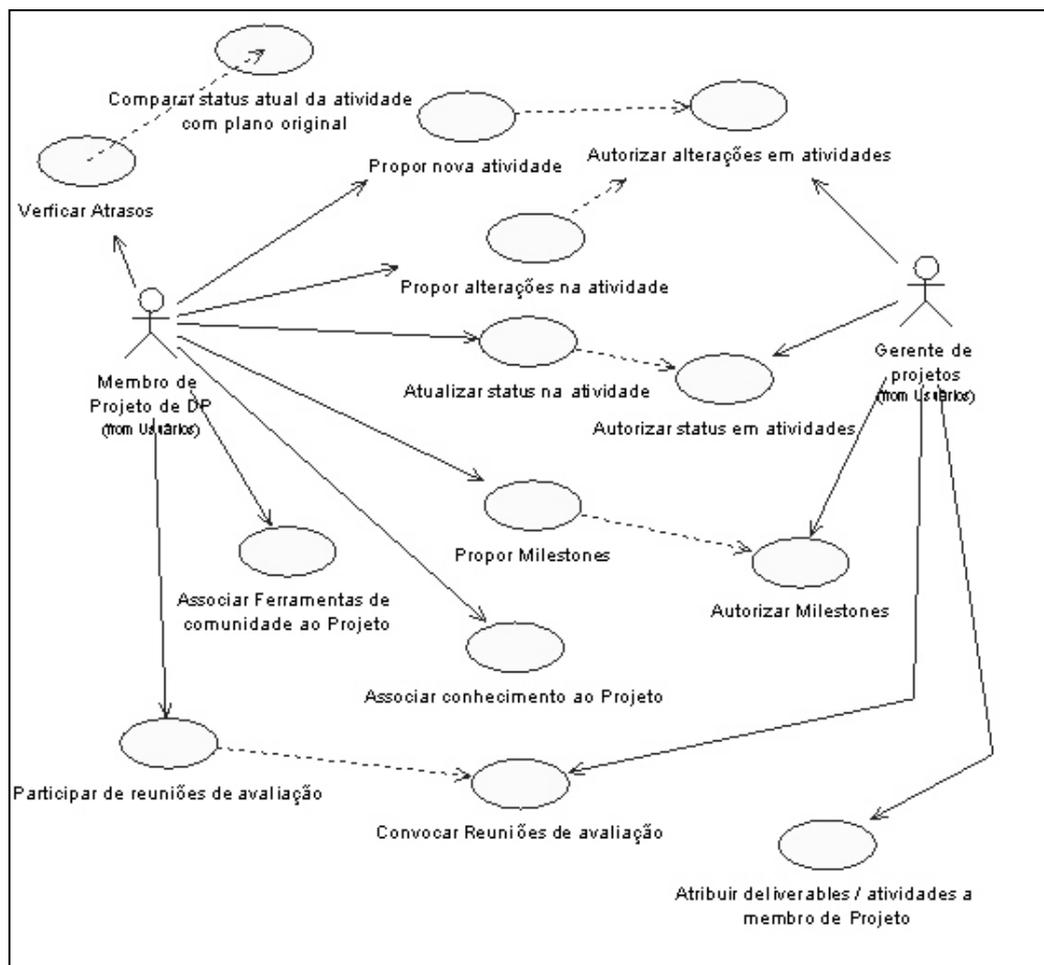


Figura 27: Diagrama de Casos de Uso: Necessidades da Equipe de Projetos DP

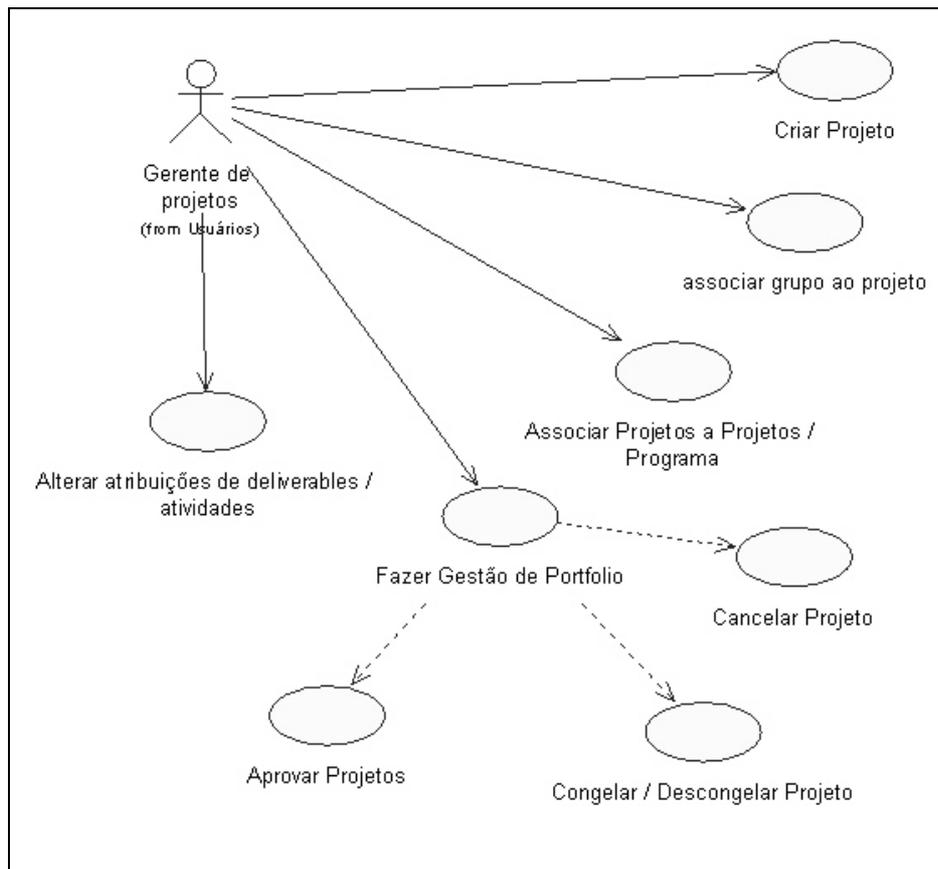


Figura 28: Diagrama de Casos de Uso: Necessidades dos Gerentes de Projeto

# Livros Grátis

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

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

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

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

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