

Marco Aurélio Cordeiro

Uma Ferramenta Automatizada
de Suporte ao Processo de
Gerenciamento de Requisitos

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre em Ciências.

Curso de Pós-graduação em Informática Aplicada –
PPGIA,

Centro de Ciências Exatas e de Tecnologia – CCET,
Pontifícia Universidade Católica do Paraná –
PUCPR.

Orientador:

Prof. Dr. Robert Carlisle Burnett

Curitiba

2002

Cordeiro, Marco Aurélio

Uma Ferramenta Automatizada de suporte ao processo de Gerenciamento de Requisitos.

Curitiba, 2002. 181p.

Dissertação (Mestrado) – Pontifícia Universidade Católica do Paraná, PUCPR. Centro de Ciências Exatas e de Tecnologia, CCET. Curso de Pós-graduação em Informática Aplicada, PPGIA.

1. Gerenciamento de Requisitos
2. CMMI
3. ISO/IEC 12.207 e 15504
4. Modelagem de Processos
5. Ambiente AGIR

Dedicatória

A meus pais, amigos e familiares, pela compreensão por muitos momentos em que estive ausente, dedicando meu tempo e atenção na elaboração deste trabalho.

E, de modo especial para:

Luciane, Lucas e Isabela

Agradecimentos

Do fundo do coração, agradecer a todos os companheiros desta difícil e trabalhosa caminhada.

Agradecimento especial ao Danilo, Luiz Carlos e Jumara, funcionários da Companhia de informática do Paraná, sem cujo apoio seria impossível o início desta jornada.

Agradeço a Pontifícia Universidade Católica do Paraná, que ofereceu a infra-estrutura necessária para que o trabalho pudesse ser realizado.

Ao Prof. Robert Carlisle Burnett, orientador deste trabalho, pela ajuda e paciência durante a realização desta dissertação.

Aos amigos da turma do mestrado, pela companhia e amizade que desfrutamos durante o decorrer do curso, pelos churrascos na casa da minha sogra, regado a muita cerveja e conversas muito agradáveis.

Em especial ao amigo Marlon Silva Marciniuk, pelo apoio moral, intelectual, por sua amizade e pelos bons e maus momentos passados ao longo do curso.

A FACET, na pessoa do Ribamar, diretor administrativo da instituição, que sempre esteve pronto a ajudar em tudo quanto fosse necessário.

Aos meus filhos, Lucas e Isabela, simplesmente por existirem e fazerem a vida ter sentido.

A minha esposa Luciane, por todo o carinho, dedicação e compreensão ao longo de nossos dezesseis anos de vida juntos.

A Deus, por estar sempre a minha frente, mostrando a caminho a seguir. Obrigado por tudo.

Sumário

1	INTRODUÇÃO.....	19
1.1	APRESENTAÇÃO.....	19
1.2	MOTIVAÇÃO.....	23
1.3	OBJETIVOS.....	24
1.4	ESTRUTURA DO TRABALHO.....	25
2	MODELOS DE QUALIDADE DE SOFTWARE.....	27
2.1	INTRODUÇÃO.....	27
2.2	METODOLOGIAS DE DESENVOLVIMENTO.....	28
2.3	MUDANDO DE DIREÇÃO.....	31
2.4	FOCO NO PROCESSO.....	32
2.4.1	DEFINIÇÃO DO PROCESSO DA ORGANIZAÇÃO.....	34
2.4.2	ATIVOS DO PROCESSO DE SOFTWARE (SOFTWARE PROCESS ASSETS).....	35
2.4.3	CONTEÚDO DO PROCESSO PADRÃO DA ORGANIZAÇÃO.....	36
2.4.4	UTILIZANDO O PROCESSO PADRÃO.....	36
2.5	OS MODELOS DE QUALIDADE.....	37
2.5.1	SW-CMM.....	38
2.5.2	NBR ISO/IEC 12.207.....	40
2.5.3	ISO/IEC 15.504.....	43
2.5.4	MODELO CMMI.....	44
2.6	MODELAGEM DE PROCESSO.....	51
2.7	AMBIENTE AGIR.....	52
2.8	CONCLUSÃO.....	54
2.9	RESUMO.....	55
3	ENGENHARIA DE REQUISITOS.....	57
3.1	INTRODUÇÃO.....	57

3.2	O QUE É UM REQUISITO	57
3.2.1	CLASSIFICAÇÃO DOS REQUISITOS	58
3.3	CONTEXTO DE DEFINIÇÃO DOS REQUISITOS.....	61
3.4	A ENGENHARIA DE REQUISITOS.....	64
3.4.1	FASES DA ENGENHARIA DE REQUISITOS	66
3.4.2	ANÁLISE E NEGOCIAÇÃO DE REQUISITOS	80
3.4.3	VALIDAÇÃO DE REQUISITOS	81
3.4.4	GERENCIAMENTO DE REQUISITOS.....	82
3.5	CONCLUSÃO.....	90
3.6	RESUMO	91
 <u>4 MODELO PROPOSTO PARA GERENCIAMENTO DE REQUISITOS.....</u>		<u>94</u>
4.1	INTRODUÇÃO	94
4.2	O MODELO PROPOSTO.....	94
4.3	BASE DE REPRESENTAÇÃO DE REQUISITOS	95
4.4	A MODELAGEM DO PROCESSO DE GERENCIAMENTO DE REQUISITOS	100
4.5	DESCRIÇÃO DO MODELO PROPOSTO DO PROCESSO DE GERENCIAMENTO DE REQUISITOS.....	102
4.6	RESUMO	106
 <u>5 A FERRAMENTA SIGERE PARA AUTOMATIZAR O PROCESSO DE GERENCIAMENTO DE REQUISITOS.....</u>		<u>108</u>
5.1	INTRODUÇÃO	108
5.2	REQUISITOS DO SISTEMA.....	108
5.3	DESCRIÇÃO DAS TELAS.....	117
5.4	RESUMO	159
 <u>6 APLICAÇÃO PRÁTICA DA MODELO DE GERENCIAMENTO DE REQUISITOS.....</u>		<u>161</u>
6.1	INTRODUÇÃO	161
6.2	PLANEJAMENTO	161
6.3	PROCESSO APLICATIVO	163

6.4	TÉCNICAS APLICADAS.....	170
6.5	RESULTADO DO PROCESSO DE GERENCIAMENTO DE REQUISITOS	170
6.6	RESUMO	171
7	<u>CONCLUSÃO</u>	<u>173</u>
7.1	INTRODUÇÃO	173
7.2	DIMENSÃO DO TRABALHO	173
7.3	CONSIDERAÇÕES FINAIS	177
7.4	TRABALHOS FUTUROS	177
	<u>REFERÊNCIA BIBLIOGRÁFICA:.....</u>	<u>179</u>

Lista de Figuras

Figura 1.1 – Fonte de Erros – Projetos da Força Aérea dos EUA.	21
Figura 1.2 – Custo Relativo de Reparo de Erro X Fase do Ciclo de Vida.	22
Figura 1.3 – Níveis de Maturidade do Modelo CMM e suas Áreas Chave.	23
Figura 2.1 – Comparativo de tempo Análise Estruturada X Engenharia da Informação	30
Figura 2.2 – Triângulo da Qualidade	33
Figura 2.4 – Adaptação do Processo Padrão.....	37
Figura 2.5 – Arquitetura da norma NBR ISO/IEC 12207.	41
Figura 2.7 – Relacionamento de GR com outras Áreas Chave do CMMI	45
Figura 2.8 – Representação dos Processos no ambiente AGIR	54
Figura 3.1 – Contexto de Definição dos Requisitos	61
Figura 3.2 – Fases da Engenharia de Requisitos	66
Figura 4.1 – Documento de Descrição de Requisitos	97
Figura 4.2 – Modelo proposto do processo de Gerenciamento de Requisitos.....	101
Figura 5.1 – Tela de Login do Sistema.....	118
Figura 5.2 – Tela inicial do sistema.....	119
Figura 5.3 – Tela Cadastrar Projetos	120
Figura 5.4 – Tela Cadastrar Requisitos.....	122
Figura 5.5 – Tela Cadastrar Atividades Dependentes dos Requisitos	123
Figura 5.7 – Tela Cadastrar Atividades	125
Figura 5.8 – Tela Cadastrar Stakeholder	126
Figura 5.9 – Tela Cadastrar Competências dos Stakeholders.....	127
Figura 5.11 – Tela Solicitar Mudança	130
Figura 5.12 – parte inicial da Tela Solicitar Mudança.....	131
Figura 5.14 – parte inicial da tela Cadastrar Requisito da Mudança	133
Figura 5.15 – parte final da tela Cadastrar Requisito da Mudança (após se teclar PgDn)	133
Figura 5.16 – Tela confirmação do cadastramento da Solicitação de Mudança.....	134
Figura 5.17 – Tela Identificar Stakeholders Pertinentes.....	135

Figura 5.18 – parte inicial (topo) da Tela Identificar Stakeholders Pertinentes	136
Figura 5.19 – parte intermediária (corpo) da Tela Identificar Stakeholders Pertinentes (após se teclar PgDn)	138
Figura 5.20 – parte final (base) da Tela Identificar Stakeholders Pertinentes (após se teclar PgDn)	139
Figura 5.21 – Tela de informação dos stakeholders (link)	140
Figura 5.22 – Tela de confirmação da função Identificar Stakeholders Pertinentes	140
Figura 5.23 – Tela Analisar Mudança nos Requisitos	141
Figura 5.24 – parte inicial (topo) da Tela Analisar Mudança nos Requisitos	142
Figura 5.25 – parte intermediária (corpo) da Tela Analisar Mudança nos Requisitos (após se teclar PgDn)	144
Figura 5.26 – parte intermediária (corpo) da Tela Analisar Mudança nos Requisitos (após se teclar PgDn)	144
Figura 5.27 – parte final (base) da Tela Analisar Mudança nos Requisitos (após se teclar PgDn)	145
Figura 5.28 – confirmação da Função Analisar Mudança nos Requisitos.....	146
Figura 5.29 – Tela Elaborar Proposta de Mudança nos Requisitos	147
Figura 5.30 – parte inicial (topo) da Tela Elaborar Proposta de Mudança nos Requisitos	148
Figura 5.31 – parte intermediária (corpo) da Tela Elaborar Proposta de Mudança nos Requisitos (após se teclar PgDn)	150
Figura 5.32 – parte intermediária e final da Tela Elaborar Proposta de Mudança nos Requisitos (após se teclar PgDn)	151
Figura 5.33 – parte final (base) da Tela Elaborar Proposta de Mudança nos Requisitos (após se teclar PgDn)	152
Figura 5.34 – parte final (base) da Tela Elaborar Proposta de Mudança nos Requisitos ..	152
Figura 5.35 –Tela Avaliar Impacto da Mudança nos Requisitos.....	153
Figura 5.36 – parte inicial (topo) da Tela Avaliar Impacto da Mudança nos Requisitos ..	154
Figura 5.37 – parte intermediária (corpo) da Tela Avaliar Impacto da Mudança nos Requisitos (após se teclar PgDn)	156

Figura 5.38 – parte intermediária (corpo) da Tela Avaliar Impacto da Mudança nos Requisitos (após se teclar PgDn)	156
Figura 5.39 – parte final (base) da Tela Avaliar Impacto da Mudança nos Requisitos (após se teclar PgDn).....	158
Figura 5.40 – parte final (base) da Tela Avaliar Impacto da Mudança nos Requisitos (após se teclar PgDn).....	158
Figura 5.41 – confirmação da Função Avaliar Impacto da Mudança nos Requisitos	159

Lista de Tabelas

Tabela 1.1 – Principais fatores que tornam um projeto crítico.....	20
Tabela 2.1 – Número de Áreas Chave por Nível de Maturidade.....	39
Tabela 5.1 – Requisito 001 - Identificar Stakeholders Pertinentes.....	109
Tabela 5.2 – Requisito 002 - Analisar a Solicitação de Mudança.....	110
Tabela 5.3 – Requisito 003 – Elaborar Proposta de Mudança.....	111
Tabela 5.4 – Requisito 004 – Avaliar Impacto da Mudança.....	112
Tabela 5.5 – Requisito 005 - Consultar a Baseline de Requisitos.....	113
Tabela 5.6 – Requisito 006 - Acompanhar a Solicitação de Mudança.....	114
Tabela 5.7 – Requisito 007 - Segurança de Acesso ao Sistema.....	115
Tabela 5.8 – Requisito 008 - Plataforma de Desenvolvimento do Distema.....	116
Tabela 6.1 – Requisito 005 - Consultar a Baseline de Requisitos – alterado na Solicitação de Mudança 001.....	164
Tabela 6.2 – Requisito 009 - Bloquear a atualização de Solicitação de Mudança finalizada – incluído na Solicitação de Mudança 001.....	165
Tabela 6.3 – Requisito 010 - Bloquear a atualização de Solicitação de Mudança finalizada – incluído na atividade Analisar Solicitação de Mudança.....	167
Tabela 6.4 – Requisito 005 - Consultar a Baseline de Requisitos - (alterado na Atividade Elaboração da Proposta de Mudança).....	169

Lista de Abreviaturas e Siglas

Item	Descrição
ABNT	Associação Brasileira de Normas Técnicas
ACP	<i>Áreas Chave de Processo</i>
AGIR	Ambiente de Gestão da Inteligência da Realidade
CMM	<i>Capability Maturity Model</i>
CMMI	<i>Capability Maturity Model Integration</i>
DFD	<i>Diagrama de Fluxo de Dados</i>
ERP	<i>Enterprise Resource Planning</i>
FUN	<i>Ambiente de Modelagem de Processos</i>
GR	<i>Gerenciamento de Requisitos</i>
IEEE	<i>Institute of Electrical and Electronics Engineers, Inc.</i>
IPVA	<i>Imposto sobre Propriedade de Veículo Automotor</i>
ISO/IEC	<i>International Standard Organization</i>
NBR	Norma Brasileira
PgDn	<i>Page Down</i>
RE	<i>Requirements Engineering</i>
SEI	<i>Software Engineering Institute</i>
SIGERE	<i>Sistema de Gerenciamento de Requisitos</i>
SPICE	<i>Software Process Improvement and Capability Evaluation</i>
SW-CMM	<i>Capability Maturity Model for Software</i>

Lista de Acrônimos

Item	Descrição
Baseline	<i>Linhas de base que definem o controle das mudanças nos requisitos</i>
Checklist	Lista de itens que devem ser verificados em uma atividade
Front-end	<i>Interface de uma aplicação de software</i>
Login	<i>Chave de acesso a um sistema de computador</i>
Middle-down	<i>Refinamentos sucessivos do Diagrama de Fluxo de Dados promovendo níveis mais detalhados</i>
Middle-up	<i>Particionamento do Diagrama de Fluxo de Dados Preliminar objetivando a geração do DFD nível 0</i>
Stakeholder	Pessoas envolvidas no projeto de desenvolvimento de software
Template	<i>Documento padrão</i>
Brainstorming	<i>Técnica de levantamento de necessidades para o desenvolvimento de sistemas</i>

Resumo

O grande desafio da Engenharia de Software é desenvolver software de qualidade assegurada, com elevada produtividade e dentro dos recursos inicialmente alocados. Novas ferramentas, métodos e metodologias aparecem freqüentemente se autodenominando soluções miraculosas que resolverão todos os nossos problemas de desenvolvimento de software.

O que realmente se percebe é que ainda continuamos a gastar muito tempo para manter o produto de software que construímos, devido a um processo fraco de gerenciamento, dificuldade de entendermos o que realmente precisamos construir para atender as necessidades de nossos clientes e a falta de controle sobre as alterações que nos são solicitadas a todo o tempo, durante o ciclo de vida de um produto de software.

Busca da qualidade do produto através do gerenciamento do processo de desenvolvimento de software pode ser considerado o último grande esforço do final do século passado em direção a melhoria da capacitação de desenvolvimento de software das organizações. Baseia-se em princípios de qualidade propostos por Walter Shewart, desenvolvidos por Edwards Deming e Joseph Juran e adaptados pelo SEI em um quadro de maturidade que estabelece uma base para gerenciamento e engenharia de projeto, visando a melhoria contínua dos processos de software. Esforço de repercussão mundial, conhecido como SW-CMM e CMMI, são estruturas que descrevem os principais elementos de um processo de software efetivo.

O conteúdo desta dissertação tem foco no gerenciamento dos requisitos de software, um dos elementos chave de um processo de desenvolvimento de software efetivo, sem perder de vista a importância de todo o ciclo de vida dos requisitos, principalmente a fase de descobrimento do problema. Todo o aspecto comportamental da abordagem junto ao cliente é discutida.

Tem como idéia central mostrar a potencialidade de um processo efetivo de gerenciamento de requisitos como forma de reduzir o abismo atual entre o desejo do cliente e o produto de software que lhe é entregue. O processo de gerenciamento de requisitos será norteado pelo modelo CMMI e por consequência pelas normas ISO/IEC 12.207 e 15.504. além das mais recentes bibliografias sobre o assunto.

Utiliza uma metodologia orientada a processo proposta pelo Ambiente de Gestão da Inteligência da Realidade (AGIR) na modelagem da área chave Gerenciamento de Requisitos.

Desta modelagem foi desenvolvida uma ferramenta para automação do processo de gerenciamento de requisitos de uma organização que utiliza esta abordagem no seu processo de desenvolvimento de software.

Por fim, são analisados criticamente os resultados obtidos na comprovação da hipótese, objetivando assegurar que um processo cuidadoso de especificação e gerenciamento de requisitos contribui fortemente para um produto de software de qualidade e aderente as necessidades do cliente.

Palavras-chave: SW-CMM, CMMI, NBR ISO/IEC 12.207, ISO/IEC 15.504, Engenharia de Requisitos, Modelagem de Processo, Avaliação de Processo de Software, Gerenciamento de Requisitos, Ambiente AGIR.

Abstract

The great challenge of the Software Engineering is to develop software of insured quality, with high productivity and inside of the resources initially allocated. New tools, methods and methodologies frequently appear if called miraculous solutions that will solve all our problems of software development.

The one that one really notices is that we still continued to spend a long time to maintain the software product that we built, due to a weak process management, difficulty of we understand what we really needed to build to assist the our customers' needs and the control lack about the alterations that are we requested at the whole time, during the cycle of life of a software product.

It looks for of the quality of the product through the process management of software development the last great effort of the end of the century passed in direction the improvement of the training of software development of the organizations can be considered. It bases on quality beginnings proposed by Walter Shewart, developed by Edwards Deming and Joseph Juran and adapted for the SEI in a maturity model that establishes a base for management and project engineering, seeking the continuous improvement of the software processes. Effort of world repercussion, well-known like SW-CMM and CMMI, they are structures that describe the main elements of an effective software process.

The content of this dissertation proposal has focus in the management of the software requirements, one of the key of the process of effective software elements, without losing of view the practical experience of the author's 16 years, that took it to place its point of view very strongly on the process of discovery of the problem. The whole aspect behavior in the aproach close to the customer is discussed.

It has as central idea to show the potentiality of an effective process of requirements management as form of reducing the current abyss among the customer's desire and the software product that it is it given. The process of requirements management will be driven for the model CMMI and for consequence for the norms ISO/IEC 12.207 and 15.504.

It uses a guided methodology it I process proposed by the Environment of Administration of the Intelligence of the Reality (AGIR) in the model of the key area Requirements Management.

Of this model a tool was developed for automation of the process of Requirements Management of an organization that uses this approach in its process of software development.

Finally, the results obtained in the confirmation of the hypothesis are analyzed critically, objectifying to assure that a careful process of specification and Requirements Management contribute strongly to a product of quality software and adherent the customer's needs.

CAPÍTULO 1

Introdução

1.1	APRESENTAÇÃO.....	19
1.2	MOTIVAÇÃO.....	23
1.3	OBJETIVOS	24
1.4	ESTRUTURA DO TRABALHO	25

1 Introdução

Nesta etapa da dissertação é apresentado o contexto do processo de desenvolvimento de software, as razões e motivação sobre o tema, uma ferramenta automatizada para suportar o processo de gerenciamento de requisitos, e a estrutura geral do trabalho.

1.1 Apresentação

Em um congresso de Software na Suíça em 1989, certa vez, o professor Daniel Teichroew da universidade de Michigan utilizou a terminologia “Aflição Crônica” para referenciar a crise da indústria de software dos últimos trinta anos, e explicou. Nós não atingimos o nível chamado crise de software. E complementou: crise é o momento decisivo, a partir do qual, solucionamos um problema. Já aflição crônica, é algo que causa sofrimento e dura um longo período de tempo. Ainda temos que chegar ao estágio de crise no desenvolvimento software[1].

Buscando identificar os fatores relevantes que contribuem para este estado de imaturidade no desenvolvimento de software, uma pesquisa americana publicada em 1994 pelo instituto Standish Group, utilizando uma amostra de 365 companhias entrevistadas, representando 8.380 aplicações, chegou às seguintes informações[2].

De acordo com a pesquisa:

- Os EUA gastam US\$ 250 bilhões por ano em desenvolvimento de aplicação de tecnologia da informação em aproximadamente 175 mil projetos;
- 31% de todos os projetos são cancelados antes do seu término, representando um desperdício da ordem de US\$ 81 bilhões;
- 53% de todos os projetos chegam ao final tendo custado 189% do valor estimado, representando US\$ 59 bilhões em custo adicional, atrasam em até 222% da estimativa

original além de serem entregues com apenas 61% das características originalmente especificadas;

- 16% são entregues no prazo e dentro do orçamento.

Além destes dados quantitativos, talvez o aspecto mais importante da pesquisa tenha sido o levantamento dos principais fatores que tornam um projeto crítico, ou seja, aqueles projetos que extrapolam prazo, custo e que são entregues com a funcionalidade prejudicada. De acordo com os entrevistados, as razões principais que levam a problemas de projeto são identificadas na tabela 1.1:

Fatores que tornam um Projeto Crítico	% de Respostas
1. Falta de Especificação do Usuário	12.8%
2. Requisitos Incompletos	12.3%
3. Mudança de Requisitos	11.8%
4. Falta de Apoio Executivo	7.5%
5. Tecnologia Imatura	7.0%
6. Falta de Recursos	6.4%
7. Expectativas irreais	5.9%
8. Objetivos obscuros	5.3%
9. Tempo irreal	4.3%
10. Tecnologia nova	3.7%
11. Outros	23.0%

Tabela 1.1 – Principais fatores que tornam um projeto crítico

Parece que a inabilidade para trabalhar mais efetivamente com usuários e entender melhor as suas necessidades (requisitos) tem sido a causa principal das falhas de software. Esta deficiência, aliada a desordem de muitos ambientes de desenvolvimento, que muitas vezes buscam a melhoria da qualidade de seus produtos, mas não investem em processos básicos, como por exemplo, a documentação dos seus sistemas é que produz esta situação de caos. [49]

Analisando os dados de projetos da Força Aérea Americana [3] sobre a fonte de erro por fase do ciclo de vida do software, relatados na figura 1.1, pode-se identificar como a principal fonte de erros no ciclo de vida de um software a fase de identificação do problema, conhecida como análise de requisitos, responsável por 41% dos erros de um software.

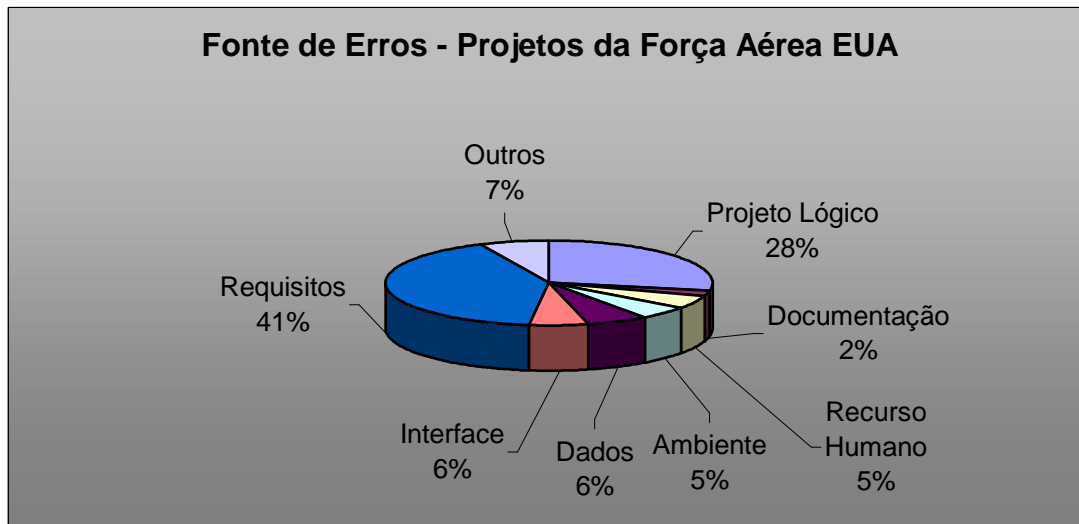


Figura 1.1 – Fonte de Erros – Projetos da Força Aérea dos EUA.

Avaliando os dados da figura 1.1 sob o foco do custo relativo de reparo de um erro por fase do ciclo de vida, demonstrado na figura 1.2, que reflete dados originados de estudos independentes das empresas GTE, TRW e IBM em meados da década de 1970[4];

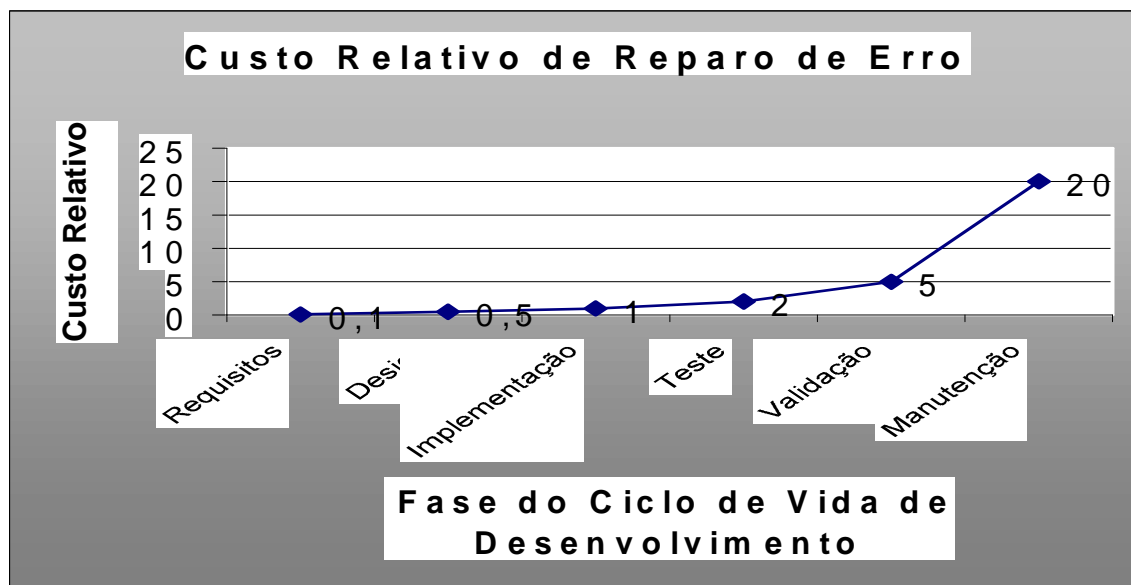


Figura 1.2 – Custo Relativo de Reparo de Erro X Fase do Ciclo de Vida.

Pode-se concluir que identificar um erro na fase de manutenção tem um custo relativo até 200 vezes maior em relação a descoberta do mesmo erro na fase de análise de requisitos, a etapa inicial do ciclo de vida. Deve-se, então, orientar os esforços na definição e melhoria contínua de processos formais a serem executados durante esta fase, como forma de aumentar a qualidade do software desenvolvido, bem como, minimizar o custo deste desenvolvimento.

Justamente, por todos estes problemas encontrados no desenvolvimento de software, no final da década de oitenta, iniciaram-se esforços orientando a solução destes problemas de construção de software através da melhoria do processo de desenvolvimento defendendo que a qualidade de um produto é reflexo da qualidade e gerenciamento do processo utilizado no desenvolvimento deste produto. Um dos modelos com foco na melhoria do processo é o SW-CMM, o qual define uma estrutura que descreve os principais elementos de um processo de software efetivo. Descreve também os estágios através dos quais, organizações de software evoluem quando definem, implementam, medem, controlam e melhoram seus processos de software. Desta forma, provê e descreve um caminho de melhoria evolutiva a partir de um processo desorganizado para um processo maduro e altamente disciplinado. Estes estágios, também conhecidos por níveis de

maturidade, são patamares evolutivos bem definidos. Cada um destes níveis possui áreas chave de processo, conforme a figura 1.3 abaixo: [5]

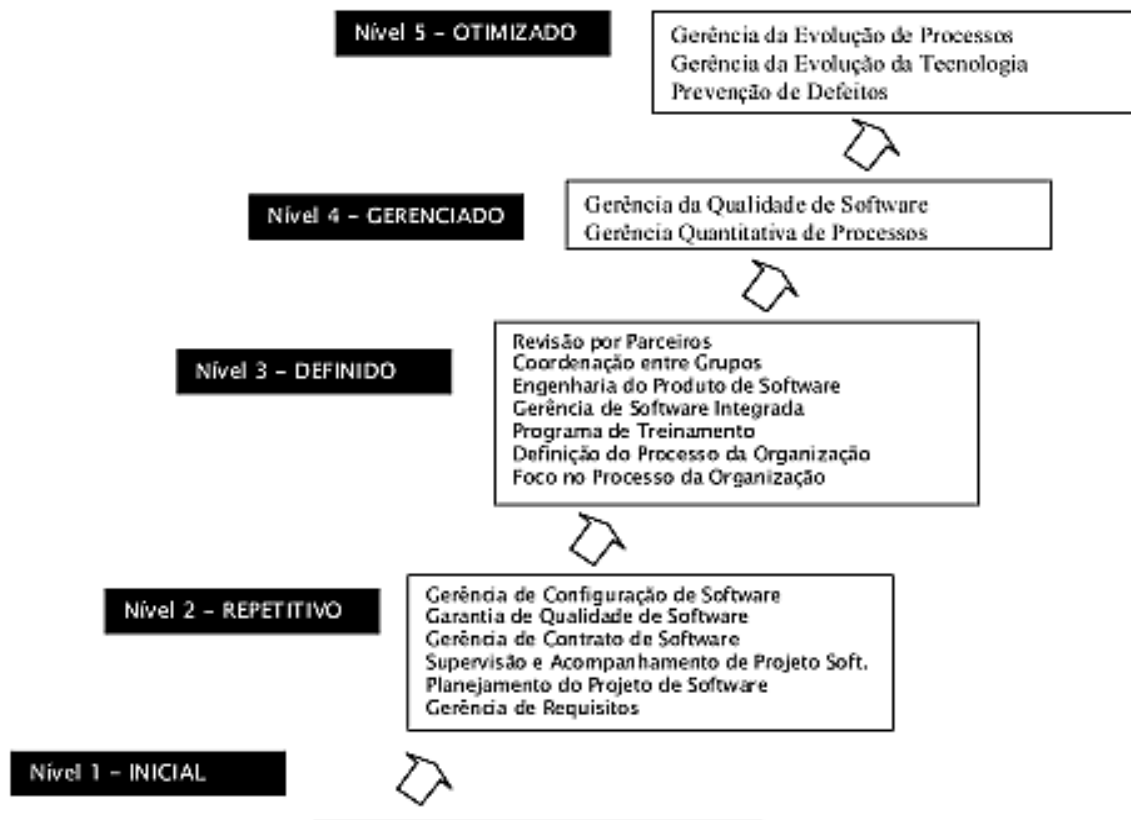


Figura 1.3 – Níveis de Maturidade do Modelo CMM e suas Áreas Chave.

Conforme podemos verificar na figura 1.3, no nível 2 do modelo CMM, conhecido por Repetitivo ou Repetível, encontramos a área chave Gerenciamento de Requisitos, foco deste trabalho.

1.2 Motivação

A motivação para a realização deste trabalho concentra-se no esforço de diminuição do abismo entre as necessidades do cliente e o produto de software construído, através da definição de um processo de gerência de requisitos que tem por finalidade estabelecer e manter um acordo com os envolvidos no projeto em relação as necessidades que o software deverá atender, se utilizando do

modelo CMMI, que integra os modelos SW-CMM e outros modelos CMM, e procura ser compatível com a norma ISO/IEC 12.207 e 15.504 (normatização internacional).

A abrangência deste trabalho tem como ponto forte a utilização de uma nova abordagem para modelagem de processos de negócio através da utilização do Ambiente AGIR para modelar o processo de Gerenciamento de Requisitos e após esta modelagem, a implementação de uma ferramenta de automação do processo modelado.

A característica específica do processo de gerenciamento de requisitos modelado é ser aderente tanto ao padrão de utilização (CMM, CMMI) quanto a norma internacional (ISO).

Além disso, tem como objetivo maior, gerar material bibliográfico sobre o tema “Requisitos”.

1.3 Objetivos

O trabalho tem como objetivos:

⇒ a nível geral,

- ✓ enfatizar a importância da área chave de processo Gerenciamento de Requisitos no sucesso de projetos de desenvolvimento de software e a necessidade de utilização de uma ferramenta automatizada para suportar tal processo, relatando os resultados obtidos com a utilização desta ferramenta;

⇒ a nível específico:

- ✓ utilizar conhecimento gerado por trabalhos anteriores na área de Requisitos;
- ✓ utilizar os Modelos de Qualidade de Software para nortear o trabalho;
- ✓ gerar material bibliográfico sobre o assunto, procurando cobrir todas as fases da Engenharia de Requisitos;
- ✓ gerar um modelo de suporte ao processo de Gerenciamento de Requisitos;
- ✓ utilizar um novo ambiente automatizado para descrição do modelo do processo de Gerenciamento de Requisitos que utilize uma abordagem orientada a processo;
- ✓ implementar uma ferramenta de suporte ao modelo proposto do processo de Gerenciamento de Requisitos.

1.4 Estrutura do Trabalho

No capítulo dois, os modelos CMM e CMMI e as normas ISO/IEC 12.207 e 15.504 são apresentados bem como todo o embasamento que deles é extraído para a definição do processo de gerenciamento de requisitos proposto. Ainda neste capítulo é apresentado o Ambiente AGIR sobre o qual o processo de gerenciamento de requisitos foi modelado.

No capítulo três é feita a apresentação da Engenharia de Requisitos e todas as fases que compõem o processo de requisitos que fundamentou este trabalho.

No capítulo quatro são apresentadas as informações que fundamentam a proposta essencial do trabalho que é o detalhamento do processo de gerenciamento de requisitos.

No capítulo cinco detalhamos a ferramenta que foi construída para automatizar o processo de gerenciamento de requisitos.

No capítulo seis, a aplicação prática da ferramenta para suportar o processo de gerenciamento de requisitos é descrita. Inicia com o planejamento das atividades, documenta a forma de aplicação, as técnicas utilizadas e um relato do acompanhamento do processo.

O capítulo sete apresenta a conclusão, a dimensão do trabalho realizado, as considerações finais e a ligação com o desenvolvimento de trabalhos futuros.

CAPÍTULO 2

Modelos de Qualidade de Software

2.1	INTRODUÇÃO	27
2.2	METODOLOGIAS DE DESENVOLVIMENTO	28
2.3	MUDANDO DE DIREÇÃO	31
2.4	FOCO NO PROCESSO.....	32
2.4.1	DEFINIÇÃO DO PROCESSO DA ORGANIZAÇÃO.....	34
2.4.2	ATIVOS DO PROCESSO DE SOFTWARE (SOFTWARE PROCESS ASSETS)	35
2.4.3	CONTEÚDO DO PROCESSO PADRÃO DA ORGANIZAÇÃO.....	36
2.4.4	UTILIZANDO O PROCESSO PADRÃO.....	36
2.5	OS MODELOS DE QUALIDADE	37
2.5.1	SW-CMM.....	38
2.5.2	NBR ISO/IEC 12.207	40
2.5.3	ISO/IEC 15.504	43
2.5.4	MODELO CMMI	44
2.6	MODELAGEM DE PROCESSO.....	51
2.7	AMBIENTE AGIR.....	52
2.8	CONCLUSÃO.....	54
2.9	RESUMO	55

2 Modelos de Qualidade de Software

Neste capítulo, apresentamos os modelos de qualidade que nortearam o trabalho e dentre eles, dedicamos especial atenção ao modelo CMMI em como ele aborda o tratamento dos requisitos de um produto de software.

2.1 Introdução

Desenvolver software de qualidade assegurada, com elevada produtividade, dentro do prazo estabelecido e sem necessitar de mais recursos que os alocados, tem sido o grande desafio da Engenharia de Software.

Já se passaram 56 anos desde que o primeiro computador entrou em operação. O ENIAC iniciou suas atividades em 1945 nos Estados Unidos. Com ele, novos conceitos deram o ar da graça. Talvez o mais duradouro destes conceitos, que se mantém firme na informática, seja o BUG, palavra originada dos primeiros problemas com que o software teve de conviver. Graças às mariposas que se alojavam nas válvulas do grande computador em busca de calor e provocavam problemas no hardware, fazendo com que o software que lá estava passasse a apresentar resultados errôneos em seus processamentos.

Com o passar do tempo, o software começou a ir se complicando na medida da evolução tecnológica dos computadores. Os programas de computador passaram a ficar maiores e mais complexos. Porém, era notório que os computadores evoluíam, demandando mais necessidade de software e esta demanda crescente tornando cada vez mais clara a dificuldade de construí-lo.

Esta dificuldade para desenvolver software, teve como reflexo um acúmulo entre demanda e produção, gerando o conhecido backlog na área de desenvolvimento. As organizações passaram a ter muita necessidade por software para suportar seus negócios e isto refletiu na busca de aumento de produtividade e qualidade no desenvolvimento de software. Produtividade era vital para se diminuir o backlog, construindo mais com menos e qualidade era fundamental para que a área de desenvolvimento utilizasse grande parte do seu tempo na construção de novos produtos ao invés de utilizá-lo na manutenção dos existentes. Mas o tempo passava e tal cenário não se concretizava.

Pelo contrário, cada novo software construído implicava menos tempo da área de desenvolvimento para construir novos produtos e mais tempo para manter os já implantados operando de forma adequada.

Este tempo elevado que era gasto na manutenção do software após a sua implantação em operação detonou uma corrida em busca da solução deste problema. As organizações viam aumentar a passos largos os seus orçamentos para a informática, que eram consumidos cada vez mais para diminuição do backlog e menos para desenvolvimento de novos produtos, implicando baixo retorno sobre o investimento.

Em meados da década de setenta uma cruzada deu início na busca da solução.

2.2 Metodologias de desenvolvimento

A grande dificuldade de transformar as necessidades do cliente em um software de qualidade orientou as pesquisas científicas na busca do desenvolvimento de uma engenharia de software similar a outras engenharias, como a de construção civil. A idéia era a quebra do paradigma vigente, cujo desenvolvimento era desestruturado, embasado em descrições textuais e dependente das pessoas envolvidas no projeto para o seu êxito, para um desenvolvimento estruturado com a elaboração de modelos que se integrassem através de um ciclo de desenvolvimento com fases bem definidas. Estava claro que o desenvolvimento de software tinha que buscar uma forma de representação de mais fácil entendimento e comunicação entre desenvolvedores e usuários como forma de melhorar o produto final.

Tal esforço iniciou com o trabalho pioneiro de Yourdon e Constantine publicado em 1978 com o desenvolvimento de modelos para elaboração do desenho e implementação do sistema[6]. A partir desta publicação, os esforços se concentraram na definição clara do ciclo de vida enfatizando as etapas iniciais de análise e projeto de sistemas estruturados. Um trabalho chave foi o de Gane e Sarson[7] publicado em 1979, concentrado na elaboração de modelos para descobrir o que deveria ser feito pelo software a ser construído e não em como fazê-lo. Defendia a abordagem top-down, por refinamentos sucessivos, de forma a enxergar primeiro a floresta e a partir daí às árvores. Pode-se definir esta, se não a maior, como uma das maiores contribuições literárias para a engenharia de

software. Alguns autores contribuíram com o trabalho de Gane e Sarson, entre eles, De Marco, Weinberg e Page-Jones. A análise estruturada pode ser considerada um grande avanço, pois, definiu um caminho a ser seguido pelos desenvolvedores de software, porém, percebeu-se sérios problemas com esta abordagem. O problema mais comum é que esta abordagem parte da modelagem do sistema físico atual que o usuário atualmente utiliza. Deste modelo, muitas vezes, pode-se aproveitar apenas 25% do seu conteúdo na transição para o modelo lógico atual. Isto se deve a redundância que o modelo físico incorpora, na medida que pode modelar uma função várias vezes, pelo fato de ser executada em diversos pontos do sistema atual além da duplicação de outros elementos de dados e as funções de verificação, validação e correção de erros, que são adequadas no sistema físico atual[14]. Esta abordagem, consumia muito tempo na elaboração dos modelos.

Em 1984 duas publicações, de James Martin e de Palmer e McMenamin[8], lançaram a análise essencial ou engenharia da informação, precursora de muitas das idéias da orientação a objetos, como encapsulamento, modularidade e a idéia de tratamento da essência do sistema, podendo ser considerada mais um aperfeiçoamento da análise estruturada que uma nova metodologia com um paradigma diferente. A idéia básica da metodologia é iniciar a construção de um modelo essencial que indica o que o sistema deve fazer para satisfazer os requisitos do usuário e procurar abstrair detalhes de implementação que se referem a forma como o sistema deve fazer. A elaboração do diagrama de contexto e lista de eventos são utilizados para derivar o modelo comportamental do sistema e então o modelo de implementação. A característica chave desta abordagem é uma transição mais suave em que a lista de eventos passa a ser modelada em DFDs e por fim os DFDs são agrupados em processos. Tal abordagem é conhecida por middle-up, pelo fato de partir dos DFDs dos eventos (DFD Preliminar) para então agrupá-los em grupos de processos, gerando o DFD nível 0 e middle-down quando inicia a expansão dos processos do nível 0, gerando níveis mais detalhados (1,2,...).

A grande contribuição da Análise Estruturada e da Análise Essencial ou Análise Estruturada Moderna[22] foi a de consolidar um ciclo de vida de desenvolvimento de software com fases bem definidas e suportadas por técnicas e ferramentas e a transcrição dos requisitos do usuário para modelos gráficos de mais fácil entendimento. Esta vantagem também trouxe impactos negativos para a fase de manutenção. Com a construção de diversos modelos, uma alteração nos requisitos do sistema, implicava a modificação dos modelos gerando um esforço adicional. Pode-se

dizer que avançamos neste período, porém, muito aquém das expectativas, como mostra a figura 2.1, que previa reduzir drasticamente o tempo dedicado à manutenção do software[9].

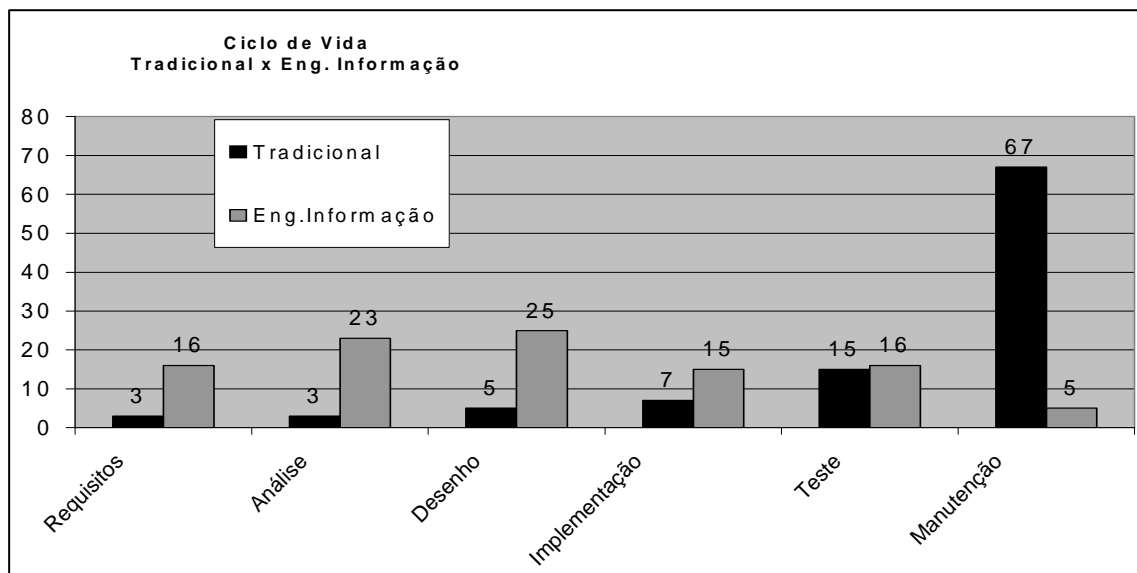


Figura 2.1 – Comparativo de tempo Análise Estruturada X Engenharia da Informação

Na prática, esta mudança drástica de tempos nunca aconteceu. As organizações continuaram com um alto backlog que implicava um percentual elevado de tempo na fase de manutenção do software. Melhoramos nossa capacidade de representar o mundo real através de modelos. Conseguimos melhorar nossa comunicação com o usuário. Mas não o suficiente para dizer que a engenharia de software atingiu um bom nível de qualidade e produtividade. Pode-se identificar entres os principais fatores desta melhora não ter sido tão significativa:

- dificuldade de identificar os requisitos do cliente;
- geração de muitos modelos durante o ciclo de vida e dificuldade de mantê-los quando o software precisa ser alterado;
- dificuldade de reutilizar componentes;
- ambientes de desenvolvimento desorganizados;
- falta de planejamento e acompanhamento do desenvolvimento;
- dificuldade de previsão de custo, prazo e recursos para desenvolvimento.

A partir de 1988, iniciaram-se esforços buscando a quebra de paradigma da análise estruturada. Surge a metodologia orientada a objetos pregando que o mundo real é orientado a objetos e que o desenvolvimento de software também deve ser encarado desta maneira.

Com certeza, a orientação a objetos nos trouxe uma nova forma de enxergar o mundo, porém, resolveu poucos dos problemas listados acima. Por quê ?

2.3 Mudando de Direção

Organizações de software em todo o mundo empregam perto de 7 milhões de técnicos e geram anualmente uma receita de mais de 600 bilhões de dólares, com taxa de crescimento anual de mais de 25% nos últimos três anos. Metade desta receita é gerada por pacotes de software de uso geral, como os ERPs e a outra metade com o desenvolvimento de produtos específicos para clientes. O segmento de software é visto atualmente como um dos mais promissores ramos da indústria com um enorme potencial futuro.

Se considerarmos o desenvolvimento de software como um projeto, então, o foco desta indústria está na execução de projetos. Assumindo que em média, cada projeto consome 7 pessoas/ano de esforço, então, a indústria de software, com seus 7 milhões de técnicos, executam algo em torno de 1 milhão de projetos por ano. Logo, desenvolver projetos de software eficientes é de fundamental importância para a indústria de software como um todo.

Os processos usados para desenvolver um projeto de software tem a maior importância na qualidade do software produzido e na produtividade alcançada pelo projeto. Por consequência, existe uma necessidade de evoluir os processos usados em uma organização para desenvolver projetos de software [19].

O principal objetivo da engenharia de software é melhorar a qualidade do software. A qualidade de produtos de software está fortemente relacionada à qualidade do processo de desenvolvimento de software[23].

A partir desta visão, na década de 90, houve uma grande preocupação com a modelagem e melhoria no processo de desenvolvimento. Abordagens importantes como as normas ISO 9000 e a ISO/IEC 12.207, os modelos CMM e o SPICE sugerem que melhorando o processo de software, podemos melhorar a qualidade dos produtos [24]. Prevê-se para os próximos anos que após terem

ajustados os seus processos para a produção de software de qualidade dentro de prazos e orçamentos confiáveis, as organizações serão pressionadas por seus concorrentes a reduzir substancialmente os prazos para a entrega de produtos. Organizações que sejam capazes de integrar e acelerar seus processos de desenvolvimento e manutenção de software, terão a primazia do mercado[25].

2.4 Foco no Processo

Um projeto de desenvolvimento de software visa a construção de um produto de software que satisfaça as necessidades do cliente e seja desenvolvido e entregue dentro do custo e prazo especificados. Em outras palavras, as três principais características de um projeto são: custo, planejamento e qualidade, onde qualidade representa quão bom o produto é e satisfaz o cliente[50].

Um projeto é um sucesso quando atinge ou excede as expectativas destas três características – custo, planejamento e qualidade. Porém, a indústria de software pode citar muitos exemplos de projetos que fracassaram. A situação vem melhorando com o passar dos anos mas ainda temos muitos projetos fracassando em termos de cronograma, custo e qualidade. Uma pesquisa utilizando dados de projetos [21] mostra que 1/3 destes projetos tem custo e cronograma além de 125% da estimativa inicial. Exemplos de projetos que saíram do controle tem sido documentados [20].

As possíveis razões para falhas de projeto incluem estimativas impróprias, falta de gerenciamento de requisitos, fraco gerenciamento de projeto, gerenciamento de risco impróprio, pobres soluções de engenharia. Muitas destas razões podem ser combinadas em uma categoria chamada “falha do processo”. Um projeto de software falha muitas vezes porque o processo seguido não é apropriado. As maiores razões para o descontrole são: objetivos pouco claros, planejamento ruim, utilização de novas tecnologias, falta de uma metodologia de gerenciamento de projeto e pessoal insuficiente [20]. Todas estas razões podem ser consideradas falha do processo. Para um projeto ter sucesso, um parâmetro chave é o conjunto de processos utilizados. Se processos apropriados são utilizados, as chances do projeto ter sucesso é extremamente alta.

Alta produtividade pode gerar redução de custo e minimizar o tempo do projeto. Qualidade e produtividade são a meta de toda a organização, porém, ainda existe um objetivo mais primordial das organizações. É a previsibilidade. Não é suficiente que um projeto tenha alta qualidade e produtividade. A organização também quer prever qual será a qualidade e produtividade atingida. Se não se consegue ter previsibilidade, é porque não se consegue realizar estimativas apropriadas e isto é o mesmo que caminhar no escuro. Nunca se sabe se está indo no caminho certo.

Como o processo tem o maior efeito sobre qualidade e produtividade, um modo de melhorar este par está na melhoria do processo usado pela organização. Por fim, vale a pena lembrar que toda organização que se considera apta para enfrentar a competição do terceiro milênio deve trabalhar com alta qualidade e produtividade. E qualidade e produtividade de uma organização depende de três fatores: processos, tecnologia e pessoas. É o chamado triângulo da qualidade [15] mostrado na figura 2.2:

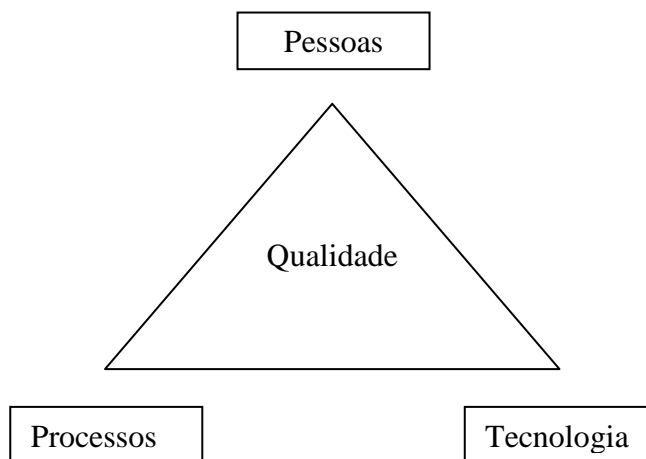


Figura 2.2 – Triângulo da Qualidade

2.4.1 Definição do Processo da Organização

Chamado pelo CMM de processo padrão da organização para desenvolvimento de projetos de software, este processo inclui processos de gerenciamento de projeto e de engenharia de software. A partir deste processo padrão, adaptações necessárias a cada projeto são realizadas, derivando o processo de software definido que cada projeto, de forma individual, irá utilizar. Este processo definido, nada mais é do que processos de gerência de projeto e de engenharia de software selecionados do processo padrão, para atender as características individuais dos projetos.

A definição e uso de processos de software envolve a complexa inter-relação de fatores organizacionais, culturais, tecnológicos e econômicos[23]. Entretanto, um único processo não pode servir a qualquer tipo de empresa e projeto. Questões relacionadas ao porte da empresa e à cultura organizacional, objetivos de projetos específicos, recursos disponíveis, tecnologias de desenvolvimento, conhecimento e experiência da equipe impõem características aos processos. Assim, processos de software necessitam ser definidos e adequados a uma determinada situação. Esta diversidade de aspectos torna cada projeto único, devendo ser este processo de desenvolvimento derivado do processo padrão, porém, específico para cada projeto, derivando então, o processo definido para o projeto.

Um processo é chamado definido quando possui entradas, padrões e procedimentos, mecanismos de verificação, saídas e critérios de aceitação e é baseado em um entendimento comum das atividades, papéis e responsabilidades dos envolvidos no projeto de desenvolvimento de software. A capacitação deixa de ser uma habilidade das pessoas e passa a ser uma habilidade da organização [18].

A idéia básica da definição do processo da organização é desenvolver um conjunto de padrões tanto de atividades de gerência quanto de engenharia de software que possam ser utilizados por todos os projetos de software da organização, porém, flexíveis o bastante para possibilitar a sua adaptação às particularidades de cada projeto.

Alguns dos principais ganhos de se investir na padronização e qualidade dos processos de desenvolvimento de software[51]:

Entendimento comum dos envolvidos (técnicos e não técnicos) sobre o processo e suas atividades e os papéis e responsabilidades de cada um no projeto;

Cada novo projeto é a consolidação do aprendizado anterior, pois as atividades a executar serão basicamente as mesmas;

Uma flexibilidade maior na movimentação de técnicos durante a execução do projeto devido a capacitação do processo ser da organização e não uma habilidade das pessoas. É claro que tirar uma pessoa de um projeto em andamento é sempre difícil pelo conhecimento do negócio que a mesma adquiriu, porém, sem um processo definido e documentado, a remoção bem como a inserção de um técnico no projeto é praticamente impossível.

Uma evolução natural do processo padrão da organização pode ocorrer a cada avaliação dos projetos realizados, através de melhorias que podem ser sugeridas, já que todo projeto tem seu processo de execução definido a partir do processo padrão. Esta é a grande vantagem de se ter na organização uma estrutura comum de trabalho. Neste caso, todos utilizam e contribuem na melhoria do processo ao passo que o trabalho isolado, em que cada um faz o que quer, contribui minimamente para a melhoria da capacitação de desenvolvimento da organização.

2.4.2 Ativos do Processo de Software (Software Process Assets)

Cada elemento deste conjunto de padrões, o CMM chama de ativos do processo os quais fazem parte da estrutura conceitual do processo de desenvolvimento de software descrita abaixo:

- **Arquitetura do Processo de Software:** é uma descrição do processo de software padrão da organização, os elementos que o compõe e o inter-relacionamento entre estes elementos;
- **Elemento do Processo de Software:** é um componente do processo padrão e descreve um conjunto de atividades com forte relacionamento. Podem ser considerados como elementos o processo de estimativa de software, revisão, design, codificação, documentação,...;

- Ciclo de Vida do Software: deve prever os possíveis ciclos de vida utilizados pelos projetos da organização. O CMM cita o cascata, cascata estendido, espiral e outros [53];
- Diretrizes e Critérios para adaptação: asseguram a existência de uma base comum para todos os projetos de software e descrevem o que pode ou não ser adaptado do processo padrão para o processo definido;
- Base de dados dos processos de software da organização: é uma base de dados utilizada para disponibilizar informações sobre os processos de software e os produtos (artefatos ou produtos intermediários) de software produzidos, tais como os dados de medições, estimativas, custo, número de defeitos, etc;
- Biblioteca de documentação de processos de software: utilizada para armazenar documentos relacionados ao processo de software padrão da organização e permitir o compartilhamento destas informações por toda a organização. Pode conter exemplos de documentos, projetos de sucesso, padrões, procedimentos, planos.

2.4.3 Conteúdo do Processo Padrão da Organização

- Decomposição do processo de desenvolvimento de software em elementos fundamentais, de tal forma que estejam bem descritos e possam ser seguidos. Cada um destes elementos é composto de atividades bem definidas e inter-relacionadas;
- Descrição do relacionamento entre os elementos do processo, apontando a ordem de execução dos elementos, interface e a interdependência.

2.4.4 Utilizando o processo padrão

A partir da definição do processo padrão da organização, este é utilizado a cada novo projeto, para a geração do processo definido do projeto que é adaptado para abranger suas características específicas. O plano de desenvolvimento de software deve ser feito a partir do

processo definido do projeto e deverá descrever as atividades que serão executadas, implementadas e controladas. Esta adaptação do processo padrão até o plano de desenvolvimento de software ocorre de acordo com a figura 2.4:

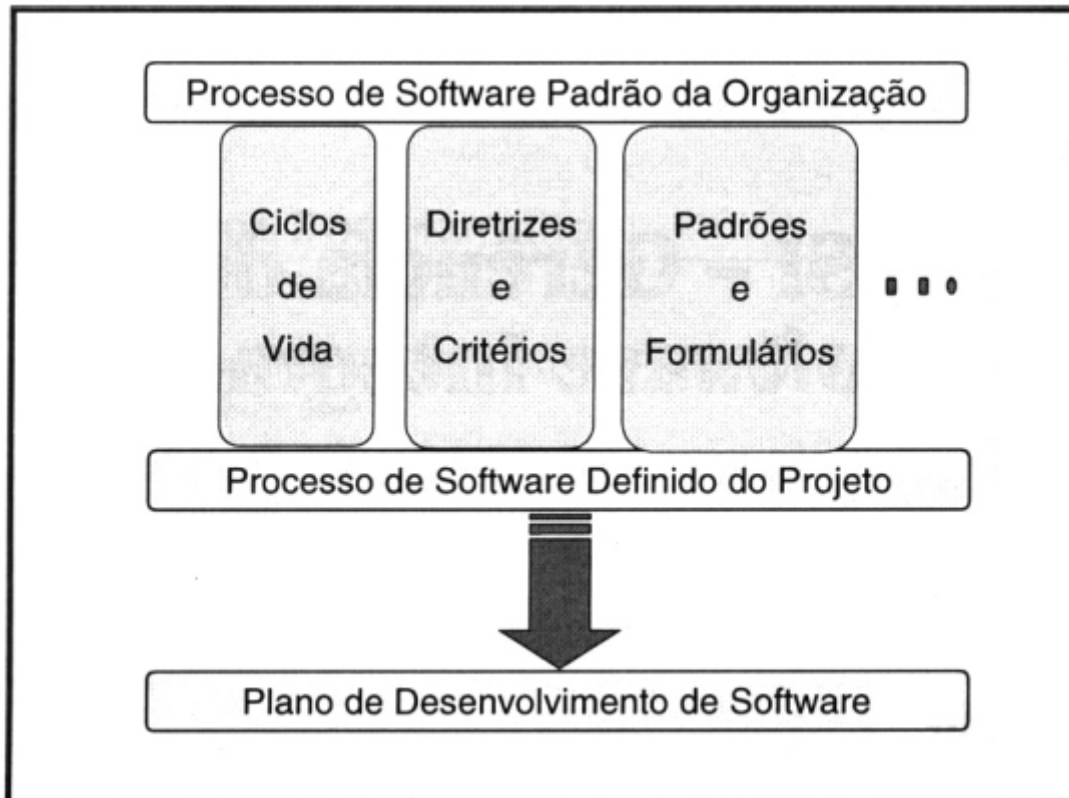


Figura 2.4 – Adaptação do Processo Padrão

2.5 Os Modelos de Qualidade

Os modelos de qualidade tratam justamente dos processos que caracterizam o nível de maturidade das organizações. O processo de gerenciamento de requisitos é tão fundamental, que os principais modelos da atualidade, o CMM e sua variante CMMI e a ISO o contemplam. Faz-se necessário neste ponto, detalharmos os modelos de qualidade, principalmente o CMMI, que possui duas áreas chave de processo para o tratamento dos requisitos de um produto de software: uma chamada Desenvolvimento de Requisitos e outra Gerenciamento de Requisitos.

2.5.1 SW-CMM

Em 1991, o SEI (Software Engineering Institute) publicou o CMM (Capability Maturity Model) para software, um framework de maturidade que estabelece fundamentos de engenharia de software e de gerenciamento de projeto para controle quantitativo do processo de software[9]. Segundo o CMM, a principal causa dos problemas de qualidade e produtividade é a falta de um processo de desenvolvimento de software claramente definido e efetivo. Conhecer os processos significa conhecer como os produtos e serviços são planejados, produzidos e entregues. Cabe ressaltar que, a partir da definição do processo, é possível definir-se medições e coletar dados de execução. Isto dá visibilidade aos gerentes e técnicos sobre o andamento dos projetos, possibilitando ações para controlar as variações do projeto e dos processos por ele utilizados. O CMM enfatiza a documentação dos processos, seguindo a premissa de que, para realizar alguma melhoria no processo, é preciso primeiro conhecê-lo e entendê-lo, e que a qualidade de um produto é reflexo da qualidade e gerenciamento do processo utilizado em seu desenvolvimento.

A Estrutura do CMM

O CMM propõe um caminho gradual que leva as organizações a se aprimorarem continuamente na busca da sua própria solução dos problemas inerentes ao desenvolvimento sistemático de software. Este caminho gradual se apresenta no CMM através de 5 níveis de maturidade que inicia no nível 1, em que a improvisação rege o processo de desenvolvimento até o nível 5 cujo foco é a melhoria sistemática do processo. Cada nível possui áreas chave de processo (ACP), que por sua vez possuem metas de melhoria do processo que são atingidas através de um conjunto de atividades realizadas para este fim (práticas chave).

Ao todo o CMM descreve 18 áreas chave de processo, distribuídas pelos 5 níveis de maturidade, conforme a tabela 2.1 abaixo:

Nível de Maturidade	Número de ACPs	Objetivo das ACPs
1 – Inicial	Não possui	
2 – Repetitivo	6	Controles básicos de gerência
3 – Definido	7	Processo padrão da organização
4 – Gerenciado	2	Controle do processo
5 – Otimizado	3	Melhoria contínua

Tabela 2.1 – Número de Áreas Chave por Nível de Maturidade

No nível 2, a preocupação do CMM está focada nos aspectos gerenciais, geralmente os problemas mais imediatos da maioria das organizações. A partir do nível 3, então, a solução de problemas técnicos, de engenharia de software, passam a ser tratados. É justamente o nível 2 do CMM em que se encontra o escopo deste trabalho.

No nível 2 o processo passa a ter um nível básico de gerenciamento do desenvolvimento de software, através de marcos de acompanhamento do andamento do projeto (milestones). Os compromissos do projeto são baseados em resultados de projetos anteriores e nos seus requisitos. É possível acompanhar custos, cronogramas e funcionalidades do software em desenvolvimento. Os problemas são tratados a medida em que vão aparecendo; é o que conhecemos por gerência reativa. Neste nível, os projetos podem seguir diferentes processos de engenharia de software, porém, são guiados por um processo de gerenciamento padrão. Podemos dizer que os projetos estão sob um controle efetivo, seguindo planos realistas, baseados em desempenhos de projetos anteriores.

Uma das ênfases do CMM é a definição do processo de software da organização, ou seja, aquilo que é realizado está documentado e de acordo com o processo padrão da organização. Este conceito de processo padrão da organização é a padronização na organização dos processos de gerenciamento de projeto e de engenharia de software. A ênfase na descrição do processo padrão se dá a partir do nível 3. Este processo padrão da organização define todos os processos básicos e comuns aos projetos de software da organização. Todos os projetos, são adaptados do processo

padrão da organização, para definir os processos que serão utilizados em cada projeto. Este conjunto de processos selecionados do processo padrão da organização para a execução de um projeto é conhecido como processo definido do projeto. A partir do momento que uma organização estabelece o seu processo padrão de desenvolvimento de software, esta passa a desenvolver projetos de forma homogênea, facilitando o entendimento dos projetos por parte de todos os envolvidos.

Uma pesquisa buscando identificar o nível de capacitação das organizações dentro do modelo CMM foi realizada pelo SEI. De acordo com a pesquisa [16] que avaliou 533 organizações de 1992 a 1996, foi revelado que 61,5% destas organizações ainda desenvolvem software de forma amadora e se encontram no nível 1 do modelo, enquanto que apenas 13,5% possuem um processo padrão de desenvolvimento. Sobre organizações brasileiras, o relatório brasileiro de 1995 [17] , baseado na avaliação de 445 empresas mostra que apenas 2,7% das empresas conheciam e estavam adotando o CMM.

2.5.2 NBR ISO/IEC 12.207

Esta norma tem por objetivo estabelecer os processos e atividades que compõem o processo de desenvolvimento de software, visando auxiliar os envolvidos na produção de software a obter um melhor entendimento das atividades que devem ser executadas e do papel de cada um no processo.

a) Arquitetura da Norma

A norma estabelece uma arquitetura de alto nível para o ciclo de vida do software que abrange desde a sua concepção até a descontinuidade. Esta arquitetura de processos segue dois princípios básicos[27], conforme figura 2.5:

- Modularidade: os processos têm alta coesão e baixo acoplamento, ou seja, todas as partes de um processo são fortemente relacionadas e o número de interfaces entre os processos é mantido ao

mínimo, visando facilitar modificações nos processos, de forma que a alteração de um processo impacte o mínimo na estrutura dos outros processos;

- Responsabilidade: cada processo na norma é de responsabilidade de uma 'parte envolvida', que pode ser uma organização ou parte dela. As partes envolvidas podem ser da mesma organização ou de organizações diferentes.

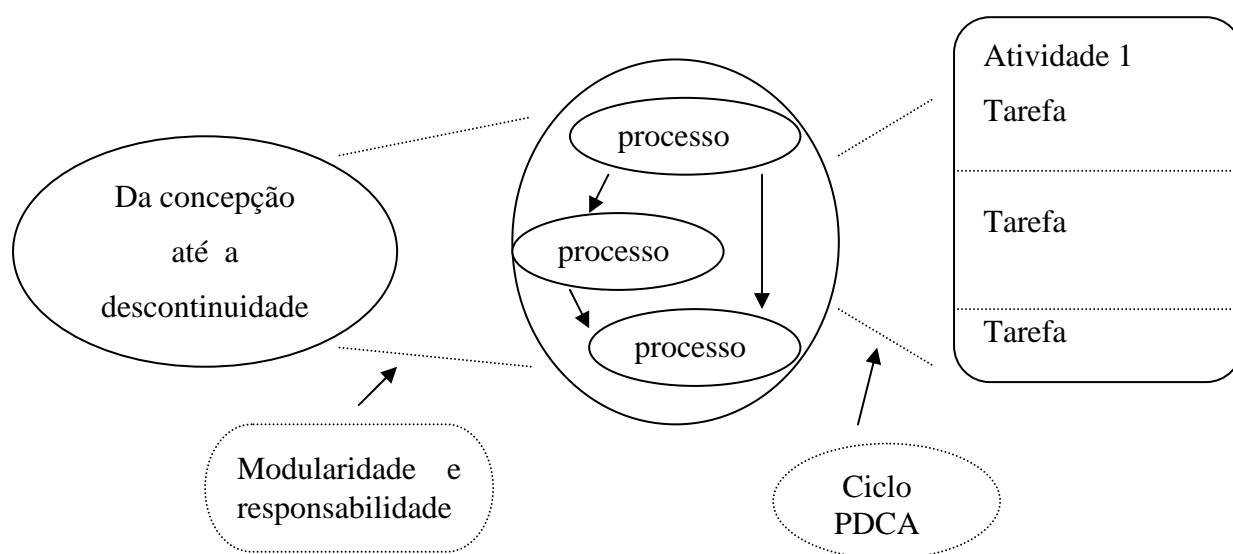


Figura 2.5 – Arquitetura da norma NBR ISO/IEC 12207.

Os processos que envolvem o ciclo de vida do software são agrupados em três classes a seguir, conforme figura 2.6:

- Processos fundamentais: compreende os processos envolvidos na execução do desenvolvimento, operação e manutenção do software durante o ciclo de vida. Esses processos são: aquisição, fornecimento, desenvolvimento, operação e manutenção. É justamente nestes processos que ocorre a atividade de gerenciamento de requisitos;
- Processos de apoio: são os processos que auxiliam o sucesso e a qualidade do projeto de software. Compreende os seguintes processos: documentação, gerência de configuração,

garantia da qualidade, verificação, validação, revisão conjunta, auditoria e resolução de problemas;

- Processos organizacionais: são os processos empregados por uma organização para estabelecer e implementar uma estrutura constituída pelos processos do ciclo de vida e pelo pessoal envolvido no desenvolvimento de software. São eles: gerência, infra-estrutura, melhoria e treinamento.

Existe ainda um processo de adaptação que define as atividades necessárias para adaptar a norma para sua aplicação em um determinado projeto, tornando-a muito flexível.

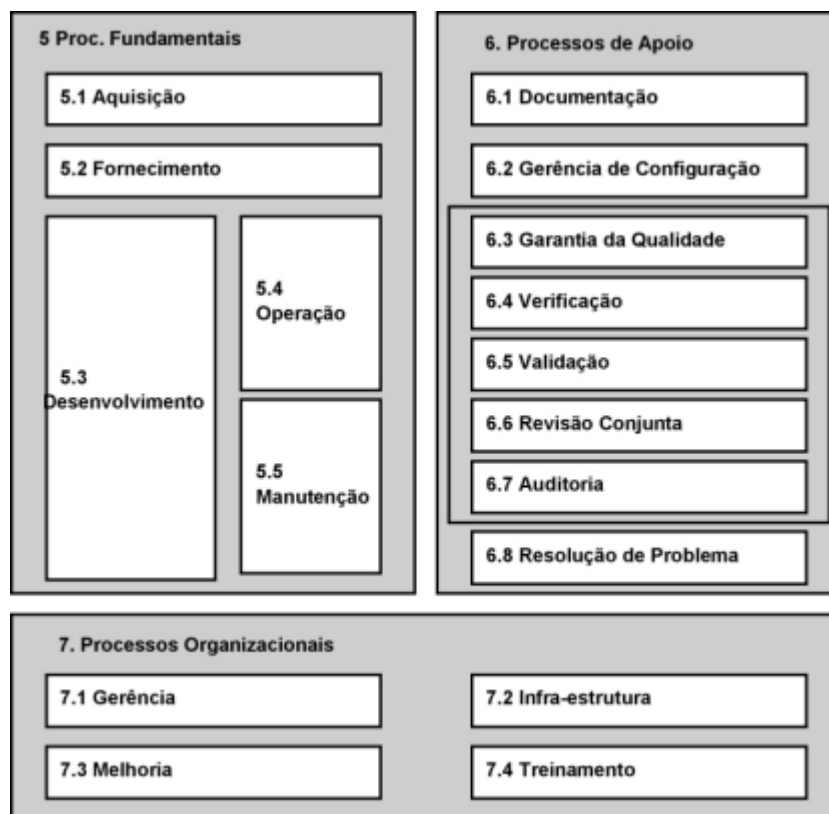


Figura 2.6 – Arquitetura da Norma NBR ISO/IEC 12.207

Uma nova versão desta norma, prevista para 2002, visa retratar as melhores práticas adotadas pela comunidade de software.

2.5.3 ISO/IEC 15.504

O projeto de elaboração da norma 15.504, também conhecido como SPICE, tem como foco a realização de avaliações de processos de software com dois objetivos: a melhoria dos processos e a determinação de capacidade de processos de uma organização. A necessidade de uma padronização global de avaliação dos processos de software motivou a sua criação [52].

Para a realização de uma avaliação é necessário que o modelo de processo utilizado seja compatível com o modelo de referência de processo da norma NBR ISO/IEC 12.207, já que este é o modelo utilizado pela norma ISO/IEC 15.504. O modelo de referência estabelecido nesta norma define duas dimensões:

a) A dimensão de processo

Agrupar os processos em cinco categorias de acordo com o tipo de atividades que executam. Cada processo é descrito em termos de um propósito que exprime um único objetivo funcional. As tarefas, atividades e práticas, as características dos artefatos produzidos são utilizados como indicadores que demonstram se determinado processo é praticado em um determinado nível de capacidade. Esses indicadores são classificados de acordo com uma escala que expressa justamente o seu nível de capacidade.

Esta dimensão, não contemplada no modelo CMM, é fundamental para que as organizações possam selecionar processos, considerados por elas mais críticos e levá-los a patamares mais elevados de capacidade, enquanto outros, possam se manter em níveis mais baixos.

b) A dimensão de capacidade

Visa identificar um conjunto de atributos que permite determinar o nível de maturidade do processo de desenvolvimento de software de uma organização. Esta dimensão de capacidade é muito similar ao proposto pelo modelo CMM e estabelece níveis de maturidade, descritos abaixo:

- Nível 0 – Incompleto: não existe evidência de que os produtos de trabalho sejam adequadamente produzidos e os resultados alcançados;
- Nível 1 – Executado: o processo gera seus produtos de trabalho evidenciando que os resultados são alcançados;
- Nível 2 – Gerenciado: o processo gera produtos de trabalho de acordo com procedimentos e padrões específicos e sua execução é planejada e acompanhada.
- Nível 3 – Estabelecido: o processo é definido a partir de um processo padrão da organização, que contempla princípios de engenharia de software e de gerência de projeto;
- Nível 4 – Previsível: o processo definido é executado dentro de limites de controle definidos, de modo a atingir as metas definidas para o processo. Medições detalhadas de desempenho são coletadas e analisadas;
- Nível 5 – Em otimização: o desempenho do processo é continuamente melhorado para atingir as necessidades atuais e futuras do negócio.

A abordagem de duas dimensões influenciou decisivamente no aparecimento do modelo CMMI, em sua variante contínua [28], cujo modelo é muito aderente a esta norma.

2.5.4 Modelo CMMI

Este modelo de qualidade foi desenvolvido para integrar três disciplinas: engenharia de software, engenharia de sistemas e desenvolvimento integrado de produto e processo. Os modelos existentes escolhidos para serem utilizados como fonte primária para o conjunto inicial de modelos CMMI são: SW-CMM v2.0 draft C, EIA/IS 731 e IPD-CMM v0.9.a. Durante o desenvolvimento do CMMI, notou-se interesse e a necessidade de assegurar que todos os produtos desenvolvidos estariam de acordo com a ISO/IEC 15.504. Como o CMM é reconhecidamente o padrão de utilização e a ISO/IEC 12.207 e 15.504 tendem a ser um padrão amplamente aceito, estar de acordo com ambos é estratégico para o projeto CMMI.

a) Áreas chave do CMMI X Gerenciamento de Requisitos

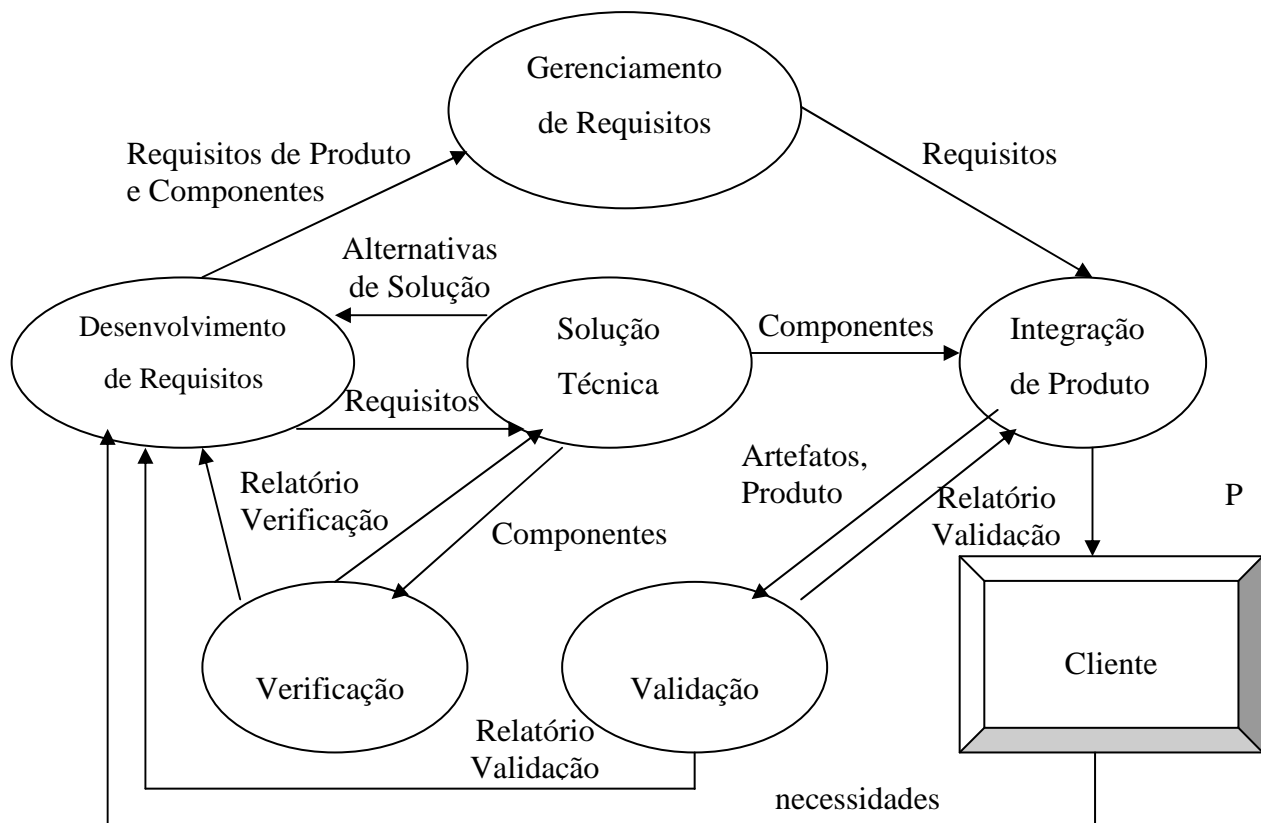


Figura 2.7 – Relacionamento de GR com outras Áreas Chave do CMMI

b) Desenvolvimento de Requisitos

A área de processo de Desenvolvimento de Requisitos, inexistente no modelo SW-CMM, inclui 3 principais grupos de práticas. O primeiro inclui aquilo que é necessário para definir um conjunto completo de requisitos do cliente para usar no desenvolvimento dos requisitos do produto. O segundo inclui aquilo que é necessário para definir um conjunto completo de requisitos de produto e componentes de produto para usar no design dos produtos e componentes. O terceiro inclui aquilo que é necessário para executar a análise para definir, derivar e entender os requisitos.

Os três grupos de práticas podem interagir recursivamente entre eles na definição de alternativas de solução e desenvolver conceitos do produto para a área de processo Solução Técnica[28].

Requisitos são desenvolvidos para ser a base do design. Isto inclui:

- Coletar e coordenar as necessidades dos stakeholders;
- Desenvolvimento do ciclo de vida dos requisitos do produto;
- Estabelecimento dos requisitos do cliente;
- Estabelecimento inicial dos requisitos do produto e componentes consistentes com os requisitos do cliente;
- Descobrimto, análise e comunicação das necessidades do cliente, expectativas e restrições para obtenção dos requisitos do cliente que constituem um entendimento do que satisfará os stakeholders.

O resultado da análise de requisitos e do conceito operacional (incluindo funcionalidade, suporte, manutenção e disponibilização) produz mais requisitos derivados, incluindo:

- Restrições de vários tipos;
- Limitação tecnológica;
- Custo e direcionadores de custo;
- Restrição de tempo e direcionadores de cronograma;
- Riscos;
- Consideração das questões envolvidas e não explicitamente declaradas pelo cliente ou usuário final;
- Fatores introduzidos pelos desenvolvedores sobre considerações de negócio, regulamentos e leis.

i. Áreas de Processo Relacionadas

Relaciona-se com a área de processo Gerenciamento de Requisitos para geração e obtenção de mais informação sobre gerenciamento de requisitos de cliente e produto, obtenção de acordo

com os fornecedores de requisitos, obtenção de compromissos com os implementadores dos requisitos e rastreabilidade.

Com a área de processo Solução Técnica na medida que as informações das saídas do Desenvolvimento de Requisitos são usadas por esta área para o desenvolvimento de soluções alternativas e de design usadas para refinar e derivar requisitos.

A área de processo Integração de Produto utiliza a informação dos requisitos de interface e gerenciamento de interface.

Relaciona-se com a área de processo Verificação que utiliza os requisitos para verificar se o produto resultante atende as necessidades dos stakeholders.

A área de processo Validação utiliza os requisitos para validação do produto construído.

Relaciona-se com a área de processo Gerenciamento de Risco que relaciona e gerencia riscos aos requisitos identificados.

Por fim, com a área de processo Gerenciamento de Configuração que deve garantir que os artefatos chave são controlados e gerenciados.

ii. Metas específicas

Desenvolver Requisitos do Cliente

As necessidades dos stakeholders (clientes, usuário final, fornecedores, construtores, testadores) são a base para se determinar os requisitos do cliente. As necessidades dos stakeholders, expectativas, restrições, interfaces, conceito operacional e conceitos do produto são analisados, harmonizados, refinados e elaborados para transformação em um conjunto de requisitos do cliente.

As subpráticas são:

- Coletar necessidades dos stakeholders;
- Descobrir necessidades;
- Transformar necessidades dos stakeholders, expectativas, restrições e interfaces em requisitos do cliente.

Desenvolvimento de Requisitos do Produto

Requisitos do cliente são refinados e elaborados para desenvolver requisitos do produto e componentes para o ciclo de vida do produto. Requisitos do cliente são analisados em conjunto com o desenvolvimento do conceito operacional para derivar em maior detalhe e precisão os requisitos do produto e dos componentes. Requisitos derivados aparecem de restrições, considerações de questões envolvidas mas não explicitamente colocadas na baseline de requisitos do cliente e fatores introduzidos pela seleção da arquitetura, design e de considerações do negócio.

As subpráticas são:

- Estabelecer requisitos do produto e componentes;
- Alocar requisitos aos componentes;
- Identificar requisitos de Interface.

Análise e Validação dos Requisitos

Análises são executadas para determinar qual impacto no ambiente operacional pretendido terá a habilidade para satisfazer as necessidades, expectativas, restrições e interface dos stakeholders. Considerações como viabilidade, missão, restrições de custo, tamanho do potencial de mercado e estratégia de aquisição devem todas ser levadas em conta, dependendo do contexto do produto. Uma definição da funcionalidade requerida é estabelecida. Todos os modos de uso específicos para o produto são considerados, e uma análise de tempo é gerada para estabelecer a seqüência crítica de tempo das funções.

O objetivo da análise é determinar requisitos candidatos para conceitos do produto que satisfarão as necessidades, expectativas e restrições dos stakeholders e então traduzir estes conceitos em requisitos. Em paralelo com esta atividade, os parâmetros que serão utilizados para avaliar a efetividade do produto são determinados baseados nas entradas do cliente e no conceito preliminar do produto.

As subpráticas são:

- Estabelecer conceitos operacionais e cenários;
- Estabelecer uma definição da funcionalidade requerida;

- Análise de requisitos;
- Avaliar custo do produto, cronograma e risco;
- Validar requisitos;
- Validar requisitos com Métodos compreensivos.

c) Gerenciamento de Requisitos

A proposta da área de processo Gerenciamento de Requisitos é gerenciar requisitos de produto e componentes de produto do projeto e identificar inconsistências entre os requisitos e os planos e artefatos do projeto.

O projeto toma passos apropriados para garantir que o acordo sobre o conjunto de requisitos é gerenciado para suportar o planejamento e execução necessários ao projeto. Quando um projeto recebe requisitos de um fornecedor de requisitos aprovado, os requisitos são revisados com os fornecedores para resolver detalhes e evitar enganos antes dos requisitos serem incorporados aos planos do projeto. Após um acordo entre os fornecedores e envolvidos sobre os requisitos, um contrato dos requisitos é obtido dos participantes do projeto que tem de realizar atividades do projeto e implementar os requisitos. O projeto deve gerenciar mudanças nos requisitos porque eles evoluem durante o projeto e identificar qualquer inconsistência que ocorrer entre os requisitos e os planos e artefatos do projeto.

Parte do gerenciamento de requisitos é capturar mudança dos requisitos, a razão e manter um rastro bidirecional dos requisitos e todos os requisitos de produto e de componente de produto.

Essa área de processo é estreitamente ligada com o Desenvolvimento de Requisitos e Solução Técnica cujos processos atuam na transformação das necessidades dos stakeholders em requisitos de produto e decidindo como alocar ou distribuir os requisitos para os componentes de produto. As práticas desta área de processo devem ser feitas em paralelo com as práticas das áreas de processo Desenvolvimento de Requisitos e Solução Técnica.

i. Áreas de Processo Relacionadas

A área de Desenvolvimento de Requisitos refere-se à transformação das necessidades dos stakeholders em requisitos de produto e decide sobre como alocar e distribuir os requisitos para os componentes.

A área Solução Técnica transforma requisitos em soluções técnicas.

A área Planejamento de Projeto preocupa-se em refletir os requisitos e mudança dos requisitos nos planos do projeto.

A área de Gerenciamento de Configuração preocupa-se em controlar e rastrear as atividades e artefatos que são baseados nos requisitos.

ii. Meta Específica

Gerenciamento de Requisitos

A meta é prover o projeto com um conjunto aprovado de requisitos durante a vida do projeto, gerenciando todas as mudanças de requisitos, assinalando o relacionamento bidirecional entre os requisitos e outras entidades afetadas e identificar inconsistências entre os requisitos e planos e artefatos do projeto. Inconsistências identificadas devem gerar ações corretivas.

- Para a Engenharia de Software – os requisitos podem ser um subconjunto dos requisitos totais do produto ou constituir todos os requisitos do produto.
- Para a Engenharia de Sistemas – Cada nível de design do componente do produto recebe os requisitos do nível anterior.

As subpráticas são:

- Obter um entendimento dos requisitos;
- Obter um acordo dos requisitos;
- Gerenciar mudança de requisitos;
- Manter rastro bidirecional dos requisitos;
- Identificar inconsistências entre os requisitos e artefatos.

2.6 Modelagem de Processo

Modelagem de processos é a expressão de conceitos que permite que cada organização compreenda e contribua para o seu próprio desenvolvimento. Os modelos tornam-se valiosos à medida que os processos e produtos de uma organização são modelados e disponibilizados para conhecimento de todos os envolvidos no processo. Os modelos também promovem o entendimento entre as diferentes áreas de processos – gerenciamento, engenharia, etc[48].

Os modelos devem ser claros o suficiente para evitar ambigüidade e representar o que realmente é executado. Tal modelagem deve abranger todas as etapas do desenvolvimento de sistemas e de gerenciamento, além de atividades que visem o próprio processo de negócio.

Dentre os itens importantes de representação de um processo de negócio, estão:

- ✓ Atividade: função que recebe determinados valores, e que a partir de determinados controles e utilizando-se de determinados recursos, gerem outros valores ao processo;
- ✓ Produto ou valor: é a informação produzida e/ou consumida pela execução das atividades. Pode ter um vínculo direto com métricas, pois se um valor é gerado, para um nível de maturidade elevado este valor deve ser medido;
- ✓ Controle: é um tipo de valor que para determinado processo se mantém inalterado durante qualquer *trace* executado;
- ✓ Perfil de conhecimento: conjunto de conhecimentos necessários para que se realize a atividade. Quanto melhor a gerência de conhecimento da organização, melhor será

realizada a alocação destes perfis. Não deve ser tratado como a pessoa em si, mas como os conhecimentos que esta deve possuir para atuar na atividade;

- ✓ Ferramenta: software ou outras ferramentas que são utilizadas para a execução da atividade;
- ✓ Infra-estrutura: equipamentos e localidades que devem ser alocados para a execução da atividade.

2.7 Ambiente AGIR

Muitas organizações ao passarem por evoluções e adequações enfrentam muitos problemas como baixa produtividade e qualidade, alto custo, desperdícios, entre outros. Basicamente, problemas como estes têm origem a partir de dois fatores: impossibilidade e dificuldade que estas organizações tem em se adaptar à mudanças no mundo dos negócios, e dificuldade que as mesmas possuem quando necessitam adquirir um novo software para informatizar seus processos.

Normalmente as organizações têm de se adaptar ao software e não o contrário, fazendo com que algo em torno de 80% dos softwares produzidos sejam desperdiçados, por não atenderem plenamente às necessidades do cliente. Além disso, muitos sistemas apresentam deficiências como: serem desenvolvidos de forma incompleta, não permitirem uma visão de como os recursos estão sendo utilizados, não estarem preparados para mudanças, as tecnologias de informação não estão integradas e o processo de integração representa recursos adicionais.

Uma forma de resolver tais problemas seria a adoção, por parte da organização, de um novo ambiente que visasse a adaptação e a evolução contínua de seus processos, o que permitiria obter uma visão totalmente real de si própria. Um ambiente propício para esta finalidade. Tal ambiente foi desenvolvido pelo Dr. Gattaz e visa superar o gargalo do desenvolvimento de software, causado pela Máquina de Turing, a qual representa uma visão estática da realidade. A sua idéia se baseia em uma Máquina Contextual que possui uma visão dinâmica da realidade, possibilitando a verificação dos efeitos das alterações nos estágios seguintes do sistema, além dos efeitos dos efeitos.

Entre as principais vantagens do ambiente AGIR, podemos citar:

- ✓ Foco no processo de negócio;
- ✓ Baseado no contexto;
- ✓ O código é independente de plataforma;
- ✓ Permite emulação;
- ✓ Possibilita a integração com controle e melhoria contínua.

O ambiente AGIR representa o funcionamento da organização através de processos enfocando três dimensões, conforme representa a figura 2.8:

- ✓ Atividade – representa as funções da organização, suas atividades relevantes e o relacionamento destas ao longo do tempo;
- ✓ Infra-estrutura – representa o comportamento dos recursos (humanos, físicos, tecnológicos, etc.) conforme requerido pelos valores a serem adicionados;
- ✓ Comunicação – representa o conjunto de regras que exprime os valores que guiam uma realidade, sendo responsável pela organização das atividades de produção e dos métodos de trabalho, de modo a maximizar a qualidade e produtividade dos processos organizacionais.

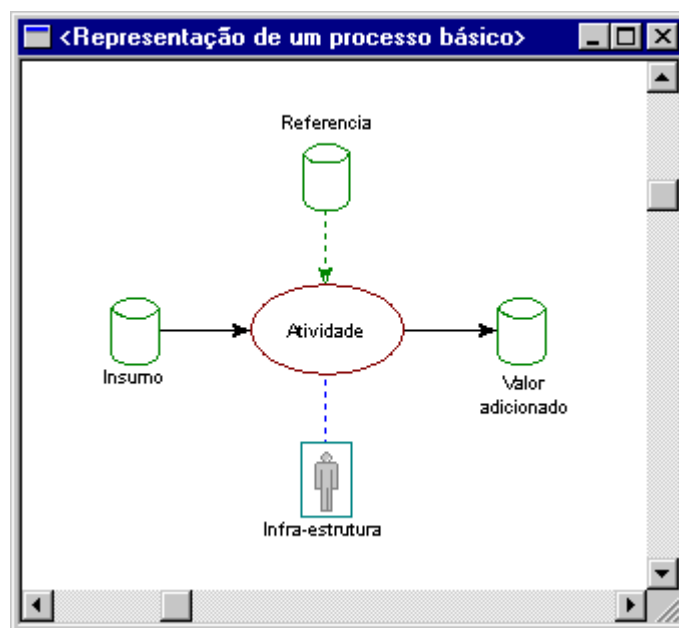


Figura 2.8 – Representação dos Processos no ambiente AGIR

A utilização do ambiente AGIR permite a organização adaptar-se e evoluir continuamente, tornando-a capaz de responder, de forma rápida e efetiva, à mudanças impostas pela atual dinâmica do mundo dos negócios.

2.8 Conclusão

O processo de desenvolvimento de software tem a maior importância na qualidade do produto construído. A utilização de modelos de qualidade de processo, como é o caso do CMMI, procurando unir o melhor do SW-CMM, padrão de utilização, com os conceitos das normas ISO/IEC 12.207 e 15.504, padrões de fato, tende a se tornar um dos mais utilizados guias para melhoria de processo. Nada melhor do que trabalhar na melhoria do processo de desenvolvimento, utilizando uma metodologia cuja abordagem é orientada a processo. Melhor ainda, se suportada por um ambiente informatizado de suporte a esta abordagem.

Modelar os processos de uma organização ajuda a institucionalizar o que deve ser feito pelos seus processos e “como” fazer a execução destes.

Por tudo que foi visto neste capítulo, podemos concluir que seguir um modelo de qualidade é fundamental para implementar processos de desenvolvimento de software, porém, com um

ambiente orientado a processo, a complexidade da manutenção do processo fica mais rápida de ser tratada, já que a modelagem torna-se visível, real, palpável.

2.9 Resumo

- ✓ O principal objetivo da engenharia de software é melhorar a qualidade do software e esta qualidade está fortemente ligada a qualidade do processo de desenvolvimento;
- ✓ O modelo CMMI tende a se tornar líder de mercado, por estar focado no atual padrão de utilização CMM e no futuro padrão de fato, a ISO/IEC 15.504;
- ✓ Organizações que sejam capazes de acelerar seus processos de desenvolvimento e manutenção de software terão a primazia do mercado;
- ✓ Utilizar uma abordagem orientada a processo para definir e manter seus processos de desenvolvimento é fundamental para se obter sucesso;
- ✓ Modelar estes processos em um Ambiente próprio para tratamento de processos, fará com que eles se tornem reais. Se tornando reais são mais fáceis de serem entendidos e seguidos, além de facilitar o processo de manutenção do processo. Para este trabalho o ambiente AGIR foi a ferramenta de modelagem utilizada.

CAPÍTULO 3

Engenharia de Requisitos

3.1	INTRODUÇÃO	57
3.2	O QUE É UM REQUISITO	57
3.2.1	CLASSIFICAÇÃO DOS REQUISITOS	58
3.3	CONTEXTO DE DEFINIÇÃO DOS REQUISITOS.....	61
3.4	A ENGENHARIA DE REQUISITOS.....	64
3.4.1	FASES DA ENGENHARIA DE REQUISITOS	66
3.4.2	ANÁLISE E NEGOCIAÇÃO DE REQUISITOS	80
3.4.3	VALIDAÇÃO DE REQUISITOS	81
3.4.4	GERENCIAMENTO DE REQUISITOS.....	82
3.5	CONCLUSÃO.....	90
3.6	RESUMO	91

3 Engenharia de Requisitos

Até aqui conseguimos verificar a importância dos modelos de qualidade de software e os processos que consideramos chave para o sucesso do desenvolvimento de sistemas. Também pudemos ter uma visão do tratamento de requisitos sob o foco destes modelos.

Nesta etapa da dissertação são detalhadas as várias abordagens de tratamento dos requisitos em termos de processos e técnicas.

3.1 Introdução

Crosby é muito claro em definir qualidade como “a conformidade com os requisitos”. Esta é uma excelente definição, na medida que os requisitos estejam corretos. A abordagem de Crosby tinha em vista seu conhecimento sobre manufatura. Não sei quão claros e corretos são os requisitos de manufatura. Porém, pode-se afirmar categoricamente que os requisitos de software raramente estão próximos da correção[29].

Os requisitos não são um fim em si mesmos, mas um meio para se atingir um fim – de fornecer valor para alguém. Se os requisitos são identificados corretamente das pessoas importantes e capturam seus valores verdadeiros, essa definição reduz precisamente à conformidade aos requisitos de Crosby. Em desenvolvimento de software, não se pode assumir essa situação ideal; assim, muito do processo está na abordagem da descoberta dos requisitos verdadeiros[30].

3.2 O que é um Requisito

Segundo Abbott, é uma função, restrição, ou outra propriedade que precisa ser fornecida, encontrada, ou atendida para satisfazer às necessidades do usuário do futuro sistema[10].

Requisito é (a) uma condição ou capacitação necessária a um usuário para solucionar um problema ou encontrar um objetivo. (b) uma condição ou capacitação que um sistema ou

componente do sistema precisa atender ou ter para satisfazer um contrato, padrão, especificação, ou outro documento formalmente estabelecido. O conjunto de todos os requisitos formam a base para o posterior desenvolvimento do sistema[11].

3.2.1 Classificação dos Requisitos

Existem diversas classificações propostas na literatura para requisitos. A mais difundida é a proposta pela ISO/IEC 9126 [38], responsável pela padronização internacional, que detalha a classificação de requisitos de qualidade de produto de software da ISO.

Segundo o Modelo de Qualidade definido na ISO/IEC 9126, especificamente nas partes 2 e 3, respectivamente tratamento de características internas e externas de produto de software, classifica os atributos de qualidade de software em seis características, que são, por sua vez, desdobradas em subcaracterísticas. Deste modelo, propositadamente, o autor separou a característica funcionalidade das demais características, apresentando-as da seguinte maneira:

Funcionais: estes requisitos estão ligados diretamente a funcionalidade do software e estão divididos em categorias.

Não Funcionais: refletem os requisitos que expressam as qualidades específicas que o software deve ter. Os requisitos não funcionais também estão divididos em categorias.

Requisitos Funcionais

√ **Adequação:** O conjunto de funções que compõem o software e devem atender as demandas estabelecidas pelo cliente.

Ex: O usuário deve ter uma função para consultar os débitos de licenciamento do seu veículo.

√ **Acurácia:** atributos que definem a precisão do resultado da execução das funções estão de acordo com o que foi especificado.

Ex: Os valores que constam no extrato de débitos do licenciamento que é impresso nas máquinas de Auto-atendimento do Banestado e Itaú devem ser idênticos aos valores informados na página da Internet do extrato de débitos do Detran/PR.

√ Interoperabilidade: atributos que identificam as facilidades do software em interagir com outros através do estabelecimento de padrões facilitadores.

Ex: O Banestado deverá enviar, diariamente, um arquivo com os pagamentos de licenciamento efetuados no dia.

√ Conformidade: atributos que identificam a aderência do software a padrões estabelecidos de desenvolvimento e às normas e leis que regem o negócio.

Ex: A página Internet para consulta dos débitos de licenciamento de veículo deverão seguir ao padrão de interface do cliente.

√ Segurança de Acesso: atributos que identificam a aderência do software em relação a procedimentos de segurança definidos pelo cliente.

Ex: O usuário somente poderá visualizar as funções do sistema que tem direitos de acesso.

Requisitos Não-Funcionais

√ Usabilidade: atributos que identificam a facilidade de se compreender o funcionamento e operação do software. Tem como subcaracterísticas: inteligibilidade, apreensibilidade, operacionalidade, atratividade e conformidade.

Ex: O software deverá possuir Help On-line com a descrição de todos os campos das telas.

√ Confiabilidade: atributos que identificam a capacidade do software em manter a sua integridade após a ocorrência de falhas não controladas. Tem como subcaracterísticas: maturidade, tolerância a falhas, recuperabilidade e conformidade.

Ex: Todas as ocorrências de falhas na consulta de licenciamento da Internet deverão ter as informações da transação armazenadas em log para tratamento estatístico.

√ Eficiência: atributos que identificam a capacidade do software em desempenhar as suas atividades de forma adequada em relação ao tempo e aos recursos alocados. Tem como subcaracterísticas: comportamento em relação ao tempo, comportamento em relação aos recursos e à conformidade.

Ex: O relatório diário de pagamentos de licenciamento deve estar disponível às 9:00h da manhã de cada dia.

√ Portabilidade: atributos que identificam a capacidade de adaptação do software quando transferido para outros ambientes e/ou plataformas. Tem como subcaracterísticas: adaptabilidade, capacidade para ser instalado, coexistência, capacidade para substituir e conformidade.

Ex: O software deverá ser construído para ser utilizado em ambiente Windows 98 ou versão superior.

√ Manutenibilidade: atributos que identificam a capacidade do software em sofrer modificações. Tem como subcaracterísticas: analisabilidade, modificabilidade, estabilidade, testabilidade e conformidade.

Ex: A função de cálculo do IPVA deve ser parametrizada para facilitar a sua manutenção.

Segundo Macedo, os requisitos não funcionais podem ser classificados em [39]:

- de Qualidade Geral
- de Abrangência
- de Operação

ou classificá-los como:

- primários
- específicos

Sommerville propõe ainda a seguinte classificação dos requisitos [37]:

- de Processo
- de Produto
- Externos

Segundo Freeman, os requisitos podem ser divididos em [40]:

- Qualidade Básica: funcionalidade, confiabilidade, facilidade de uso, economia e segurança de uso.
- Qualidade Extra: flexibilidade, facilidade de reparo, adaptabilidade, facilidade de entendimento, boa documentação e facilidade de adicionar melhorias.

3.3 Contexto de Definição dos Requisitos

Um dos modelos mais abrangente do contexto de definição de requisitos é o apresentado em [41], conforme a figura 3.1 apresentada a seguir:

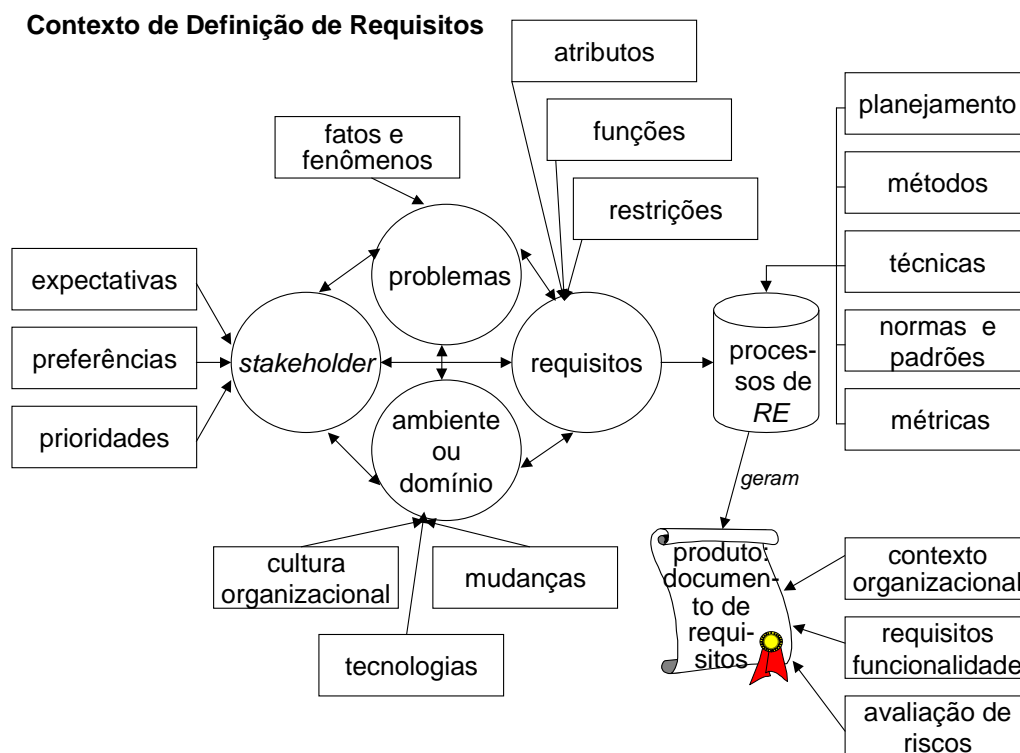


Figura 3.1 – Contexto de Definição dos Requisitos

Descrição dos elementos da figura 3.1:

Elementos fundamentais da abordagem de requisitos

√ Ambiente ou domínio da aplicação

O ambiente ou domínio da aplicação é onde ocorrem os fenômenos que caracterizam os problemas referentes aos requisitos do cliente[46]. É o primeiro elemento a ser conhecido e representado pelo engenheiro de requisitos. Incluem os aspectos sociais, econômicos e políticos em que se insere a organização.

√ Problemas

Um problema é a diferença de algo como desejado em relação a algo como percebido[34].

√ Requisitos

Requisito é uma declaração descritiva de exigências, do ponto de vista de alguém, sobre a qual será provida tecnologia da informação para a solução de problemas.

√ Stakeholder

Stakeholder compreende o conjunto de pessoas que direta ou indiretamente são afetadas pelo sistema a ser construído para a solução de problemas [47]. Também chamado por “envolvidos no projeto”, e consiste do: cliente, usuários, desenvolvedores, líder do projeto, responsável do cliente, etc.

Características associadas ao Ambiente ou Domínio da Aplicação

√ Cultura Organizacional

Referem-se às regras e normas que regulamentam a organização, comportamentos, hábitos e costumes.

√ Mudanças

Refere-se à dinâmica social e organizacional do elemento humano como agente de mudança do ambiente.

√ Tecnologias

Referem-se aos avanços tecnológicos e aos impactos sobre o ambiente organizacional.

Características associadas aos Problemas

√ Fatos ou Fenômenos

Um fato é uma verdade simples acerca do mundo [46]. Um fenômeno refere-se à forma de ver o mundo, depende de interpretação do contexto e do impacto que causa, sob o ponto de vista de quem o interpreta. O conhecimento de ambos e a identificação de quem os relata é que permite o entendimento do problema.

Características associadas aos Requisitos

√ Funções

As funções são ações nas quais o requisito é declarado. Descrevem o que se tem de fazer para atender a finalidade proposta.

√ Atributos

Os atributos são dimensões das características de funcionalidade e de não funcionalidade dos requisitos.

√ Restrições

As restrições são limitações que delinham o espaço de solução do problema. Tornam-se critérios de aprovação ou recusa para um produto [30].

Características associadas aos Stakeholders

√ Preferências

As preferências são condições desejáveis e particulares do cliente, porém opcionais. São condicionadas à definição prévia dos atributos e das restrições dos requisitos, ou seja, são circunscritas no espaço de solução do problema [30].

√ Expectativas

As expectativas são declarações do cliente quanto à forma de ver atendida uma demanda. São originadas do conhecimento do problema e do ambiente, cuja satisfação refere-se à solução [30].

√ Prioridade

A definição do que é prioritário pelos envolvidos é uma condição essencial no processo de desenvolvimento de software. O processo é essencialmente limitado pela disponibilidade de recursos (humanos, financeiros, tecnológicos,...) e pelo fator custo de produção.

Obter o nível de exigência de solução para produtos e serviços mais prioritários, além de acelerar a entrega do resultado, fazendo antes o que é realmente essencial, contribui para a negociação da forma de trabalho e para a medição de resultado da satisfação do cliente de forma gradativa.

Aplicação dos Processos e utilização de Técnicas

A aplicação dos processos e a utilização de técnicas de Engenharia de Requisitos deve iniciar antes da definição do software a ser construído e basear-se no conhecimento inicial do problema, fase identificada como “descobrimto de requisitos”.

Produto da Abordagem de Descobrimto de Requisitos

O resultado da aplicação dos processos da engenharia de requisitos caracteriza-se pelo Documento de Requisitos.

3.4 A Engenharia de Requisitos

O principal objetivo do processo de definição de requisitos é obter um acordo entre quem solicita e quem desenvolve, estabelecendo clara e rigorosamente o que deverá ser produzido. Segundo Brackett, o processo de definição de requisitos compreende as seguintes atividades[12]:

- Identificação dos requisitos: são descobertos os requisitos através da interação com os stakeholders, ou derivando-os a partir dos requisitos de sistema mais abrangentes;
- Identificação das restrições de desenvolvimento de software: são identificadas as restrições no processo de desenvolvimento de software. Por exemplo, custo e características de hardware a ser utilizado;

- Análise de requisitos: são avaliados os problemas potenciais dos requisitos, (por exemplo, viabilidade, ambigüidade e inconsistência) classificados (por exemplo, por prioridade) e avaliados quanto a viabilidade e riscos;
- Representação de requisitos: os resultados da identificação dos requisitos são retratados, entre outros, através de texto, diagramas, modelos ou regras de prototipação;
- Comunicação de requisitos: os resultados da definição de requisitos são apresentados aos stakeholders para revisão e aprovação;
- Preparação para validação de requisitos de software: são estabelecidos os critérios de aceitação e técnicas para garantir que o software, quando produzido, satisfaça os requisitos dos stakeholders;
- Gerenciar o processo de definição dos requisitos: durante todo o ciclo de vida do projeto desde a alocação até a implantação, as mudanças de requisitos precisam ser avaliadas quanto ao custo e ao impacto no cronograma (atividades realizadas ou não e produtos gerados).

3.4.1 Fases da Engenharia de Requisitos

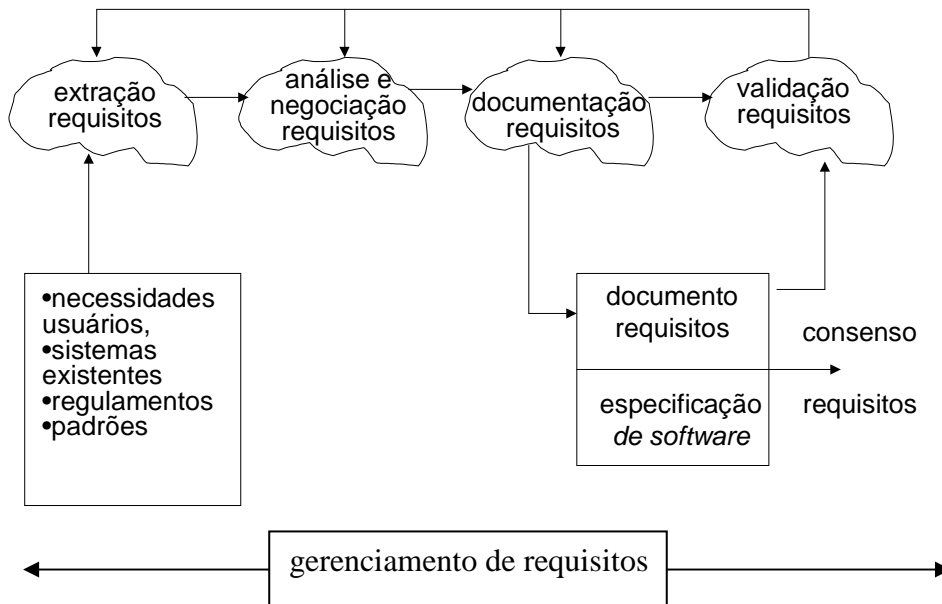


Figura 3.2 – Fases da Engenharia de Requisitos

a) Extração de Requisitos

É a etapa do processo de engenharia de requisitos responsável por descobrir os requisitos do sistema, sejam eles funcionais ou de qualidade (não funcionais), o que favorece um número maior de soluções, pelo fato de se conhecer mais sobre o problema a ser resolvido. Durante muito tempo, requisitos de qualidade apareciam em fases tardias do ciclo de vida de desenvolvimento o que, muitas vezes, implicava um grande esforço para implementação, obrigando a mudanças no projeto.

É um processo cuidadoso de interação com pessoas aliado a necessidade de avaliação da organização, do domínio da aplicação e dos processos de negócio em que o sistema operará. Pode parecer apenas um processo de transferência de conhecimento entre pessoas, e seria, se o cliente soubesse exatamente o que está precisando, fato que raramente encontramos no mundo real.

As quatro dimensões para o descobrimento dos requisitos, segundo Kotonya [36]:

- Domínio da Aplicação

Conhecimento do domínio da aplicação consiste em conhecer o negócio sobre o qual o sistema será aplicado. Isto significa que para entender os requisitos de um sistema de locação de veículos, você tem que conhecer o negócio de locar veículos.

- Problema

Conhecimento dos detalhes específicos do problema do cliente é fundamental para a definição correta dos requisitos verdadeiros do sistema. Para isso, é necessário o conhecimento da forma de operação do negócio de locação de veículos do cliente.

- Negócio

Conhecer como o sistema afetará as diferentes partes do negócio do cliente e as contribuições que o mesmo fará é fundamental no descobrimento dos requisitos verdadeiros.

- A necessidade e restrições do cliente e seus usuários

Entender a necessidade e as restrições das pessoas afetadas pelo sistema implica entender os processos do negócio que serão suportados pelo sistema e o papel dos sistemas existentes nestes processos de negócio.

A qualidade do produto a ser construído dependerá diretamente da qualidade dos requisitos identificados. Isto quer dizer que construir coisas de forma correta, não significa que elas sejam as coisas certas. A implicação desta afirmação torna chave um processo eficiente para identificação do problema e suas possíveis soluções, pelo fato de, se você descobrir os requisitos errados, ainda que o seu produto atenda tais requisitos, ele não será o produto que o cliente está esperando receber.

Uma das grandes dificuldades na exploração inicial dos requisitos é a habilidade de saber o que perguntar e a identificação das pessoas chave, a quem serão feitas as perguntas, a fim de se obter a definição do problema.

As Ferramentas Disponíveis

A grande dificuldade para descobrir o que realmente precisa ser feito, conforme já demonstrado na pesquisa do Standish Group e nos apontamentos de Weinberg, nos remete ao grande problema que é descobrir o problema e sua solução. Existem algumas ferramentas

fundamentais que o Analista deve dominar, se quer ter êxito no processo de descobrimento dos requisitos que são:

- Identificar as pessoas certas;
- Fazer as perguntas certas;
- Realizar as reuniões certas.

b) Identificando as pessoas certas

Quando não se conhece o ambiente e as pessoas para as quais irá desenvolver um sistema, duas figuras são imprescindíveis de se identificar. A primeira, é se a pessoa que realmente decide está presente. Isto é fundamental tendo em vista que muitas vezes, existe um cliente, do tipo sabe tudo, que não está presente, pelo seu pouco tempo disponível, muitas vezes puro reflexo de seu posicionamento centralizador, motivado pelo fato de achar que é mais inteligente ou sabe mais que os outros. Este comportamento pode acabar refletindo negativamente no projeto da seguinte maneira. Este ser superior envia um representante para tomar as decisões em seu lugar. Após muitas horas da equipe de trabalho, em que diversas decisões foram tomadas, pode acontecer de se voltar praticamente ao início, devido a discordância de pontos de vista em relação a esta pessoa. Por esta razão, é um ponto chave para o sucesso do projeto que a pessoa que decide esteja participando, afinal, é muito provável que ela seja o cliente responsável pelo pagamento do trabalho de definição dos requisitos.

A outra figura que deve ser identificada é o “mal intencionado”, que sempre vai estar pensando em si e não na equipe de trabalho. Um grande problema que este ser pode adicionar ao projeto é a discórdia, visto que o mesmo estará interessado em aparecer, se promover e pouco fará para o bom andamento das coisas. Identificar este tipo de pessoa, que é semelhante a um vírus, é fundamental para que seja possível se resguardar e a sua equipe de trabalho. Um cuidado adicional que se deve ter com este tipo de pessoa é que elas geralmente são inteligentes, só que não utilizam esta capacidade para o bem comum. Geralmente estas pessoas ficam procurando por erros no projeto, mas não com o intuito de corrigi-los, e sim, de apontá-los tardiamente no processo, de

forma a desbancar o líder do projeto. Inocular o seu poder de ação é o melhor a fazer e para isso será necessário bastante habilidade no trato com estas pessoas.

Uma outra observação importante é entender a diferença entre cliente e usuário. Ambos precisam estar no grupo de trabalho. O cliente porque é quem paga pelo valor do produto e o usuário porque é o responsável pelo sucesso ou fracasso do produto. De acordo com Weinberg [30] 75% das ferramentas de software adquiridas nunca são usadas e o principal motivo é que os requisitos para a elaboração destas ferramentas são esboçados por um grupo de pessoas que exclui os usuários alvo. As ferramentas não somente não atendem aos requisitos como os usuários em potencial não se sentem envolvidos no processo, motivo pelo qual não se comprometem a dar bom uso às ferramentas.

Um princípio que se deve ter em mente, quando se atua em um projeto que envolve o desenvolvimento de software é o de Veblen que diz o seguinte: “Não existe mudança, não importa quão ruim, que não beneficie algumas pessoas, nem mudança, não importa quão boa, que não prejudique alguém”. As observações de Veblen nos estimulam a pensar a respeito dos usuários não somente como beneficiários dos produtos, mas como pessoas que serão afetadas. Se não pensarmos a respeito disso antecipadamente, é provável que elas surjam posteriormente e se oponham ao produto, algumas vezes com grande sucesso.

c) As Perguntas Certas

A maioria das pessoas passa grande parte de suas vidas fazendo perguntas ou a elas respondendo. No desenvolvimento de sistemas, fazer perguntas faz parte do trabalho a ser feito. No entanto, são poucas as pessoas que realmente são conscientes dos diferentes tipos de perguntas que podem ser feitas e quando cada tipo deve ser usado. Também são poucas as que estão conscientes das respostas que resultarão de suas perguntas. O sucesso do profissional de sistemas depende em grande parte de fazer as perguntas certas e de uma avaliação cuidadosa das respostas de modo a se tomar as decisões corretas. Fazemos perguntas por múltiplas razões:

- Obter mais informações;
- Dar início a uma discussão ou a um debate;

- Esclarecer pontos de vista;
- Buscar cooperação;
- Envolver e motivar as pessoas;
- Ajudar as pessoas a refletir e a aprender.

As perguntas podem ser divididas em duas categorias: abertas e fechadas. As abertas estimulam o entrevistado a formular a resposta com suas próprias palavras. As fechadas, em geral, exigem respostas monossilábicas e o trabalho maior é da pessoa que pergunta. Fazer boas perguntas é fundamental a todos que trabalham com pessoas, pois é desta forma que se vai descobrir as coisas.

Há uma técnica para fazer as perguntas certas, ou seja, aquelas capazes de extrair o que estamos querendo delas [31]. A pergunta certa ajuda o entrevistado a descobrir que ele tem as respostas dentro de si mesmo, ainda que antes da pergunta ele não soubesse disso.

i. Perguntas Abertas

São aquelas que estimulam a outra pessoa a falar e a expressar suas idéias. Elas forçam o entrevistado a pensar, e neste processo, ele freqüentemente encontra idéias, pensamentos ou conhecimentos dentro de si, que nunca imaginou que existissem. Mas existe algo tão importante quanto fazer as perguntas certas. Você precisa estar preparado e atento para interpretar as respostas, para pelo menos saber prosseguir, provavelmente com mais perguntas. Uma análise verdadeiramente eficiente conduzirá a diferentes níveis de avaliação e terá como base as respostas às seguintes perguntas:

- O que o entrevistado, na verdade está dizendo ?
- O que ele parece estar tentando comunicar ?
- O que se pode inferir por meio da maneira como se comunica e das palavras que usa ?
- O que ele transmite por intermédio de seu estilo de abordar o assunto ?

A observação apontada no livro de Pressman [1] ilustra bem a dificuldade da verdadeira análise que deve ser feita das respostas obtidas: “Sei que você acredita que entendeu o que acha que eu disse, mas não estou certo de que percebe que aquilo que ouviu não é o que eu pretendia dizer”.

A maior desvantagem de se usar perguntas abertas é que, ao estimular o entrevistado a falar, o mesmo pode se desviar do assunto, bem como falar além do necessário. Nesta hora, a discussão deve ser reconduzida de forma hábil pelo entrevistador. Tipos de perguntas Abertas:

ii. Perguntas Investigadoras

As perguntas investigadoras são próprias para buscar informações a um nível maior de profundidade. O principal objetivo desse tipo de pergunta é conseguir algo além de respostas superficiais e investigar mais detalhadamente o assunto. Cuidado para não transformar a entrevista num interrogatório. Estas perguntas exigem um ouvinte metuculoso para serem realmente eficientes. Estas perguntas ainda se subdividem em :

iii. Perguntas Simples

Aquelas que são usadas pelas crianças, do tipo “Por quê ?” ou “Por que não ?”. Tais perguntas são utilizadas para investigar mais profundamente as razões de uma resposta anterior específica. Para não ter efeito de um interrogatório, deve-se suavizar o efeito de um seco “Por quê?” fazendo perguntas mais proveitosas adicionando uma expressão facial e tom de voz amenos, como:

- Por que você disse isso ?
- O quê leva você a dizer isso ?

iv. Perguntas Comparativas

Permitem ao entrevistador não apenas avaliar sobre uma base “antes e depois”, mas também fazer o entrevistado pensar. Este tipo de pergunta ajuda no desenvolvimento e ordenação de idéias através da comparação entre momentos distintos, como:

- O que esta experiência alterou a sua maneira de pensar ?

- Até que ponto o novo sistema de controle de pagamentos melhorou a administração do departamento ?
- Como você analisa a rotina do seu departamento com a implantação do novo sistema?

v. Perguntas Hipotéticas

Este tipo de pergunta dão ao entrevistado a oportunidade de testar seus conhecimentos, sua experiência e sua habilidade em relação ao assunto em questão. Ela dá a oportunidade de se lidar com suposições e a adotar uma abordagem analítica da solução do problema. Basicamente, tais perguntas devem ser resumidas em duas palavras: “E se...?”.

- E se tivéssemos recursos ilimitados que tecnologias poderiam ser utilizadas ?
- E se você estivesse no meu lugar, o que faria ?
- E se o pacote de software X fosse adquirido, o que melhoraria o seu trabalho ?

vi. Perguntas Extensivas e Precisas

Tal pergunta deve ser utilizada para estimular uma resposta mais completa quando informações precisam ser acrescentadas sobre o assunto em pauta. Usadas em entrevistas, estas perguntas podem produzir uma resposta mais abrangente, principalmente se o entrevistado for acessível.

- O que o faz dizer isso ?
- O que você quer dizer quando diz...?
- Você poderia ser um pouco mais específico ?
- Pode explicar isso de forma um pouco mais detalhada ?

vii. Perguntas Fundamentadas

Este tipo de pergunta é utilizada para explorar as opiniões ou atitudes do entrevistado de forma que o mesmo aprofunde o assunto, esboçando seus pensamentos, na profundidade que o mesmo achar adequada. Caso o entrevistado não desça na profundidade necessária, o entrevistador pode dar seqüência a uma pergunta fundamentada com uma pergunta investigadora. A avaliação das perguntas fundamentadas deve ser meticulosa, inclusive com a avaliação do nível de emoção envolvido e o próprio tom de voz da resposta que podem sinalizar novos questionamentos.

- Como você se sente em relação ao novo sistema ?
- Qual o seu ponto de vista sobre a atual situação do seu departamento ?
- O que você pensa a respeito do desenvolvimento deste sistema ?

viii. Perguntas Reflexivas

Deve ser utilizada para explorar em detalhes as atitudes e opiniões do entrevistado. Estas perguntas são capazes de produzir informações muito valiosas a cerca do assunto tratado. É importante que o entrevistador mantenha uma postura neutra. Isto quer dizer que a imparcialidade é o ponto chave para se usar tais perguntas.

- Todos dizem que vai dar certo. Eles sabem sobre o que estão falando, não sabem ?
- Você está satisfeito com o sistema ou não ?

ix. Perguntas Conclusivas

O importante deste tipo de pergunta é a avaliação da resposta, com foco no conteúdo e não nas emoções do entrevistado. Devem ser utilizadas para fazer uma breve revisão, checar o entendimento que o entrevistador teve dos fatos e até mesmo para esclarecer os pensamentos do próprio entrevistado. Também conhecidas como perguntas de confirmação.

- Se estou entendendo, você quer dizer que ...?
- Então, você está dizendo que o principal problema foi a falta de recursos ?

x. Perguntas Fechadas

O objetivo principal deste tipo de pergunta é obter itens específicos de informação. É bastante similar às perguntas conclusivas. São perguntas simples de serem respondidas, geralmente por “Sim ou Não”.

- Você já consultou o Manual de Procedimentos do departamento ?

Geralmente, uma pergunta fechada é seguida por uma pergunta exploratória simples.

- Você imagina descobrir os requisitos do sistema sem a realização de reuniões ?
- Sim.
- Como você fará ?

Perguntas que começam com “Quem, Quando, Onde, Quantos” são perguntas fechadas, embora não possam ter como resposta um simples “sim ou não”.

- Quantas pessoas fazem parte da equipe de desenvolvimento ?
- Quando você pode iniciar a definição dos requisitos do sistema ?
- Quem será o responsável pelo projeto no cliente ?

Perguntar é a alma do negócio de desenvolvimento de sistemas. Temos que ser exímios utilizadores desta ferramenta se quisermos ter êxito nesta profissão.

d) As Reuniões Certas

Em qualquer projeto de desenvolvimento de sistemas, haverá um grupo de pessoas que serão responsáveis por identificar os problemas e propor soluções para resolvê-los, utilizando seus múltiplos pontos de vista. Toda esta interação entre as pessoas se dará, na maioria das vezes, através de reuniões. Se pensarmos que quanto maior o projeto, provavelmente maior será o número de

peças envolvidas, e se estas peças fazem parte do grupo, é porque muito provavelmente elas têm conhecimento sobre o assunto. Então, existe um custo, que não é baixo, gasto em reuniões de trabalho. Se olharmos do ponto de vista econômico, do custo envolvido em reuniões, é evidente que elas têm de ser tão produtivas quanto possível. Se olharmos ao nosso redor, notamos que muitas vezes, para não dizer na maioria das vezes, não é isso que testemunhamos. E o segredo para o melhor aproveitamento desta ferramenta chamada reunião, qual será ?

i. Fazendo a coisa certa

Tudo começa antes da reunião. A primeira coisa a fazer é decidir a respeito de sua finalidade. O próximo passo é identificar as pessoas certas para participar da reunião. Após estas ações, publique uma agenda da reunião, que deve ter horário de início e término, além dos assuntos que serão abordados e o tempo previsto para discussão de cada item da pauta. Após publicada a agenda, procure não fugir da pauta, caso contrário, poderá desmontar todo o planejamento feito, que poderá seguir por caminhos que fujam do objetivo ou a discussão de temas que necessitem de pessoas que não estão presentes, pois para a pauta prevista as mesmas não eram necessárias. Se por acaso surgirem assuntos não previstos, anote para discussão oportuna. Nunca planeje uma reunião para tratar assuntos diversos, em que parte dos participantes tenham interesse em alguns pontos e partes em outros pontos. Neste caso, planeje duas reuniões menores [32].

Durante a reunião, se faz necessário a figura do secretário, responsável pela elaboração da ata e do líder, responsável pela condução da reunião. Como líder, você deve montar a estratégia para o êxito, que em parte se dará pela seleção das pessoas certas, do planejamento da reunião e da elaboração das perguntas certas. Para o sucesso absoluto, atenção extrema não somente para o que as pessoas falam, mas principalmente para como elas falam. Isto é fundamental para se fazer as observações certas.

Ao final da reunião, procure revisar o que foi anotado pelo secretário com o grupo. Após a reunião, elabore a ata, que deverá conter o horário de realização, os participantes, a finalidade, os tópicos tratados, dúvidas que possam ter ficado, encaminhamentos para lembrar quem

ficou de fazer o quê, e de preferência, já agende a próxima reunião. Encaminhe a ata a todos os participantes e àqueles que você julga necessário o conhecimento do que foi discutido.

ii. O primeiro encontro de verdade - a 1ª reunião

No levantamento de dados de um cliente, algumas questões livres de contexto, ou seja, que podem ser utilizadas em qualquer contexto (qualquer projeto), devem ser feitas no início de um projeto para se obter informações a respeito das propriedades do problema que se tem e suas possíveis soluções. Tais questões são apropriadas para qualquer produto a ser projetado. Elas são um conjunto à parte do leque de ferramentas do analista, que podem ser usadas antes de se envolver em detalhes enganosos[30].

Algumas destas questões relativas ao processo, são:

- Quem é o cliente do projeto ?
- O que uma solução bem sucedida significa para este cliente ?
- Qual é a razão real para desejar solucionar o problema ?
- Quem deve estar na equipe do projeto ?
- Quanto tempo teremos para este projeto ?
- Qual a relação entre tempo e custo para o projeto ?
- Onde mais pode ser obtida a solução para este problema ? Podemos copiar algo que já exista ?

Algumas destas questões relativas ao produto, são:

- Que problemas este sistema resolve ?
- Que problemas este sistema pode criar ?
- Que ambiente esse sistema provavelmente encontrará ?
- Qual o grau de precisão necessário ou desejado no produto ?

Algumas destas questões relativas as questões, são:

- Eu estou lhe fazendo muitas perguntas ?
- As minhas perguntas parecem relevantes ?
- As suas respostas são oficiais ?
- Existe mais alguém que possa dar informações úteis ?
- Existe algum material adicional que possa conter informações úteis ?
- Existe algum lugar onde eu possa ver o ambiente no qual este produto será usado ?
- Existe algo mais que eu possa perguntar ?
- Existe algo mais que você queira me perguntar ?

Uma grande vantagem de se utilizar questões livres de contexto é que as mesmas podem ser preparadas com antecedência além delas ajudarem a vencer o embaraço inicial relacionado com um novo projeto e novas relações.

iii. Características do Líder

Numa certa altura da peça “Muito barulho por nada”, de Shakespeare, aparece a frase: “Quando dois homens montam no mesmo cavalo, um deles tem que ir na garupa” [33]. É assim que acontece com as reuniões, ou seja, alguém tem de estar na direção, e cumprir as seguintes tarefas, nem sempre fáceis:

- Impor respeito àqueles que participam;
- Vir preparado;
- Cumprir o horário;
- Iniciar na hora marcada;
- Conduzir a reunião dentro da agenda;
- Definir as regras da reunião logo no início;

- Definir o secretário da reunião para que o mesmo proceda as anotações para elaboração da ata;
- Fazer a apresentação das pessoas, se necessário;
- Controlar as discussões tendo em vista as diferentes personalidades presentes;
- Encorajar contribuições quando necessário;
- Assegurar que todos tenham se pronunciado;
- Manter a reunião dentro do assunto;
- Sumarizar claramente a discussão em intervalos regulares;
- Levar a discussão a uma conclusão;
- Repassar com o grupo as anotações do secretário no final da reunião;
- Encaminhar a ata da reunião para todos os presentes e interessados;
- Providenciar as ações de acompanhamento, para que as pendências sejam resolvidas.

iv. Definindo as Regras da Reunião

A definição das regras de uma reunião, logo no seu início, é fundamental para evitar problemas de diversas ordens. Isso garantirá a segurança necessária para que as pessoas participem plenamente, afinal, é muito mais confortável se trabalhar sobre regras conhecidas e claras. A seguir, a lista das principais regras a serem acordadas entre os participantes:

- Estabelecer uma política de interrupção, para que as pessoas não interrompam nem sejam interrompidas durante suas explicações. Isto ajudará o líder a silenciar os que interrompem, de forma polida. Esta política de interrupção estará atrelada a regra de limite de tempo de fala;
- Consiga também um compromisso em relação ao tempo de fala de cada um, que no caso é uma pré-condição do compromisso em relação às interrupções. Frequentemente, os que interrompem ficam preocupados que alguém nunca pare de falar. Acordando, digamos um limite de dois a quatro minutos para cada pessoa, o ambiente se tornará seguro evitando as interrupções indesejáveis;

- Comprometa as pessoas em relação a proibição de ataques pessoais e rispidez. É um compromisso fácil de se conseguir no início de uma reunião, porém, impossível após a mesma ter virado uma batalha campal;
- Obtenha a concordância do grupo para que assuntos relacionados, porém, não pertencentes a pauta de discussão, que por ventura apareçam durante a reunião, sejam anotados para discussão posterior.

v. Colocando as Ferramentas em prática

É fato que a utilização das Pessoas, Perguntas e Reuniões ocorre de forma integrada e o sucesso na utilização é quase uma arte, porém, só se chega a esse estado se você tem um processo para seguir na definição das pessoas certas, na elaboração das perguntas certas e condução certa de reuniões certas. Defina o seu processo, baseando-se nas informações anteriores, e a chance de êxito será significativamente maior.

Na verdade, todo trabalho de requisitos é precedido por algum tipo de processo de iniciação. Não importa a origem, a idéia é o ponto inicial para o processo de requisitos. Segundo Weinberg, devido ao fato de as pessoas terem suas formas individuais de pensar, pode haver muitas interpretações diferentes ou formas de idéias, e daí, muitos pontos iniciais. Isso pode ser desastroso para o processo de definição de requisitos. Uma solução possível para se ter uma plataforma sólida para explorar requisitos é examinar cada projeto como uma tentativa de solucionar algum problema e então, reduzir cada ponto inicial a uma forma comum de descrição do problema, o qual pode ser definido como sendo a diferença entre as coisas conforme são percebidas e as coisas conforme desejadas.

“Um documento padrão para descrição dos requisitos é fundamental no processo de descobrimento do problema.”

3.4.2 Análise e Negociação de Requisitos

Esta fase tem como objetivo verificar o documento de requisitos produzido, procurando identificar se todos os problemas são resolvidos com os requisitos descritos, se os mesmos não entram em conflito ou se sobrepõem. Além da resolução de conflitos e sobreposição, o resultado desta etapa é um documento de requisitos acordado entre os participantes.

A utilização de um checklist facilita esta fase, tendo em vista ser este processo trabalhoso e que envolve pessoas experientes da equipe de trabalho, o que passa a implicar em alto custo. Ele funciona como um lembrete para o processo, de forma que se reduz as chances de esquecer um determinado item a ser verificado.

Segundo Sommerville [37], alguns itens que devem ser verificados nesta fase:

- Existe alguma informação em relação a forma como o sistema será implementado ?
- A descrição dos requisitos descrevem um único requisito ou existem requisitos que podem ser desmembrados ?
- Os requisitos são realmente necessários ou somente uma perfumaria ?
- Os requisitos estão de acordo com as metas do negócio do cliente ?
- Os requisitos podem ser interpretados de maneira diferente por diferentes usuários (ambigüidade) ?
- Os requisitos são possíveis de ser implementados no sistema ?
- Os requisitos podem ser testados para verificar se são satisfeitos pelo sistema ?

Uma outra ferramenta bastante útil para suportar o processo de análise e validação é a utilização de um protótipo, devido aos seguintes aspectos:

- Pode ser considerado como uma versão inicial do sistema;
- É algo bastante concreto para os envolvidos no projeto, principalmente para o usuário final e o cliente, que poderão “ver” o produto desejado, antes de ser empenhado os altos custos de desenvolvimento;
- Foca basicamente a funcionalidade desejada. Esta característica torna a sua construção relativamente rápida.

Esta fase termina com a geração do documento dos Requisitos verdadeiros do Cliente, ou seja, aqueles que devem guiar o processo de implementação. Também conhecido como requisitos alocados.

3.4.3 Validação de Requisitos

Após a documentação dos requisitos ter sido produzida, inicia-se o processo de validação, buscando checar se os requisitos estão certos, ou seja, descritos de forma apropriada, procurando eliminar problemas de incompleta, ambigüidade ou inconsistência.

A preocupação maior desta fase é com a qualidade do documento de requisitos produzido. Para esta fase também é recomendado o uso de checklist que deve identificar, entre outras coisas:

- Os requisitos podem ser entendidos claramente ?
- Os requisitos não possuem informação repetida desnecessariamente ?
- Os requisitos atendem completamente as necessidades do cliente ?
- Existe alguma informação faltante que deveria estar descrita no documento ?
- Os requisitos podem ser interpretados de forma diferente por diferentes usuários ?
- Os requisitos não geram contradição entre si ?
- Os requisitos estão organizados de forma adequada ?
- Os requisitos estão em conformidade com os padrões estabelecidos ?
- Os requisitos podem ser rastreados, possuem ligações com outros requisitos que possuem relação e a razão de sua existência está documentada ?

O objetivo final desta fase é liberar o documento de requisitos que deve representar de forma clara e consistente o que deve ser implementado. Tal documento será o guia para as etapas seguintes do processo de desenvolvimento do sistema.

Porém, enquanto problemas forem detectados nesta fase, o documento de saída será a lista de problemas encontrados. Esta lista de problemas retornará para as etapas anteriores do ciclo de vida do processo de desenvolvimento de requisitos, ficando neste loop até que o documento atinja o padrão de qualidade adequado.

3.4.4 Gerenciamento de Requisitos

Responsável por controlar a evolução dos requisitos de um sistema, seja por constatação de novas necessidades, seja por constatação de deficiências nos requisitos registrados até o momento. [13]

Sempre que os requisitos alocados forem alterados, os planos de software, os artefatos, e as atividades afetadas devem sofrer ajustes para continuarem consistentes. O ponto chave é que os requisitos são ativos e estão em uso durante todo o ciclo de vida sendo a base para a modelagem do sistema.

Uma das grandes verdades do desenvolvimento de sistemas é: “os requisitos mudam”. Por este motivo, agilidade no processo de tratamento das mudanças é fundamental . Esta agilidade se torna possível a partir do momento em que podemos rastrear os requisitos.

O rastreamento de requisitos [42,43] refere-se à habilidade de descrever e seguir a vida de um requisito. Desde a fonte de origem até o desenvolvimento e especificação. É uma técnica que permite a visibilidade do relacionamento de dependência entre os requisitos, identificando relacionamentos hierárquicos do tipo “pai-filho” entre os requisitos. Além da dependência entre os requisitos, deve ser possível identificar a dependência entre os requisitos e os artefatos que eles afetam.

Segundo Pinheiro [44], a motivação para o rastreamento de requisitos é justificada pelas seguintes observações:

- os requisitos evoluem durante a vida de um software;

- os requisitos são contextualizados e dependem de detalhes da situação particular do contexto em que surgem;

- o rastreamento de requisitos também é contextualizado. Isto implica que um rastreamento poderá produzir objetos significativos para determinada situação;

- os requisitos são parte intrínseca do processo de desenvolvimento e o rastreamento de artefatos do projeto é útil por todo o ciclo de vida dos requisitos e do desenvolvimento.

Existem várias razões que motivam a mudança, mas sem dúvida, o principal fato é que durante o desenvolvimento os envolvidos vão entendendo melhor suas necessidades, refletindo em “mudanças”. Além disso, a economia mundial cada vez mais globalizada, caracterizada por um forte acoplamento, incentiva a ocorrência do “efeito dominó”, em que uma das peças, com frequência, é o sistema que está em desenvolvimento. Isso significa que um desaquecimento da economia dos Estados Unidos, ou problemas econômicos com a Argentina ou ainda atentados terroristas ou o mal da vaca louca no Japão e Inglaterra, tem reflexos diretos na economia do planeta e nas organizações de todo o mundo. Como os sistemas dão suporte em grande parte aos negócios das empresas, sofrem impacto direto de qualquer desestabilização mundial.

Por este motivo que o ciclo de vida dos requisitos assim como o de desenvolvimento de sistemas se adaptam melhor a uma abordagem em espiral. A todo momento estamos aprendendo e tendo que voltar em etapas anteriores, adaptar ou mudar e seguir novamente o fluxo. Esta é a nova ordem mundial do desenvolvimento e pode ter certeza: “sem um gerenciamento de requisitos eficiente, tal abordagem é tecnicamente impossível”.

Congelar os requisitos após a etapa de validação é utopia [27], já que os negócios não são estáveis. Como eles se adaptam às mudanças, os sistemas também têm de se adaptar. A capacidade de adaptação do processo de desenvolvimento é hoje um diferencial estratégico entre as empresas de software. Esta capacidade de adaptação é mérito em grande parte do processo de gerenciamento de requisitos.

a) Atividades de Gerenciamento de Requisitos

- √ Garantir a integridade do documento de requisitos (baseline);
- √ Rastrear o relacionamento entre os requisitos;
- √ Rastrear os requisitos em relação aos outros artefatos do projeto;
- √ Gerenciar mudanças;
- √ Aplicar os requisitos na solução técnica;
- √ Guiar o design e implementação;
- √ Suportar o particionamento do trabalho entre os grupos;
- √ Suportar o processo de teste e validação do produto;

- √ Suportar o desenvolvimento iterativo;
- √ Suportar o gerenciamento dos marcos de projeto;
- √ Suportar o gerenciamento das interfaces com outros sistemas.

i. Garantir a integridade do documento de requisitos

Muitos projetos falham, todavia, a solução é tecnicamente de boa qualidade. Se um produto não faz o que o cliente deseja, então, pouco importa a sua qualidade técnica. O papel chave dos requisitos é mostrar aos desenvolvedores e usuários qual é a necessidade do sistema utilizando uma linguagem comum aos envolvidos. O papel principal dos requisitos é a comunicação. Para que esta comunicação seja adequada do início ao fim do ciclo de vida do projeto, é fundamental um processo de gerenciamento de requisitos adequado que garanta a integridade do documento de requisitos que é o mapa da mina dos envolvidos no projeto.

ii. Rastrear o relacionamento entre os requisitos

O objetivo principal do processo de gerenciamento de requisitos é prover informação para a atividade de avaliação de impacto da mudança em um projeto. Uma das informações fundamentais para suportar esta atividade é a identificação do relacionamento entre os requisitos, conceito conhecido como rastreabilidade.

Quando uma requisição solicita a mudança de um requisito, é importante conhecer a posição deste requisito na hierarquia. A partir deste reconhecimento é possível avaliar se os requisitos superiores ou inferiores na hierarquia também deverão ser modificados. Somente a partir desta avaliação hierárquica se consegue identificar quais os requisitos que deverão sofrer modificação caso a mudança seja aprovada.

Este é um dos motivos que torna a avaliação de impacto uma atividade trabalhosa, muito maior do que na realidade aparenta. A necessidade de se registrar diversas informações reforça a importância de um sistema com banco de dados específico para este fim, pois tal sistema possibilita

armazenamento de grande quantidade de informação específica do desenvolvimento, com múltiplos relacionamentos e agilidade na disponibilização destas informações.

iii. Rastrear a dependência entre os requisitos e os artefatos do projeto

Prover informação de rastreabilidade entre os requisitos e os artefatos do projeto que sofrem a sua influência é de fundamental importância para dar suporte a atividade de avaliação de impacto da mudança. Esta informação é necessária para agilizar a estimativa de quantidade de esforço e custo para modificar os artefatos do projeto de forma a acomodá-los em relação a mudança requisitada.

Esta informação de rastreabilidade deve ser possível em ambas as direções: “dos requisitos para os artefatos e vice-versa”.

iv. Gerenciar mudanças

Mudanças ocorrem durante o ciclo de vida do sistema e normalmente alterações de requisitos afetam todas as fases do ciclo de vida e seus artefatos. A menos que conheçamos o relacionamento entre os requisitos e os demais componentes do desenvolvimento, não teremos base racional para decidir sobre a mudança. Um bom conjunto de requisitos e seus relacionamentos definem precisamente a base para a avaliação da mudança.

Quando novos requisitos emergem ou quando é solicitada uma mudança nos requisitos existentes ou quando uma mescla das duas situações anteriores acontece, caracterizamos esta ocorrência por “mudança”.

O que ocorre após a primeira passada pelo processo de validação dos requisitos é a geração da versão um do documento de requisitos (início da baseline). Após esta etapa, a atividade responsável por acomodar novos requisitos ou alteração nos existentes é chamada de gerenciamento da mudança. Esta atividade inicia com o recebimento da requisição de mudança e vai até a sua rejeição, caso em que a mudança não será implementada, ou sua aprovação para implementação,

refletindo na mudança de versão do documento de requisitos (nova versão da baseline), que passa a ser o novo acordo entre os envolvidos no projeto, ou seja, o novo mapa para o desenvolvimento do sistema.

O grande objetivo desta atividade é prover informação para a avaliação do impacto de esforço e custo da mudança no projeto e minimizar os chamados efeitos negativos da mudança no desenvolvimento do sistema. É uma atividade delicada. Qualquer falha na avaliação pode refletir em um produto que tecnicamente não atenda às necessidades do cliente ou um custo e esforço de implementação da mudança muito maiores do que os previstos.

O processo para execução desta atividade deve prever:

- como a requisição de mudança será feita;
- qual o procedimento para avaliar o impacto da mudança;
- quem é o responsável pelo parecer da avaliação;
- quem é o responsável por autorizar ou rejeitar a mudança;
- como a mudança deve ser disseminada para os outros processos do desenvolvimento de sistema.

O líder do projeto é geralmente o responsável por garantir a incorporação da mudança no projeto.

As entradas desta atividade são:

- histórico das requisições de mudança;
- relacionamento entre os requisitos;
- dependência entre os requisitos e os artefatos do sistema;
- a requisição de mudança;
- os artefatos já produzidos;
- plano de desenvolvimento.

O produto intermediário desta atividade é:

- relatório de impacto da mudança.

As saídas desta atividade são:

- Parecer da requisição de mudança (aprovação ou rejeição);
- Nova versão do documento de requisitos (em caso de aprovação).

A partir das saídas do processo, caso a mudança seja aprovada, o plano do projeto é revisado e os artefatos são modificados para atender a mudança. Para facilitar a atividade de avaliação de impacto das novas requisições de mudança todas as solicitações, inclusive as rejeitadas, devem ter as informações registradas, já que é comum os requisitos rejeitados tornarem a ser propostos. Há uma lógica que motiva este fato. Quando proposto pela primeira vez, havia uma razão para a proposição e outra para a rejeição. De posse das informações históricas, a avaliação tem apenas de verificar se a razão da proposição ainda é válida e se a razão da rejeição também. Se a razão da proposição não for mais válida ou se a razão da rejeição ainda continua, então, a rejeição da requisição persiste, caso contrário, a avaliação deve verificar outras razões para rejeitá-la ou alocá-la para ser implementada.

As informações relevantes da solicitação de mudança:

- Número da Requisição (identificador único para cada requisição);
- Uma breve descrição da mudança;
- A razão da mudança;
- Datas relevantes;
- Parecer da Avaliação;
- Responsáveis pela avaliação;
- Responsáveis pela aceitação ou rejeição;
- Situação da requisição.

A avaliação da mudança deve identificar os artefatos que precisarão ser alterados e a quantidade de esforço e custo de implementação. É importante também avaliar os riscos do projeto

em relação à mudança. Com estas informações o líder do projeto e o responsável do cliente podem dar seu parecer final, pois conseguirão visualizar a mudança como um todo, no contexto do projeto.

Um gerenciamento de mudança eficiente precisa de uma ferramenta adequada para dar o suporte de informação necessária. Um software com utilização de um banco de dados próprio para realizar esta atividade é imprescindível. Pode-se partir para utilização de software de prateleira, como o Excel, para a realização do trabalho, mas é como pregar um prego usando um coco. Não parece tão difícil, mas geralmente, alguém se machuca ou se estraga o coco ou se inutiliza o prego. Quando se procura gerenciar qualquer coisa, deve-se ter sempre em mente: “ou você está no controle do processo ou é controlado por ele”.

v. Aplicar os requisitos na solução técnica

Requisitos da solução representam o passo seguinte à definição do problema. Eles formam um modelo abstrato do que precisa ser produzido, expresso como funcionalidade e outras características não funcionais que permitem aos desenvolvedores enxergar como o seu trabalho se encaixa no sistema como um todo.

vi. Guiar o design e implementação

A partir dos requisitos da solução definidos, as fases seguintes do ciclo de vida, de design e implementação são executadas. É fundamental para a qualidade do produto final que, qualquer mudança nos requisitos seja imediatamente remetida para estas fases, que fatalmente, terão alterações a implementar.

vii. Suportar o particionamento do trabalho entre os grupos

Como o documento de requisitos é o acordo entre todos os envolvidos no projeto, as mudanças de requisitos, refletirão uma nova versão do documento de requisitos, que deve ser um novo acordo entre todos os grupos de trabalho envolvidos.

viii. Suportar o processo de teste e validação do produto

O atributo principal de um requisito é ser testável. Este motivo torna o documento de requisitos a fonte de inspiração para a elaboração do plano de testes e a conseqüente validação dos componentes do produto, e após a integração dos componentes, proceder a validação completa do produto.

ix. Suportar o desenvolvimento iterativo

O processo de desenvolvimento de software será cada dia mais baseado no conceito de evolução [45], ou seja, estaremos sempre modificando algum software já existente ou em construção. Este fato implica que, solicitações de mudança serão cada dia mais freqüentes, tornando o gerenciamento de requisitos um processo fundamental no controle do desenvolvimento iterativo.

x. Suportar o gerenciamento dos marcos de projeto

A base para o acompanhamento dos marcos de projeto será a última versão do documento de requisitos para checagem da qualidade do desenvolvimento até aquele ponto. Qualquer mudança nos requisitos implicará acionar o planejamento do projeto para rever marcos e atividades de cronograma.

xi. Suportar o gerenciamento das interfaces com outros sistemas

O documento de requisitos deve documentar as interfaces com outros sistemas. Mudança de requisitos que afetem interface com outros sistemas implicará acionar o processo de design e implementação para implementar as mudanças.

3.5 Conclusão

A fase “Análise de Requisitos” do ciclo de vida de desenvolvimento de sistema é a fase que compreende o processo completo de requisitos, iniciando na etapa descobrimento de requisitos e finalizando o ciclo desta fase com a versão inicial do documento de requisitos do sistema validado. Vale lembrar que a etapa de gerenciamento de requisitos ficará em estado latente durante todo o ciclo de desenvolvimento do sistema, acordando a cada solicitação de mudança, ou, provendo informação histórica e de rastreabilidade sobre os requisitos. A fase Análise de Requisitos acontecerá em paralelo com as demais fases do ciclo de desenvolvimento, principalmente com o design (projeto lógico e físico). É provável que esta espiral da execução das fases perdure até a descontinuação do produto.

Esta espiral vem de encontro ao forte conceito atual de Software Crescente, em que se procura desenvolver o produto em módulos, de tal forma a evitar o desgaste entre desenvolvedor e cliente, que passa a ver mais rapidamente o retorno do seu investimento. O trabalho em módulos proporciona entrega do produto, ainda que em partes, muito mais rapidamente, diminuindo a pressão do cliente para a entrega do produto, já que de tempos em tempos novos módulos vão sendo liberados. É claro que existem projetos que não tem a possibilidade de modularizar o produto final, porém, tais projetos ocorrem em pequenas quantidades, se comparados aos projetos que podem se utilizar do conceito de modularização.

No caso da impossibilidade da modularização, cuidado para não atropelar o processo de desenvolvimento. Lembre-se: “A qualidade do produto final depende diretamente da qualidade dos requisitos”. Na hora de cortar tempo de cronograma do projeto, critério para cortar a coisa certa.

3.6 Resumo

- Requisito é, em última análise, a definição do que deve ser feito pelo sistema;
- O principal atributo de um requisito é ser testável. Se ele não possuir este atributo, então, isto é um forte sinal que ele tem problema;
- O ciclo de vida recomendado compreende as seguintes fases: descobrimento, análise e negociação, documentação, validação e gerenciamento de requisitos;
- A fase de descobrimento e a de análise e negociação são fortemente integradas e tem como objetivo o reconhecimento dos requisitos verdadeiros do cliente para a solução do problema em questão;
- A fase de documentação e validação são responsáveis pela transformação dos requisitos do cliente em requisitos do sistema;
- A fase de gerenciamento de requisitos é responsável por controlar as mudanças nos requisitos, além de prover visibilidade do relacionamento entre os requisitos e com outros artefatos do projeto. Esta visibilidade é conhecida por rastreabilidade;
- o produto final da análise de requisitos é o documento de requisitos do sistema, que passa a ser o acordo entre cliente e desenvolvedor do que deve ser construído. Este documento guiará as fases seguintes do ciclo de vida de desenvolvimento. A qualidade do produto será diretamente proporcional a qualidade do documento de requisitos;
- É fundamental para o sucesso de qualquer projeto, inclusive de desenvolvimento de sistemas, um líder técnico com habilidade de formular e interpretar perguntas e respostas, de identificar as pessoas certas que devem ser envolvidas no projeto, pois, a elas serão dirigidas as perguntas e se avaliará as respostas. Além disso, tirar o máximo proveito de reuniões de trabalho também é fundamental, já que é a maior ferramenta conhecida para extração de requisitos e este processo, fatalmente envolverá um grupo de pessoas de diversos segmentos;

- Dois autores tem um foco muito centrado na fase de descobrimento de requisitos em suas bibliografias: Weinberg [29, 30, 34] e Macaulay [35];
- Sommerville [37] é leitura obrigatória para a fase de gerenciamento de requisitos, devido aos sólidos conceitos que nos proporciona, apesar de ressalvas em relação a algumas ferramentas que sugere aplicação;
- Como abordagem geral sobre Engenharia de Requisitos, é fundamental a leitura de Kotonya [36], uma excelente bibliografia.

CAPÍTULO 4

Modelo Proposto para Gerenciamento de Requisitos

4.1	INTRODUÇÃO	94
4.2	O MODELO PROPOSTO.....	94
4.3	BASE DE REPRESENTAÇÃO DE REQUISITOS	95
4.4	A MODELAGEM DO PROCESSO DE GERENCIAMENTO DE REQUISITOS	100
4.5	DESCRIÇÃO DO MODELO PROPOSTO DO PROCESSO DE GERENCIAMENTO DE REQUISITOS.....	102
4.6	RESUMO	106

4 Modelo Proposto para Gerenciamento de Requisitos

Nesta etapa da dissertação está documentada a proposta central do trabalho. É constituída de um modelo de Gerenciamento de Requisitos que deu origem a uma ferramenta automatizada de suporte a este modelo. A modelagem foi realizada no ambiente AGIR, mais especificamente em uma de suas três ferramentas, conhecida por FUN, cuja função é prover suporte a modelos de processos de negócio. A partir deste modelo, foi iniciada a construção de uma ferramenta para suportar o processo de Gerenciamento de Requisitos, objetivo maior deste trabalho.

4.1 Introdução

A abordagem de Gerenciamento de Requisitos, inicia a partir das informações levantadas na fase de extração de requisitos. Por este motivo, esta etapa é fundamental ao processo de GR, pois são as informações capturadas nesta fase que dão sustentação ao processo.

Desta forma, a primeira ação antes de iniciar a definição do modelo foi definir um documento padrão para fazer o levantamento dos requisitos. Partimos do documento proposto pela Edna [41] e fizemos algumas alterações, de forma a adequar todas as informações necessárias ao processo de GR.

Após a geração do documento padrão para levantamento dos requisitos, passamos ao estudo do entendimento dos modelos de qualidade e das mais diversificadas referências bibliográficas e chegamos ao modelo proposto, descrito a seguir.

4.2 O Modelo Proposto

O modelo constitui a proposta básica do trabalho aqui apresentado. Representa uma abordagem para consolidar a idéia dos modelos de qualidade sobre a importância da área chave Gerenciamento de Requisitos.

Para que o GR seja efetivo, há necessidade de se ter um documento padrão de descrição de requisitos o qual produz muitas das informações que são manipuladas durante a avaliação de impacto das mudanças.

Após a primeira passada pelas fases do ciclo de vida dos requisitos, gera-se a versão 001 da baseline de requisitos. A partir desta versão, qualquer alteração nos requisitos deve passar por um processo formal de avaliação que inicia com o preenchimento de uma solicitação de mudança, segundo o CMMI de responsabilidade da área chave “Desenvolvimento de Requisitos” e encerra com a avaliação de impacto da mudança, atividade responsável por aceitá-la ou rejeitá-la.

O próximo passo, após a elaboração da solicitação de mudança, consiste na seleção dos stakeholders que farão a análise da mudança. Nesta etapa a mudança é verificada em termos de conteúdo dos requisitos, procurando verificar se todos os requisitos que compõem a mudança foram identificados.

Em seguida, caso os stakeholders responsáveis pela análise da mudança a considerem pertinente, passa-se para a atividade de elaboração da proposta da mudança, responsável por verificar se os requisitos da mudança estão descritos de maneira apropriada. Além disso, a mudança é avaliada para se estimar tempo, custo e recursos necessários.

A partir daí, a proposta de mudança vai para o líder de projeto e o responsável do cliente para que eles procedam a avaliação de impacto da mudança no projeto. Neste ponto a mudança é aceita ou rejeitada.

Em caso de aceitação da solicitação de mudança, gera-se uma nova versão da baseline de requisitos e os responsáveis por atividades que possam ser afetadas pela nova versão recebem um aviso para avaliar o impacto da mudança em suas atividades.

A medida que novas mudanças são solicitadas, passa a ser possível fazer uma análise analítica do processo de GR, procurando identificar com base nas informações históricas, os requisitos que mais sofrem modificações, quem é a fonte da informação, o tempo médio de implementação e muitas outras informações que servirão de base para a melhoria do próprio processo de GR e do processo de desenvolvimento de software como um todo.

4.3 Base de Representação de Requisitos

A base de representação das informações deve ser um meio que permita a comunicação mais facilitada para o entendimento do problema. A linguagem adotada para esta

representação deve fazer parte do contexto, referir-se ao assunto que se está tratando. A simbologia adotada deve corresponder à utilizada no ambiente. O treinamento prévio a ser feito deverá ser o do engenheiro de requisitos para se adequar ao ambiente e prover a leitura e interpretação. Enfim, o que estiver documentado deve representar e ser interpretado à luz do ambiente cultural que o originou. A linguagem é a expressão do contexto, com seus sinais e símbolos. Para promover o entendimento comum é necessário prover a comunicação recíproca entre as partes.

Segundo Leite [54], o Léxico Ampliado da Linguagem (LAL), é uma técnica que procura descrever os símbolos de uma linguagem. A idéia central é a existência de uma linguagem da aplicação. A representação da linguagem fundamenta-se em dois aspectos: a noção dos fatos ou fenômenos do ambiente e os impactos decorrentes da aplicação destes elementos.

No processo de descoberta dos requisitos além de observar-se a linguagem é necessário representá-la na forma de registro escrito e interpretável pela fonte de informação como sua expressão verdadeira.

A opção adotada pelo modelo proposto é representar requisitos através de sentenças simples em linguagem natural. Em princípio, talvez tenha-se um grande volume, com ambigüidades a eliminar e dados a complementar. No entanto, por motivo da abordagem de tratamento da informação ser orientada ao tratamento de problemas, isto deve facilitar o surgimento de idéias de forma mais espontânea.

A estrutura da linguagem adotada pelo modelo na representação da informação tem um rigor maior nas sentenças que descrevem os requisitos. Baseia-se na funcionalidade do requisito que representa a transformação de uma entrada em saída ou uma mudança de estado.

A forma de expressão do requisito é proposta no modelo pelo Documento de Descrição de Requisitos, representado na seqüência. Deve seguir a estrutura: nome (sujeito) + verbo (ação) + objeto (sobre o quê) + complemento que caracterize a funcionalidade do requisito associada ao problema.

O Documento de Descrição de Requisito, conforme figura 4.1, é a base de geração de informações para o processo de Gerenciamento de Requisitos.

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: _____
2. Nome do Requisito: _____
3. Responsável pela Informação: _____ Área: _____ data: _____
4. Característica do Requisito: _____ (1 – funcional, 2 – não funcional)
5. Área de Origem: _____ (1 – interna, 2 – externa, 3 – ordem legal)
6. Qualificação Funcional: _____ (1 – operacional, 2 – gerencial, 3 – estratégico)
7. Tipo do Requisito: _____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):

9. Problema Identificado:

10. Produto Esperado:

11. Aplicação:

12. Atributos

13. Restrições

14. Preferências

15. Expectativas

Figura 4.1 – Documento de Descrição de Requisitos

Descrição explicativa: **Documento de Descrição de Requisitos**

1. Identificação do Requisito:

Corresponde ao número atribuído para controle e referência do requisito, em ordem seqüencial de registro.

2. Nome do Requisito:

É a descrição sucinta do requisito.

3. Responsável pela Informação:

É a identificação de quem está prestando a informação: nome, área em que trabalha e a data de inclusão do requisito.

4. Característica do Requisito:

Identifica se o requisito especifica uma funcionalidade ou não.

5. Área de Origem:

Identifica a área de origem do requisito, a área demandante:

1 – Interna (origem interna da organização)

2 – externa (origem externa à organização)

3 – de ordem legal (origem devido a aplicação da legislação)

6. Qualificação Funcional:

Caracteriza o nível funcional do requisito

1 – operacional

2 – gerencial

3 – estratégico

7. Tipo do Requisito:

Caracteriza o grau de cuidado com o requisito

1 – normal (requisito sem característica específica)

2 – global (requisito que afeta o projeto como um todo)

3 – volátil (requisito que tende a mudança constante)

4 – emergente (requisito que apareceu durante o projeto)

8. Descrição do Requisito:

Refere-se a descrição do requisito de forma clara e comunicativa. É a condição ou a exigência expressa pelo cliente para satisfação dos objetivos relacionados a solução do problema. O requisito deve ser consistente, confiável e completo.

9. Problema Identificado:

Refere-se ao problema ou necessidade que originou o requisito.

10. Produto Esperado:

Refere-se ao produto que se deseja obter para a satisfação do requisito. Corresponde a forma de solução para um problema específico.

11. Aplicação:

Refere-se à forma de aplicação do produto gerado pela implementação do requisito. A quem se destina a solução.

12. Atributos:

Refere-se à forma ou conteúdo do produto para atendimento ao requisito. Os atributos são dimensões das características de funcionalidade e de qualidade dos requisitos no domínio da aplicação.

13. Restrições:

Refere-se às restrições de ordem social, política, econômica, organizacional ou legal que delimitam o atendimento ao requisito demandado. As restrições são limitações que delineiam o espaço de solução do problema. As mais comuns referem-se às limitações para o conhecimento da informação e ao acesso ao universo da fonte de informação atual e potencial futuro.

14. Preferências:

Refere-se às preferências do cliente quanto ao atendimento do requisito (do que). As preferências são condições desejáveis e particulares do cliente, porém, opcionais. São condicionadas a definição prévia dos atributos e das restrições, ou seja, são circunscritas no espaço de solução do problema.

15. Expectativas:

Refere-se às expectativas a serem atendidas no domínio conhecido pelo cliente (para quando, tipo de satisfação). As expectativas são formas de expressão de desejo do cliente. São originadas do conhecimento do problema e do ambiente, cuja satisfação se refere à esperança de solução.

4.4 A Modelagem do Processo de Gerenciamento de Requisitos

O processo de gerenciamento de requisitos foi estruturado em diversas atividades distintas, iterativas e distribuídas em várias etapas do processo, gerando versões complementares do documento de requisitos.

A figura 4.2 representa a modelagem do processo de GR e o ciclo evolutivo de controle de mudança no documento de requisitos.

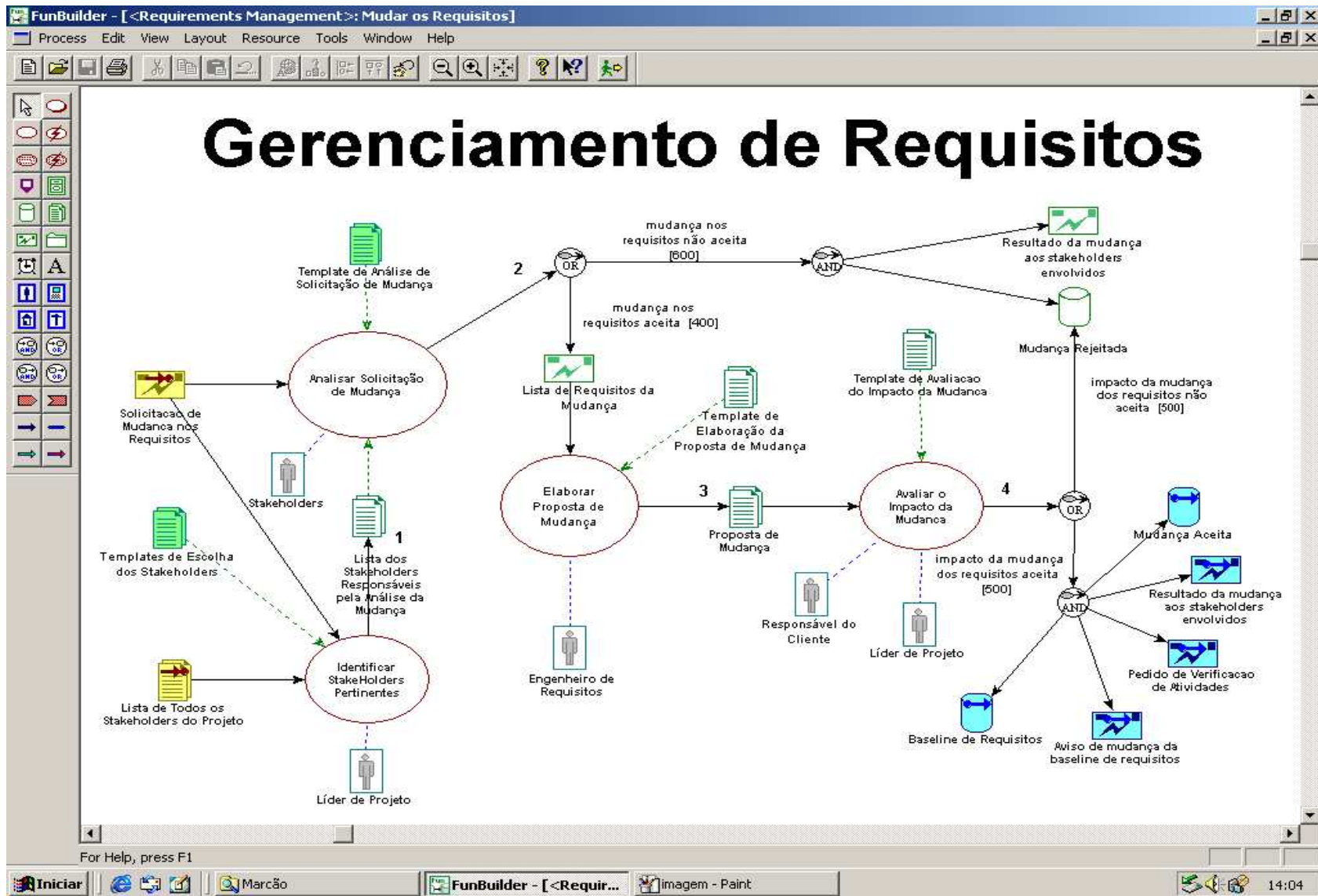


Figura 4.2 – Modelo proposto do processo de Gerenciamento de Requisitos

4.5 Descrição do Modelo Proposto do Processo de Gerenciamento de Requisitos

Descrição das Atividades

Atividade 1:

Identificar Stakeholders Pertinentes

Descrição:

Identificar os stakeholders que serão responsáveis por analisar a Solicitação de Mudança nos requisitos.

Referência:

Template de Escolha dos Stakeholders

- formulário a ser preenchido com o nome dos stakeholders escolhidos para realizar a Análise da Solicitação de Mudança.

Entrada:

Lista de Todos os Stakeholders do Projeto

- lista de todos os stakeholders do projeto.

Solicitação de Mudança nos Requisitos

- solicitação de mudança nos requisitos proposta por um stakeholder do projeto.

Saída:

Lista dos Stakeholders Responsáveis pela Análise da Mudança

- lista dos stakeholders escolhidos para procederem a Análise da Solicitação de Mudança.

Recurso necessário:

Líder de Projeto

- responsável pela escolha dos stakeholders que farão a Análise da Solicitação de Mudança.

Descrição das Atividades

Atividade 2:

Analisar Solicitação de Mudança

Descrição:

Analisar os requisitos da solicitação de mudança procurando identificar se faltam requisitos que serão afetados pela mudança, se estão consistentes e não ambíguos.

Referência:

Template de Análise da Solicitação de Mudança

- formulário a ser preenchido com as informações da Análise da Solicitação de Mudança.

Entrada:

Lista dos Stakeholders Responsáveis pela Avaliação da Mudança

- lista dos stakeholders escolhidos para procederem a Análise da Solicitação de Mudança.
- Solicitação de Mudança nos Requisitos
- solicitação de mudança nos requisitos proposta por um stakeholder do projeto.

Saída:

Mudança nos requisitos não aceita

- Resultado da Mudança aos stakeholders envolvidos.
- Arquivamento da Mudança rejeitada.

Mudança nos requisitos aceita

- Lista de Requisitos da Mudança.

Recurso necessário:

Stakeholders

- Stakeholders escolhidos para procederem Análise da Solicitação de Mudança.

Descrição das Atividades

Atividade 3:

Elaborar a Proposta de Mudança

Descrição:

Avaliar os requisitos da solicitação de mudança analisada, sob a luz da Engenharia de Requisitos, procurando identificar se os requisitos estão completos, corretos, consistentes e sem ambigüidade.

Referência:

Template de Elaboração da Proposta de Mudança

- formulário a ser preenchido com as informações da Proposta de Mudança.

Entrada:

Lista de Requisitos da Mudança

- lista dos requisitos da Solicitação de Mudança analisada.

Saída:

Proposta de Mudança

- os requisitos da Solicitação de Mudança analisada são descritos de forma técnica.

Recurso necessário:

Engenheiro de Requisitos

- responsável por avaliar a lista de requisitos da mudança, procurando identificar incompletude, ambigüidade, problema de entendimento e descrevê-los em um nível adequado de qualidade.

Descrição das Atividades

Atividade 4:

Avaliar o Impacto da Mudança

Descrição:

Avaliar o impacto da mudança nos requisitos sobre os fatos atuais, procurando verificar se há infra-estrutura necessária disponível e interesse na implementação da mudança.

Referência:

Template de Avaliação de Impacto da Mudança

- formulário a ser preenchido com as informações da Avaliação de Impacto da Mudança.

Entrada:

Proposta de Mudança

- os requisitos da Solicitação de Mudança analisada são descritos de forma técnica.

Saída:

Impacto da Mudança nos Requisitos não aceita

- resultado da mudança aos stakeholders envolvidos.
- arquivamento da mudança rejeitada.

Impacto da Mudança nos Requisitos aceita

- resultado da mudança aos stakeholders envolvidos.
- atualização da Baseline de Requisitos.
- arquivamento da mudança aceita.
- Aviso de mudança da Baseline de Requisitos aos envolvidos no projeto.
- Pedido de verificação das Atividades do projeto.

Recurso necessário:

Líder de Projeto

- responsável por avaliar o impacto da mudança.

Responsável do cliente

- responsável por avaliar o impacto da mudança.

4.6 Resumo

Neste capítulo apresentamos a descrição completa do Modelo de Gerenciamento de Requisitos proposto e suas atividades, além de detalhar o modelo do documento para descrição dos requisitos.

CAPÍTULO 5

A Ferramenta SIGERE para automatizar o processo de Gerenciamento de Requisitos

5.1	INTRODUÇÃO	108
5.2	REQUISITOS DO SISTEMA.....	108
5.3	DESCRIÇÃO DAS TELAS.....	117
5.4	RESUMO	159

5 A Ferramenta SIGERE para automatizar o processo de Gerenciamento de Requisitos

Nesta etapa da dissertação está documentada a ferramenta SIGERE construída para suportar o processo de gerenciamento de requisitos proposto.

5.1 Introdução

A partir do modelo proposto para gerenciamento de requisitos de uma organização que utiliza a metodologia orientada a processo em seu desenvolvimento de software, derivou-se uma ferramenta de suporte a este processo. A ferramenta foi desenvolvida para ambiente Internet, utilizando a linguagem ASP no desenvolvimento do Front-end e banco de dados ACCESS, via ODBC, visando facilitar a migração do aplicativo para outros bancos de dados.

5.2 Requisitos do Sistema

A seguir são descritos os requisitos do sistema. Estes requisitos formam a base sobre a qual a ferramenta está fundamentada.

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 001

2. Nome do Requisito: Identificar Stakeholders Pertinentes _____

3. Responsável pela Informação: Marco _____ Área: desenvolvimento ___ data: 01/06/2001

4. Característica do Requisito: 1 _____ (1 – funcional, 2 – não funcional)

5. Área de Origem: 1 _____ (1 – interna, 2 – externa, 3 – ordem legal)

6. Qualificação Funcional: 1 _____ (1 – operacional, 2 – gerencial, 3 – estratégico)

7. Tipo do Requisito: 1 _____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):

_O líder de projeto deve identificar os stakeholders que serão responsáveis por efetuar a Análise da Solicitação de Mudança.

9. Problema Identificado:

_Existe a necessidade de que a Análise da Solicitação de Mudança seja realizada por pessoas que estejam mais capacitadas para executar esta atividade.

10. Produto Esperado:

_A lista dos stakeholders que serão responsáveis por proceder a Análise da Solicitação de Mudança.

11. Aplicação:

_Racionalizar o processo de seleção dos stakeholders responsáveis pela Análise da Solicitação de Mudança.

12. Atributos

_ Nome dos stakeholders selecionados.

13. Restrições

_ Somente os stakeholders do projeto poderão ser selecionados.

14. Preferências

_ Possibilidade de consultar todas as informações dos stakeholders durante o processo de seleção dos stakeholders que serão responsáveis por Analisar a Solicitação de Mudança.

15. Expectativas

_ Ter toda a informação da Solicitação de Mudança durante o processo de seleção dos stakeholders responsáveis.

Tabela 5.1 – Requisito 001 - Identificar Stakeholders Pertinentes

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 002
2. Nome do Requisito: Analisar a Solicitação de Mudança
3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/06/2001
4. Característica do Requisito: 1_____ (1 – funcional, 2 – não funcional)
5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):

_Os stakeholders responsáveis deverão efetuar a Análise da Solicitação de Mudança.

9. Problema Identificado:

_Existe a necessidade de que a Análise da Solicitação de Mudança seja realizada para verificar se a solicitação é pertinente e se os requisitos da mudança contemplam completamente a mudança desejada.

10. Produto Esperado:

_A lista completa dos requisitos da mudança.

11. Aplicação:

_Racionalizar o processo de Análise da Solicitação de Mudança.

12. Atributos

_Informação completa dos requisitos da mudança e o parecer da atividade de Análise da Solicitação de Mudança.

13. Restrições

_Somente os stakeholders responsáveis pela Análise poderão executar esta Atividade.

14. Preferências

_Possibilidade de consultar todas as informações dos requisitos da mudança durante o processo de Análise da Solicitação de Mudança.

15. Expectativas

_Ter toda a informação da Solicitação de Mudança durante o processo de Análise da Solicitação de Mudança.

Tabela 5.2 – Requisito 002 - Analisar a Solicitação de Mudança

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 003

2. Nome do Requisito: Elaborar Proposta de Mudança

3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/06/2001

4. Característica do Requisito: 1_____ (1 – funcional, 2 – não funcional)

5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)

6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)

7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):

_ O Engenheiro de Requisitos deve avaliar os requisitos da solicitação de mudança analisada, sob a luz da Engenharia de Requisitos, procurando identificar se os requisitos da mudança estão completos, corretos, consistentes e sem ambigüidade.

9. Problema Identificado:

_ Existe a necessidade de que a Solicitação de Mudança Analisada seja verificada em relação a qualidade da descrição dos requisitos da mudança, de forma a eliminar problemas de entendimento.

10. Produto Esperado:

_ A Proposta de Mudança.

11. Aplicação:

_ Racionalizar o processo de Elaboração da Proposta de Mudança.

12. Atributos

_ Informação completa dos requisitos da mudança com o parecer da atividade de Análise da Solicitação de Mudança e as informações sobre custo, tempo e recursos necessários para implementação da mudança.

13. Restrições

_ Somente um engenheiro de requisitos poderá realizar esta atividade.

14. Preferências

_ Possibilidade de consultar todas as informações dos requisitos da mudança durante o processo de Elaboração da Proposta de Mudança.

15. Expectativas

_ Ter toda a informação da Solicitação de Mudança durante o processo de Elaboração da Proposta de Mudança.

Tabela 5.3 – Requisito 003 – Elaborar Proposta de Mudança

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 004
2. Nome do Requisito: Avaliar Impacto da Mudança
3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/06/2001
4. Característica do Requisito: 1_____ (1 – funcional, 2 – não funcional)
5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):

_ O Líder de Projeto e o Responsável do Cliente devem avaliar o impacto da mudança nos requisitos sobre os fatos atuais, procurando verificar se há infra-estrutura necessária disponível e interesse na implementação da mudança.

9. Problema Identificado:

_ Existe a necessidade de que qualquer mudança no escopo do projeto, a partir da aprovação da versão 001 da Baseline de Requisitos, passe por um processo formal de aprovação, que avalie o impacto da mudança desejada.

10. Produto Esperado:

_ Resultado da Solicitação de Mudança (Aceita ou Rejeitada).

11. Aplicação:

_ Racionalizar o processo de Avaliação de Impacto da Mudança.

12. Atributos

_ Informação completa dos requisitos da mudança com o parecer da atividade de Análise da Solicitação de Mudança, as informações sobre custo, tempo e recursos necessários para implementação e o parecer da atividade de Avaliação de Impacto da Mudança.

13. Restrições

_ Somente o Líder de Projeto e o Responsável do Cliente poderão realizar esta atividade.

14. Preferências

_ Possibilidade de consultar todas as informações dos requisitos da mudança durante o processo de Avaliação de Impacto da Mudança.

15. Expectativas

_ Ter toda a informação da Solicitação de Mudança durante o processo de Avaliação de Impacto da Mudança.

Tabela 5.4 – Requisito 004 – Avaliar Impacto da Mudança

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 005
2. Nome do Requisito: Consultar a Baseline de Requisitos
3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/06/2001
4. Característica do Requisito: 1_____ (1 – funcional, 2 – não funcional)
5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):
_ Os stakeholders do projeto querem consultar a baseline de requisitos.

9. Problema Identificado:
_ Existe a necessidade de se consultar os requisitos do sistema.

10. Produto Esperado:
_ Informação dos requisitos do sistema.

11. Aplicação:
_ Disponibilidade da informação dos requisitos do sistema.

12. Atributos
_ Informação completa dos requisitos do sistema.

13. Restrições

14. Preferências
_ Fácil recuperação das informações.

15. Expectativas

Tabela 5.5 – Requisito 005 - Consultar a Baseline de Requisitos

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 006
2. Nome do Requisito: Acompanhar a Solicitação de Mudança
3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/06/2001
4. Característica do Requisito: 1_____ (1 – funcional, 2 – não funcional)
5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):

_ Os stakeholders do projeto querem acompanhar a Solicitação de Mudança, desde o seu cadastramento até o seu parecer final.

9. Problema Identificado:

_ Existe a necessidade de se acompanhar o andamento da solicitação de mudança.

10. Produto Esperado:

_ informação da situação da Solicitação de Mudança.

11. Aplicação:

_ Acompanhar o andamento da Solicitação de Mudança.

12. Atributos

_ Informação completa da Solicitação de Mudança e a situação em que esta se encontra.

13. Restrições

14. Preferências

_ Fácil recuperação das informações.

15. Expectativas

Tabela 5.6 – Requisito 006 - Acompanhar a Solicitação de Mudança

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 007
 2. Nome do Requisito: Segurança de Acesso ao Sistema
 3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/06/2001
 4. Característica do Requisito: 2_____ (1 – funcional, 2 – não funcional)
 5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
 6. Qualificação Funcional: 2_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
 7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)
-
8. Descrição do Requisito (sujeito + verbo + objeto):
_ O sistema deve ser acessado somente por pessoas autorizadas.
 9. Problema Identificado:
_ Existe a necessidade de se bloquear o acesso a pessoas não autorizadas.
 10. Produto Esperado:
_ Tela de Login de acesso ao sistema.
 11. Aplicação:
_ Segurança de acesso às informações do sistema.
 12. Atributos
_ login e senha.
 13. Restrições
_
 14. Preferências
_ Tela inicial do Aplicativo.
 15. Expectativas
_ bloquear o acesso direto às páginas Web do sistema, caso o usuário não entrado pela Tela de login.

Tabela 5.7 – Requisito 007 - Segurança de Acesso ao Sistema

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 008
 2. Nome do Requisito: Plataforma de Desenvolvimento do Sistema
 3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/06/2001
 4. Característica do Requisito: 2_____ (1 – funcional, 2 – não funcional)
 5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
 6. Qualificação Funcional: 2_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
 7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)
-
8. Descrição do Requisito (sujeito + verbo + objeto):
_ A plataforma do sistema deve ser em ambiente Web e banco de dados Access.
 9. Problema Identificado:
_ O ambiente em que o sistema operará é imposição do cliente.
 10. Produto Esperado:
_ O sistema desenvolvido para rodar em ambiente Web e com acesso ao banco de dados Access, porém, com fácil possibilidade de migração para outros bancos de dados do mercado (SQL, Sybase, Oracle).
 11. Aplicação:
_ Atendimento à vontade do cliente.
 12. Atributos
 13. Restrições
_ limitação do ambiente de operação e desenvolvimento do sistema.
 14. Preferências
 15. Expectativas

Tabela 5.8 – Requisito 008 - Plataforma de Desenvolvimento do Sistema

5.3 Descrição das Telas

As telas são descritas uma a uma, seguindo a lógica em que as funções devem ser executadas, de forma a facilitar o entendimento. Para páginas Web com tamanho maior que o de uma tela de vídeo, a descrição será apresentada tela a tela, se utilizando da tecla avançar página, conhecida por PgDn (page down).

Fluxo de cadastramento das informações no sistema:

- Cadastrar os stakeholders da organização;
- Cadastrar a competência dos stakeholders na organização;
- Cadastrar o projeto para o qual se fará o gerenciamento de requisitos;
- Cadastrar a competência dos stakeholders no projeto;
- Cadastrar os requisitos de sistema do projeto;
- Cadastrar a dependência dos requisitos;
- Cadastrar as atividades dependentes dos requisitos.

Tela de Login – Figura 5.1

Todo o usuário necessita de um login e senha para ter acesso ao sistema. Isto significa que o sistema bloqueia usuários não autorizados. Ainda que se tente entrar no sistema através do endereço de outras páginas do aplicativo, o sistema bloqueia o usuário se o mesmo não estiver efetuado o login.

Campos da Tela:

- Login: informar o login do usuário para acesso ao sistema;
- Senha: informar a senha do usuário (até 8 caracteres alfanuméricos).

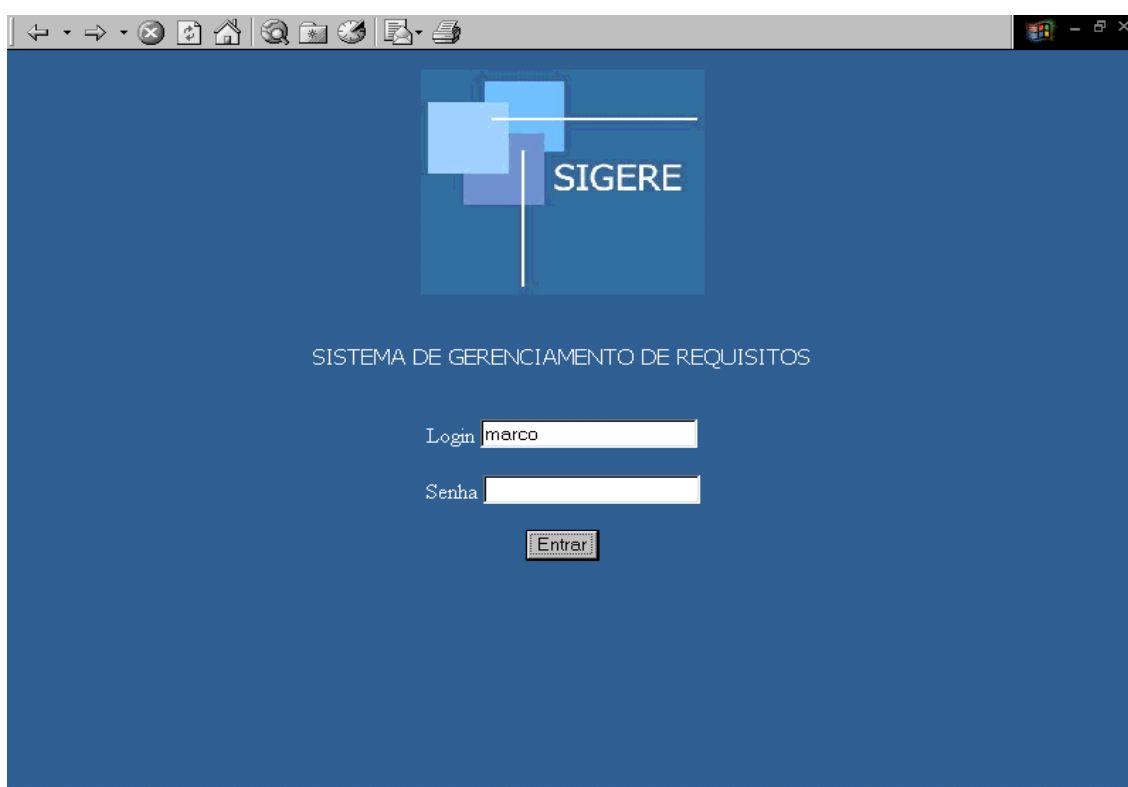


Figura 5.1 – Tela de Login do Sistema

Tela Inicial do Sistema – Figura 5.2

Mostra o conceito da interface utilizada. O menu principal, que contém a funcionalidade do aplicativo, permanece visível em todas as telas. A tela inicial apresenta os envolvidos no projeto que patrocinaram a criação da ferramenta.

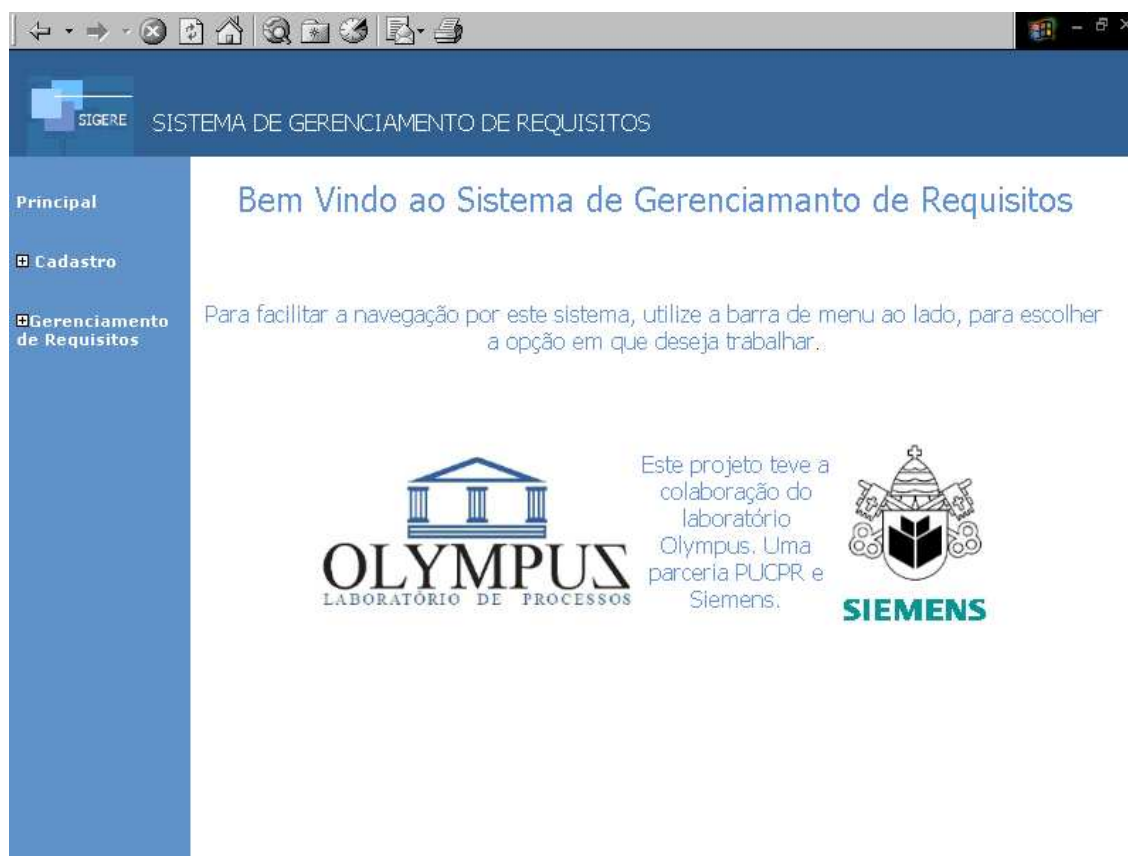


Figura 5.2 – Tela inicial do sistema

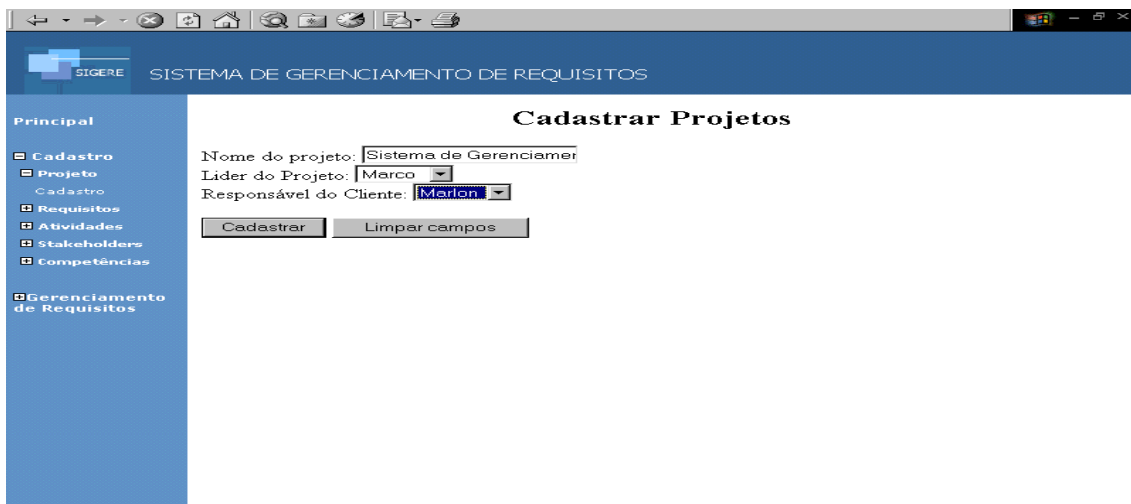
Função Cadastrar Projetos – Figura 5.3

Função que permite o cadastramento dos projetos da organização para os quais se fará o gerenciamento dos requisitos.

Campos da Tela:

- Nome do Projeto: nome do projeto;
- Líder do Projeto: selecionar um dos nomes da lista do tipo Combobox que contém os nomes de todos os stakeholders da organização;
- Responsável do Cliente: selecionar um dos nomes da lista do tipo Combobox que contém os nomes de todos os stakeholders da organização.

Instrução: caso o stakeholder que será o Líder do Projeto ou o Responsável do Cliente não apareçam na lista de seleção, cadastrá-los primeiro como um stakeholder da organização, através da função Cadastrar Stakeholder.



A imagem mostra a interface de usuário do sistema SIGERE, especificamente a tela "Cadastrar Projetos". O cabeçalho do sistema indica "SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS". À esquerda, há um menu de navegação com opções como "Principal", "Cadastro", "Projeto", "Requisitos", "Atividades", "Stakeholders", "Competências" e "Gerenciamento de Requisitos". O formulário principal contém os seguintes campos e controles:

- Nome do projeto: Sistema de Gerenciamei
- Líder do Projeto: Marco
- Responsável do Cliente: Marlon
- Botões: Cadastrar, Limpar campos

Figura 5.3 – Tela Cadastrar Projetos

Função Cadastrar Requisitos – Figura 5.4

Esta função permite o cadastramento dos requisitos da versão 001 da baseline que se origina no processo Desenvolvimento de Requisitos, conforme diz o CMMI. A partir desta versão, qualquer mudança nos requisitos ocorrerá de forma controlada, através de uma solicitação de mudança.

Campos da Tela

- Projeto: selecionar o projeto ao qual pertence o requisito;
- Stakeholder Responsável: selecionar dentre os stakeholders do projeto, aquele que é a fonte principal da informação deste requisito;
 - Requisito: informar nome do requisito;
 - Data de Inclusão: data de origem do requisito no projeto;
 - Qualificação Funcional: selecionar o tipo de qualificação funcional do requisito (operacional, gerencial e estratégico);
 - Área de Origem: selecionar a área de origem do requisito (interna, externa ou de ordem legal);
 - Característica: selecionar a característica do requisito (funcional, usabilidade, confiabilidade, eficiência, portabilidade e manutenibilidade);
 - Tipo do Requisito: selecionar o tipo do requisito (volátil, emergente, normal, global);
 - Descrição: informar a descrição completa do requisito;
 - Problema Identificado: informar o problema que o requisito está solucionando;
 - Produto Esperado: informar o produto que se espera obter para atendimento do requisito;
 - Aplicação do Requisito: informar aplicação do produto da demanda do requisito;
 - Atributos: informar se existem atributos desejados no conteúdo do produto para atendimento do requisito;

- Restrições: informar se existem restrições de ordem social, política, econômica que delimitam o atendimento ao requisito;
- Preferências: informar se existem condições desejáveis e particulares do stakeholder quanto ao atendimento do requisito;
- Expectativas: informar se existem expectativas a serem atendidas no domínio conhecido pelo stakeholder para atendimento do requisito;
- Situação: informar se o requisito cadastrado foi aceito ou rejeitado.

Obs: ao clicar no botão LIMPAR CAMPOS, o requisito informado é cadastrado e os campos são limpos, visando o cadastramento de outros requisitos.

The screenshot shows a web browser window displaying the 'SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS' interface. The main content area is titled 'Cadastrar Requisitos'. On the left, there is a navigation menu with categories like 'Principal', 'Cadastro', 'Projeto', 'Requisitos', 'Atividades', 'Stakeholders', 'Competências', and 'Gerenciamento de Requisitos'. The form fields are as follows:

- Projeto: Sistema de Gerenciamento de Requisitos (dropdown)
- StakeHolder Responsável: Marlon (dropdown)
- Requisto: [text input]
- Data de Inclusão (dd/mm/aaaa): [text input]
- Qualificação Funcional: Operacional (dropdown)
- Área de Origem: Interna (dropdown)
- Característica: Funcional (dropdown)
- Tipo do Requisito: Volátil (dropdown)
- Descrição: [text input]
- Problema Identificado: [text input]
- Produto Esperado: [text input]
- Aplicação dos Requisitos: [text input]
- Atributos Desejados: [text input]
- Restrições de Implementação: [text input]
- Referências de Implementação: [text input]
- Expectativas: [text input]
- Situação: Proposto (dropdown)

At the bottom of the form, there is an 'enviar' button and a 'Limpa campos' button.

Figura 5.4 – Tela Cadastrar Requisitos

Função Cadastrar Atividades Dependentes dos Requisitos – Figura 5.5

Esta função permite o cadastramento do relacionamento de dependência entre os requisitos e as atividades do projeto. Este relacionamento se apresenta na estrutura hierárquica, tipo pai-filho.

Campos da Tela:

- Projeto: selecionar o projeto para o qual será determinado o relacionamento de dependência entre os requisitos e as atividades do projeto;
- Requisito: selecionar o requisito que será o pai no relacionamento, ou seja, o requisito do qual dependerá a atividade;
- Atividade: selecionar a atividade do projeto que é dependente do requisito.

A imagem mostra a interface de usuário do sistema SIGERE. No topo, há uma barra azul com o logo 'SIGERE' e o texto 'SISTEMA DE GERENCIAMENTO DE REQUISITOS'. À esquerda, um menu lateral azul contém as opções: 'Principal', 'Cadastro' (com subitens 'Projeto', 'Requisitos', 'Atividades Dependentes'), 'Dependência' (com subitens 'Atividades', 'Stakeholders', 'Competências') e 'Gerenciamento de Requisitos'. O conteúdo principal da tela é o formulário 'Cadastrar Atividades Dependentes dos Requisitos'. Este formulário possui um campo 'Projeto' com um menu suspenso selecionando 'Sistema de Gerenciamento de Requisitos', um botão 'Selecionar', dois campos de menu suspenso para 'Requisito' e 'Atividade', um botão 'Ok' e um link 'Voltar' em roxo.

Figura 5.5 – Tela Cadastrar Atividades Dependentes dos Requisitos

Função Cadastrar Dependência de Requisitos – Figura 5.6

Esta função permite o cadastramento do relacionamento de dependência entre os requisitos. Este relacionamento se apresenta na estrutura hierárquica, tipo pai-filho.

Campos da Tela:

- Projeto: selecionar o projeto para o qual será determinado o relacionamento de dependência dos requisitos;
- Requisito Pai: selecionar o requisito que será o pai no relacionamento;
- Requisito Filho: selecionar o requisito que será o filho no relacionamento.

A imagem mostra a interface de usuário do sistema SIGERE (Sistema de Gerenciamento de Requisitos). O cabeçalho azul contém o logo SIGERE e o nome do sistema. À esquerda, há um menu de navegação com opções como 'Cadastro', 'Atividades', 'Stakeholders' e 'Gerenciamento de Requisitos'. O formulário principal, intitulado 'Cadastrar Dependência de Requisitos', possui os seguintes elementos:

- Projeto: [Sistema de Gerenciamento de Requisitos] (menu suspenso)
- Selecionar (botão)
- Requisito Pai: [] (menu suspenso)
- Requisito Filho: [] (menu suspenso)
- Ok (botão)
- Voltar (link)

Figura 5.6 – Tela Cadastrar Dependência de Requisitos

Função Cadastrar Atividades – Figura 5.7

Esta função permite o cadastramento das atividades de gerenciamento de projeto e de engenharia de software executadas pela organização.

Campos da Tela:

- Descrição: informar o nome da atividade executada na organização;
- Tipo: selecionar o tipo da atividade.

A imagem mostra a interface de usuário do sistema SIGERE, especificamente a tela de "Cadastrar Atividades". No topo, há uma barra azul com o logotipo "SIGERE" e o texto "SISTEMA DE GERENCIAMENTO DE REQUISITOS". À esquerda, um menu lateral azul contém opções como "Principal", "Cadastro", "Projeto", "Requisitos", "Atividades", "Stakeholders", "Competências" e "Gerenciamento de Requisitos". O formulário principal, intitulado "Cadastrar Atividades", possui os seguintes campos: "Descrição:" com o texto "Selecionar Stakeholders", "Tipo:" com um menu suspenso selecionando "Manual", e "Modelo:" com um menu suspenso selecionando "Negócio". Abaixo dos campos, há dois botões: "Cadastrar" e "Limpa campos".

Figura 5.7 – Tela Cadastrar Atividades

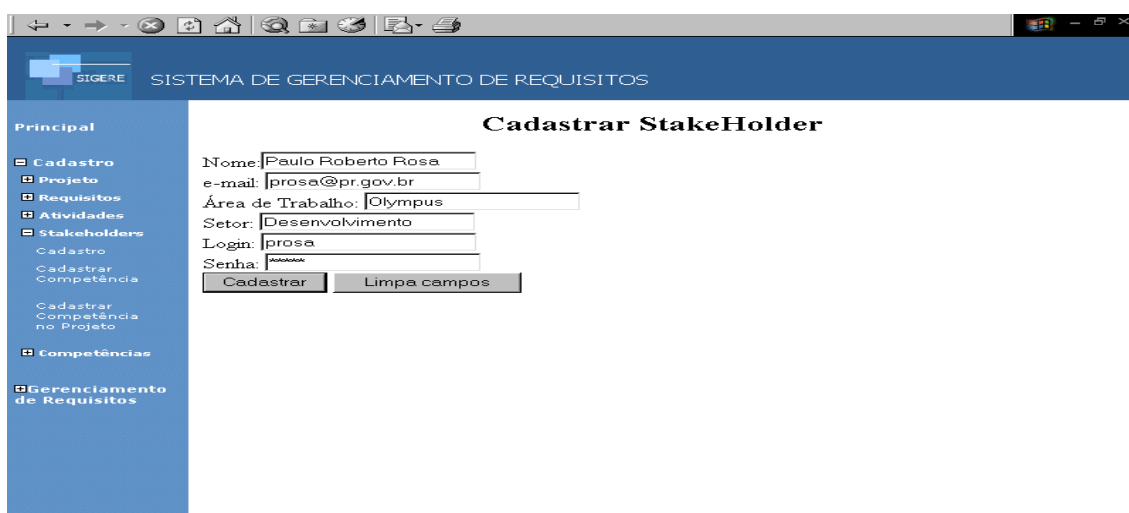
Antes de se cadastrar o projeto de desenvolvimento para o qual se vai gerenciar seus requisitos, devemos cadastrar os stakeholders da organização.

Função Cadastrar Stakeholder – Figura 5.8

Esta função permite o cadastramento de todos os stakeholders da organização. Todos os participantes de qualquer etapa do ciclo de mudança dos requisitos deve ser cadastrado nesta função.

Campos da Tela:

- Nome: informar o nome do stakeholder;
- E-mail: informar o e-mail do stakeholder;
- Área de Trabalho: informar a organização a que pertence o stakeholder;
- Setor: informar o setor na organização em que o stakeholder está lotado;
- Login: informar o login para acesso do stakeholder ao sistema;
- Senha: informar a senha do stakeholder para acesso ao sistema.



The screenshot shows a web browser window displaying the 'SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS' interface. The main content area is titled 'Cadastrar StakeHolder' and contains a form with the following fields and values:

- Nome: Paulo Roberto Rosa
- e-mail: prosa@pr.gov.br
- Área de Trabalho: Olympus
- Setor: Desenvolvimento
- Login: prosa
- Senha: [masked]

At the bottom of the form are two buttons: 'Cadastrar' and 'Limpa campos'. On the left side, there is a navigation menu with the following items:

- Principal
- Cadastro
 - Projeto
 - Requisitos
 - Atividades
 - Stakeholders
 - Cadastro
 - Cadastrar Competência
 - Cadastrar Competência no Projeto
- Competências
- Gerenciamento de Requisitos

Figura 5.8 – Tela Cadastrar Stakeholder

Função Cadastrar Competências dos Stakeholders – Figura 5.9

Esta função permite o cadastramento das competências dos stakeholders necessárias aos projetos da organização.

Campos da Tela:

- Stakeholder: selecionar da lista de stakeholders, aquele para o qual se vai cadastrar suas competências;
- Competências: selecionar as competências que o stakeholder possui na organização.

The screenshot displays the 'Cadastrar Competências dos Stakeholders' form within the SIGERE (SISTEMA DE GERENCIAMENTO DE REQUISITOS) application. The form includes a 'StakeHolder' dropdown menu currently set to 'Marlon'. Below this, there is a list of eight competencies, each with a checked checkbox: 'Analista de Sistemas', 'Programador', 'Testador', 'Usuário', 'Cliente', 'Engenheiro de Software', 'Engenheiro de Processo', and 'Engenheiro de Requisitos'. A 'Cadastrar' button is positioned at the bottom of the form. The left sidebar contains a navigation menu with categories like 'Principal', 'Cadastro', 'Projeto', 'Requisitos', 'Atividades', 'Stakeholders', 'Competências', and 'Gerenciamento de Requisitos'. The 'Stakeholders' section is expanded, showing sub-options for 'Cadastro', 'Cadastrar Competência', and 'Cadastrar Competência no Projeto'.

Figura 5.9 – Tela Cadastrar Competências dos Stakeholders

Função Cadastrar Competências dos Stakeholders no Projeto – Figura 5.10

Esta função permite o cadastramento das competências dos stakeholders necessárias aos projetos da organização.

Campos da Tela:

- StakeHolder: selecionar da lista de stakeholders, aquele para o qual se vai cadastrar suas competências;
- Competências: selecionar as competências que o stakeholder desempenhará no projeto;
- Projeto: selecionar o projeto para o qual serão determinadas as competências necessárias dos stakeholders.

A imagem mostra a interface de usuário do sistema SIGERE (Sistema de Gerenciamento de Requisitos). O título da tela é "Cadastrar Competências dos Stakeholders no Projeto".

Na barra de navegação lateral esquerda, há as seguintes opções:

- Principal
- Cadastro
 - Projeto
 - Requisitos
 - Atividades
 - Stakeholders
 - Cadastro
 - Cadastrar Competência
 - Cadastrar Competência no Projeto
- Competências
- Gerenciamento de Requisitos

Os campos principais da tela são:

- Stakeholder:** Um campo de seleção com um botão "Selecionar" abaixo dele.
- Competências:** Uma lista de seleção com o item "Cliente" selecionado. A lista contém: Analista de Sistemas, Usuário, Cliente, Engenheiro de Software, Engenheiro de Processo, Engenheiro de Requisitos, Analista de Sistemas, Programador, Testador, Usuário.
- Projeto:** Um campo de seleção com o item "Sistema de Gerenciamento de Requisitos" selecionado.

Na base da tela, há botões "Ok" e "Cancelar".

Figura 5.10 - Cadastrar Competências dos Stakeholders no Projeto

Fluxo de Cadastramento e Avaliação da Mudança no sistema:

- Solicitar Mudança
- Identificar os Stakeholders Pertinentes a Mudança
- Analisar a Solicitação de Mudança
- Elaborar a Proposta de Mudança
- Avaliar Impacto da Mudança

Função Solicitar Mudança – Figura 5.11

Esta função permite que os stakeholders do projeto possam solicitar uma mudança nos requisitos. Esta mudança pode significar uma alteração e/ ou exclusão em requisitos existentes e/ou a inclusão de novos requisitos.

Campos da Tela:

- Projeto: selecionar o projeto para o qual a mudança é pertinente;
- Mudança: informar a descrição da mudança;
- Justificativa: informar a justificativa da mudança.

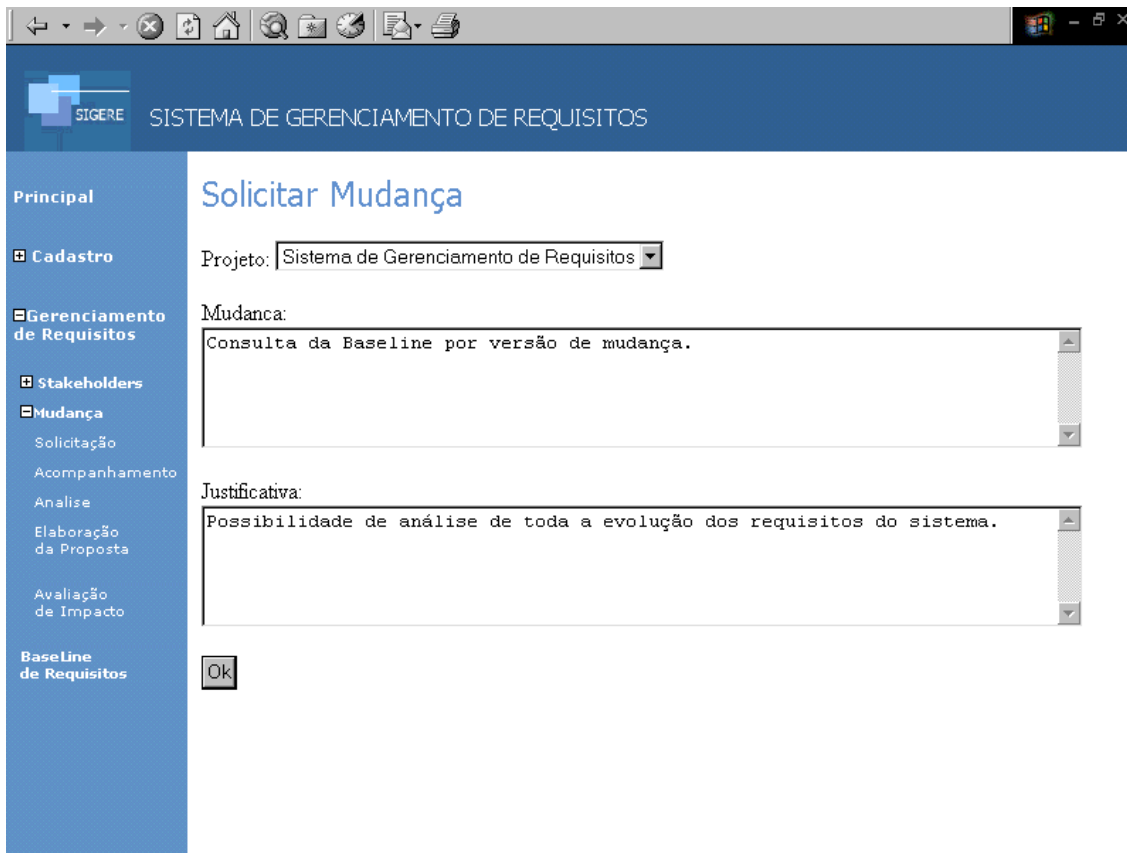


Figura 5.11 – Tela Solicitar Mudança

Ao clicar no botão OK, aparecerá a tela representada pelas figuras 5.12 e 5.13:

Esta tela, parte da função Solicitar Mudança, permite ao stakeholder clicar no botão incluir novo requisito ou selecionar um requisito da baseline para alteração ou exclusão.

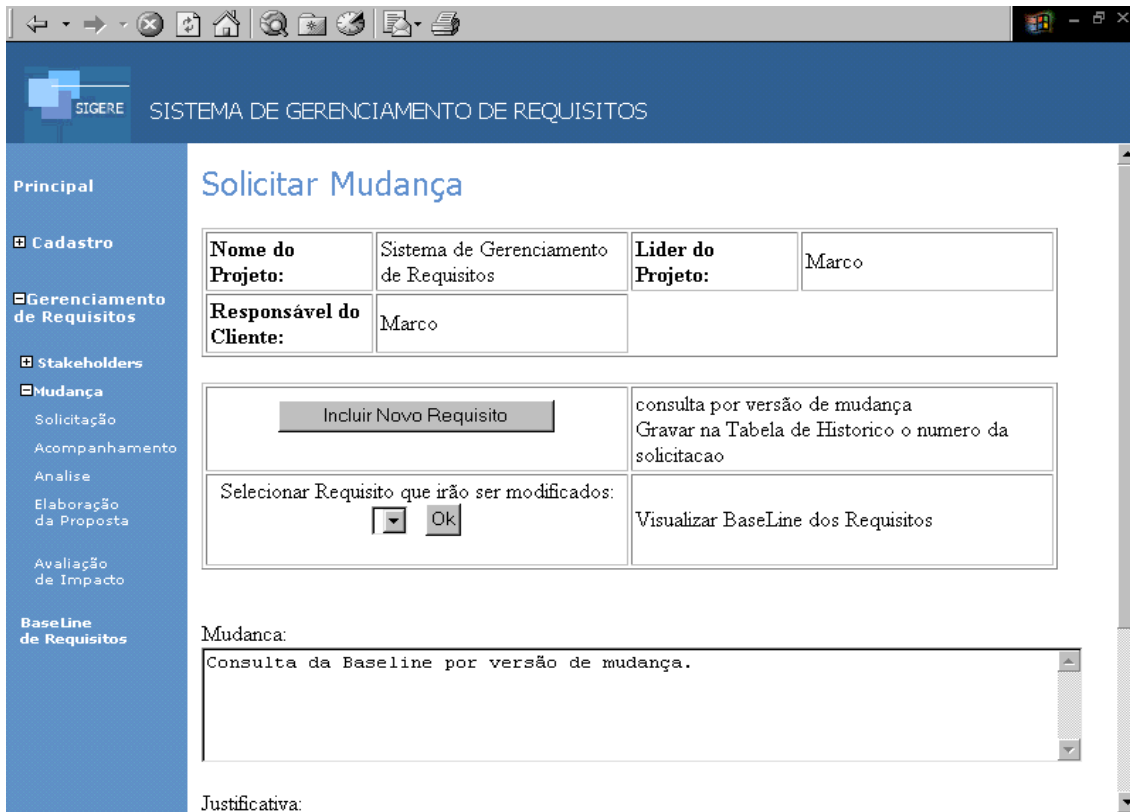


Figura 5.12 – parte inicial da Tela Solicitar Mudança

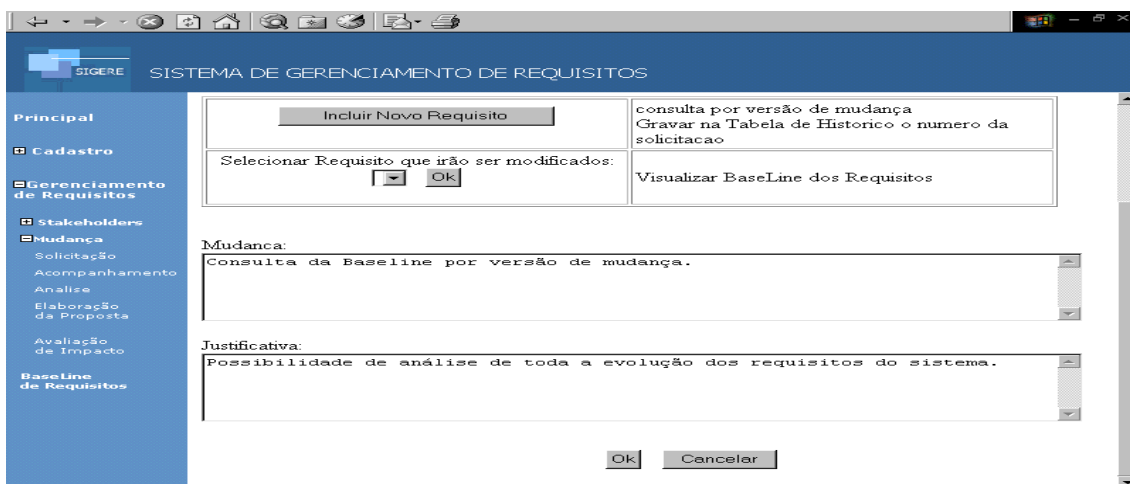


Figura 5.13 – parte final da Tela Solicitar Mudança (após se teclar PgDn)

A tela Cadastrar Requisito da Mudança, figuras 5.14 e 5.15, permite ao stakeholder a inclusão de um novo requisito ao projeto. Nesta tela, deve-se informar os dados do requisito e clicar no botão Cadastrar.

Campos da Tela:

- Nome do Projeto: nome do projeto ao qual pertence o requisito;
- Líder do Projeto: nome do líder técnico do projeto;
- Responsável do Cliente: nome do responsável pelo projeto no cliente;
- Requisito: informar nome do requisito;
- Descrição do Requisito: informar a descrição completa do requisito;
- Problema Identificado: informar o problema que o requisito está solucionando;
- Produto Esperado: informar o produto que se espera obter para atendimento do requisito;
- Aplicação do Requisito: informar aplicação do produto da demanda do requisito;
- Atributos: informar se existem atributos desejados no conteúdo do produto para atendimento do requisito;
 - Restrições: informar se existem restrições de ordem social, política, econômica que delimitam o atendimento ao requisito;
 - Preferências: informar se existem condições desejáveis e particulares do stakeholder quanto ao atendimento do requisito;
 - Expectativas: informar se existem expectativas a serem atendidas no domínio conhecido pelo stakeholder para atendimento do requisito;
 - Qualificação Funcional: selecionar o tipo de qualificação funcional do requisito (operacional, gerencial e estratégico);
 - Área de Origem: selecionar a área de origem do requisito (interna, externa ou de ordem legal);
 - Característica: selecionar a característica do requisito (funcional, usabilidade, confiabilidade, eficiência, portabilidade e manutenibilidade);
 - Tipo do Requisito: selecionar o tipo do requisito (volátil, emergente, normal, global).

SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS

Principal

- Cadastro
- Gerenciamento de Requisitos
- Stakeholders
- Mudança
 - Solicitação
 - Acompanhamento
 - Análise
 - Elaboração da Proposta
 - Avaliação de Impacto
- BaseLine de Requisitos

Cadastrar Requisito da Mudança

Nome do Projeto:	Sistema de Gerenciamento de Requisitos	Lider do Projeto:	Marco
Responsável do Cliente:	Marco		

Requisito:	consulta por versão de mudança	Descrição do Requisito:	O sistema deve prover consulta da baseline, por versão de mudança.
Expectativas:	Facilidade de navegação	Problema Identificado:	Necessidade de consultar a baseline de requisitos por mudança aprovada.
Produto Esperado:	baseline com possibilidade de seleção dos requisitos antes de uma determinada mudança	Aplicação:	rastreabilidade dos requisitos

Figura 5.14 – parte inicial da tela Cadastrar Requisito da Mudança

SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS

Principal

- Cadastro
- Gerenciamento de Requisitos
- Stakeholders
- Mudança
 - Solicitação
 - Acompanhamento
 - Análise
 - Elaboração da Proposta
 - Avaliação de Impacto
- BaseLine de Requisitos

Expectativas:	Facilidade de navegação	Problema Identificado:	Necessidade de consultar a baseline de requisitos por mudança aprovada.
Produto Esperado:	baseline com possibilidade de seleção dos requisitos antes de uma determinada mudança	Aplicação:	rastreabilidade dos requisitos
Atributos:	dados dos requisitos.	Restrições:	grande quantidade de informação
Preferências:	Help de ajuda de utilização da tela		

Figura 5.15 – parte final da tela Cadastrar Requisito da Mudança (após se teclar PgDn)

Ao término do cadastramento de novos requisitos ou a alteração e/ou exclusão dos requisitos já existentes, ao se clicar no botão OK da tela Solicitar Mudança, confirma-se a solicitação de mudança, aparecendo a tela abaixo, figura 5.16:

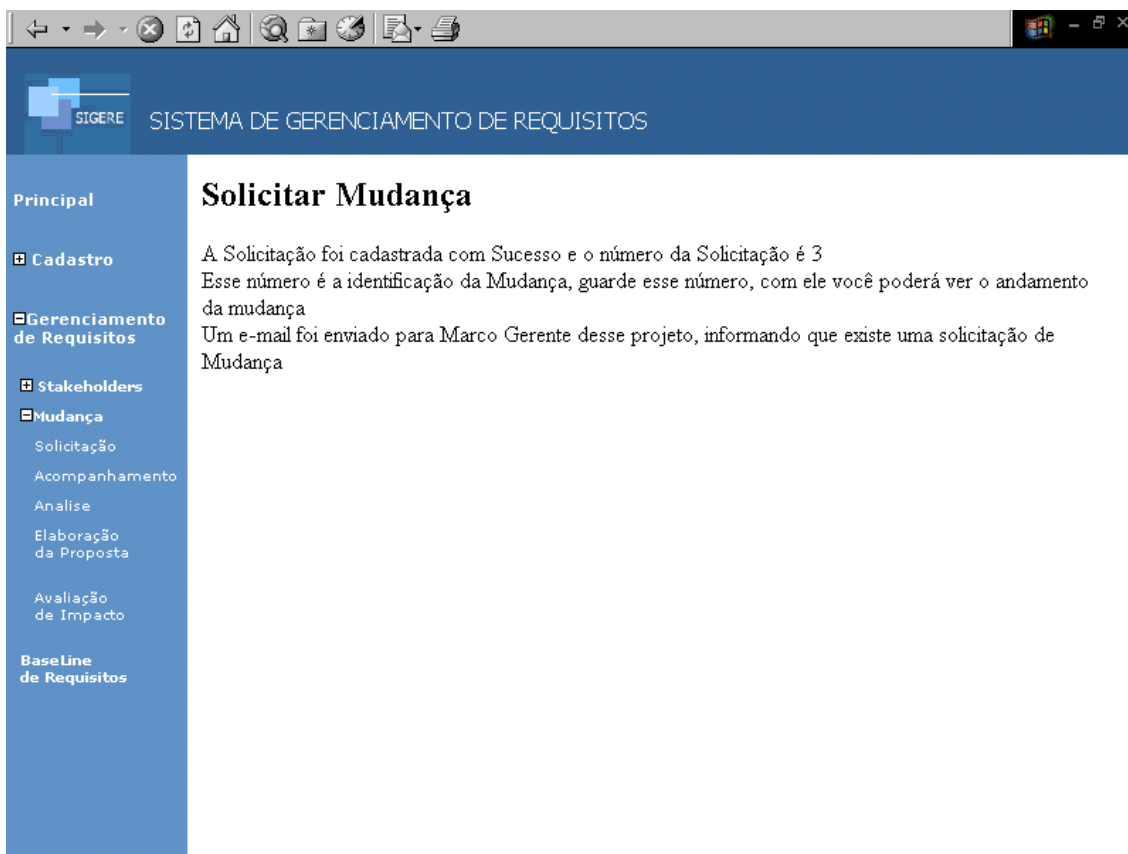


Figura 5.16 – Tela confirmação do cadastramento da Solicitação de Mudança

Função Identificar Stakeholders Pertinentes – Figura 5.17

Esta função permite selecionar os stakeholders que serão responsáveis por analisar a Solicitação da Mudança.

Campos da Tela:

- Projeto: selecionar o projeto que possui uma solicitação de mudança para ser avaliada;
- Mudanças: selecionar a mudança para a qual se deseja indicar os stakeholders que serão responsáveis por avaliá-la.

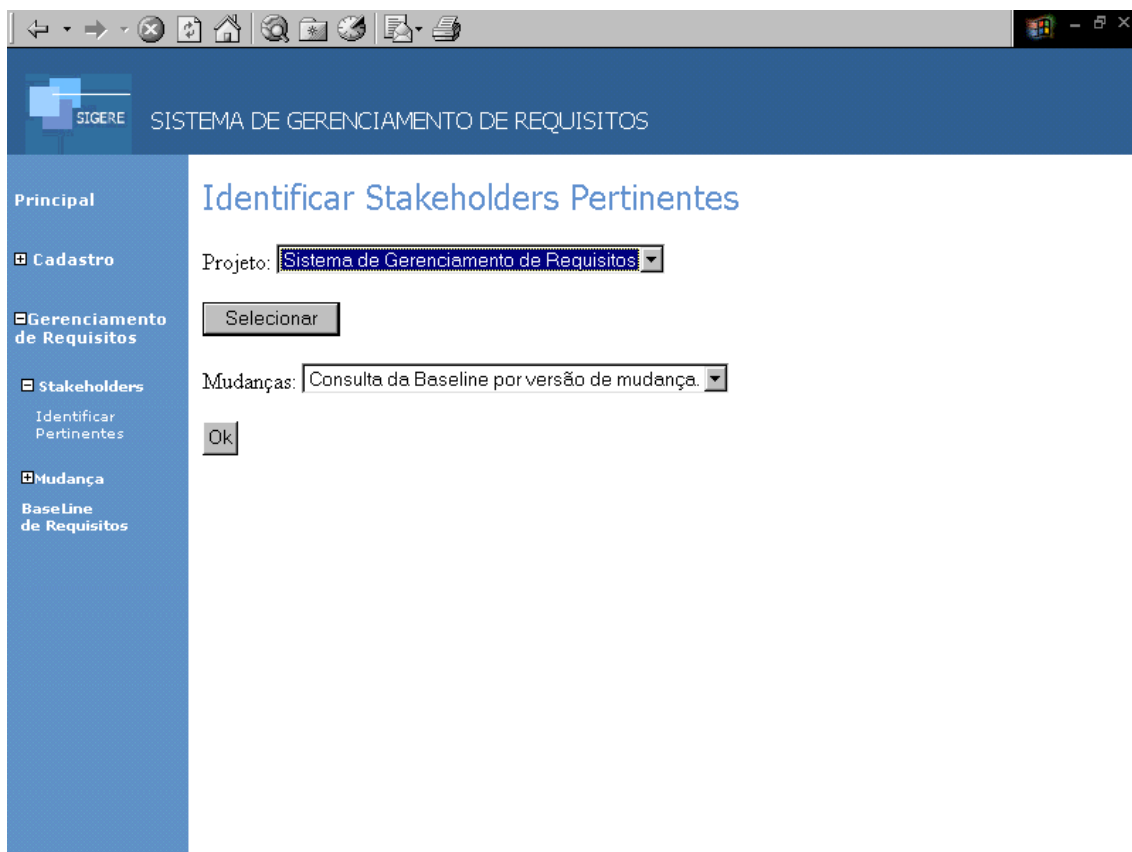


Figura 5.17 – Tela Identificar Stakeholders Pertinentes

Ao clicar no botão OK, aparecerá a tela abaixo que mostra toda a solicitação de mudança, com todos os requisitos que foram alterados, excluídos ou incluídos e no seu final, aparece todos os stakeholders que foram a Fonte de Informação dos Requisitos e aqueles que fazem parte do projeto, porém, não estão envolvidos com nenhum dos requisitos da solicitação de mudança. Nesta tela, no seu final, deverão ser selecionados os stakeholders que serão os responsáveis pela Análise da solicitação de mudança.

Campos da Tela:

No Topo da tela

- Nome do Projeto: nome do projeto ao qual pertence o requisito;
- Líder do Projeto: nome do líder técnico do projeto;
- Responsável do Cliente: nome do responsável pelo projeto no cliente;
- Número da Solicitação: número seqüencial da solicitação da mudança.

SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS

Principal

- Cadastro
- Gerenciamento de Requisitos
- Stakeholders
 - Identificar Pertinentes
- Mudança
 - Baseline de Requisitos

Identificar Stakeholders Pertinentes

Nome do Projeto:	Sistema de Gerenciamento de Requisitos	Número da Solicitação	3
Gerente do Projeto:	Marco		
Responsável do Cliente:	Marco		

Requisito:	consulta por versão de mudança	Stakeholder Responsável pela Mudança	Marco
Descrição do Requisito:	O sistema deve prover consulta da baseline, por versão de mudança.	Tipo de Mudança:	Inclusão

Dependência de Requisitos
 Histórico do Requisito

Atividades Dependentes
 Requisitos Não Dependentes

Informações da Situação Atual do Requisito
[Informações Complementares da Mudança do Requisito](#)

Figura 5.18 – parte inicial (topo) da Tela Identificar Stakeholders Pertinentes

No Corpo da tela (aparece tantas vezes quantos forem os requisitos que fazem parte da Solicitação da Mudança), figura5.19,

- Requisito: informar nome do requisito;
- Descrição do Requisito: informar a descrição completa do requisito;
- Stakeholder Responsável pela Mudança: stakeholder que está propondo a mudança;
- Tipo da Mudança: inclusão, alteração ou exclusão do requisito.
- Mudança: descrição da mudança no requisito;
- Justificativa da Mudança: justificativa da mudança no requisito.

Links que aparecem no corpo da tela, para cada requisito da solicitação de mudança:

- Dependência de Requisitos: este link possibilita ao usuário verificar a dependência deste requisito em relação aos demais requisitos do projeto (ligações do tipo pai-filho);
- Atividades Dependentes: este link possibilita ao usuário verificar as Atividades do Projeto que são dependentes deste requisito;
- Informações da Situação Atual do Requisito: este link possibilita ao usuário verificar todas as informações deste requisito antes da solicitação de mudança;
- Histórico do Requisito: este link possibilita ao usuário verificar as informações históricas deste requisito. É o conceito aplicado da rastreabilidade;
- Requisitos Não Dependentes: possibilita ao usuário verificar informações de requisitos que não tem relação com este requisito;
- Informações Complementares da Mudança do Requisito: este link possibilita ao usuário verificar as demais informações deste requisito na solicitação de mudança (problema identificado, produto esperado,...).

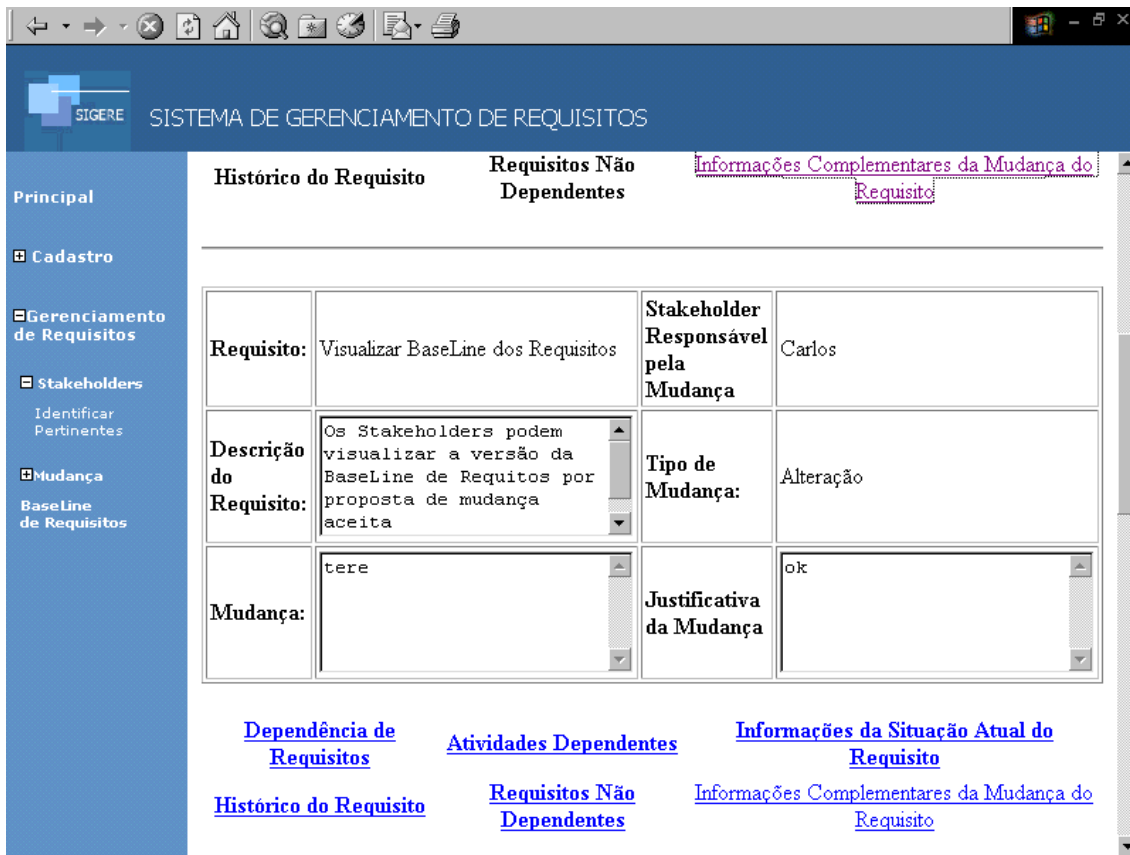


Figura 5.19 – parte intermediária (corpo) da Tela Identificar Stakeholders Pertinentes (após se teclar PgDn)

Na Base da Tela, figura 5.20;

- Stakeholders do Requisito: aparece uma lista com todos os stakeholders que tem relação com algum dos requisitos da solicitação de mudança;
- Outros Stakeholders: aparece uma lista com todos os stakeholders que não tem relação com os requisitos da solicitação de mudança.

Nesta tela o usuário deverá selecionar os stakeholders que serão responsáveis por Analisar a solicitação de mudança.

Link que aparece na base da tela, ao lado do nome dos stakeholders

- Informações: este link possibilita ao usuário verificar todas as informações do stakeholder.

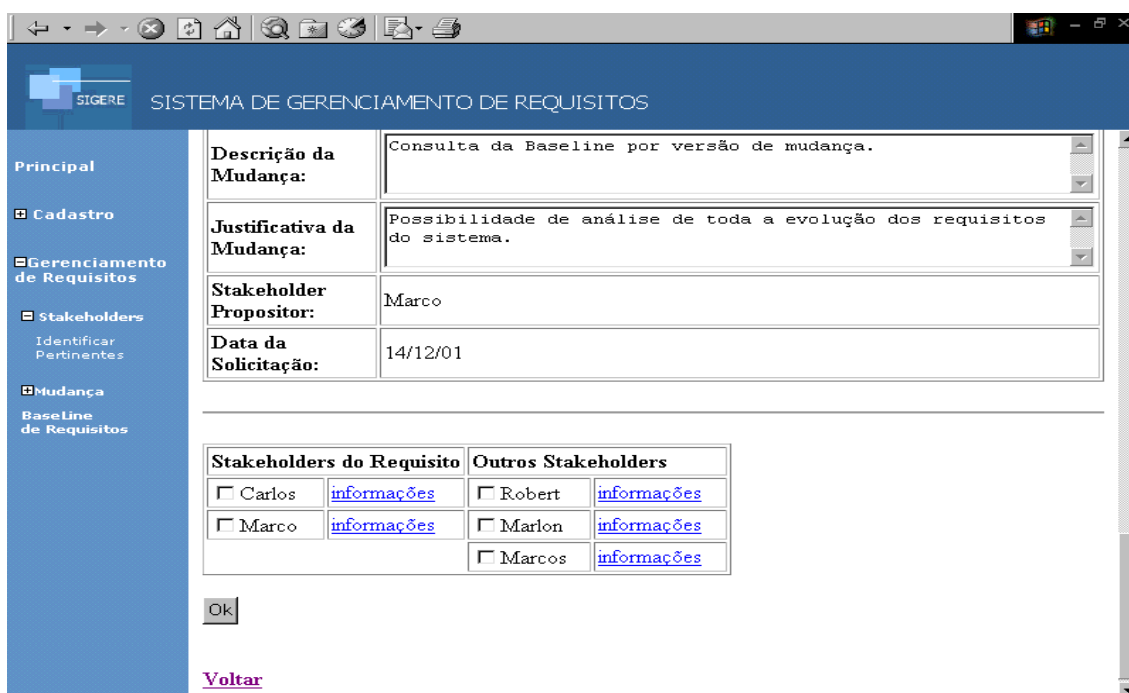


Figura 5.20 – parte final (base) da Tela Identificar Stakeholders Pertinentes (após se teclar PgDn)

Tela de informação dos stakeholders (link) – Figura 5.21

Nome:	Marco	Correio Eletrônico:	cordeiro@pr.gov.br
Área Trabalho:	Desenvolvimento	Sector:	Olimpus
Competências no Projeto		Outras Competências	
		Analista de Sistemas Testador Usuário Engenheiro de Software Engenheiro de Requisitos	

Figura 5.21 – Tela de informação dos stakeholders (link)

Após a confirmação da tela Stakeholders Pertinentes, aparece a tela abaixo, figura 5.22, informando do êxito do cadastramento da função.

SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS

Principal

- Cadastro**
- Gerenciamento de Requisitos**
- Stakeholders**
 - Identificar Pertinentes
- Mudança**
 - BaseLine de Requisitos

Os Stakeholders da mudança foram cadastrados com sucesso.
Um e-mail foi enviado para todos os Stakeholders Selecionados informando que existe uma nova mudança para ser analisada

Figura 5.22 – Tela de confirmação da função Identificar Stakeholders Pertinentes

Função Analisar Solicitação de Mudança – Figura 5.23

Esta função permite aos stakeholders responsáveis procederem a Análise da Solicitação de Mudança.

Campos da Tela:

- Projeto: selecionar o projeto que possui uma solicitação de mudança para ser avaliada;
- Mudanças: selecionar a mudança para a qual se deseja proceder a Análise da Solicitação de Mudança.

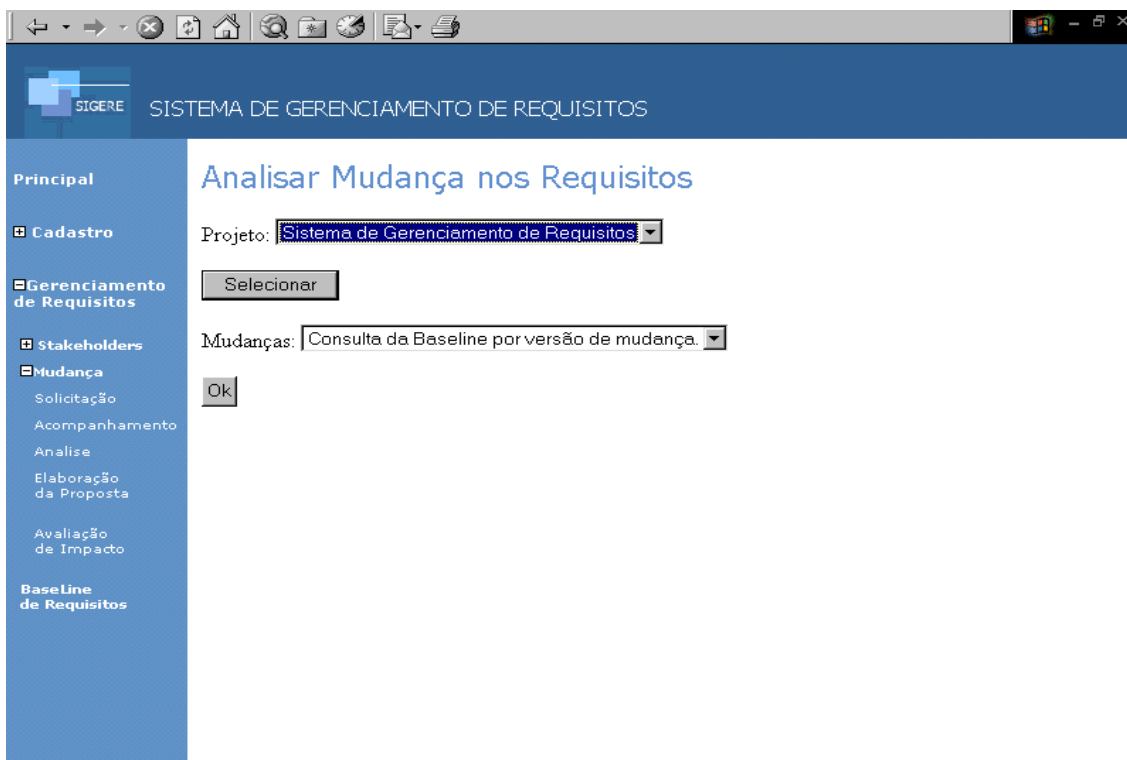


Figura 5.23 – Tela Analisar Mudança nos Requisitos

Ao clicar no botão OK, aparece a tela abaixo, figura 5.24, que mostra toda a solicitação de mudança, com todos os requisitos que foram alterados, excluídos ou incluídos e no seu final, aparece os campos para se informar o parecer da Solicitação de Mudança.

Campos da Tela:

No Topo da tela

- Nome do Projeto: nome do projeto ao qual pertence o requisito;
- Líder do Projeto: nome do líder técnico do projeto;
- Responsável do Cliente: nome do responsável pelo projeto no cliente;
- Número da Solicitação: número seqüencial da solicitação da mudança.

Nome do Projeto:	Sistema de Gerenciamento de Requisitos	Número da Solicitação	3
Gerente do Projeto:	Marco		
Responsável do Cliente:	Marco		

Requisito:	Visualizar BaseLine dos Requisitos	Stakeholder Responsável pela Mudança	Carlos
Descrição do Requisito:	Os Stakeholders podem visualizar a versão da BaseLine de Requisitos por proposta de mudança aceita	Tipo de Mudança:	Alteração
Mudança:	tere	Justificativa da Mudança	ok

Figura 5.24 – parte inicial (topo) da Tela Analisar Mudança nos Requisitos

No Corpo da tela, (aparece tantas vezes quantos forem os requisitos que fazem parte da Solicitação da Mudança), figuras 5.25 e 5.26.

- Requisito: informar nome do requisito;
- Descrição do Requisito: informar a descrição completa do requisito;
- Stakeholder Responsável pela Mudança: stakeholder que está propondo a mudança;
- Tipo da Mudança: inclusão, alteração ou exclusão do requisito;
- Mudança: descrição da mudança no requisito;
- Justificativa da Mudança: justificativa da mudança no requisito.

Links que aparecem no corpo da tela, para cada requisito da solicitação de mudança:

- Dependência de Requisitos: este link possibilita ao usuário verificar a dependência deste requisito em relação aos demais requisitos do projeto (ligações do tipo pai-filho);
- Atividades Dependentes: este link possibilita ao usuário verificar as Atividades do Projeto que são dependentes deste requisito;
- Informações da Situação Atual do Requisito: este link possibilita ao usuário verificar todas as informações deste requisito antes da solicitação de mudança;
- Histórico do Requisito: este link possibilita ao usuário verificar as informações históricas deste requisito. É o conceito aplicado da rastreabilidade;
- Requisitos Não Dependentes: possibilita ao usuário verificar informações de requisitos que não tem relação com este requisito;
- Informações Complementares da Mudança do Requisito: este link possibilita ao usuário verificar as demais informações deste requisito na solicitação de mudança (problema identificado, produto esperado,...).

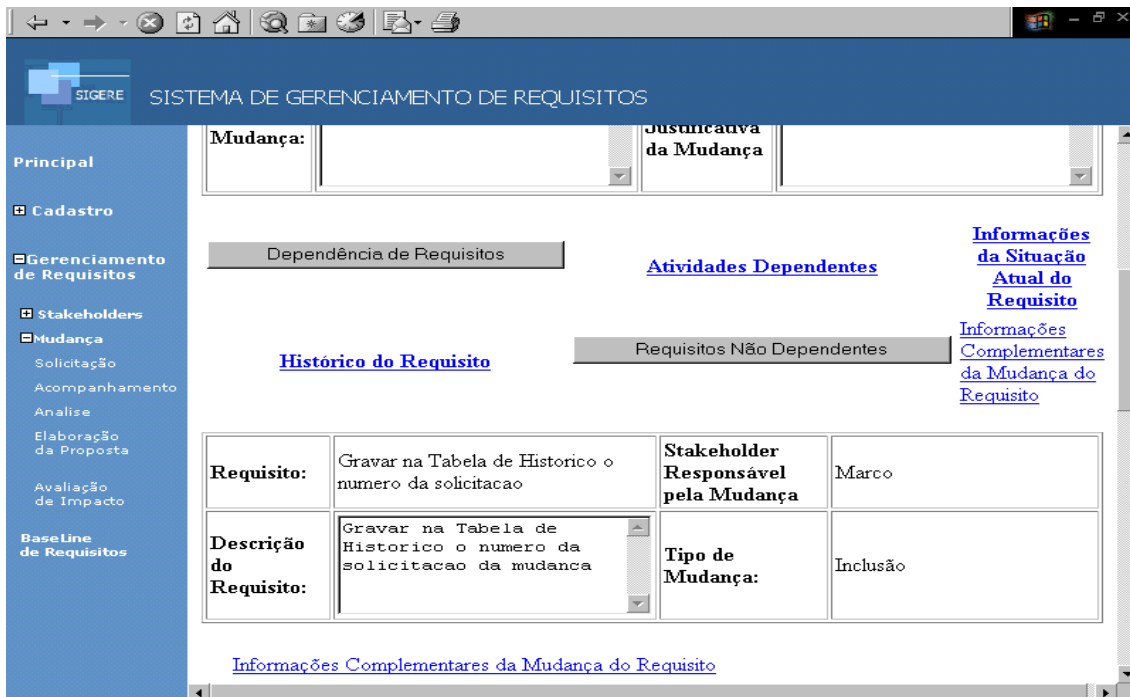


Figura 5.25 – parte intermediária (corpo) da Tela Analisar Mudança nos Requisitos (após se teclar PgDn)

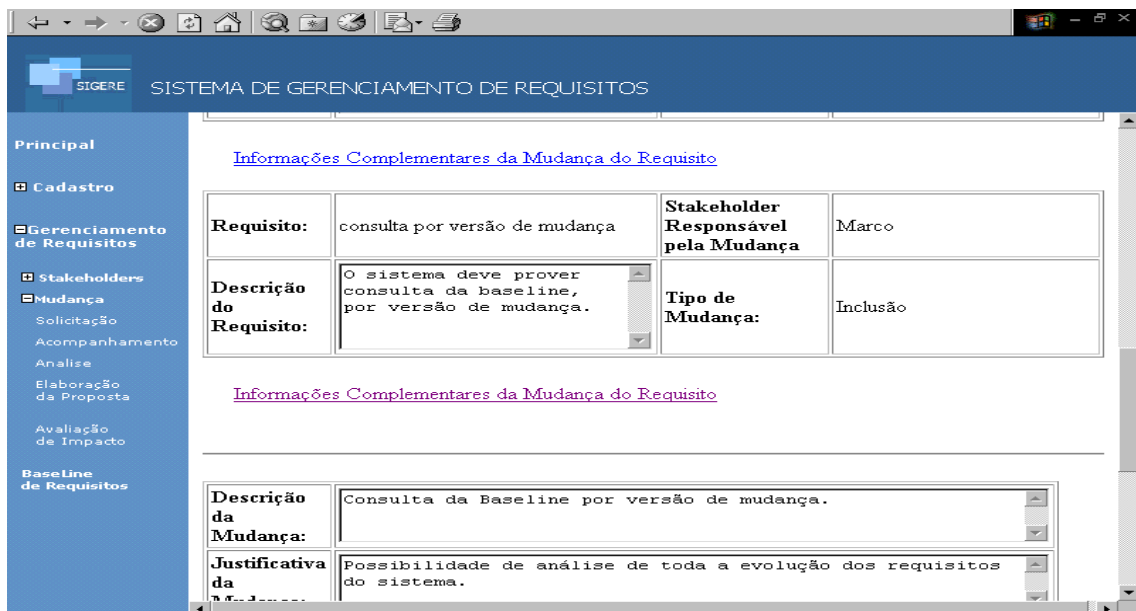


Figura 5.26 – parte intermediária (corpo) da Tela Analisar Mudança nos Requisitos (após se teclar PgDn)

Na Base da Tela, figura 5.27:

- Parecer da Análise: informar o parecer da Análise sobre a Solicitação de Mudança;
- Situação do Parecer da Análise: selecionar a opção desejada (Aceitar, Rejeitar ou Rejeitar Parcial).

A situação “Rejeitar Parcial” indica que uma parte da mudança foi aprovada enquanto outra foi reprovada.

The screenshot displays the 'SISTEMA DE GERENCIAMENTO DE REQUISITOS' interface. The left sidebar contains a navigation menu with the following items: Principal, Cadastro, Gerenciamento de Requisitos, Stakeholders, Mudança (with sub-items: Solicitação, Acompanhamento, Análise, and Elaboração da Proposta), Avaliação de Impacto, and BaseLine de Requisitos. The main content area shows a form for 'Análise de Mudança' with the following fields: 'Justificativa da Mudança' (text area with 'Possibilidade de análise de toda a evolução dos requisitos do sistema.'), 'Stakeholder Propositor' (text field with 'Marco'), and 'Data da Solicitação' (text field with '14/12/01'). Below these fields is a button labeled 'Incluir Novo Requisito'. Further down is a 'Parecer da Análise' section with a dropdown menu currently showing 'OK'. At the bottom, there are three radio buttons: 'Aceitar' (selected), 'Rejeitar', and 'Rejeitar Parcial'. Below the radio buttons are 'Ok' and 'Cancelar' buttons.

Figura 5.27 – parte final (base) da Tela Analisar Mudança nos Requisitos (após se teclar PgDn)

Após se clicar no botão OK, aparecerá a tela abaixo, figura 5.28, indicando que a Análise da Mudança foi realizada com sucesso.

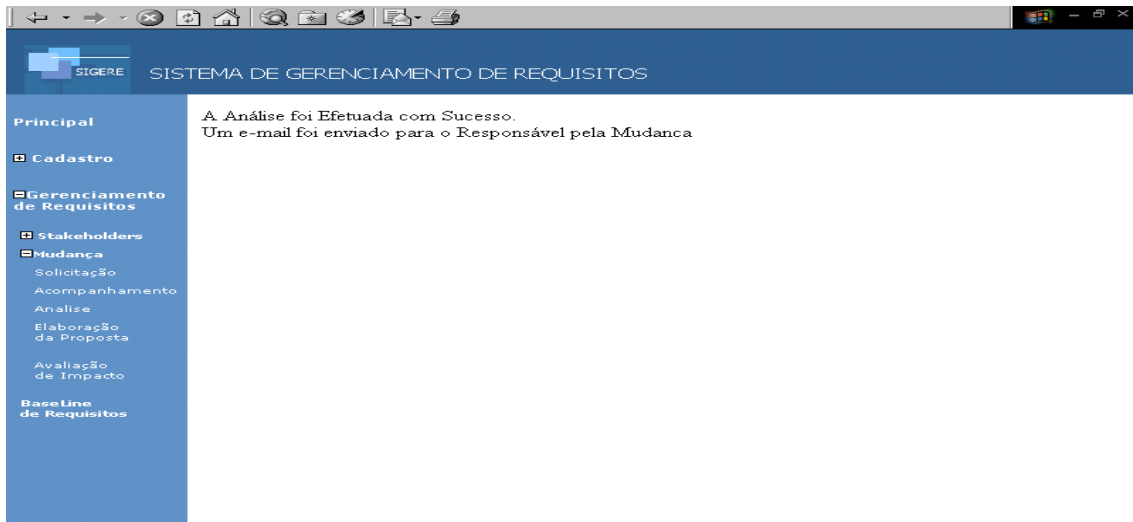


Figura 5.28 – confirmação da Função Analisar Mudança nos Requisitos

Função Elaborar Proposta de Mudança – Figura 5.29

Esta função permite ao responsável pela atividade que proceda a Elaboração da Proposta de Mudança. Esta atividade consiste em descrever os requisitos tecnicamente de forma adequada. Recomenda-se que esta atividade seja executada por um Engenheiro de Requisitos ou um Engenheiro de Software.

Campos da Tela

- Projeto: selecionar o projeto que possui uma solicitação de mudança Analisada para a qual será elaborada uma proposta de mudança;
- Mudanças: selecionar a mudança para a qual se deseja proceder a Elaboração de Proposta de Mudança.

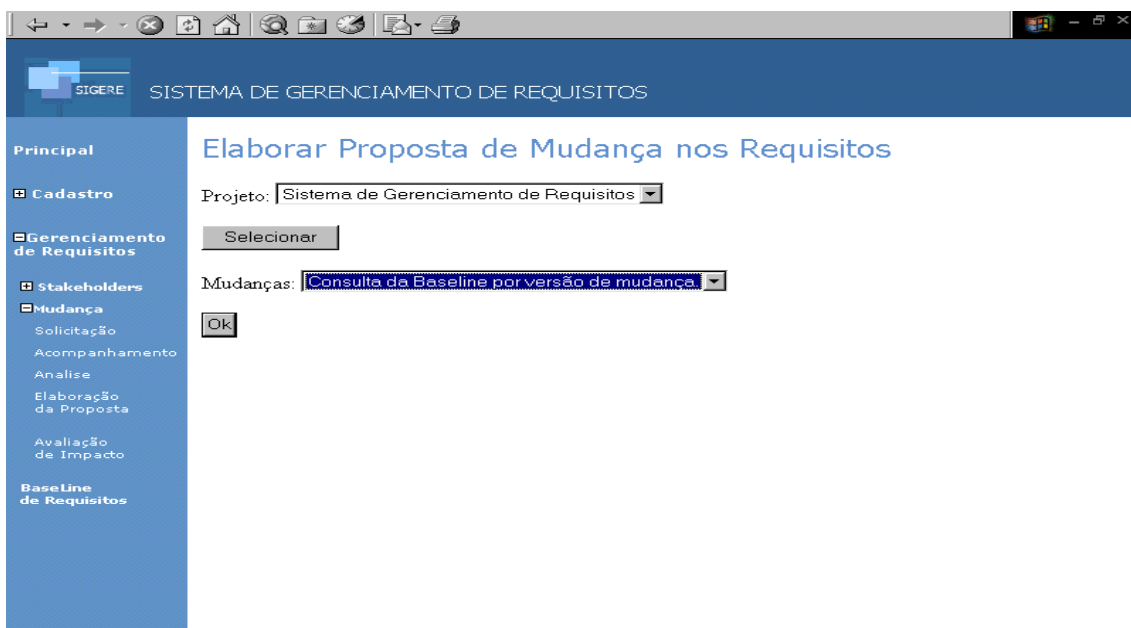


Figura 5.29 – Tela Elaborar Proposta de Mudança nos Requisitos

Ao clicar no botão OK, aparece a tela abaixo, figuras 5.30, 5.31, 5.32 e 5.33, que mostra toda a solicitação de mudança Analisada, com todos os requisitos que foram alterados, excluídos ou incluídos e no seu final, aparece os campos de informações da Elaboração da Proposta.

Campos da Tela:

No Topo da tela

- Nome do Projeto: nome do projeto ao qual pertence o requisito;
- Líder do Projeto: nome do líder técnico do projeto;
- Responsável do Cliente: nome do responsável pelo projeto no cliente;
- Número da Solicitação: número sequencial da solicitação da mudança.

SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS

Elaborar Proposta de Mudança nos Requisitos

Nome do Projeto:	Sistema de Gerenciamento de Requisitos	Número da Solicitação
Gerente do Projeto:	Marco	3
Responsável do Cliente:	Marco	

Requisito:	consulta por versão de mudança	Stakeholder Responsável pela Mudança	Marco
Descrição do Requisito:	O sistema deve prover consulta da baseline, por versão de mudança.	Tipo de Mudança:	Inclusão
Status da Análise	Ácito	Parecer da Análise	

Dependência de

Figura 5.30 – parte inicial (topo) da Tela Elaborar Proposta de Mudança nos Requisitos

No Corpo da tela (aparece tantas vezes quantos forem os requisitos que fazem parte da Solicitação da Mudança), figuras 5.31 e 5.32:

- Requisito: informar nome do requisito;
- Descrição do Requisito: informar a descrição completa do requisito;
- Stakeholder Responsável pela Mudança: stakeholder que está propondo a mudança;
- Tipo da Mudança: inclusão, alteração ou exclusão do requisito;
- Mudança: descrição da mudança no requisito;
- Justificativa da Mudança: justificativa da mudança no requisito.
- Status da Análise: indica se o requisito foi aceito ou rejeitado na Análise da Solicitação da Mudança;
- Parecer da Análise: parecer da Análise sobre a mudança no requisito.

Links que aparecem no corpo da tela, para cada requisito da solicitação de mudança:

- Dependência de Requisitos: este link possibilita ao usuário verificar a dependência deste requisito em relação aos demais requisitos do projeto (ligações do tipo pai-filho);
 - Atividades Dependentes: este link possibilita ao usuário verificar as Atividades do Projeto que são dependentes deste requisito;
 - Informações da Situação Atual do Requisito: este link possibilita ao usuário verificar todas as informações deste requisito antes da solicitação de mudança;
 - Histórico do Requisito: este link possibilita ao usuário verificar as informações históricas deste requisito. É o conceito aplicado da rastreabilidade;
 - Requisitos Não Dependentes: possibilita ao usuário verificar informações de requisitos que não tem relação com este requisito;
 - Informações Complementares da Mudança do Requisito: este link possibilita ao usuário verificar as demais informações deste requisito na solicitação de mudança (problema identificado, produto esperado,...).

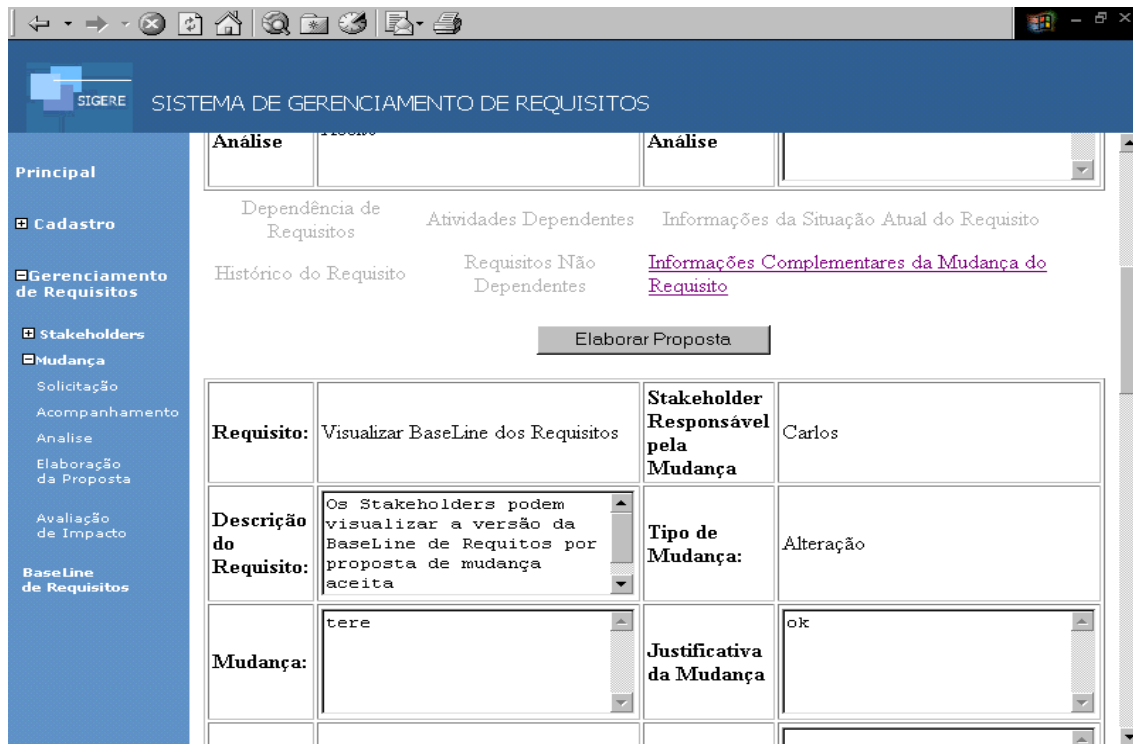


Figura 5.31 – parte intermediária (corpo) da Tela Elaborar Proposta de Mudança nos Requisitos (após se teclar PgDn)

Na Base da Tela, figuras 5.32 e 5.33:

- Mudança: descrição da mudança no requisito;
- Justificativa da Mudança: justificativa da mudança no requisito.
- Stakeholder Propositor: stakeholder que propôs a mudança;
- Data da Solicitação: data em que a Solicitação da Mudança foi cadastrada;
- Observação: deve ser utilizado pelo responsável pela Elaboração da Proposta para fazer alguma observação que considere necessária;
 - Custo/Esforço da Mudança: informar, no sistema métrico da organização (PF, LOC,...), a quantidade de esforço para implementar a Mudança;
 - Tempo para a Mudança: informar a quantidade de tempo necessário para implementar a Mudança;
 - Recursos para a Mudança: informar a quantidade de recursos necessários para implementar a Mudança no Tempo previsto.

The screenshot shows the 'SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS' interface. The left sidebar contains a navigation menu with options like 'Principal', 'Cadastro', 'Gerenciamento de Requisitos', 'Stakeholders', 'Mudança', 'Solicitação', 'Acompanhamento', 'Análise', 'Elaboração da Proposta', 'Avaliação de Impacto', and 'BaseLine de Requisitos'. The main content area displays the following information:

Status da Análise	Áceito	Parecer da Análise	
Dependência de Requisitos		Atividades Dependentes	Informações da Situação Atual do Requisito
Histórico do Requisito		Requisitos Não Dependentes	Informações Complementares da Mudança do Requisito
Descrição da Mudança:	Consulta da Baseline por versão de mudança.		
Justificativa da Mudança:	Possibilidade de análise de toda a evolução dos requisitos do sistema.		
Stakeholder Propositor:	Marco		
Data da Solicitação:	14/12/01		

Figura 5.32 – parte intermediária e final da Tela Elaborar Proposta de Mudança nos Requisitos (após se teclar PgDn)

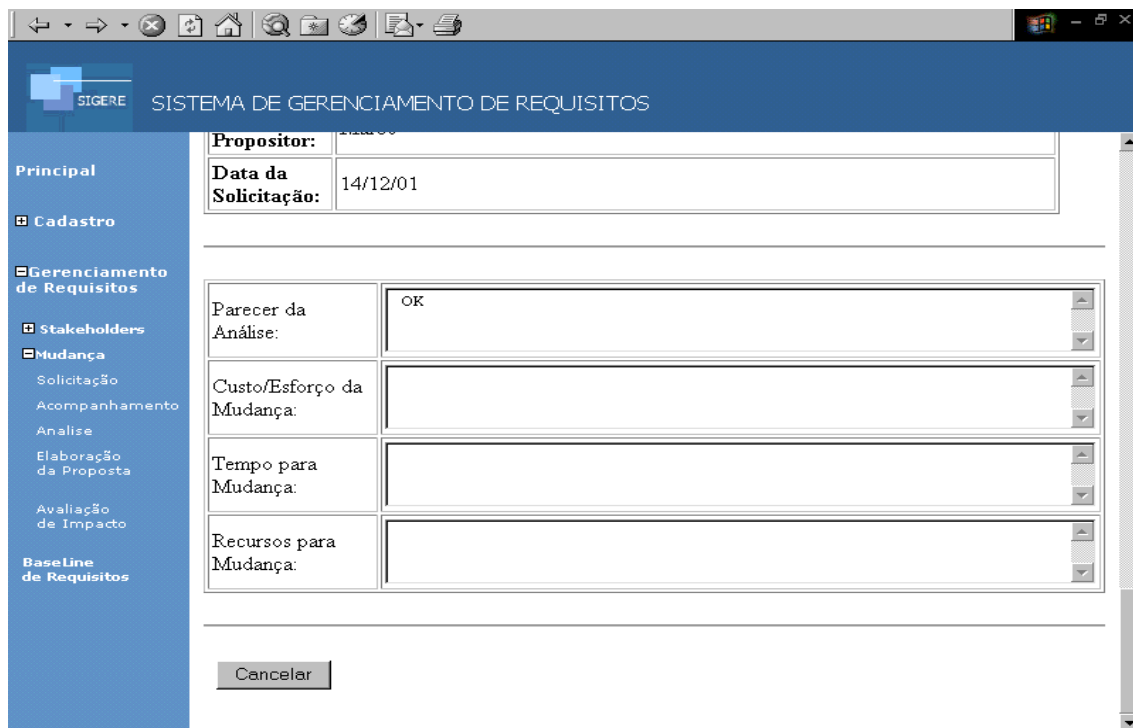


Figura 5.33 – parte final (base) da Tela Elaborar Proposta de Mudança nos Requisitos (após se teclar PgDn)

Após se clicar no botão OK, aparecerá a tela abaixo, figura 5.34, indicando que a Elaboração da Proposta da Mudança foi realizada com sucesso.

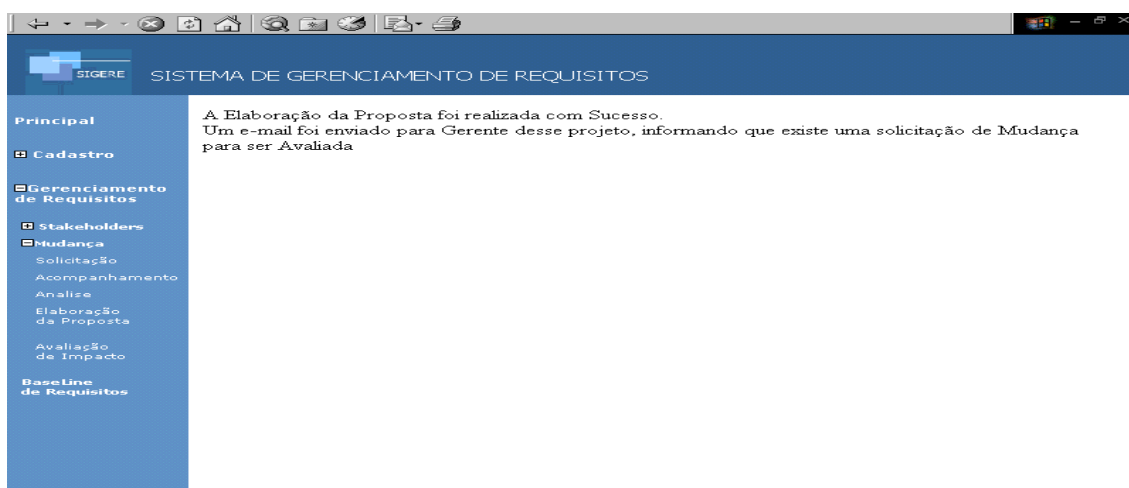


Figura 5.34 – parte final (base) da Tela Elaborar Proposta de Mudança nos Requisitos

Função Avaliar Impacto da Mudança – Figura 5.35

Esta função permite ao Líder de Projeto proceder a Avaliação de Impacto da Mudança. Neste ponto, a mudança pode ser aprovada para implementação ou rejeitada.

Campos da Tela

- Projeto: selecionar o projeto que possui uma Proposta de mudança para ser avaliada;
- Mudanças: selecionar a mudança para a qual se deseja proceder a Avaliação de Impacto da Mudança.

The screenshot shows a web browser window displaying the SIGERE (Sistema de Gerenciamento de Requisitos) application. The page title is 'Avaliar Impacto das Mudanças nos Requisitos'. On the left, there is a blue sidebar menu with the following items: 'Principal', 'Cadastro', 'Gerenciamento de Requisitos', 'Stakeholders', 'Mudança' (with sub-items: 'Solicitação', 'Acompanhamento', 'Análise', 'Elaboração da Proposta', 'Avaliação de Impacto'), and 'BaseLine de Requisitos'. The main content area contains two dropdown menus: 'Projeto:' with the value 'Sistema de Gerenciamento de Requisitos' and a 'Selecionar' button below it; and 'Mudanças:' with the value 'Consulta da Baseline por versão de mudança' and an 'Ok' button below it. The browser's address bar and toolbar are visible at the top.

Figura 5.35 –Tela Avaliar Impacto da Mudança nos Requisitos

Ao clicar no botão OK, aparece a tela abaixo, figuras 5.36, 5.37, 5.38, 5.39 e 5.40, que mostra toda a Proposta de Mudança, com todos os requisitos que foram alterados, excluídos ou incluídos e no seu final, aparece os campos para se informar o parecer da Avaliação de Impacto.

Campos da Tela:

No Topo da tela, figura 5.36:

- Nome do Projeto: nome do projeto ao qual pertence o requisito;
- Líder do Projeto: nome do líder técnico do projeto;
- Responsável do Cliente: nome do responsável pelo projeto no cliente;
- Número da Solicitação: número sequencial da solicitação da mudança.

SIGERE SISTEMA DE GERENCIAMENTO DE REQUISITOS

Avaliar Impacto das Mudanças nos Requisitos

Nome do Projeto:	Sistema de Gerenciamento de Requisitos	Número da Solicitação
Gerente do Projeto:	Marco	3
Responsável do Cliente:	Marco	

Requisito:	consulta por versão de mudança	Stakeholder Responsável pela Mudança	Marco
Descrição do Requisito:	O sistema deve prover consulta da baseline, por versão de mudança.	Tipo de Mudança:	Inclusão
Status da Análise	Aceito	Parecer da Análise	

Dependência de

Figura 5.36 – parte inicial (topo) da Tela Avaliar Impacto da Mudança nos Requisitos

No Corpo da tela (aparece tantas vezes quantos forem os requisitos que fazem parte da Solicitação da Mudança), figuras 5.37 e 5.38:

- Requisito: informar nome do requisito;
- Descrição do Requisito: informar a descrição completa do requisito;
- Stakeholder Responsável pela Mudança: stakeholder que está propondo a mudança;
- Tipo da Mudança: inclusão, alteração ou exclusão do requisito;
- Mudança: descrição da mudança no requisito;
- Justificativa da Mudança: justificativa da mudança no requisito.
- Status da Análise: indica se o requisito foi aceito ou rejeitado na Análise da Solicitação da Mudança;
- Parecer da Análise: parecer da Análise sobre a mudança no requisito.

Links que aparecem no corpo da tela, para cada requisito da solicitação de mudança:

- Dependência de Requisitos: este link possibilita ao usuário verificar a dependência deste requisito em relação aos demais requisitos do projeto (ligações do tipo pai-filho);
 - Atividades Dependentes: este link possibilita ao usuário verificar as Atividades do Projeto que são dependentes deste requisito;
 - Informações da Situação Atual do Requisito: este link possibilita ao usuário verificar todas as informações deste requisito antes da solicitação da mudança;
 - Histórico do Requisito: este link possibilita ao usuário verificar as informações históricas deste requisito. É o conceito aplicado da rastreabilidade;
 - Requisitos Não Dependentes: possibilita ao usuário verificar informações de requisitos que não tem relação com este requisito;
 - Informações Complementares da Mudança do Requisito: este link possibilita ao usuário verificar as demais informações deste requisito na solicitação de mudança (problema identificado, produto esperado,...).

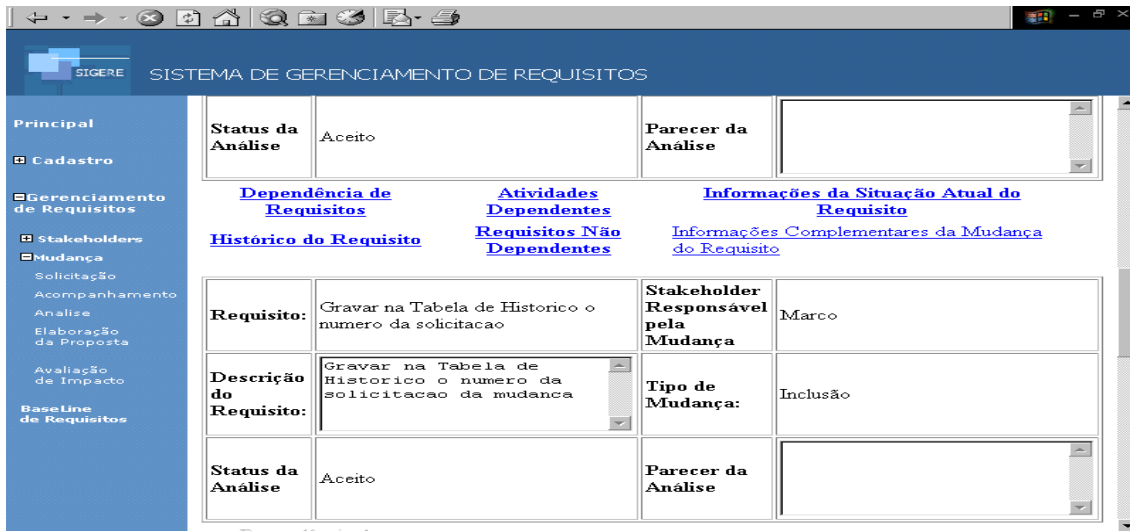


Figura 5.37 – parte intermediária (corpo) da Tela Avaliar Impacto da Mudança nos Requisitos (após se teclar PgDn)

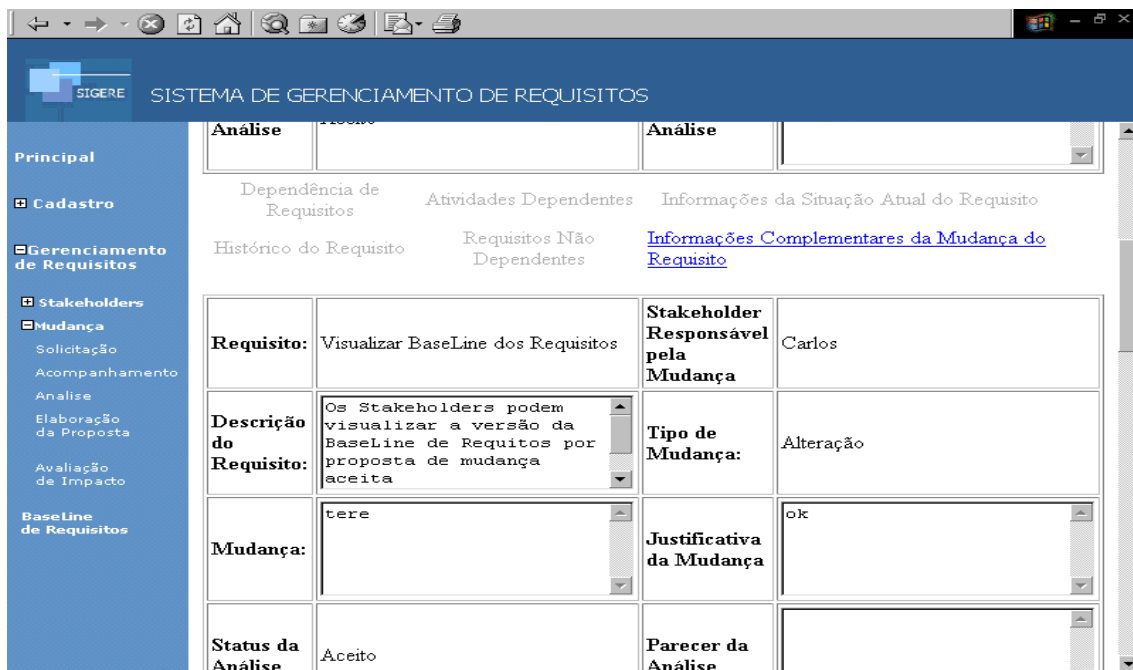


Figura 5.38 – parte intermediária (corpo) da Tela Avaliar Impacto da Mudança nos Requisitos (após se teclar PgDn)

Na Base da Tela, figura 5.39 e 5.40

- Mudança: descrição da mudança no requisito;
- Justificativa da Mudança: justificativa da mudança no requisito.
- Stakeholder Propositor: stakeholder que propôs a mudança;
- Data da Solicitação: data em que a Solicitação da Mudança foi cadastrada;
- Custo/Esforço da Mudança: a quantidade de esforço necessária para implementar a Mudança;
- Tempo para a Mudança: a quantidade de tempo necessária para implementar a Mudança;
- Recursos para a Mudança: a quantidade de recursos necessária para implementar a Mudança no Tempo previsto;
- Parecer da Avaliação de Impacto: informar o parecer da Avaliação de Impacto da Mudança;
- Situação da Avaliação de Impacto: selecionar a opção desejada (Aceitar ou Rejeitar).

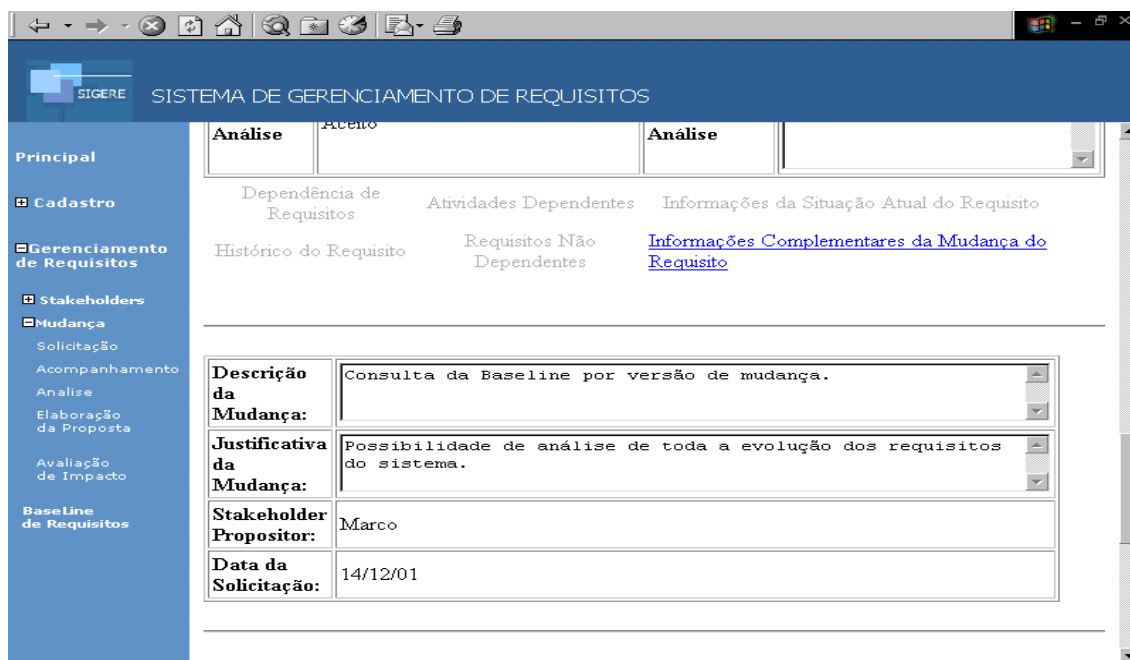


Figura 5.39 – parte final (base) da Tela Avaliar Impacto da Mudança nos Requisitos (após se teclar PgDn)

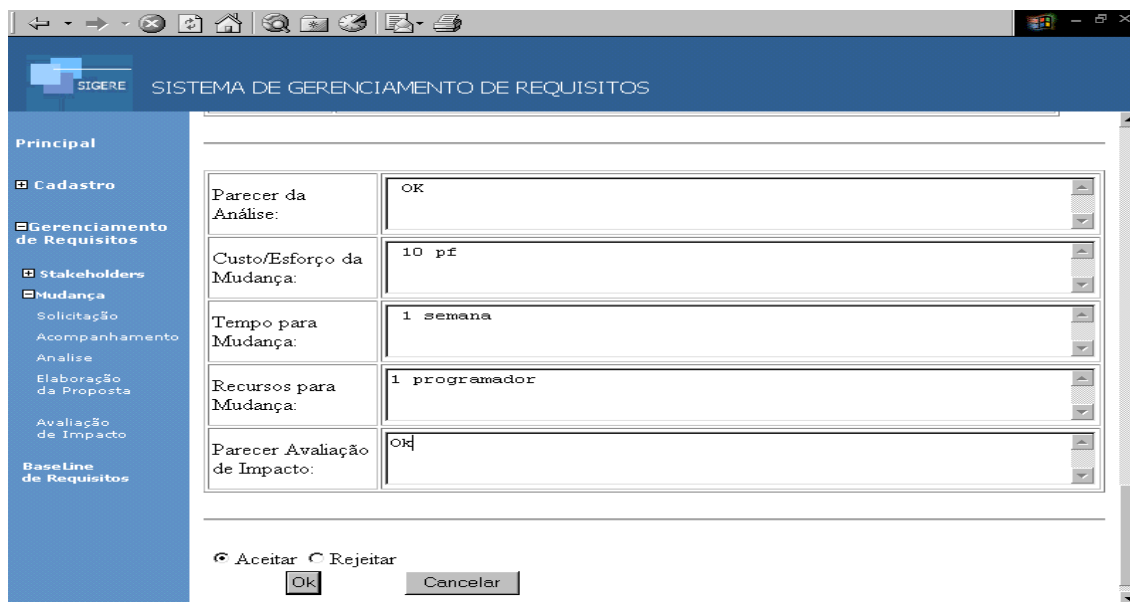


Figura 5.40 – parte final (base) da Tela Avaliar Impacto da Mudança nos Requisitos (após se teclar PgDn)

Após se clicar no botão OK, aparece a tela abaixo, figura 5.41, indicando que a atividade de Avaliação de Impacto foi realizada com sucesso.

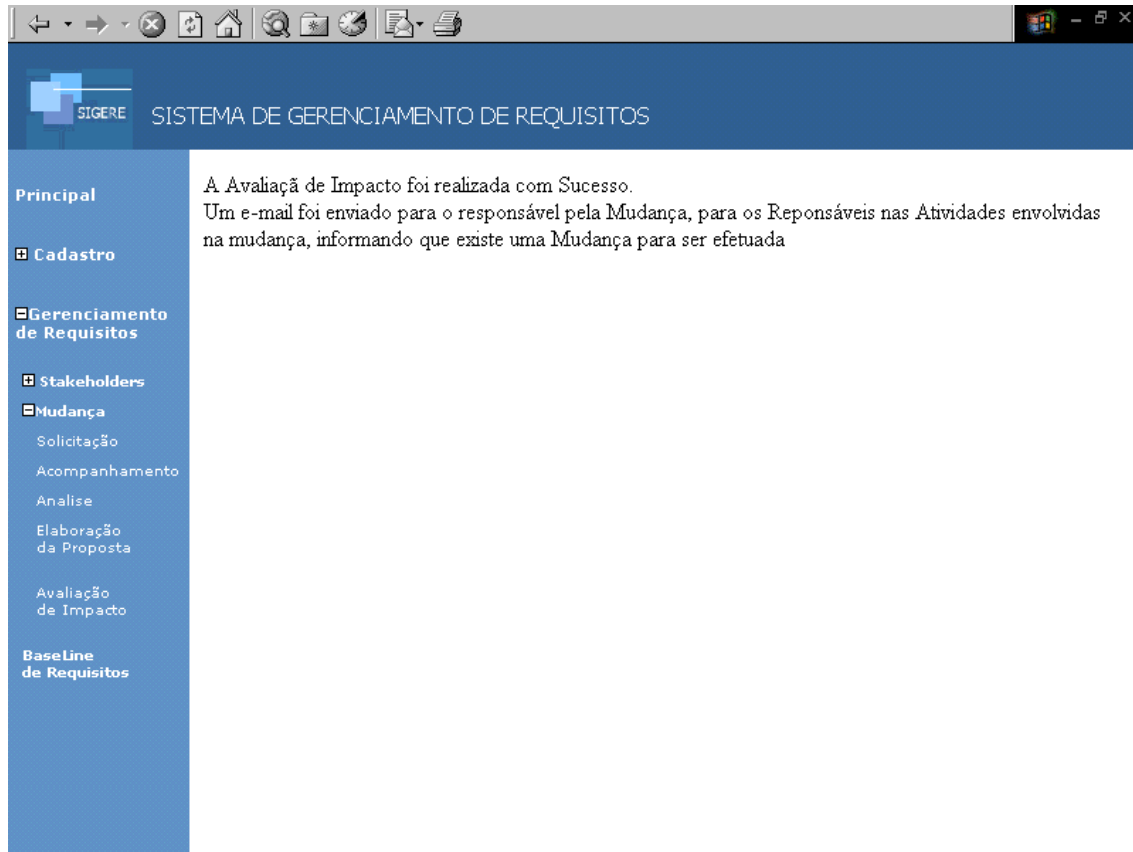


Figura 5.41 – confirmação da Função Avaliar Impacto da Mudança nos Requisitos

5.4 Resumo

Neste capítulo foi descrita a Ferramenta SIGERE que dá suporte completo ao modelo de GR, e suas funcionalidades. No próximo capítulo, apresentamos a aplicação prática do modelo.

CAPÍTULO 6

Aplicação Prática da Modelo de Gerenciamento de Requisitos

6.1	INTRODUÇÃO	161
6.2	PLANEJAMENTO	161
6.3	PROCESSO APLICATIVO	163
6.4	TÉCNICAS APLICADAS.....	170
6.5	RESULTADO DO PROCESSO DE GERENCIAMENTO DE REQUISITOS	170
6.6	RESUMO	171

6 Aplicação Prática da Modelo de Gerenciamento de Requisitos

Nesta etapa da dissertação estão documentados os processos e os produtos resultantes da aplicação prática do modelo proposto de Gerenciamento de Requisitos. Além do relatório de acompanhamento faz-se uma análise dos resultados obtidos.

6.1 Introdução

A aplicação prática do modelo de Gerenciamento de Requisitos abrange o próprio ambiente em que a ferramenta de GR foi desenvolvida.

O relato da aplicação prática compreende o detalhamento do planejamento das atividades, o desenvolvimento do processo aplicativo, o detalhamento e acompanhamento do processo e os resultados obtidos.

O planejamento das atividades detalha a seleção do grupo de trabalho a ser envolvido no processo de aplicação do modelo.

Todo o trabalho é relatado em termos de técnicas aplicadas e feito um registro do acompanhamento do processo aplicativo.

Ao final, é descrito como foi a obtenção dos resultados da aplicação do modelo proposto de Gerenciamento de Requisitos.

6.2 Planejamento

Para início dos trabalhos foram identificados os stakeholders que seriam responsáveis pela execução das atividades do processo.

Como o processo de Gerenciamento de Requisitos foi modelado para ser utilizado no laboratório Olympus da PUCPR, um centro de pesquisa em Engenharia de Software, foram selecionados dois stakeholders deste local para atuarem, um no papel de usuário e outro no papel de responsável do cliente.

O autor ficou responsável por atuar nos papéis de Engenheiro de Requisitos e Líder de Projeto.

As atividades de execução do trabalho foram definidas a partir do objetivo expresso de validar o modelo de GR proposto.

O planejamento das atividades para aplicação do modelo de GR compreendeu os seguintes tópicos:

- identificação dos stakeholders que atuariam no processo;
- agenda de reunião conjunta do pessoal do laboratório Olympus para desenvolvimento dos trabalhos;
- distribuição de material do modelo proposto;
- treinamento dos stakeholders no preenchimento do documento de requisitos;
- treinamento dos stakeholders na utilização de checklist;
- apresentação aos stakeholders do modelo proposto de GR e da ferramenta SIGERE;
- distribuição da versão 001 da Baseline de Requisitos do sistema;
- elaboração em conjunto de uma Solicitação de Mudança;
- aplicação de checklist na atividade de Análise da Solicitação de Mudança;
- aplicação de checklist na atividade de Elaboração da Proposta de Mudança;
- conclusão da aplicação do processo de GR;
- relato de experiência e resultados obtidos.

A forma de participação do grupo compreendeu duas fases principais: na 1ª fase (uma semana), ocorreu o treinamento no preenchimento do documento de requisitos e utilização de checklist, além da apresentação do modelo de GR proposto e da ferramenta de suporte ao processo, denominada SIGERE; na 2ª fase (uma semana) executamos as atividades do modelo proposto do processo de GR.

A contratação de participação das pessoas selecionadas foi feita através de duas maneiras: primeiro, através de conversa individual com o coordenador do laboratório Olympus, explicando os motivos e objetivos do trabalho e segundo, em uma reunião em grupo com todos os participantes do processo.

6.3 Processo Aplicativo

A aplicação prática do modelo de GR iniciou com a definição do grupo de trabalho que ficou estabelecido da seguinte forma:

- 1 usuário (laboratório Olympus);
- 1 responsável do cliente (laboratório Olympus);
- 1 líder de projeto e engenheiro de requisitos (o autor).

Após a definição do grupo de trabalho, todo o processo foi realizado em conjunto. Iniciou com a distribuição do documento de requisitos da versão 001 da Baseline de Requisitos do sistema SIGERE, conforme item 5.2-Requisitos do Sistema na página 108.

Em seguida, foi feito um Brainstorming para levantar o conteúdo da Solicitação de Mudança que seria objeto de estudo. Desta atividade, como resultado, obtivemos a Solicitação de Mudança 001, composta pelos seguintes requisitos:

- alteração do requisito 005, Consultar a Baseline de Requisitos, de forma a eliminar a ambigüidade de entendimento da atual descrição que consta na tabela 5.5, de forma a explicitar que os requisitos vigentes e não vigentes poderiam ser consultados, conforme nova descrição do requisito na tabela 6.1;
- inclusão do requisito 009, Bloquear a Atualização de Solicitação de Mudança Finalizada, conforme descrito na tabela 6.2.

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 005
2. Nome do Requisito: Consultar a Baseline de Requisitos
3. Responsável pela Informação: Marlon_____ Área: negócio___ data: 01/09/2001
4. Característica do Requisito: 1_____ (1 – funcional, 2 – não funcional)
5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):

_ Os stakeholders do projeto querem consultar a baseline de requisitos.

9. Problema Identificado:

_ Existe a necessidade de se consultar os requisitos **vigentes e não vigentes** de um sistema.

10. Produto Esperado:

_ Informação de todos os requisitos do sistema, **vigentes e não vigentes**.

11. Aplicação:

_ Disponibilidade das informações dos requisitos do sistema.

12. Atributos

_ Informação completa dos requisitos **vigentes e não vigentes** do sistema.

13. Restrições

14. Preferências

_ Fácil recuperação das informações.

15. Expectativas

Tabela 6.1 – Requisito 005 - Consultar a Baseline de Requisitos – alterado na Solicitação de Mudança 001

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 009
2. Nome do Requisito: Bloquear Atualização de Solicitação de Mudança Finalizada
3. Responsável pela Informação: Arthur_____ Área: negócio___ data: 01/09/2001
4. Característica do Requisito: 2_____ (1 – funcional, 2 – não funcional)
5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
7. Tipo do Requisito: 4_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):
_Os stakeholders do projeto não podem modificar nenhuma Solicitação de Mudança finalizada, ou seja, aquelas que já tenham sido aceitas ou rejeitadas.

9. Problema Identificado:
_Existe a necessidade de se manter a integridade das informações do sistema.

10. Produto Esperado:
_Informações íntegras providas pelo sistema.

11. Aplicação:
_Garantir a integridade das informações do sistema.

12. Atributos

–

13. Restrições

–

14. Preferências

–

15. Expectativas

–

Tabela 6.2 – Requisito 009 - Bloquear a atualização de Solicitação de Mudança finalizada
– incluído na Solicitação de Mudança 001

Na seqüência, as informações do documento de requisitos da Solicitação de Mudança 001 foram transcritas para o sistema SIGERE.

A próxima atividade, realizada individualmente por todos os integrantes do grupo de trabalho, foi a Análise da Solicitação de Mudança proposta, através da aplicação do checklist apresentado no item 3.4.2-Análise e Negociação de Requisitos na página 80.

Nesta atividade, com a aplicação do checklist, detectamos o seguinte problema:

- falta do requisito 010 na Solicitação de Mudança, que complementaria o requisito 009. O requisito 010 formalizaria uma restrição do sistema, em que somente uma única Solicitação de Mudança poderia estar em aberto no sistema, conforme descrito na tabela 6.3.

Após a Análise da Solicitação de Mudança, promovemos as seguintes alterações na Lista de Requisitos da Mudança:

DOCUMENTO de DESCRIÇÃO de REQUISITOS
<p>1. Identificação do Requisito: 010</p> <p>2. Nome do Requisito: Bloquear a abertura de mais de uma Solicitação de Mudança</p> <p>3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/09/2001</p> <p>4. Característica do Requisito: 2_____ (1 – funcional, 2 – não funcional)</p> <p>5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)</p> <p>6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)</p> <p>7. Tipo do Requisito: 4_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)</p>
<p>8. Descrição do Requisito (sujeito + verbo + objeto): _Os stakeholders do projeto não podem iniciar uma nova Solicitação de Mudança, antes que todas as anteriores tenham sido finalizadas.</p> <p>9. Problema Identificado: _Existe a necessidade de se manter a integridade das informações do sistema.</p> <p>10. Produto Esperado: _Informações íntegras providas pelo sistema.</p> <p>11. Aplicação: _Garantir a integridade das informações do sistema.</p> <p>12. Atributos –</p> <p>13. Restrições –</p> <p>14. Preferências –</p> <p>15. Expectativas –</p>

Tabela 6.3 – Requisito 010 - Bloquear a atualização de Solicitação de Mudança finalizada – incluído na atividade Analisar Solicitação de Mudança

Concluída a atividade de Análise da Solicitação de Mudança, o passo seguinte foi executar a atividade de Elaboração da Proposta de Mudança. A execução desta atividade foi desempenhada pelo Engenheiro de Requisitos, se utilizando do checklist apresentado no item 3.4.3-Validação de Requisitos na página 81.

Esta atividade promoveu alteração no requisito 005, devido a sua incompleteza. Após a alteração, este requisito ficou documentado conforme descrito na tabela 6.4

DOCUMENTO de DESCRIÇÃO de REQUISITOS

1. Identificação do Requisito: 005
2. Nome do Requisito: Consultar a Baseline de Requisitos
3. Responsável pela Informação: Marco_____ Área: desenvolvimento___ data: 01/09/2001
4. Característica do Requisito: 1_____ (1 – funcional, 2 – não funcional)
5. Área de Origem: 1_____ (1 – interna, 2 – externa, 3 – ordem legal)
6. Qualificação Funcional: 1_____ (1 – operacional, 2 – gerencial, 3 – estratégico)
7. Tipo do Requisito: 1_____ (1 – normal, 2 – global, 3 – volátil, 4 – emergente)

8. Descrição do Requisito (sujeito + verbo + objeto):

_ Os stakeholders do projeto querem consultar a baseline de requisitos.

9. Problema Identificado:

_ Existe a necessidade de se consultar os requisitos **vigentes e não vigentes** de um sistema, **além de suas informações históricas**.

10. Produto Esperado:

_ Informação dos requisitos **vigentes e não vigentes** do sistema **com suas informações históricas**.

11. Aplicação:

_ Facilitar o processo de consulta das informações dos requisitos do sistema.

12. Atributos

_ Informação completa dos requisitos **vigentes e não vigentes** do sistema e **informações históricas**.

13. Restrições

14. Preferências

_ Fácil recuperação das informações.

15. Expectativas

Tabela 6.4 – Requisito 005 - Consultar a Baseline de Requisitos - (alterado na Atividade Elaboração da Proposta de Mudança)

Na última etapa do processo, executamos em conjunto a atividade de Avaliação de Impacto da Mudança. Esta atividade foi a de menor esforço para o grupo, que trabalhou afinado desde o início, finalizando com a aceitação da mudança para implementação. Desta forma, a versão da Baseline de Requisitos foi modificada, ficando da seguinte forma os requisitos da versão 002 da baseline:

- Requisitos 001, 002, 003, 004, 006, 007, 008 ficaram idênticos a versão 001 da baseline;
- Requisito 005 ficou conforme descrito na tabela 6.4;
- Requisito 009 e 010 ficaram conforme descritos nas tabelas 6.2 e 6.3 respectivamente.

6.4 Técnicas Aplicadas

Foram aplicadas técnicas de reuniões em grupo para esclarecimento e execução do trabalho.

Nos documentos foram expostos de que forma se desejava obter as informações, induzindo sempre se pensar no problema ou necessidade para se chegar a um requisito, sem a preocupação com a função ou atividade da pessoa.

Foram aplicadas técnicas de utilização de checklist para verificação de problemas com a descrição dos requisitos.

Os participantes no grupo de trabalho pertencentes ao laboratório Olympus tiveram de ser treinados no preenchimento do documento de requisitos e na utilização do sistema SIGERE.

6.5 Resultado do Processo de Gerenciamento de Requisitos

O resultado do processo de GR foi bastante representativo. Gerou reação imediata dos representantes do laboratório em questões que poderiam ser melhoradas no seu atual processo de trabalho com a utilização do modelo de GR proposto e da ferramenta SIGERE.

A aplicação de checklist se mostrou bastante eficaz nas atividades de Análise da Solicitação de Mudança e de Elaboração da Proposta de Mudança.

Talvez a maior dificuldade apresentada no processo tenha sido a utilização do documento de requisitos que se apresentou muito diferente do modelo atualmente aplicado no laboratório.

Ficou evidente a motivação dos representantes do Olympus com o modelo de GR proposto, que já possui certificação SIEMENS, nível 2,25, que representa no CMM estar entre o nível Repetitivo e o Definido, ou no CMMI ou na ISO/IEC 15.504, representa estar entre o nível Gerenciado e Definido/ Estabelecido.

O processo de GR é fundamental para o laboratório ou qualquer organização que tenha como ambição atingir patamares mais elevados de qualidade no desenvolvimento de software.

6.6 Resumo

Neste capítulo foi abordada a aplicação prática do modelo de gerenciamento de requisitos no ambiente de desenvolvimento da ferramenta SIGERE. Foram relatados o planejamento das atividades, a forma de aplicação do modelo, as técnicas utilizadas, um relato do acompanhamento da aplicação do processo de GR e a avaliação do produto final.

CAPÍTULO 7

Conclusão

7.1	INTRODUÇÃO	173
7.2	DIMENSÃO DO TRABALHO	173
7.3	CONSIDERAÇÕES FINAIS	177
7.4	TRABALHOS FUTUROS	177

7 Conclusão

Nesta etapa da dissertação é feita a conclusão do trabalho, relacionando os diversos assuntos tratados, associando considerações finais e desencadeando trabalhos futuros.

7.1 Introdução

O domínio do conhecimento de requisitos com o enfoque para a solução de problemas, objeto da engenharia de requisitos, requer um processo iterativo de descobrimento, análise e negociação, documentação, validação e gerenciamento de requisitos.

7.2 Dimensão do Trabalho

O impacto no software, decorrente de problemas com requisitos abrange cerca de 41% do total dos erros encontrados.

O custo relativo para consertar um erro de requisitos encontrado na fase de manutenção do software é até 200 vezes maior do que se encontrado na fase de análise de requisitos.

Requisitos corretos ou verdadeiros são tão essenciais ao processo de desenvolvimento de qualquer produto, principalmente software, que da evolução do modelo CMM para o CMMI, foi criada mais uma área chave de processo, chamada “Desenvolvimento de Requisitos”, específica para tratar este assunto.

Assim como descobrir os requisitos verdadeiros, é tão ou mais importante, gerenciá-los, de forma que as mudanças sejam documentadas e avaliadas em relação ao impacto que causarão no projeto, em caso de implementação, e que esta mudança, após aceita, seja disseminada para todos os grupos de trabalho, garantindo que o software em construção esteja sempre aderente a última versão da baseline de requisitos.

O modelo de qualidade CMM deu início, na teoria e na prática, à abordagem orientada ao processo de desenvolvimento ao invés de focar o produto em construção.

A norma ISO/IEC 15.504 deu uma nova visão aos modelos de qualidade, a dimensão “de processo”. Esta dimensão, não contemplada no modelo CMM, é fundamental para que organizações

possam atuar nos seus processos mais críticos, levando-os a patamares mais elevados de capacidade, enquanto outros, sejam mantidos em níveis mais baixos. Já a dimensão “de capacidade” é muito similar ao CMM.

Influenciado decisivamente pela ISO, o CMM evoluiu para CMMI, incorporando a dimensão de processo, além de expandir o número de áreas chave e níveis de maturidade. Na expansão do número de áreas chave, incluíram uma para tratar especificamente o levantamento dos requisitos. A esta nova área chave chamaram “Desenvolvimento de Requisitos” que veio se juntar a já existente “Gerenciamento de Requisitos”, de forma a tratar por completo o ciclo de vida dos requisitos nos projetos de desenvolvimento de sistemas.

Os modelos de qualidade forneceram para este trabalho a abordagem à processo e a visão de integração da área chave “Gerenciamento de Requisitos” com as demais áreas chave de processo. Em relação a forma como diferencia Requisitos de Cliente, de Produto e Componentes, o autor não considera adequado o tratamento, e preferiu simplesmente a abordagem de Requisitos de Sistema neste trabalho.

A partir da abordagem orientada a processo, modelamos o processo e o sistema de Gerenciamento de Requisitos no Ambiente AGIR, através da ferramenta FUN. Esta ferramenta se mostrou bastante aderente à modelagem, dando vida aos processos, através do mapeamento das atividades que o compõe e do inter-relacionamento delas no processo. Se preocupa também em modelar a infra-estrutura e recursos humanos necessários para a realização das atividades.

A ferramenta se mostrou ótima para mapear funções e controles, porém, não tão boa para o mapeamento dos dados. Fazendo uma comparação com a Análise Essencial, podemos verificar claramente que a modelagem de processo é muito melhor do que a representação dos DFD's e Lista de Eventos, porém, falta-lhe uma forma de representação dos dados, como o Diagrama de Entidades e Relacionamentos.

Em relação a Engenharia de Requisitos, este trabalho procurou tratar todo o ciclo de vida dos requisitos, trazendo informações importantes para se iniciar bem o trabalho de descobrimento dos requisitos.

Na fase de descobrimento, além de salientar a importância de um documento padrão de descrição dos requisitos, procurou trazer uma abordagem para se iniciar bem um projeto, tomando cuidados para identificar as pessoas certas para participar da etapa de levantamento de requisitos,

além de esclarecer um pouco da arte de se fazer perguntas e interpretá-las. Também procurou mostrar quão necessário é a realização de “reuniões produtivas” que são o foro principal do descobrimento dos requisitos.

Na fase de análise e negociação, tratou a importância da seleção dos stakeholders para o êxito desta etapa e da necessidade de um checklist à mão para facilitar o trabalho.

Na fase de validação, apontou a necessidade do trabalho do Engenheiro de Requisitos para que o documento de requisitos seja descrito adequadamente. Nesta fase, também é importante um checklist para guiar o processo.

Na fase de documentação se mostrou a necessidade de um documento de requisitos padrão, pois, a partir das informações nele contidas é que se alimenta o processo de Gerenciamento de Requisitos.

Na fase de Gerenciamento de Requisitos se apontou a responsabilidade por controlar a mudança nos requisitos e prover informação de rastreabilidade. Este foi o foco principal deste trabalho, que mapeou, modelou e gerou uma ferramenta de suporte ao processo. A base de conhecimento para a elaboração da ferramenta foi a literatura atual sobre o assunto e os modelos de qualidade.

Com este trabalho, conclui-se que todo o ciclo de vida dos requisitos deve ser bem trabalhado para que se tenha êxito no Gerenciamento de Requisitos. Somente com requisitos verdadeiros que se constrói um produto de software de qualidade.

Para que os requisitos permaneçam verdadeiros durante todo o ciclo de vida de um software é necessário um Gerenciamento de Requisitos efetivo que mantenha o documento de requisitos refletindo as mudanças ocorridas durante o processo.

Uma grande contribuição do trabalho foi a análise crítica da abordagem de requisitos dada pelo CMMI, e a proposta de como este tratamento poderia ser facilitado se a interação entre as áreas chave ocorresse de acordo com a figura 7.1, a seguir:

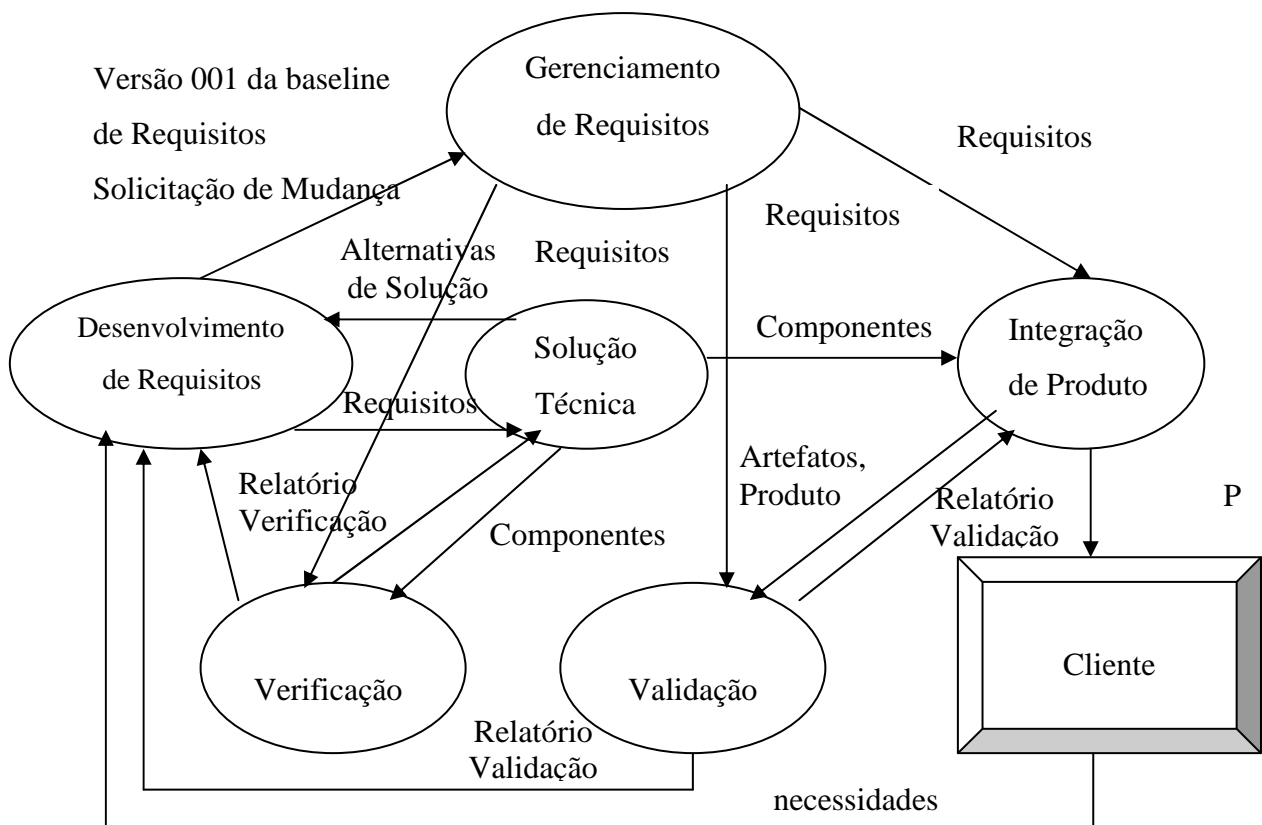


Figura 7.1 – Relacionamento de GR com outras Áreas Chave do CMMI, na visão do trabalho

Pode-se notar que todas as metas específicas apontadas no CMMI são contempladas no modelo proposto de GR, conforme demonstrado na tabela 7.1, a seguir:

Atividade das Meta Específica do CMMI	Atividade Contemplada no Trabalho
Obter um entendimento dos requisitos	X
Obter um acordo dos requisitos	X
Gerenciar mudança dos requisitos	X
Manter um rastro bi-direcional dos requisitos	X
Identificar inconsistências entre requisitos e artefatos	X

Tabela 7.1 – Atividades de GR no CMMI x Atividades contempladas pelo modelo de GR

7.3 Considerações Finais

O Modelo proposto do processo de Gerenciamento de Requisitos é aplicável a uma organização cujo desenvolvimento de sistemas utiliza a abordagem orientada a processo, em que o foco da metodologia se concentra nas “atividades” que são executadas no processo.

Apesar disso, verifica-se que a ferramenta pode ser facilmente alterada para se adaptar a abordagem da análise essencial.

Uma grande dificuldade encontrada no tratamento de requisitos se verifica na atividade de determinação do relacionamento entre os requisitos. A ferramenta permite o mapeamento hierárquico, tipo pai-filho, entre os requisitos, porém, não possui uma característica relevante que facilite a execução desta atividade.

Conclui-se, deste trabalho, que o documento final de requisitos é realmente um objeto de contratação de software, com um maior grau de certeza de aplicação e garantia de atendimento aos requisitos dos stakeholders.

É evidente que o Gerenciamento de Requisitos é fundamental para que o documento final de requisitos sempre esteja refletindo as mudanças ocorridas durante o ciclo de vida do software.

Requisitos corretos e gerenciados estreitarão muito o atual abismo entre a necessidade do cliente e o software que lhe é entregue.

7.4 Trabalhos Futuros

Como proposta para trabalhos futuros, podemos elencar os seguintes itens:

- Implementar mecanismos que facilitem a determinação do relacionamento dos requisitos;
- Implementar mecanismos que facilitem a determinação do relacionamento dos requisitos com as atividades do processo de negócio;
- Implementar mecanismos que facilitem a priorização dos requisitos;
- Implementar métricas para previsão de custo e esforço de implementação de mudança nos requisitos;

- implementar tratamento analítico das informações armazenadas pelo sistema de Gerenciamento de Requisitos como forma de melhorar o próprio processo de tratamento dos requisitos e do processo de desenvolvimento como um todo;
- fazer uma comparação do sistema de Gerenciamento de Requisitos implementado neste trabalho com outros sistemas disponíveis no mercado, a fim de verificar pontos fortes e fracos da ferramenta.

O que pode-se constatar é que esta área chave Gerenciamento de Requisitos ainda está longe da realidade da maioria das organizações, tanto no Brasil como em qualquer outra parte do mundo. Isto se mostrou bastante real, já que este trabalho, era para ser um levantamento da utilização desta área chave nas organizações, onde se percebeu que um mínimo de empresas tinham incorporado o Gerenciamento de Requisitos em seu processo de desenvolvimento de software.

Ainda temos um longo caminho até que o Gerenciamento de Requisitos se torne uma prática nas nossas organizações.

Acredita-se que a necessidade de certificações de empresas desenvolvedoras de software, fruto da exigência do mercado, venha a alavancar a utilização do processo de Gerenciamento de Requisitos nas organizações.

É bastante promissora a possibilidade de melhora na capacitação das empresas de desenvolvimento de software nesta década resultante da forte pressão do mercado por melhora na qualidade e diminuição de custo.

Referência Bibliográfica:

- [1] PRESSMAN, Roger S., *Software Engineering: A Practitioner's Approach*, 3ª edição, Editora McGraw-Hill inc., 1995
- [2] CHAOS, *The Standish Group International*, Inc., Dennis, MA, 1994.
- [3] Sheldon, F. et al, *Reliability Measurement from Theory to Practice*, IEEE Software, 9,4 (Julho 1992)
- [4] DAVIS, Alan M., *Software Requirements – objects, functions and states*, editora Prentice Hall, 1993
- [5] PAULK, M.C.; *The Capability Maturity Model – Guidelines for Improving the Software Process*, Addison Wesley, SEI series, 1995.
- [6] YOURDON, E. e Constantine, L. *Structured Design*. Englewood Clifts, Prentice Hall, 1979.
- [7] GANE, Chris e SARSON, Trish. *Structured System Analysis: tools and techniques*, Prentice Hall, 1979
- [8] PALMER, John e MCMENAMIN, Steve. *Essential System Analysis*, Nova Iorque, Yourdon Press, 1984.
- [9] PAULK, M.C., et al *The Capability Maturity Model for Software*, Versão 1.0, SEI, CMU/SEI-91-TR-24, 1991.
- [10] ABBOTT, R.J.; *An Integrated Approach to Software Development*; Nova Iorque:John Wiley, 1986.
- [11] IEEE *Standard Glossary of Software Engineering Terminology*, Nova Iorque: IEEE, ANSI/IEEE Std 729.1983, 1983.
- [12] BRACKETT, J.W.; *Software Requirements*; CMU/SEI-19-1.2, Software Engineering Institute, 1990.
- [13] THAYER, R.H.; Dorfman, M.; *Software Requirements Engineering*; Segunda edição, IEEE Computer Society, 1997.
- [14] YOURDON, Edward; *Managing the Systems Life Cicle*, Nova Iorque, Yourdon Press, 1982

- [15] S.Masters. *CMM based appraisal for internal process improvement (CBA-IPI): Method Description*, SEI, CMU/SEI-96-TR-007, 1996.
- [16] Software Engineering Institute. *Process Maturity Profile of the Software Community 1996 Year End Update*, 1997.
- [17] WEBER, K.C.; Luca, J.C.M.; *Qualidade e Produtividade em Software*, Makron Books, 2ª edição, 1997.
- [18] FIORINI, Soeli; *Engenharia de Software com CMM*, Brasport Livros e Multimídia Ltda, 1998.
- [19] JALOTE, Pankaj. *CMM in Practice: Processes for Executing Software Projects at Infosys*. USA: Addison Wesley Longman, Inc., 1999.
- [20] R.L. Glass. *Software Runaways, Lessons Learned from Massive Software Project Failures*. USA: Prentice Hall PTR, 1998.
- [21] L.H. Putnam and W. Myers. *Industrial Strength Software – Effective Management Using Measurement*. USA: IEEE Computer Society Press, 1997.
- [22] YOURDON, Edward *Modern Structured Analysis*, Prentice Hall, 1989.
- [23] FUGGETA, A. *Software process: a roadmap*. In: *The Future of Software Engineering*, A. Finkelstein (ed),2000.
- [24] PFLEEGER, S.L. *Software Engineering – theory and practice*. Prentice Hall, 1998.
- [25] CURTIS, B. *The global pursuit of process maturity*. IEEE Software, jul/ago 2000.
- [marlon1]
- [27] ROCHA, Ana R.C. et al. *Qualidade de Software – teoria e prática*. Prentice Hall, 2001.
- [28] CMMI Project. *CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development – Continous Representation*, Version 1.01. SEI, 2000.
- [29] WEINBERG, Gerald M. *Quality Software Management*, Dorset House Publishing, 1992.
- [30] GAUSE, Don, Winberg, Gerald M. *Exploring Requirements*, Dorset House Publishing, 1989.
- [31] MACKAY, Ian *Asking Questions*, Institute of Personnel and Development, 2000.

- [32] DOYLE, Michael, Straus, David *How to Make Meetings Work: The New Interaction Method*, Playboy Press, 1986.
- [33] FORSYTH, Patrick, *Making Meetings Work*, Institute of Personnel and Development, 1996.
- [34] WEINBERG, Gerald M.; *Are your lights on ?*, Dorset House Publishing, 1992.
- [35] MACAULAY, L.A., *Requirements Engineering*, Springer, 1996.
- [36] KOTONYA, Gerald *Requirements Engineering: Processes and Techniques*, John Wiley, 1998.
- [37] SOMMERVILLE, Ian *Requirements Engineering – A good practice guide*, John Wiley, 1997.
- [38] ISO/IEC 9126. *Information Technology – Software Product Evaluation – Quality Characteristics and Guidelines for their use*, 1991.
- [39] MACEDO, N.M. e Leite, J.C.S.P. *Integrando requisitos não funcionais aos requisitos baseados em ações concretas*. Costa Rica, Anais do Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software, CYTED & Instituto Tecnológico de Costa Rica (TEC), 1999.
- [40] FREEMAN, P.A. *Software perspectives: the system is the message*. Massachusetts, Addison-Wesley Publishing Company, 1987.
- [41] ZANLORENCI, Edna Pacheco *Descrição e Qualificação de Requisitos: um Modelo Aplicável à Análise e Validação da Informação*. Curitiba, 1999.
- [42] GOTEL, Orlena; FINKELSTEIN, Anthony *Extended Requirements Traceability: Results of an Industrial Case Study*. ISRE'97 Third International Symposium on Requirements Engineering. 1ed. USA:IEEE CSP, Los Alamitos, CA. Proceedings, 1997.
- [43] HAMMER, T. *et al Requirement Metrics – Value Added*. ISRE'97 Third International Symposium on Requirements Engineering. 1ed. USA:IEEE CSP, Los Alamitos, CA. Proceedings, 1997.
- [44] PINHEIRO, Francisco A.C.; GOGUEN, Joseph A. *An Object Oriented Tool for Tracing Requirements*. 1ed. USA:IEEE Software, 1996.
- [45] LEITE, J.C.S.P. *Software Evolution, The requirements engineering view*. Buenos Aires, Argentina, Keynote address 26 Jaio, Proceedings SoST'97, JAIIO, SADIO, 1997.

- [46] JACKSON, Michael *Software Requirements and Specifications: A lexicon of practice, principles and prejudices*. 1ed USA, Massachussets: Addison Wesley, Reading, 1995.
- [47] RYAN, Kevin. *Requirements Engineering – getting value for money*. SBES'98, XII Simpósio Brasileiro de Engenharia de Software, Maringá, Paraná. 1ed. Brasil : SBC Sociedade Brasileira de Computação, 1998.
- [48] MARCINIUK, Marlon S. et al. *Modelagem de Processos: Contribuindo para a Maturidade da Organização*, SIMPROS, 2001.
- [49] CORDEIRO, Marco Aurélio *O panorama atual da Indústria de software*, revista bate-byte, n.97, pg 22, Celepar,2000.
- [50] CORDEIRO, Marco Aurélio *Foco no processo*, revista bate-byte, n.100, pg 26, Celepar,2000.
- [51] CORDEIRO, Marco Aurélio *Modelos de qualidade de desenvolvimento de software*, revista bate-byte, n.99, pg 31, Celepar,2000.
- [52] MARCINIUK, Marlon S. et al. *CMMI – Atendendo à ISO TR 15504-2*, CITS, 2001.
- [53] *CMM Baised Appraisal Project; Software Capability Evolution Version 2.0*, SEI 1994.
- [54] LEITE, Júlio C.S.P. *Engenharia de Requisitos*. 1ed. Brasil : PUC-RIO, Rio de Janeiro, RJ. (Notas de Aula), 1994, 63 p.