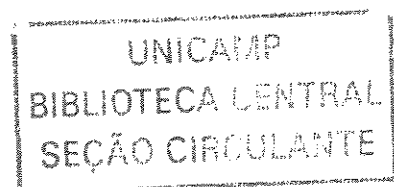


**Um estudo sobre a relevância de boas práticas de
Engenharia de Requisitos**

Andreza Polcelli

Trabalho Final de Mestrado Profissional



**Instituto de Computação
Universidade Estadual de Campinas**

**Um estudo sobre a relevância de boas práticas de
Engenharia de Requisitos**

Andreza Polcelli

Fevereiro de 2005 ✓

Banca Examinadora:

Prof. Dr. Mário Lúcio Côrtes (Orientador)
Instituto de Computação - UNICAMP

Prof^a. Dr^a. Ariadne Maria Brito Rizzoni Carvalho
Instituto de Computação - UNICAMP

Prof^a. Dr^a. Ana Cervigni Guerra
Centro de Pesquisas Renato Archer – CenPRA/MCT

Prof^a. Dr^a. Maria Cecília Calani Baranauskas (Suplente)
Instituto de Computação - UNICAMP

NIDADE BC
 N.º CHAMADA 711/UNICAMP
P757e
 EX
 OMBO BC/ 64897
 ROC. 16-P-00086105
 C ☐ D ☒
 PREÇO R\$11,00
 DATA 21-07-05
 Nº CPD
Bib. id. 359302

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Polcelli, Andreza

P757e

Um estudo sobre a relevância de boas práticas de engenharia de requisitos / Andreza Polcelli -- Campinas, [S.P. :s.n.], 2005.

Orientador : Mário Lúcio Côrtes

Trabalho final (mestrado profissional) - Universidade Estadual de Campinas, Instituto de Computação.

1. Software – Qualidade. 2. Engenharia de Software. 3. Engenharia de Software – Normas. 4. Software – Desenvolvimento. I. Côrtes, Mário Lúcio. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Um estudo sobre a relevância de boas práticas de Engenharia de Requisitos

Este exemplar corresponde à redação final do Trabalho Final devidamente corrigido e defendido por Andreza Polcelli e aprovada pela Banca Examinadora.

Campinas, 14 de fevereiro de 2005.



Prof. Dr. Mário Lúcio Côrtes
(Orientador)

Trabalho Final apresentado ao Instituto de Computação da UNICAMP, como requisito parcial para obtenção do título de Mestre em Computação na área de Engenharia de Software.

TERMO DE APROVAÇÃO

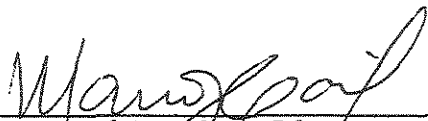
Trabalho Final Escrito defendido e aprovado em 14 de fevereiro de 2005, pela
Banca Examinadora composta pelos Professores Doutores:



Profa. Dra. Ana Cervigni Guerra
CenPRA



Profa. Dra. Ariadne Maria Brito Rizzoni Carvalho
IC - UNICAMP



Prof. Dr. Mário Lúcio Côrtes
IC - UNICAMP

© Andreza Polcelli, 2005.
Todos os Direitos Reservados

*"A árvore plantada junto à corrente de
águas, no devido tempo, dá o seu fruto,
e as suas folhas não murcham..."*

AGRADECIMENTOS

Ao soberano e eterno Deus que tudo criou e que, especialmente, me capacitou.

Aos meus pais por toda demonstração de amor, apoio, incentivo e confiança.

A todos familiares e amigos por demonstrarem seu apoio e por me encorajarem a completar este trabalho.

Aos meus notáveis pequeninos pelas grandiosas demonstrações de amor, carinho e amizade.

Ao meu orientador Prof. Dr. Mário Lúcio Côrtes, pela oportunidade e por acreditar em meu potencial.

RESUMO

A engenharia de requisitos descreve atividades fundamentais no processo de produção de sistemas de software. Os requisitos são o alicerce sob o qual são implementadas as fases do desenvolvimento de software. Ao longo das fases do ciclo de vida do software, a importância de ter requisitos bem definidos e controlados é bem maior do que nas fases anteriores. Um problema nos requisitos inseridos na fase de requisitos pode introduzir outros problemas nas fases seguintes, gerando problemas no software e aumentando o retrabalho, custo, prazo e esforço do projeto de software e também aumentando a insatisfação do cliente.

Vários modelos de qualidade de software e normas apresentam um conjunto de boas práticas em engenharia de requisitos que podem ser usadas pelas organizações para enfrentar esses problemas. Este trabalho examina a ocorrência dessas práticas nas normas e modelos e as correlaciona, e estuda a relevância do uso dessas boas práticas de requisitos em projetos de software reais.

O estudo experimental mostrou que grande parte (de 31% a 55%) dos problemas encontrados em software são consequências de deficiências nos requisitos. Assim, confirma dados da literatura que mostram que as deficiências nas práticas de requisitos propagam problemas para as fases seguintes do processo de desenvolvimento, aumentando o custo e prazo de projetos de software e também são a maior causa dos problemas encontrados nos sistemas de software. O estudo experimental mostrou que há correlação entre o uso de boas práticas de requisitos e a ocorrência desses problemas de software. Entretanto, não foi possível obter uma correlação significativa entre a ocorrências de problemas e o efeito disso sobre o desempenho dos projetos de software. Embora a porcentagem de ocorrências de problemas, contabilizada em cada uma das práticas de requisitos, seja diferente, não se observou prevalência significativa na importância entre as práticas de requisitos.

As boas práticas e o resultado mostrado por este estudo experimental, aliado à experiência no dia a dia de trabalho com requisitos de sistema de software, permitiram apresentar algumas recomendações que podem agregar valor na implementação das práticas definidas nos modelos e normas de qualidade.

ABSTRACT

Requirements engineering describes the fundamentals activities in software systems production processes. Requirements are the foundation under which phases of the software development are implemented. Throughout the phases of the software life cycle, the importance of having well defined and controlled requirements is much larger than in the prior phases. A requirements problem inserted in the requirements phase can introduce other problems in the following phases, causing software problems and increasing rework, costs, schedule slippage and effort of the software project, and also reducing customer satisfaction.

Several software quality models and standards present best practices in requirements engineering that can be used by organizations to address these problems. This dissertation examines the occurrence of these requirements practices in some models and quality standards, correlates them and studies the relevance of the use of these best practices in real software projects.

The experimental study showed that significant part (from 31% to 55%) of problems found in software is consequence of requirement problems. This confirms literature information that indicates that the deficiencies in requirements practices propagate problems to the following phases in the development process, having adverse effect on project cost and schedule, and also are the main reason of problems found in software systems. The experimental study has showed that the use of best practices is related to software problems occurrences. However, a significant correlation between the problems occurrences and the effect of this on the performance of the software projects was not found. Although the percentages of software problems occurrences, accounted in each one of the requirements practices, are different, it was not observed significant prevalence in the importance of the requirements practices.

The best practices and the results presented in this experimental study, associated to the day by day job experience with software requirements, allowed to present some recommendations that can aggregate value in the implementation of the practices defined in the models and quality standards.

Índice

CAPÍTULO 1	INTRODUÇÃO	5
1.1	MOTIVAÇÃO E OBJETIVOS	5
1.2	ESTRUTURA.....	7
CAPÍTULO 2	CONCEITOS GERAIS SOBRE A ENGENHARIA DE REQUISITOS	9
2.1	ENGENHARIA DE SOFTWARE E ENGENHARIA DE REQUISITOS	9
2.2	PRINCIPAIS TIPOS DE REQUISITOS	11
2.2.1	Requisitos de contrato.....	11
2.2.2	Requisitos do cliente	11
2.2.3	Requisitos de sistema	11
2.2.4	Requisitos de software	12
2.2.5	Requisitos técnicos.....	12
2.2.6	Requisitos não-técnicos.....	12
2.2.7	Requisitos funcionais.....	12
2.2.8	Requisitos não-funcionais.....	12
2.3	DEFINIÇÃO DE DEFEITO, ERRO, FALHA, E PROBLEMA	13
2.4	CONSIDERAÇÕES FINAIS.....	13
CAPÍTULO 3	TRABALHOS CORRELATOS	15
3.1	CONSIDERAÇÕES FINAIS.....	22
CAPÍTULO 4	MODELOS CMM E CMMI	23
4.1	MODELO SW-CMM.....	23
4.1.1	Conceitos básicos: Níveis de maturidade	24
4.1.2	Conceitos básicos: Áreas chave de processo.....	25
4.1.3	A questão requisito e o modelo SW-CMM	29
4.2	MODELO CMMI	33
4.2.1	Conceitos básicos: CMMI	33
4.2.2	Áreas de processo do CMMI	35
4.2.3	A questão requisito e o modelo CMMI	39
4.3	CONSIDERAÇÕES FINAIS.....	41
CAPÍTULO 5	NORMAS DA FAMÍLIA ISO 9000	43
5.1	NORMA ISO 9000.....	44
5.2	NORMA ISO 9001	45
5.2.1	A questão requisitos e a norma ISO 9001	45
5.3	GUIA ISO 9000-3	48
5.3.1	A questão requisitos e o Guia ISO 9000-3.....	48
5.4	NORMA ISO 9004.....	50
5.4.1	A questão requisitos e a norma ISO 9004	50
5.5	CONSIDERAÇÕES FINAIS.....	52
CAPÍTULO 6	OUTROS MODELOS E NORMAS	53
6.1	NORMA NBR ISO/IEC 9126-1	53
6.1.1	A questão requisitos e a norma NBR ISO/IEC 9126-1	54
6.2	NORMA NBR ISO/IEC 12207	55
6.2.1	A questão requisitos e a norma NBR ISO/IEC 12207	55
6.3	NORMA IEEE 830.....	63

6.3.1	A questão requisitos e a norma IEEE 830	63
6.4	MODELO PMBOK	64
6.4.1	A questão requisitos e o modelo PMBOK	65
6.5	CONSIDERAÇÕES FINAIS	67
CAPÍTULO 7	CORRELAÇÕES ENTRE NORMAS E MODELOS	69
7.1	PRINCIPAIS DEFINIÇÕES E TERMINOLOGIAS	70
7.1.1	Terminologias SW-CMM	70
7.1.2	Terminologias CMMI	71
7.1.3	Terminologias da família ISO 9000	72
7.1.4	Terminologias ISO/IEC 9126	73
7.1.5	Terminologias ISO/IEC 12207	74
7.1.6	Terminologias IEEE 830	74
7.1.7	Terminologias PMBOK	75
7.2	CORRELAÇÕES ENTRE NORMAS, PADRÕES E MODELOS	76
7.3	CONSIDERAÇÕES FINAIS	80
CAPÍTULO 8	ANÁLISE DA RELEVÂNCIA DE PRÁTICAS DE REQUISITOS A PARTIR DE DADOS EXPERIMENTAIS	81
8.1	METODOLOGIA	81
8.2	EXECUÇÃO	84
8.2.1	Passo 1 – Seleção de projetos de software	84
8.2.2	Passo 2 – Cenários e Características dos projetos	85
8.2.3	Passo 3 – Desempenho dos projetos	91
8.2.4	Passo 4 – Análise comparativa do desempenho dos projetos	92
8.2.5	Passo 5 – Obtenção dos registros de solicitações	93
8.2.6	Passo 6 – Classificação e Contabilização das solicitações	93
8.2.7	Passo 7 – Análise comparativa entre as solicitações e o desempenho dos projetos selecionados	95
8.2.8	Passo 8 – Correlação com boas práticas de requisitos	97
8.2.9	Passo 9 – Análise dos resultados	99
8.2.10	Passo 10 – Recomendações	101
8.3	CONSIDERAÇÕES FINAIS	102
CAPÍTULO 9	CONCLUSÃO	103
9.1	PERSPECTIVAS FUTURAS	104
9.2	CONCLUSÕES FINAIS	104
CAPÍTULO 10	REFERÊNCIAS	107
10.1	OUTRAS PUBLICAÇÕES	111
10.2	REFERÊNCIAS WEB	113
CAPÍTULO 11	ANEXOS	115
11.1	ANEXO A: SW-CMM	115
11.1.1	CMM Nível 2	115
11.1.2	CMM Nível 3	119
11.1.3	CMM - Outros Níveis	122
11.2	ANEXO B: CMMI	123
11.2.1	CMMI Nível 2	123
11.2.2	CMMI Nível 3	126
11.3	ANEXO C: NORMAS DA FAMÍLIA ISO 9000	130

11.3.1	ISO 9000	130
11.3.2	ISO 9000-3	130
11.3.3	ISO 9001	133
11.3.4	ISO 9004	136
11.4	ANEXO D: ISO 9126	138
11.5	ANEXO E: ISO 12207	141
11.6	ANEXO F: IEEE 830	148
11.7	ANEXO G: PMBOK	151
11.8	ANEXO H: BIBLIOGRAFIA	153

Índice de Tabelas

Tabela 4-1 – Resumo das Áreas Chave de Processo do Nível 2.....	26
Tabela 4-2 – Resumo das Áreas Chave de Processo do Nível 3.....	27
Tabela 4-3 – Resumo das Áreas Chave de Processo do Nível 4.....	28
Tabela 4-4 – Resumo das Áreas Chave de Processo do Nível 5.....	29
Tabela 4-5 – CMMI: Representação por Estágios	33
Tabela 4-6 – CMMI: Representação Contínua.....	34
Tabela 4-7 – Exemplo Nível 2 Representação por estágios	34
Tabela 4-8 – Exemplo Nível 3 Representação por estágios	35
Tabela 4-9 – Exemplo Representação contínua	35
Tabela 4-10 – Resumo das Áreas de Processo do Nível 2.....	36
Tabela 4-11 – Resumo das Áreas de Processo do Nível 3.....	37
Tabela 4-12 – Resumo das Áreas de Processo do Nível 4.....	38
Tabela 4-13 – Resumo das Áreas de Processo do Nível 5.....	39
Tabela 7-1 – Tabela de Correlação de Práticas de Requisitos	77
Tabela 7-2 – Legenda da Tabela 7-1	78
Tabela 8-1 – Resumo dos cenários dos projetos	86
Tabela 8-2 – Esforço/Prazo dos Projetos.....	91
Tabela 8-3 – Variação de Esforço/Prazo dos Projetos.....	92
Tabela 8-4 – Tamanho dos Projetos.....	92
Tabela 8-5 – Solicitações por Projeto	94
Tabela 8-6 – Ocorrência de Solicitação por Tipo de Solicitações e por Projeto	95
Tabela 8-7 – Normalização.....	95
Tabela 8-8 – Boas Práticas de Requisitos e Solicitações	98
Tabela 8-9 – Classificação por boas práticas	98

Índice de Gráficos

Gráfico 8.1 – Desvio de Esforço	96
Gráfico 8.2 – Desvio de Prazo	96
Gráfico 8.3 – Relação de Erros por Práticas de Requisitos	99

Índice de Figuras

Figura 2.1 – Processo de engenharia de requisitos [SOMME03].....	10
Figura 3.1 – Porcentagem de defeitos por atividade (Figura adaptada [FLORA92])	16
Figura 3.2 – Recursos gastos em dólares (Figura adaptada [HERBS94]).....	17
Figura 3.3 – Porcentagem de custo (Figura adaptada [HERBS94]).....	17
Figura 3.4 – Defeitos por atividade do ciclo de vida (Figura adaptada [BAUME92])	18
Figura 3.5 – Defeitos encontrados em revisão de pares (Figura adaptada [BAUME92])...19	
Figura 3.6 – Defeitos inseridos e detectados (Figura adaptada [BAUME92]).....	20
Figura 3.7 – Custo para correção de falhas (Figura adaptada [SEMAT95]).....	21

Capítulo 1 Introdução

Em todo o mundo, as organizações envolvidas com desenvolvimento de sistemas de software buscam atender os seus principais objetivos. Alguns desses objetivos se configuram na busca em obter sucesso no desenvolvimento de sistemas de software. Relacionados ao termo sucesso, neste caso, podem ser citados a produção de sistemas de software de alta qualidade, muito bem gerenciados com alto controle de custo, esforço e prazo, e que atendem as expectativas do cliente e aos seus requisitos preestabelecidos. Nessa busca, torna-se necessário minimizar ou evitar problemas¹ que surgem durante todo o processo de desenvolvimento de sistemas de software, desde as fases iniciais do ciclo de vida, como por exemplo, a dificuldade de executar o levantamento de requisitos, até as suas fases finais.

1.1 Motivação e objetivos

Na construção de uma casa ou edifício o projeto de engenharia da obra a ser realizada é a parte fundamental da obra, é a base onde todo o projeto da construção está fundamentado, portanto a complexidade e importância do projeto de engenharia da obra são potencializadas pelo seu valor agregado. A partir desse projeto de engenharia da obra será construído o alicerce da casa ou edifício, base de sustentação de toda a construção, depois serão construídas as paredes e afins e por fim o telhado e acabamento. Assim também acontece quando se constrói um sistema de software. Se compararmos o desenvolvimento de sistemas de software com a construção de uma casa ou edifício, assim como é necessário fazer o projeto de engenharia da obra, teremos o planejamento do desenvolvimento do projeto de software, e os requisitos do software são como a pedra fundamental, base naquilo em que o projeto de software inteiro está fundamentado. Falar em requisito é também lembrar de qualidade e satisfação do cliente que são parte das peças chaves do sucesso das organizações de software.

Dentre os muitos problemas que podem acontecer durante todo o processo de desenvolvimento de sistemas de software, os problemas relacionados à base da

¹ O termo “problema” está definido no item 2.3.

construção de software (os requisitos) podem gerar grandes impactos negativos nos resultados finais dessa construção. Como exemplo, pode ser citado o caso *Ariane Flight 501* que, no dia quatro de junho de 1996, o voo inicial do foguete *Europe's Ariane 5* terminou em um grande fracasso com a destruição completa do foguete. A investigação do problema [BUREN98] demonstrou que os requisitos de software estavam incompletos e as atividades de análise de requisitos e as atividades do processo de verificação e validação de requisitos foram deficientes em descobrir as falhas nos requisitos. Esse exemplo mostra porque a área de requisitos é tão importante e essencialmente necessária à manutenção da qualidade nas atividades relacionadas aos projetos de software.

Por essa importância, que está associada à área de requisitos, as organizações podem aplicar as boas práticas de requisitos definidas nos modelos de qualidade e processo de software a fim de evitar ou minimizar os problemas no software.

Por se tratar de uma área que tem complexidade, importância e grande valor agregado, a área de requisitos foi escolhida para estudo e elaboração deste trabalho. Essa mesma motivação aumenta por não haver muitas publicações que mostrem uma visão global das principais práticas de requisitos em diferentes modelos ou normas de qualidade ou que apresentem uma análise da relevância do uso dessas práticas em projetos de software. Além disso, a área de requisitos é a área de experiência e aplicação prática do autor que busca melhorar a qualidade de suas tarefas e também busca contribuir para trabalhos futuros na área de engenharia de requisitos.

Esta dissertação de mestrado tem como tema central a área de engenharia de requisitos e está baseada nos seguintes objetivos:

1. Estudo da engenharia de requisitos aplicada nos principais modelos de processo e qualidade de software.
2. Correlação entre as práticas de requisitos descritas nesses modelos de processo e qualidade de software.
3. Análise da relevância do uso das boas práticas de requisitos resultante de uma contabilização de problemas apresentados em uma amostra de projetos de software reais.

4. Recomendações gerais de uso prático para minimizar os problemas relacionados à questão requisito e também, por consequência, diminuir as falhas² relacionadas a requisitos encontradas no software.

1.2 Estrutura

Esta dissertação está estruturada em dois grandes blocos. No primeiro bloco, são apresentados conceitos e informações extraídas da literatura, como se segue:

- O Capítulo 2 expõe conceitos básicos relacionados ao assunto do trabalho.
- O Capítulo 3 apresenta trabalhos correlatos ao assunto deste trabalho.
- Os capítulos 4, 5 e 6 apresentam os aspectos associados à área de engenharia de requisitos das normas e modelos produzidos pelo *SEI* (*Software Engineering Institute*), normas da família ISO 9000 e outros, respectivamente.

No segundo bloco, são apresentadas as principais contribuições deste trabalho:

- O Capítulo 7 apresenta uma correlação entre os modelos apresentados nos capítulos anteriores resultando em definições de práticas globais aplicadas aos modelos e normas apresentados. O Capítulo 8 apresenta um estudo com dados experimentais de quatro projetos de software reais sobre a relevância do uso das práticas globais de requisitos definidas no Capítulo 7.
- O Capítulo 9 apresenta as conclusões deste trabalho.
- O Capítulo 10 lista as referências bibliográficas. E o Capítulo 11 apresenta os anexos.

² Os termos “defeito”, “falha” e “erro” estão definidos no item 2.3.

Capítulo 2 Conceitos gerais sobre a Engenharia de Requisitos

O objetivo deste capítulo é descrever uma visão geral dos conceitos utilizados neste trabalho, introduzindo assuntos relacionados a requisitos, como engenharia de software e engenharia de requisitos, para o bom entendimento dos textos descritos nos capítulos seguintes. São apresentados conceitos gerais e um breve glossário de termos e expressões mais relevantes para este trabalho.

2.1 Engenharia de Software e Engenharia de Requisitos

Engenharia de software pode ser entendida como a disciplina de engenharia aplicada ao desenvolvimento de software. Segundo Pressman [PRESS95], a engenharia de software compreende um conjunto de etapas que envolvem métodos, ferramentas e procedimentos que possibilitam ao gerente o controle do processo de desenvolvimento de software e oferecem ao profissional uma base para a construção de software de alta qualidade produtivamente. E segundo Sommerville [SOMME03], a engenharia de software é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação.

A engenharia de requisitos pode ser vista como uma sub-área da engenharia de software. Thayer e Dorfman [THAYER97] definem engenharia de requisitos como uma ciência (disciplina) que objetiva a elicitacão¹ e a documentação dos requisitos de software. A engenharia de requisitos consiste dos seguintes estágios: elicitacão dos requisitos de software, análise, especificação, verificação e gestão.

De acordo com o IEEE [IEEE84], o processo de aquisição, refinamento e verificação das necessidades do cliente é chamado de engenharia de requisitos e tem como um dos seus principais objetivos a obtenção de uma especificação correta e completa dos requisitos. Boehm [BOEHM89] define engenharia de requisitos como uma

¹ Embora a palavra "elicitacão" não conste dos principais dicionários da língua portuguesa, utiliza-se a palavra "elicitacão" para traduzir a palavra inglesa "*elicitation*" que significa extração ou levantamento de requisitos.

disciplina para desenvolver uma especificação completa, consistente e não ambígua que sirva como base para um acordo entre todas as partes envolvidas descrevendo o que o produto de software irá fazer.

Sommerville [SOMME03] afirma que a engenharia de requisitos é um processo que envolve todas as atividades exigidas para criar e manter o documento de requisitos de sistema, como ilustrado na Figura 2.1.

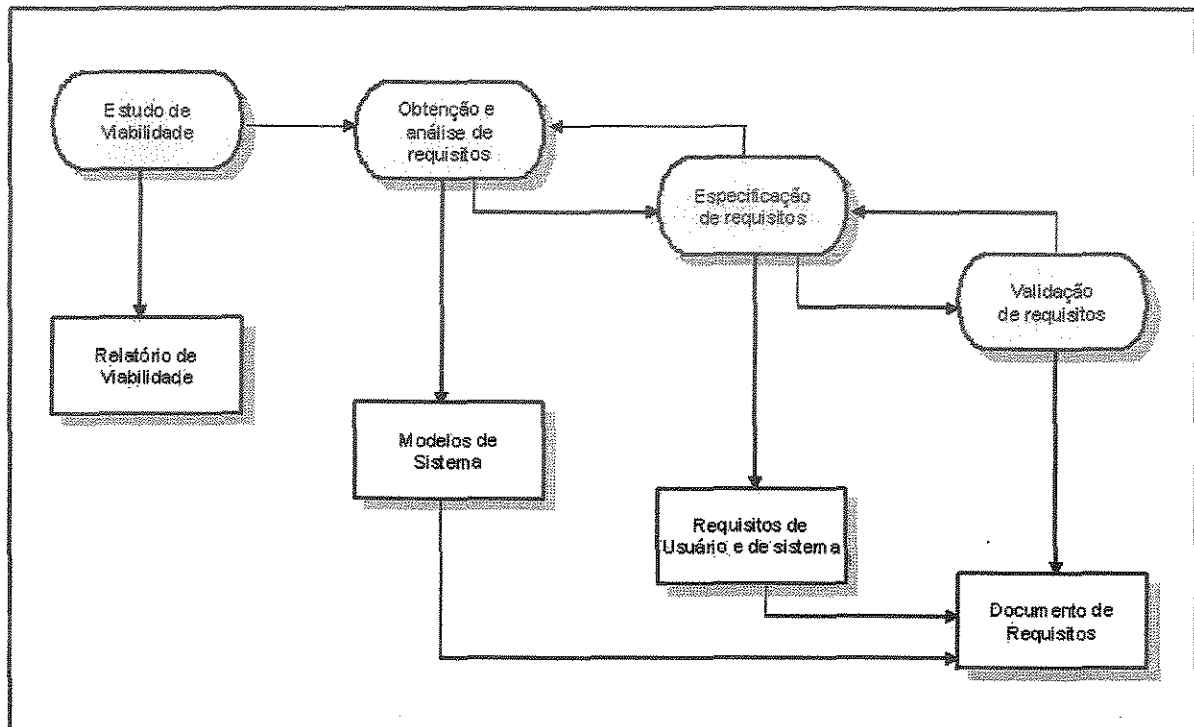


Figura 2.1 – Processo de engenharia de requisitos [SOMME03]

A Figura 2.1 mostra as atividades genéricas do processo de engenharia de requisitos e os documentos gerados por cada uma dessas atividades:

- **Estudo de viabilidade:** Para sistemas novos, o processo de engenharia de requisitos deve começar com um estudo de viabilidade. A entrada para o estudo de viabilidade é uma descrição geral do sistema e de como ele será utilizado dentro da organização. O resultado do estudo de viabilidade deve ser um relatório. Esse relatório de viabilidade deve recomendar se o desenvolvimento do sistema deve continuar ou não, podendo propor

mudanças no enfoque, no orçamento e no cronograma, além de sugerir outros requisitos.

- **Obtenção e análise de requisitos:** Os membros da equipe técnica de desenvolvimento de software trabalham junto com o cliente e os usuários finais do sistema para descobrir mais informações sobre o domínio da aplicação. Essa atividade pode envolver diferentes tipos de pessoas.
- **Especificação de requisitos:** Essa atividade deve definir os requisitos do usuário e os requisitos do sistema (descrições mais detalhadas dos requisitos do usuário).
- **Validação de requisitos:** Essa atividade se ocupa de mostrar que os requisitos realmente definem o sistema que o cliente deseja. Essa atividade deve gerar o documento completo de requisitos.

2.2 Principais tipos de requisitos

Seguem as descrições genéricas mais utilizadas relacionadas a requisitos:

2.2.1 Requisitos de contrato

Requisitos de contrato são requisitos estabelecidos na fase de aquisição e contratação de produtos e serviços. São requisitos citados e definidos no documento de contrato entre o cliente e o fornecedor. Todos os requisitos técnicos e não-técnicos relacionados na aquisição [TR02902].

2.2.2 Requisitos do cliente

Requisitos do cliente são declarações, em linguagem natural e também em diagramas, sobre as funções que o sistema deve fornecer [SOMME03].

2.2.3 Requisitos de sistema

Requisitos de sistema estabelecem detalhadamente as funções e as restrições de sistema [SOMME03].

2.2.4 Requisitos de software

Requisitos de software são os requisitos atribuídos ao software, quer sejam eles definidos pelos clientes ou definidos pelas áreas envolvidas em um projeto de software. A necessidade de um usuário em solucionar um problema sendo suprida por requisitos de software [TR02593].

2.2.5 Requisitos técnicos

Requisitos técnicos são todos os requisitos de cunho técnico, quer sejam requisitos funcionais ou não-funcionais. Requisitos que descrevem o que o software deve fazer e suas regras operacionais [TR02593].

2.2.6 Requisitos não-técnicos

Requisitos não-técnicos são requisitos que não apresentam características técnicas de um projeto de software, mas sim características gerais do projeto, como cronogramas, prazos de entrega e outros. Requisitos não-técnicos podem ser definidos pelo cliente ou pelas áreas envolvidas em um projeto de software. Acordos, regras, condições e termos que afetam os produtos ou serviços que estão sendo adquiridos [TR02902].

2.2.7 Requisitos funcionais

Requisitos funcionais são declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações [SOMME03].

2.2.8 Requisitos não-funcionais

Requisitos não-funcionais são restrições sobre os serviços ou as funções oferecidas pelo sistema. Entre eles destacam-se restrições de tempo, restrições sobre o processo de desenvolvimento, padrões entre outros [SOMME03].

2.3 Definição de defeito, erro, falha, e problema

De acordo com o padrão IEEE 610.12-1990, os termos “defeito”, “engano”, “erro” e “falha” são definidos da seguinte maneira [VINCE04]:

- Defeito (*fault*): Passo, processo ou definição de dados incorretos, como por exemplo, uma instrução ou comando incorreto.
- Engano (*mistake*): Ação humana que produz um resultado incorreto.
- Erro (*error*): Diferença entre o valor obtido e o valor esperado, que ocorre como estado intermediário do programa.
- Falha (*failure*): Produção de uma saída incorreta com relação à especificação.

Este trabalho usa a expressão “problemas de software” com o significado de defeitos, erros ou falhas que estejam relacionados ao software por falta de especificação, por especificação incorreta ou por implementação incorreta, entre outros.

2.4 Considerações finais

Neste capítulo vimos alguns conceitos gerais relacionados à engenharia de requisitos e aos tipos de requisitos para entender e estabelecer a terminologia a ser usada neste trabalho.

Capítulo 3 Trabalhos correlatos

Este capítulo apresenta um resumo de alguns artigos publicados na literatura e outros, como material de treinamento de empresas, que tratam assuntos relacionados às boas práticas de requisitos descritas em modelos e normas de qualidade e também que tratam os efeitos da utilização ou não das melhores práticas de engenharia de requisitos em projetos de software.

O artigo [STARK98] apresenta alguns aspectos interessantes:

- Esse artigo apresenta dados relacionados a requisitos de 40 *releases* de software (108 alterações de requisitos foram feitas durante 40 *releases* de software, sendo que 26 dos 40 *releases*, ou seja, 65% tiveram seus requisitos alterados, sendo que oito desses 26 *releases* apresentaram mais de 50% de alterações de requisitos).
- O artigo apresenta dados que mostram que 20 alterações de requisitos foram feitas em um *release* de software (com duração de 14 meses desde o início do projeto). Apenas seis dos requisitos entregues não foram alterados, e as nove alterações (de um total de vinte) foram feitas nos últimos cinco meses acarretando em grande impacto no esforço de implementação.
- Declara também que o *release* teve uma duração bem maior que a duração planejada (mais que o dobro) e que teve seu custo aumentado em 22%. Com base nesses dados, declara que as mudanças nos requisitos não são a única razão de aumento de custo e cronograma, mas são os maiores ofensores.
- Como recomendação descreve que os requisitos devem ser mais claros, bem esclarecidos e compreendidos pela equipe de desenvolvimento e as alterações de requisitos acordadas devem ser mais formalmente gerenciadas pela equipe de gestão de configuração (responsável pelas *releases* do software).

O artigo “*Software Quality Measurement: A framework for Counting Problems and Defect*” [FLORA92] apresenta um mecanismo (modelo) que pode ser utilizado para obter medições de software. A razão fundamental para obter medições de software e processo de software é obter dados que possam auxiliar na melhoria do controle de cronograma, custo e qualidade do produto de software. Esse artigo declara que os dados do exemplo da Figura 3.1 mostram que o processo de inspeção nas atividades iniciais do ciclo de vida encontram aproximadamente 50% de defeitos injetados e conclui que inspeções e revisões mais disciplinadas são necessárias.

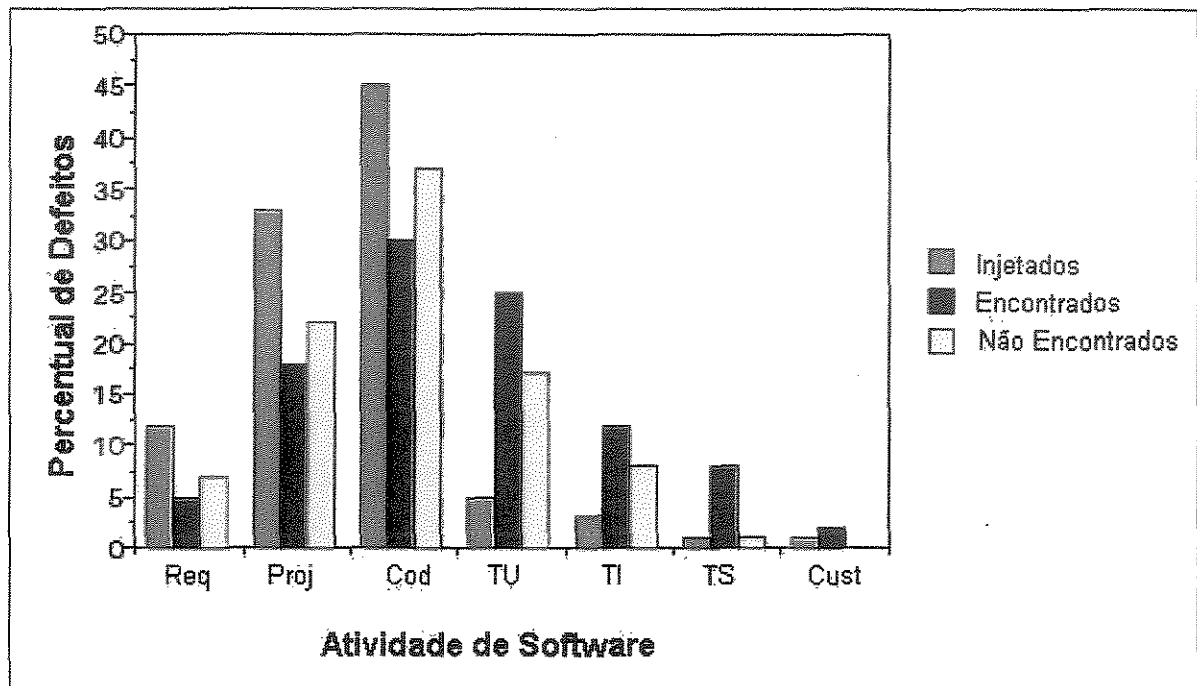


Figura 3.1 – Porcentagem de defeitos por atividade (Figura adaptada [FLORA92])

O artigo [HERBS94] traz um estudo de caso da empresa *Texas Instruments* que desenvolveu o mesmo projeto de software, porém utilizando métodos diferentes e assim obteve alguns indicadores de custo. O projeto A foi desenvolvido sob gestão tipo *ad hoc* (os requisitos não foram documentados de maneira correta). O projeto B seguiu orientações para definir requisitos, testar os requisitos e realizar inspeção de projeto e código. Vide Figura 3.2 e Figura 3.3 que mostram os gráficos com resultados de dois projetos.

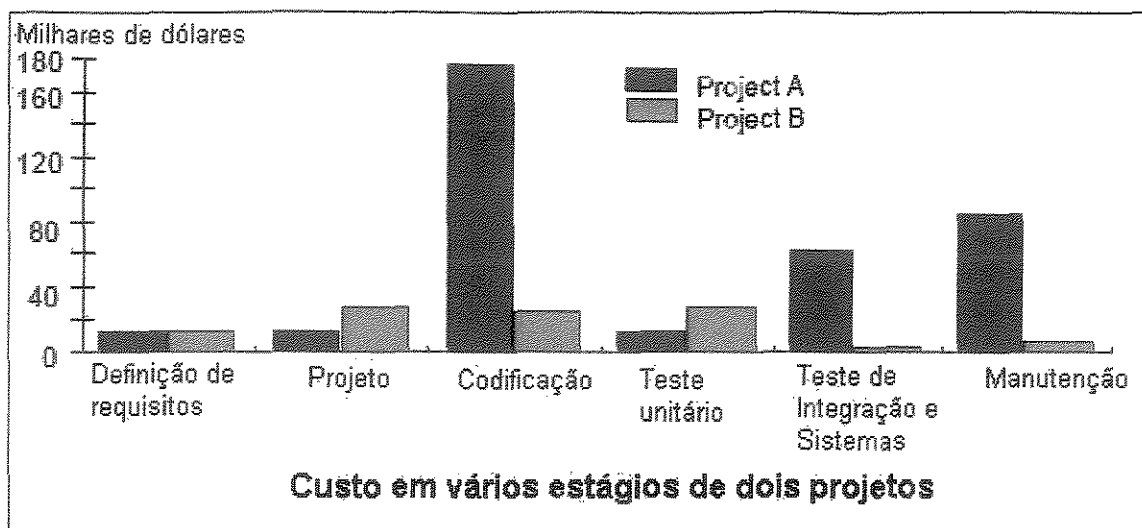


Figura 3.2 – Recursos gastos em dólares (Figura adaptada [HERBS94])

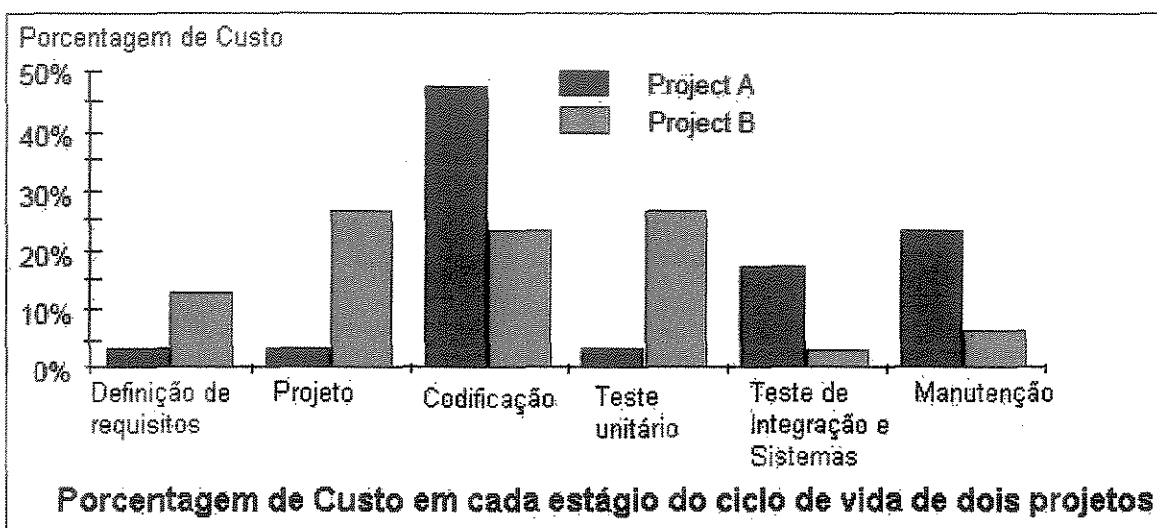


Figura 3.3 – Porcentagem de custo (Figura adaptada [HERBS94])

A Figura 3.3 mostra uma elevação no custo (esforço) das fases de análise e projeto em contraste com uma considerável redução de custo na codificação do Projeto B, assim como um aumento de custo na fase de teste unitário contrastando com uma grande redução nas fases de testes de integração e sistêmicos. A partir dessa amostra conclui-se que o Projeto B, que se preocupou mais com atividades de requisitos utilizando boas práticas como inspeção, custou menos que o Projeto A que não utilizou boas práticas como documentação de requisitos.

O artigo [BAUME92] descreve um conjunto de medições de software que são compatíveis com as práticas descritas no modelo **SW-CMM** (*Software – Capability Maturity Model*). No nível 3 (*Defined*) na atividade de revisão entre pares. Esse artigo apresenta gráficos relacionando defeitos encontrados nas atividade do ciclo de vida de software (Figura 3.4 e Figura 3.5).

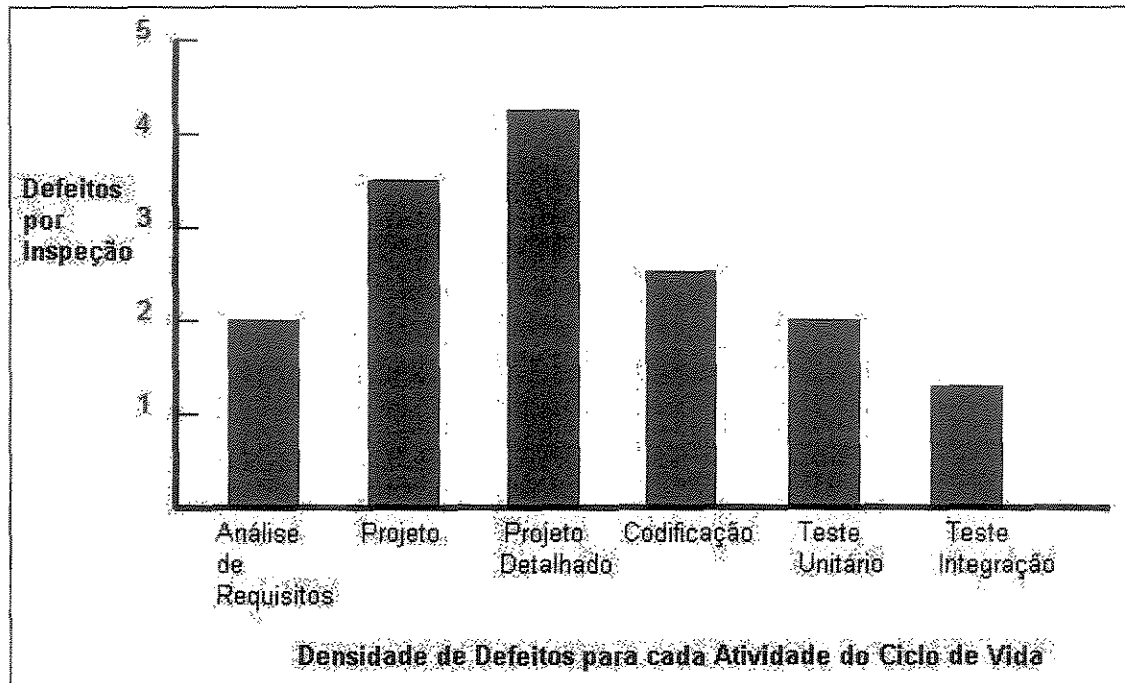


Figura 3.4 – Defeitos por atividade do ciclo de vida (Figura adaptada [BAUME92])

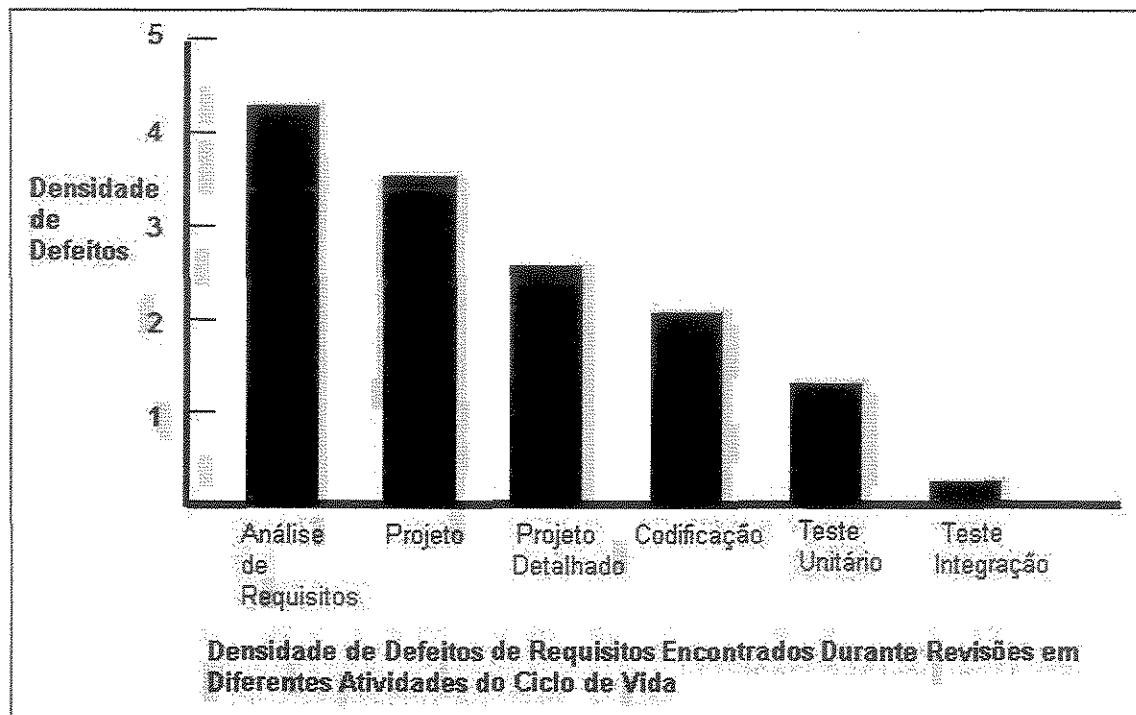


Figura 3.5 – Defeitos encontrados em revisão de pares (Figura adaptada [BAUME92])

O gráfico ilustrado na Figura 3.4 mostra a densidade de defeitos por inspeção observada em cada atividade do ciclo de vida e o gráfico da Figura 3.5 mostra o número de defeitos detectados em requisitos durante cada atividade do ciclo de vida. Quando a maioria dos defeitos é encontrada nas fases mais avançadas do ciclo de vida, deve-se melhorar o processo de definição de requisitos e realizar revisões (*peer review*) nos estágios iniciais do ciclo de vida.

Prevenção de defeitos [BAUME92] deve ser o objetivo de todos os envolvidos em um projeto. Como fazer a prevenção de defeitos: As causas dos defeitos no produto que são recorrentes nas atividades do projeto de software são identificadas e eliminadas. O quadro ilustrado na Figura 3.6 mostra onde os defeitos são injetados e detectados nas atividades do ciclo de vida, por exemplo 43 defeitos detectados em requisitos e 106 (soma da primeira coluna) defeitos injetados por requisitos durante as fases do ciclo de vida.

Inseridos Detectados	Requisitos	Projeto	Projeto Detalhado	Código
Requisitos	43			
Projeto	27	32		
Projeto Detalhado	14	17	55	
Código	3	7	26	73
Teste Unitário	3	4	17	58
Teste de Integração	4	5	4	18
Teste de Aceitação	12	10	2	12

Defeitos Inseridos e Detectados pela Atividades do Ciclo de Vida

Figura 3.6 – Defeitos inseridos e detectados (Figura adaptada [BAUME92])

A Sematech [SEMAT95] contém uma apresentação institucional sobre o tema: desenvolvimento e re-usabilidade de software de alta-qualidade. Na seção relacionada a requisitos são apresentados gráficos (vide Figura 3.7) que mostram quando os defeitos foram introduzidos (inclusive requisitos) e quando os defeitos foram encontrados e declara que 25 a 80% das falhas estão relacionadas a erros em definição de requisitos.

25% a 80% das falhas são relacionadas a erros na definição

Custo para Corrigir uma Falha (Falha criada durante a fase de requisitos)

Durante Fase de Requisitos	\$ 195
Durante Fase de Projeto	\$ 489
Durante Fase de Codificação	\$ 997
Durante Fase de Teste	\$7.136
Durante Fase de Manutenção	Muito mais!

SEMATECH

Figura 3.7 – Custo para correção de falhas (Figura adaptada [SEMAT95])

A Figura 3.7 também apresenta uma tabela que mostra quanto custa corrigir uma falha em determinadas fases de um projeto. Para ilustrar, a tabela da Figura 3.7 diz que para corrigir uma falha na fase de requisitos custa \$195 e para corrigir a falha na fase de teste custa \$7.136, ou seja, quanto antes a falha for reparada menor o custo para o projeto. Assim sendo, podemos analisar que as boas práticas de requisitos que procuram evitar a propagação de falhas para outras fases do projeto, práticas como inspeção, revisão e documentação de requisitos, podem trazer benefícios para o projeto e um desses benefícios pode estar diretamente relacionado ao custo do projeto como ilustrado na Figura 3.7.

O artigo [BUREN98] não relata dados experimentais em projetos de software, mas descreve teoricamente as atividades de requisitos citando alguns aspectos importantes e relacionando-os com as normas e modelos de qualidade e processo de software, como segue:

- **Gestão de requisitos:** define a atividade de gestão de requisitos citando a importância de sua relação com a atividade de elicitação (uma boa elicitação pode reduzir retrabalho pela diminuição das mudanças em requisitos). Como tecnologias de gestão de requisitos esse artigo cita o modelo **SW-CMM** e referencia os principais pontos relacionados à gestão de requisitos nos níveis de maturidade 2 e 3.

-
- Documentação de Requisitos: define o que é documentação de requisitos citando padrões de especificação de requisitos conhecidos como os modelos IEEE 12207 e IEEE-STD 830-1993.

3.1 Considerações finais

Este capítulo apresentou um resumo de alguns trabalhos relacionados ao tema central e aos objetivos desta dissertação. Como resultado obtido dessa pesquisa podem ser citados alguns aspectos importantes:

- Boas práticas relacionadas à engenharia de requisitos podem ser encontradas em modelos e normas de qualidade.
- A aplicação de boas práticas de engenharia de requisitos em projetos de software tende a diminuir problemas encontrados no software.
- As boas práticas relacionadas à engenharia de requisitos também podem evitar um grande aumento no custo, prazo e cronograma de projetos de software.

A análise do resultado dessa pesquisa ajudou na consolidação da idéia desenvolvida em torno dos objetivos pretendidos (descritos no Capítulo 1). A ausência de publicações que tratam diretamente sobre a relevância das diversas práticas de requisitos através de dados experimentais reforçou a motivação de realizar a investigação apresentada no Capítulo 8.

Capítulo 4 Modelos CMM e CMMI

Relacionados com o desenvolvimento de software bem sucedido (desenvolver software planejado, de baixo custo, cumprindo prazo, com qualidade e principalmente atendendo aos requisitos), alguns modelos de qualidade e processo de software foram desenvolvidos a fim de auxiliar as organizações a obter o sucesso no desenvolvimento de sistemas de software. Dois desses modelos foram desenvolvidos pelo **SEI** (*Software Engineering Institute*): o **SW-CMM** (*Software - Capability Maturity Model*) e o **CMMI** (*Capability Maturity Model Integration*).

Este capítulo apresenta um resumo dos modelos **SW-CMM** e **CMMI** descrevendo suas principais características, os objetivos de cada área chave de processo, e mais detalhadamente as questões diretamente relacionadas a requisitos por se tratar do tema central deste trabalho.

4.1 Modelo SW-CMM

SW-CMM é o modelo de maturidade de capacidade¹ de software e foi originado a partir de iniciativa do **DoD** (*Department of Defense*) norte-americano. A Universidade de *Carnegie Mellon* em Pittsburgh, nos Estados Unidos, foi escolhida para administrar o **SEI**. A criação desse modelo está baseada na visão de Watts Humphrey, primeiro diretor do Programa de Processo de Software do **SEI**. As primeiras versões estruturadas de Humphrey foram descritas em relatórios técnicos do **SEI** [HUMP87a] e [HUMP87b], artigos [HUMPH88] e em seu livro "*Managing the Software Process*" [HUMPH89]. Em 1993 foi publicada a versão 1.1 do **SW-CMM**, em vigor até hoje [TR02493] e [TR02593].

O propósito do **SW-CMM** é fornecer um guia que instrui as organizações a controlar processos de desenvolvimento e manutenção de software. Baseado em melhoria contínua do processo, visa conduzir projetos com sucesso atendendo aos requisitos de prazo, custo e qualidade.

O modelo **SW-CMM** objetiva ajudar as organizações de software a melhorar a maturidade de seus processos de software. Esse modelo descreve práticas que podem

¹ O termo em inglês é *capability*, neste texto está traduzido para o termo "capacidade".

ser utilizadas pelas organizações com a finalidade de controlar processos para desenvolver e manter software e também pode ser utilizado como um guia para evoluir em gestão e engenharia de software.

4.1.1 Conceitos básicos: Níveis de maturidade

O modelo **SW-CMM** está estruturado em níveis de maturidade definidos do nível 1 ao nível 5. E cada nível de maturidade é subdividido em áreas chave de processo [TR02493].

Os níveis de maturidade definem uma escala ordinal para medir a maturidade do processo de software de uma organização. Para uma organização alcançar um determinado nível de maturidade em seus processos, essa organização deve ter previamente alcançado os níveis anteriores, pois os níveis de maturidade são cumulativos.

O nível 1 é definido no modelo **SW-CMM** como nível inicial, ou seja, toda organização que não possui processos definidos e controlados encontram-se no nível 1. Nesse nível, o processo de software é caracterizado como *ad hoc* e o sucesso depende de esforço individual.

O nível 2 é definido no modelo **SW-CMM** como nível repetível. Esse nível é alcançado quando os processos básicos de gestão de projeto são estabelecidos para acompanhar custo, cronograma e requisitos, a fim de que se possam repetir sucessos anteriores em projetos com aplicações similares.

O nível 3 é definido no modelo **SW-CMM** como o nível definido, e é alcançado quando o processo de software para as atividades técnicas e atividades de gestão é documentado, padronizado e integrado em um processo de software padrão institucionalizado. Todos os projetos utilizam uma versão aprovada do processo de software padrão da organização para desenvolver e manter o software.

O nível 4 é definido no modelo **SW-CMM** como o nível gerenciado. Esse nível é alcançado quando medições detalhadas do processo de software e da qualidade do produto são coletadas e o processo de software e seus produtos são quantitativamente compreendidos e controlados.

O nível 5 é definido no modelo **SW-CMM** como o nível em otimização. Nesse nível a melhoria contínua do processo é propiciada pelo feedback quantitativo do processo e pelas idéias e tecnologias inovadoras.

4.1.2 Conceitos básicos: Áreas chave de processo

Com exceção feita ao nível 1 do modelo **SW-CMM**, cada nível de maturidade é decomposto em várias áreas chave de processo, que indicam as áreas nas quais uma organização deve focar para a melhoria de seu processo de software. As áreas chave de processo identificam os assuntos que devem ser tratados para se obter um determinado nível de maturidade.

O modelo **SW-CMM** define que cada área chave de processo identifica um grupo de atividades relacionadas, as quais, quando executadas coletivamente, atingem um conjunto de objetivos considerados importantes para aumentar a capacidade do processo [TR02593]. O caminho para o cumprimento dos objetivos² de uma área chave de processo pode diferir entre projetos, de acordo com as diferenças de domínios de aplicação ou de ambientes. Todos os objetivos de uma área chave de processo devem ser atingidos para que se satisfaça aquela área chave de processo. Quando os objetivos de uma área chave de processo são cumpridos de maneira continuada em um projeto, a organização pode ser considerada como tendo institucionalizado a capacidade do processo caracterizada por esta área chave de processo.

4.1.2.1 Áreas chave de processo do nível 1

Como o nível 1 do **SW-CMM** reflete o nível inicial, onde as organizações ainda não têm processos definidos, não há áreas chave descritas para esse nível.

² O termo em inglês é *GOAL* (neste trabalho encontra-se traduzido como “objetivo” ou “meta”).

4.1.2.2 Áreas chave de processo do nível 2

As áreas chave de processo do nível 2 do **SW-CMM** são focadas em assuntos de projeto de software relacionados com o estabelecimento de controles básicos de gestão de projeto. Os objetivos de cada área chave de processo para o nível 2 são descritos na Tabela 4-1.

Tabela 4-1 – Resumo das Áreas Chave de Processo do Nível 2

Área Chave de Processo	Objetivo
Gestão de Requisitos (<i>Requirements Management</i>)	Estabelecer um entendimento comum entre o cliente e o projeto de software sobre os requisitos do cliente que serão tratados pelo projeto de software.
Planejamento de Projeto de Software (<i>Software Project Planning</i>)	Estabelecer planos para a execução das atividades técnicas de software e para a gestão do projeto de software.
Acompanhamento de Projeto de Software (<i>Software Project Tracking and Oversight</i>)	Estabelecer visibilidade adequada sobre o andamento do projeto, de tal maneira que a gerência possa tomar ações efetivas quando o desempenho do projeto de software sofre um desvio significativo com relação ao planejado.
Gestão de Subcontratação de Software (<i>Software Subcontract Management</i>)	Selecionar subcontratados qualificados e gerenciá-los de modo efetivo.
Garantia de Qualidade de Software (<i>Software Quality Assurance</i>)	Fornecer um controle com visibilidade apropriada sobre os processos em uso pelo projeto de software e sobre os produtos desenvolvidos. A Garantia de Qualidade de Software é parte integrante da maioria dos processos de gestão de software.
Gestão de Configuração de Software (<i>Software Configuration Management</i>)	Estabelecer e manter a integridade dos produtos do projeto de software ao longo de todo o seu ciclo de vida.

4.1.2.3 Áreas chave de processo do nível 3

As áreas chave de processo do nível 3 do **SW-CMM** abordam questões de projeto e questões da organização referente à institucionalização de processos. Os objetivos de cada uma das áreas chave de processo do nível 3 são descritos na Tabela 4-2.

Tabela 4-2 – Resumo das Áreas Chave de Processo do Nível 3

Área Chave de Processo	Objetivo
Foco Sobre os Processos da Organização (<i>Organization Process Focus</i>)	Estabelecer a responsabilidade organizacional pelas atividades de processo de software que melhorem a capacidade geral dos processos de software da organização.
Definição do Processo da Organização (<i>Organization Process Definition</i>)	Desenvolver e manter um conjunto de artefatos de processo de software que melhorem o desempenho do processo nos projetos e forneça uma base para benefícios cumulativos e de longo prazo para a organização. Esses recursos fornecem uma base estável que pode ser institucionalizada por meio de mecanismos tais como treinamento, que será descrito em Programa de Treinamento.
Programa de Treinamento (<i>Training Program</i>)	Desenvolver as habilidades e conhecimento dos indivíduos, de tal modo que eles possam executar suas tarefas de modo eficaz e eficiente.
Gestão Integrada de Software (<i>Integrated Software Management</i>)	Integrar as atividades de gestão e de engenharia de software em um processo de software coerente e definido que é desenvolvido adequadamente a partir do processo de software padrão da organização.
Engenharia de Produto de Software (<i>Software Product Engineering</i>)	Executar de maneira consistente um processo de engenharia bem definido que integre todas as atividades de engenharia de software para produzir produtos de software corretos e consistentes de modo eficiente.
Coordenação Intergrupos (<i>Intergroup Coordination</i>)	Estabelecer um meio para que o grupo de engenharia de software participe ativamente com outros grupos de modo que o projeto seja capaz de satisfazer as necessidades do cliente, de maneira eficiente e eficaz.
Revisão entre Pares (<i>Peer Reviews</i>)	Remover defeitos dos produtos de software eficientemente e o mais cedo possível durante seu ciclo de vida.

4.1.2.4 Áreas chave de processo do nível 4

As áreas chave de processo do nível 4 do **SW-CMM** objetivam o estabelecimento de um entendimento quantitativo sobre o processo de software e sobre os produtos de software que estão sendo construídos. As duas áreas chave de processo desse nível de maturidade, Gestão Quantitativa do Processo e Gestão da Qualidade de Software, são altamente interdependentes, como descrito na Tabela 4-3.

Tabela 4-3 – Resumo das Áreas Chave de Processo do Nível 4

Área Chave de Processo	Objetivo
Gestão Quantitativa do Processo (<i>Quantitative Process Management</i>)	Controlar quantitativamente o desempenho do processo do projeto de software. O desempenho de um processo de software representa os resultados reais obtidos seguindo-se um processo de software.
Gestão da Qualidade de Software (<i>Software Quality Management</i>)	Desenvolver um entendimento quantitativo da qualidade dos produtos de software do projeto e obter objetivos de qualidade específicos.

4.1.2.5 Áreas chave de processo do nível 5

As áreas chave de processo do nível 5 do **SW-CMM** cobrem as questões que tanto a organização quanto o projeto devem tratar para implementar a melhoria de processo de software de forma contínua e que possa ser medida. Os objetivos de cada uma das áreas chave de processo para o nível 5 são descritos na Tabela 4-4.

Tabela 4-4 – Resumo das Áreas Chave de Processo do Nível 5

Área Chave de Processo	Objetivo
Prevenção de Defeitos (<i>Defect Prevention</i>)	Identificar as causas de defeitos e prevenir a sua recorrência.
Gestão de Mudança de Tecnologia (<i>Technology Change Management</i>)	Identificar novas tecnologias vantajosas (ferramentas, métodos e processos) e transferi-las para a organização de maneira ordenada, como descrito na Gestão de Alteração de Processo. O foco da Gestão de Alteração de Tecnologia é inovar de modo eficiente num mundo em constante alteração.
Gestão de Mudança de Processo (<i>Process Change Management</i>)	Melhorar continuamente o processo de software utilizado na organização com a intenção de melhorar a qualidade do software, aumentar a produtividade e diminuir o tempo de desenvolvimento do produto.

4.1.3 A questão requisito e o modelo SW-CMM

Como visto nos itens anteriores, o modelo **SW-CMM** é dividido em níveis de maturidade e áreas chave de processo. Nesse contexto, o modelo **SW-CMM** possui níveis de maturidade e algumas áreas chave de processo que apresentam maior relacionamento com a questão requisito. No nível 2 do **SW-CMM** há uma área chave de processo (Gestão de Requisitos) exclusiva para tratar gestão de requisitos e no nível 3 há uma área chave de processo (Engenharia de Produto de Software) que envolve atividades relacionadas a requisitos. Outros níveis de maturidade e outras áreas chave de processo também fazem menção às práticas que envolvem a questão requisito, porém com menor intensidade.

Na área chave de processo de Gestão de Requisitos no nível 2 do **SW-CMM** são definidos dois objetivos a serem alcançados. O primeiro objetivo reflete a necessidade dos requisitos de software (requisitos do sistema alocados ao software) serem controlados com a finalidade de se estabelecer uma *baseline* para o desenvolvimento do software e também para uso gerencial. O segundo objetivo reflete a necessidade dos planos, produtos e atividades de software serem mantidos consistentes com os requisitos de software. Uma organização que deseja obter o nível 2 deve atingir completamente esses dois objetivos entre outros. E para atingir esses objetivos, o modelo **SW-CMM**

define um conjunto de boas práticas de gestão de requisitos. A seguir são listadas as práticas que tratam a questão requisito no modelo **SW-CMM**.

4.1.3.1 Práticas do Nível 2: Gestão de Requisitos

- Commitment1:* O projeto deve seguir uma política estabelecida pela organização para o gerenciamento de requisitos de sistema alocados ao software. Os requisitos alocados devem ser documentados, revisados e os artefatos do projeto (planos, os produtos e as atividades de software) devem manter-se consistentes com os requisitos alocados.
- Ability1:* Para cada projeto, é definida a responsabilidade pelos requisitos de sistema e a alocação dos mesmos ao hardware, software e a outros componentes do sistema. Essa responsabilidade envolve o gerenciamento, a documentação e a alocação dos requisitos e a efetivação de alterações dos requisitos de sistema como também a alocação dos mesmos.
- Ability2:* Os requisitos alocados (requisitos não técnicos, requisitos técnicos e critérios de aceitação) são documentados.
- Ability3:* Recursos e orçamentos adequados são providos para a gerência dos requisitos alocados. Pessoas que têm experiência e competência no domínio da aplicação e em engenharia de software são designadas para gerenciar os requisitos alocados e ferramentas para suportar as atividades de gestão de requisitos são disponibilizadas.
- Ability4:* Os membros da equipe de engenharia de software e outros grupos afetados são treinados para realizar suas atividades de gestão de requisitos.
- Activity1:* A equipe de engenharia de software revisa os requisitos alocados antes dos mesmos serem incorporados ao projeto de software.
- Activity2:* A equipe de engenharia de software deve utilizar os requisitos alocados como base para os planos, produtos e atividades de software. Essa prática descreve a importância em gerenciar requisitos, pois as atividades de desenvolvimento requerem os requisitos do projeto como insumo, e, portanto é necessário que os requisitos alocados sejam gerenciados e controlados.
- Activity3:* As alterações, nos requisitos alocados, são revisadas e incorporadas ao projeto de software. Esta prática reflete a preocupação em manter qualidade e realizar alterações em requisitos de maneira controlada a fim de manter os artefatos do projeto aderente às alterações sofridas.
- Measurement1:* As medições são realizadas e utilizadas para determinar a situação das atividades de gestão de requisitos alocados.

- Verification1:* As atividades de gestão de requisitos alocados são analisadas periodicamente por um gerente superior.
- Verification2:* As atividades de gestão de requisitos alocados são analisadas periodicamente por um gerente de projeto periodicamente ou motivadas por um evento.
- Verification3:* A equipe de garantia da qualidade de software deve analisar e proceder a auditoria das atividades e dos produtos de gestão dos requisitos alocados e reportar os resultados.

4.1.3.2 Práticas do Nível 3: Engenharia de Produto de Software

- Commitment1:* O projeto segue uma política organizacional definida para executar as atividades de engenharia de software. Os planos, produtos e tarefas de software são rastreáveis para os requisitos de sistema alocados ao software.
- Activity2:* Os requisitos de software são desenvolvidos, mantidos, verificados e documentados através de uma análise sistemática dos requisitos alocados de acordo com os processos de software definidos para o projeto. Métodos efetivos de análise de requisitos são usados para identificar e derivar os requisitos de software. O resultado da análise dos requisitos é documentado. Os requisitos de software são analisados para assegurar que são factíveis e apropriados para serem implementados em software, são claros e completos, consistentes uns com os outros e testáveis. Os requisitos de software são documentados. O grupo responsável pelo teste de sistema e pelo teste de aceitação do software analisa cada um dos requisitos de software para verificar se podem ser testados. Os métodos para verificação e validação de cada requisito de software são identificados e documentados. O documento de requisitos de software deve sofrer a revisão entre pares antes de ser considerado completo. O documento de requisitos de software é revisado e aprovado e quando apropriado, também é revisado pelo cliente e usuários finais. O documento de requisitos de software é colocado sob gestão de configuração. E os requisitos de software são alterados quando os requisitos alocados são alterados.
- Activity3:* O projeto de software é desenvolvido, mantido, documentado e verificado de acordo com o processo de software definido para o projeto, para acomodar os requisitos de software a fim de auxiliar a codificação. O documento de projeto de software é alterado quando um requisito de software é alterado.

- Activity4:* A codificação do software é desenvolvida, mantida, documentada e verificada, de acordo com o processo de software definido para o projeto, para implementar os requisitos de software e o projeto de software. O código é alterado quando requisitos de software ou projeto de software são alterados.
- Activity5:* Testes de software são executados de acordo com o processo de software definido para o projeto. Planos de testes, procedimentos de testes e casos de testes são alterados quando requisitos alocados, requisitos de software, projeto de software ou código são alterados.
- Activity6:* Testes de integração de software são planejados e executados de acordo com o processo de software definido para o projeto. Os casos de teste de integração e procedimentos de testes são revisados pelo indivíduo responsável pelos requisitos de software, projeto de software, testes de sistema e testes de aceitação.
- Activity7:* Os testes de sistema e aceitação de software são planejados e executados para demonstrar que o software satisfaz aos requisitos. Os resultados dos testes são documentados e usados como base para determinar se o software satisfaz aos requisitos.
- Activity10:* A consistência dos produtos de trabalho de software é mantida, incluindo planejamento de software, descrições de processos, requisitos alocados, requisitos de software, projeto de software, código, planos de teste e procedimentos de teste. Os requisitos de software, projeto, código e casos de teste são rastreados até a fonte de onde são derivados e para os produtos resultantes das atividades de engenharia de software.
- Verification3:* O grupo de garantia da qualidade de software realiza revisão e auditoria das atividades e produtos de trabalho da Engenharia de Produto de Software e relata os resultados. Os requisitos de software são revisados para assegurar que estão completos, consistentes, factíveis e testáveis.

Outras práticas e sub-práticas também relacionadas a requisitos, definidas no modelo **SW-CMM**, são listadas e identificadas no ANEXO A: SW-CMM.

4.2 Modelo CMMI

CMMI é o modelo de maturidade de capacidade de software integrado desenvolvido pelo **SEI** (*Software Engineering Institute*). O propósito desse modelo é fornecer um guia que instrui as organizações a desenvolver melhorias nos processos organizacionais e melhorias na habilidade de gerenciar a aquisição, desenvolvimento e manutenção de produtos e serviços. Este modelo possui duas representações, por estágios (*Staged Representation*) [TR02902] ou continuada (*Continuous Representation*) [TR02802]. Estas duas representações apresentam o mesmo conteúdo principal, porém estão estruturadas de maneira diferente. Também é possível verificar os níveis de maturidade definidos nesse modelo. Cada nível de maturidade é subdividido em áreas de processo.

Desde 1991 o modelo **CMM** desenvolveu várias disciplinas. Algumas das mais notáveis incluem engenharia de sistemas, engenharia de software, aquisição de software, gerência de recursos e desenvolvimento, e produto integrado e desenvolvimento de processo. O **SEI** chegou a anunciar algumas vezes a versão 2.0 do **SW-CMM**, mas devido à necessidade de integrar essas diversas disciplinas, o modelo **CMMI** foi desenvolvido para solucionar o problema do uso de múltiplos modelos **CMM**.

4.2.1 Conceitos básicos: CMMI

Para obter um melhor entendimento das práticas descritas pelo modelo **CMMI** é importante conhecer sua estrutura. A Tabela 4-5 e a Tabela 4-6 apresentam os níveis de maturidade e capacidade do modelo **CMMI** respectivamente.

Tabela 4-5 – CMMI: Representação por Estágios

<i>Maturity Level</i>	<i>Staged Representation Maturity Level</i>
1	<i>Initial</i>
2	<i>Managed</i>
3	<i>Defined</i>
4	<i>Quantitatively Managed</i>
5	<i>Optimizing</i>

Tabela 4-6 – CMMI: Representação Contínua

<i>Capability Level</i>	<i>Continuous Representation Capability Level</i>
0	<i>Incomplete</i>
1	<i>Performed</i>
2	<i>Managed</i>
3	<i>Defined</i>
4	<i>Quantitatively Managed</i>
5	<i>Optimizing</i>

O Nível de Maturidade (*Maturity Level*³) representa a maturidade da organização. O Nível de Capacidade (*Capability Level*⁴) representa a capacidade de um determinado processo. Para exemplificar, segue a ilustração abaixo, composta dos exemplos 1 e 2:

Exemplo 1: Representação por estágios

Supondo que as PAs⁵ do nível 2 sejam as PAs 1, 2 e 3, as PAs do nível 3 sejam as PAs 4 e 5 e as PAs do nível 4 sejam as PAs 6 e 7. E considerando que:

SG = *Specific Goal*

GG1 = *Generic Goal* de nível 1

GG2 = *Generic Goal* de nível 2

GG3 = *Generic Goal* de nível 3

GG4 = *Generic Goal* de nível 4

GG5 = *Generic Goal* de nível 5

Para a organização obter o nível 2 de maturidade, os seguintes itens marcados na Tabela 4-7 devem ser atingidos, ou seja, as PAs do nível 2 devem ter satisfeitos os objetivos específicos (SG) de cada PA e os objetivos genéricos dos níveis 1 e 2 (GG1 e GG2).

Tabela 4-7 – Exemplo Nível 2 Representação por estágios

	PA1	PA2	PA3	PA4	PA5	PA6	PA7
SG	X	X	X				
GG1	X	X	X				
GG2	X	X	X				
GG3							
GG4							
GG5							

³ *Maturity levels, which belong to the staged representation, apply to an organization's overall maturity. There are five maturity levels, numbered 1 through 5. Each maturity level comprises a predefined set of process areas [TR02802].*

⁴ *Capability levels, which belong to the continuous representation, apply to an organization's process-improvement achievement for each process area. There are six capability levels, numbered 0 through 5. Each capability level corresponds to a generic goal and a set of generic and specific practices [TR02802].*

⁵ PA = Process Area

Para a organização obter o nível 3 de maturidade, os seguintes itens marcados na Tabela 4-8 devem ser atingidos, ou seja, as PAs do nível 2 e 3 devem ter satisfeitos os objetivos específicos (SG) de cada PA e os objetivos genéricos dos níveis 1, 2 e 3 (GG1, GG2 e GG3).

Tabela 4-8 – Exemplo Nível 3 Representação por estágios

	PA1	PA2	PA3	PA4	PA5	PA6	PA7
SG	X	X	X	X	X		
GG1	X	X	X	X	X		
GG2	X	X	X	X	X		
GG3	X	X	X	X	X		
GG4							
GG5							

Exemplo 2: Representação contínua

Para atingir:

O nível de capacidade 2 para a PA1: satisfazer SG e GG1 e GG2 da PA1.

O nível de capacidade 3 para a PA4: satisfazer SG, GG1, GG2 e GG3 da PA4.

O nível de capacidade 5 para a PA7: satisfazer SG, GG1, GG2, GG3, GG4 e GG5 da PA4.

Sendo que os seguintes itens marcados na Tabela 4-9 devem ser atingidos.

Tabela 4-9 – Exemplo Representação contínua

	PA1	PA2	PA3	PA4	PA5	PA6	PA7
SG	X			X			X
GG1	X			X			X
GG2	X			X			X
GG3				X			X
GG4							X
GG5							X

4.2.2 Áreas de processo do CMMI

Na representação por estágios, áreas de processo são agrupadas em níveis de maturidade [TR02902].

4.2.2.1 CMMI Nível 1 - Inicial

No nível 1 do **CMMI**, os processos são caóticos e tipicamente “*ad hoc*”. Como este nível reflete o nível inicial, onde as organizações ainda não têm processos definidos, não há áreas de processo descritas para esse nível.

4.2.2.2 CMMI Nível 2 - Gerenciado

No nível 2 do **CMMI**, os requisitos são gerenciados e os processos são planejados, executados, medidos e controlados. Nesse nível, requisitos, processos, produtos de trabalho e serviços são gerenciados. Os principais objetivos das áreas de processo do nível 2 são descritos na Tabela 4-10.

Tabela 4-10 – Resumo das Áreas de Processo do Nível 2

Área de Processo	Objetivo
Gestão de Requisitos (<i>Requirements Management</i>)	Gerenciar os requisitos dos produtos e componentes de produto de projetos e identificar inconsistências entre esses requisitos e os planos e produtos de trabalho de projetos.
Planejamento de Projeto (<i>Project Planning</i>)	Estabelecer e manter planos que definem atividades de projeto.
Monitoramento e Controle de Projeto (<i>Project Monitoring and Control</i>)	Prover uma compreensão do progresso do projeto de forma que, ações corretivas apropriadas, possam ser tomadas quando o desempenho do projeto divergir significativamente do plano.
Gestão de Contratos de Clientes (<i>Supplier Agreement Management</i>)	Gerenciar a aquisição de produtos de clientes para qual existem contrato formal.
Medição e Análise (<i>Measurement and Analysis</i>)	Desenvolver e sustentar uma capacidade de medida que é usada para apoiar necessidades de informação de gerência.
Garantia da Qualidade de Processo e Produto (<i>Process and Product Quality Assurance</i>)	Prover pessoal e gerência com objetivos internos aos processos e produtos de trabalho associados.
Gestão de Configuração (<i>Configuration Management</i>)	Estabelecer e manter a integridade de produtos de trabalho que usam identificação de configuração, controle de configuração, estado de configuração, e auditorias de configuração.

4.2.2.3 CMMI Nível 3 - Definido

No nível 3 do **CMMI**, os processos são bem caracterizados e entendidos, e são descritos em padrões, procedimentos, ferramentas e métodos. As áreas de processo do nível 3 são descritas na Tabela 4-11.

Tabela 4-11 – Resumo das Áreas de Processo do Nível 3

Área de Processo	Objetivo
Desenvolvimento de Requisitos (<i>Requirement Development</i>)	Produzir e analisar requisitos do cliente, requisitos de produto e componente de produto.
Solução Técnica (<i>Technical Solution</i>)	Projetar, desenvolver, e implementar soluções para os requisitos. Soluções, projetos, e implementações de produtos, componentes de produto, e produtos relacionados ao processo do ciclo de vida.
Integração de Produto (<i>Product Integration</i>)	Agrupar o produto a partir dos componentes de produto, assegurando que o produto integrado funciona corretamente e possa ser entregue.
Verificação (<i>Verification</i>)	Assegurar que os produtos de trabalho satisfazem seus requisitos.
Validação (<i>Validation</i>)	Demonstrar que os produtos ou componentes de produtos cumprem completamente seu uso planejado quando colocado sob ambiente planejado.
Foco Sobre os Processos da Organização (<i>Organizational Process Focus</i>)	Planejar e implementar melhorias no processo organizacional baseado em uma compreensão completa dos pontos fortes e pontos fracos dos processos da organização e processos ativos.
Definição do Processo da Organização (<i>Organizational Process Definition</i>)	Estabelecer e manter um conjunto de processos organizacionais ativos e utilizáveis.
Treinamento Organizacional (<i>Organization Training</i>)	Desenvolver as habilidades e conhecimento das pessoas de modo que elas possam executar seus papéis de maneira efetiva e eficaz.

Área de Processo	Objetivo
Gestão Integrada de Projeto (<i>Integrated Project Management</i>)	Estabelecer e gerenciar o projeto e o pessoal envolvido de acordo com um processo definido que é advindo dos padrões de processos da organização.
Gestão de Risco (<i>Risk Management</i>)	Identificar problemas antes que eles ocorram, de forma que as atividades de controle de riscos possam ser planejadas e executadas quando necessário durante a vida do projeto.
Análise de Decisão e Resolução (<i>Decision Analysis and Resolution</i>)	Analisar possíveis decisões usando um processo de avaliação formal que avalia alternativas identificadas contra critérios estabelecidos.

4.2.2.4 CMMI Nível 4 - Quantitativamente gerenciado

No nível 4 do **CMMI**, são selecionados subprocessos que significativamente contribuem para o desempenho global do processo. Estes subprocessos selecionados são controlados usando técnicas quantitativas e estatísticas. As áreas de processo do nível 4 são descritas na Tabela 4-12.

Tabela 4-12 – Resumo das Áreas de Processo do Nível 4

Área de Processo	Objetivo
Desempenho de Processo Organizacional (<i>Organizational Process Performance</i>)	Estabelecer e manter uma compreensão quantitativa do desempenho de um conjunto de processos padrões da organização em defesa da qualidade e objetivos de desempenho de processo, e prover os dados de desempenho de processo, <i>baselines</i> , e modelos para gerenciar os projetos da organização quantitativamente.
Gestão Quantitativa de Projeto (<i>Quantitative Project Management</i>)	Gerenciar, quantitativamente, os processos definidos do projeto para alcançar a qualidade estabelecida do projeto e os objetivos de desempenho do processo.

4.2.2.5 CMMI Nível 5 - Em otimização

No nível 5 do **CMMI**, os processos são continuamente melhorados baseado em uma compreensão quantitativa das causas comuns da variação inerente nestes processos. As áreas de processo do nível 5 são descritas na Tabela 4-13.

Tabela 4-13 – Resumo das Áreas de Processo do Nível 5

Área de Processo	Objetivo
Desenvolvimento e Inovação Organizacional (<i>Organizational and Innovation Deployment</i>)	Selecionar e desenvolver melhorias incrementais e inovadoras que melhorem as medidas dos processos da organização e tecnologias. As melhorias apóiam os objetivos da qualidade e desempenho de processo da organização como derivado dos objetivos de negócio da organização.
Análise Causal e Resolução (<i>Causal Analysis and Resolution</i>)	Identificar causas de defeitos e outros problemas e tomar ações para preveni-los e impedir de acontecer no futuro.

4.2.3 A questão requisito e o modelo CMMI

O nível 2 do **CMMI** apresenta uma área de processo diretamente relacionada à gestão de requisitos. Dentro dessa área de processo é definido um objetivo específico a ser alcançado. Esse objetivo reflete a necessidade do gerenciamento dos requisitos e a necessidade de identificação das inconsistências entre os planos e produtos de trabalho e os requisitos. E para atingir esse objetivo, o modelo **CMMI** define um conjunto de boas práticas de gestão de requisitos (Área de Processo *Requirements Management*) que estão descritas a seguir:

Área de processo: Requirements Management

- SP1.1: Obter o entendimento dos requisitos. Desenvolver o significado dos requisitos com quem os provê.
- SP1.2: Obter comprometimento aos requisitos. Obter comprometimento aos requisitos dos participantes do projeto.
- SP1.3: Gerenciar alterações nos requisitos.
- SP1.4: Manter rastreabilidade bidirecional entre os requisitos e os planos e produtos de trabalho do projeto.

- SP1.5: Identificar inconsistências entre os planos e produtos de trabalho do projeto e os requisitos.

O nível 3 do **CMMI** apresenta uma área de processo diretamente relacionada a desenvolvimento de requisitos. Dentro dessa área de processo são definidos três objetivos específicos a serem alcançados. O primeiro objetivo reflete a necessidade de desenvolver os requisitos do cliente. O segundo objetivo reflete a necessidade de desenvolver os requisitos do produto. E o terceiro objetivo reflete a necessidade de analisar e validar requisitos. E para atingir esses objetivos, o modelo **CMMI** define um conjunto de boas práticas de desenvolvimento de requisitos (*Área de Processo Requirements Development*) que estão descritas a seguir:

Área de processo: Requirements Development

- SP1.1: Extrair as necessidades dos envolvidos, as expectativas, regras, e interfaces para todas as fases do ciclo de vida de produto.
- SP1.2: Desenvolver os requisitos do cliente. Transformar as necessidades dos envolvidos, as expectativas, regras, e interfaces em requisitos de cliente.
- SP2.1: Estabelecer e manter requisitos de produto e de componente de produto, que estão baseados nos requisitos do cliente.
- SP2.2: Alocar requisitos para cada componente de produto.
- SP2.3: Identificar requisitos de interface.
- SP3.1: Estabelecer e manter conceitos operacionais e cenários associados.
- SP3.2: Estabelecer e manter uma definição da funcionalidade requerida.
- SP3.3: Analisar requisitos para assegurar que eles são necessários e suficientes.
- SP3.4: Analisar requisitos para equilibrar as necessidades dos envolvidos e regras.
- SP3.5: Validar requisitos usando diversas técnicas, quando apropriado, para assegurar que o produto resultante executará, como planejado, no ambiente do usuário.

Outras práticas e sub-práticas também relacionadas a requisitos, definidas no modelo **CMMI**, são listadas e identificadas no ANEXO B: CMMI.

4.3 Considerações finais

Neste capítulo vimos um resumo das principais práticas, relacionadas à questão requisitos, descritas nos modelos **SW-CMM** e **CMMI**.

Capítulo 5 Normas da família ISO 9000

Acrescentando a idéia de relacionar as práticas de requisitos dos modelos *SW-CMM* e *CMMI* vistos no capítulo anterior (Capítulo 4), este capítulo apresenta um resumo das normas de qualidade da família ISO 9000 e suas aplicações na área de requisitos. Nesse contexto, serão tratadas as normas NBR ISO 9000 [ISO9000], ISO/IEC 9000-3 [ISO9000-3], NBR ISO 9001 [ISO9001] e a norma NBR ISO 9004 [ISO9004].

As normas da família ISO 9000 foram desenvolvidas para apoiar organizações, de todos os tipos e tamanhos, na implementação e operação de sistemas eficazes de gestão da qualidade. A norma NBR ISO 9000 descreve os fundamentos de sistemas de gestão da qualidade e estabelece a terminologia para esses sistemas. A ISO/IEC 9000-3 fornece um guia para as organizações na aplicação da norma ISO 9001:2000 para aquisição, provisão, desenvolvimento, operação e manutenção de software de computador.

A norma NBR ISO 9001 está focada na eficácia do sistema de gestão da qualidade em atender aos requisitos do cliente. A norma NBR ISO 9004 fornece uma orientação para um sistema de gestão da qualidade com objetivos mais amplos do que a NBR ISO 9001, especificamente no que tange à melhoria contínua do desempenho global de uma organização e sua eficiência, bem como à sua eficácia. A norma NBR ISO 9004 é recomendada como uma orientação para organizações cuja alta direção deseja ir além dos requisitos estabelecidos na norma NBR ISO 9001, buscando melhoria contínua de desempenho, entretanto não tem propósito de certificação ou finalidade contratual.

Organizações com sistema de gestão da qualidade para desenvolver, operar e manter software baseado na ISO/IEC 9000-3 podem escolher utilizar orientações adicionais da ISO/IEC 12207. Muitos dos processos da ISO/IEC 12207 [ISO12207] (apresentada no Capítulo 6) podem ser usados para apoiar ou complementar os processos de NBR ISO 9001 a contento da organização.

5.1 Norma ISO 9000

A norma NBR ISO 9000 pode ser utilizada por organizações que buscam vantagens através da implementação de um sistema de gestão da qualidade ou que buscam a confiança nos seus fornecedores de que os requisitos de seus produtos serão atendidos. Oito princípios de gestão da qualidade formam a base para a norma NBR ISO 9000, como descrito a seguir:

- Foco no cliente: organizações dependem de seus clientes e, portanto, é recomendável que entendam as necessidades atuais e futuras do cliente, e seus requisitos, e procurem exceder suas expectativas.
- Liderança: líderes estabelecem objetivos e o rumo da organização. É recomendável que eles criem e mantenham um ambiente interno, no qual as pessoas possam estar totalmente envolvidas no propósito de atingir os objetivos da organização.
- Envolvimento de pessoas: pessoas de diferentes habilidades são a essência de uma organização e seu total envolvimento possibilita que as suas habilidades sejam usadas para o benefício da organização.
- Abordagem de processo: um resultado desejado é alcançado mais eficientemente quando as atividades e os recursos relacionados são gerenciados como um processo.
- Abordagem sistêmica para gestão: identificar, entender e gerenciar processos inter-relacionados como um sistema que contribui para a eficácia e eficiência da organização no sentido desta atingir os seus objetivos.
- Melhoria contínua: é recomendável que a melhoria contínua do desempenho global da organização seja seu objetivo permanente.
- Abordagem factual para tomada de decisão: decisões eficazes são baseadas na análise de dados e informações.
- Benefícios mútuos nas relações com fornecedores: uma organização e seus fornecedores são interdependentes, e uma relação de benefícios mútuos aumenta a habilidade de ambos em agregar valor.

A norma NBR ISO 9000 descreve os fundamentos de sistemas de gestão de qualidade que constituem o objeto da família NBR ISO 9000. Neste contexto os termos e definições relacionados a requisitos e aplicados a esta norma estão descritos no Capítulo 7, item 7.1.3.

5.2 Norma ISO 9001

A norma NBR ISO 9001:2000 promove a adoção de uma abordagem de processo para o desenvolvimento, implementação e melhoria da eficácia de um sistema de gestão de qualidade para aumentar a satisfação do cliente pelo atendimento aos seus requisitos. Essa norma especifica requisitos para um sistema de gestão de qualidade, quando uma organização necessita demonstrar sua capacidade para fornecer de forma coerente produtos que atendam os requisitos do cliente e os requisitos regulamentares aplicáveis, e quando uma organização objetiva aumentar a satisfação do cliente por meio da efetiva aplicação do sistema, incluindo processos para melhoria contínua do sistema e garantia da conformidade com os requisitos do cliente e requisitos regulamentares aplicáveis.

5.2.1 A questão requisitos e a norma ISO 9001

A norma NBR ISO 9001 determina quais são as implicações da direção de uma organização em relação ao bom atendimento aos requisitos e satisfação do cliente. O resumo das práticas relacionadas a requisitos está descrito a seguir:

Item 5.1 Comprometimento da direção:	A alta direção de uma organização deve fornecer evidência de seu comprometimento com o desenvolvimento e com a implantação do sistema de gestão de qualidade e com a melhoria contínua de sua eficácia mediante a comunicação à organização da importância em atender aos requisitos dos clientes, como também aos requisitos regulamentares e estatutários (requisitos em conformidade com leis).
Item 5.2 Foco no cliente:	A alta direção de uma organização deve assegurar que os requisitos dos clientes são determinados e atendidos com o propósito de aumentar a satisfação do cliente e desta maneira manter o foco no cliente.
Item 5.3 Política de qualidade:	A alta direção deve assegurar que política de qualidade inclui um comprometimento com o atendimento aos requisitos e com a melhoria contínua da eficácia do sistema de gestão da qualidade.

Item 5.6.3 Saídas da análise crítica:	As saídas da análise crítica pela direção devem incluir quaisquer decisões e ações relacionadas à melhoria do produto em relação aos requisitos do cliente.
Item 6.3 Infra-estrutura	A organização deve determinar, prover e manter a infra-estrutura necessária para alcançar a conformidade com os requisitos do produto.
Item 6.4 Ambiente de trabalho	A organização deve determinar e gerenciar as condições do ambiente de trabalho necessárias para alcançar a conformidade com os requisitos do produto.
Item 7.1 Planejamento da realização do produto	A organização, ao planejar a realização do produto, deve determinar como objetivos da qualidade e requisitos para o produto, critérios de aceitação do produto e deve determinar os registros necessários para fornecer evidência de que os processos de realização e o produto resultante atendem aos requisitos.
Item 7.2.1 Determinação de requisitos relacionados ao produto	A organização deve determinar os requisitos especificados pelo cliente e deve determinar os requisitos não declarados pelo cliente, requisitos estatutários e regulamentares relacionados ao produto, e qualquer requisito adicional determinado pela organização.
Item 7.2.2 Análise crítica dos requisitos relacionados ao produto	A organização deve analisar criticamente os requisitos relacionados ao produto, e esta análise crítica deve ser realizada antes da organização assumir o compromisso de fornecer um produto para o cliente e deve assegurar que os requisitos do produto estão definidos, e os requisitos de contrato ou pedidos que difiram daqueles previamente mencionados estão resolvidos e a organização tem a capacidade de atender aos requisitos definidos. Quanto à prática de realização de análise crítica, pode-se entender como atividade de revisão. A organização deve confirmar os requisitos do cliente, antes da aceitação, quando o cliente não fornecer uma declaração documentada dos requisitos. E para manter documentação e requisitos atualizados, quando os requisitos de produto forem alterados, a organização deve assegurar que os documentos pertinentes são complementados e que o pessoal afetado é alertado sobre os requisitos alterados.
Item 7.3.2 Entradas de projeto e desenvolvimento	As entradas relativas a requisitos de produto devem ser determinadas e registros devem ser mantidos, e devem incluir requisitos de funcionamento e desempenho, requisitos estatutários e regulamentares e outros requisitos essenciais para o projeto e desenvolvimento. Para a norma NBR ISO 9001, requisitos devem ser completos, sem ambigüidades e não conflitantes entre si.

Item 7.3.3 Saídas de projeto e desenvolvimento	As saídas do projeto e desenvolvimento devem atender aos requisitos de entrada para o projeto e desenvolvimento e conter ou referenciar critérios de aceitação do produto.
Item 7.3.4 Análise crítica de projeto e desenvolvimento	Na intenção de garantir qualidade e certificar que o produto desenvolvido atende aos requisitos, a NBR ISO 9001, declara que revisões ou análises críticas sistemáticas de projeto e desenvolvimento devem ser realizadas para avaliar a capacidade dos resultados do projeto e desenvolvimento em atender aos requisitos.
Item 7.3.5 Verificação de projeto e desenvolvimento	A verificação deve ser executada conforme disposições planejadas, para assegurar que as saídas do projeto e desenvolvimento estejam atendendo aos requisitos de entrada do projeto e desenvolvimento.
Item 7.3.6 Validação de projeto e desenvolvimento	A validação do projeto e desenvolvimento deve ser executada conforme disposições planejadas, para assegurar que o produto resultante é capaz de atender aos requisitos para a aplicação especificada.
Item 7.4.1 Processo de aquisição	Quanto aos processos e as informações de aquisição, a norma NBR ISO 9001, relaciona práticas de requisitos, como por exemplo, as informações de aquisição devem descrever o produto a ser adquirido e incluir, quando apropriado, requisitos para aprovação de produto, procedimentos, processos e equipamento, requisitos de qualificação pessoal e sistema de gestão da qualidade.
Item 7.4.2 Informações de aquisição	A organização deve assegurar a adequação dos requisitos de aquisição especificados antes da sua comunicação ao fornecedor.
Item 7.4.3 Verificação do produto adquirido	A organização deve estabelecer e implementar inspeção ou outras atividades necessárias para assegurar que o produto adquirido atende aos requisitos de aquisição especificados.

As práticas, relacionadas a requisitos, definidas na norma NBR ISO 9001, são listadas e identificadas no ANEXO C: Normas da família ISO 9000.

5.3 Guia ISO 9000-3

A ISO/IEC 9000-3 fornece um guia para as organizações na aplicação da norma ISO 9001:2000 para aquisição, provisão, desenvolvimento, operação e manutenção de software de computador. O Guia ISO/IEC 9000-3 identifica as práticas que devem ser executadas e são independentes de tecnologias, modelos de ciclo de vida, processos de desenvolvimento, seqüência de atividades ou estrutura organizacional utilizadas por uma organização.

5.3.1 A questão requisitos e o Guia ISO 9000-3

O Guia ISO/IEC 9000-3 implementa a norma NBR ISO 9001 (descrita no item 5.2) para questões relacionadas a software. Neste contexto, as práticas da NBR ISO 9001 estão embutidas no Guia ISO/IEC 9000-3.

A ISO/IEC 9000-3 indica detalhamento de algumas práticas da NBR ISO 9001 ou indica algumas práticas adicionais a NBR ISO 9001.

- Determinação de requisitos relacionados ao produto: nas práticas relacionadas a requisitos, a ISO/IEC 9000-3 acrescenta que a organização também deve, quando apropriado, estabelecer métodos para acordar requisitos durante o desenvolvimento, desenvolver os requisitos envolvendo cliente e usuários (exemplo: provendo definição de termos e fazendo explanação de requisitos), obter a aprovação do cliente sobre os requisitos, estabelecer um método de rastreabilidade¹ de requisitos até o final do produto (matriz de rastreabilidade de requisitos). Esta norma orienta que as mudanças nos requisitos devem ser controladas e aditivos ao contrato pode ser necessário quando há alterações nos requisitos. Os requisitos devem ser expressos de forma clara e sem ambigüidade para facilitar a validação durante a aceitação do produto. Requisitos devem ser rastreáveis ao longo do ciclo de vida do desenvolvimento. E no caso de projeto e desenvolvimento ser do tipo incremental, o objetivo deve ser agrupar um conjunto de requisitos em detalhes suficientes para começar cada incremento.

¹ Termo original em inglês é *traceability*

- Análise crítica dos requisitos relacionados ao produto: a ISO/IEC ISO 9000-3 também se aplica às atividades obrigatórias de revisão (análise crítica) de requisitos relacionados ao produto. Quando os requisitos do produto são alterados, a organização deve assegurar que os documentos relevantes são modificados e que o pessoal envolvido é notificado. Riscos técnicos também devem ser incluídos quando os requisitos relacionados ao produto são revisados. A revisão dos requisitos pode ser executada por organizações internas ou externas.
- Planejamento do projeto e desenvolvimento: para as fases de projeto e desenvolvimento, o Guia ISO/IEC 9000-3, determina que os planos de desenvolvimento devem conter, quando apropriado, atividades de análise de requisitos.
- Entradas de projeto e desenvolvimento: as revisões devem ser feitas enquanto os requisitos estiverem inconsistentes ou incompletos, enquanto os requisitos não puderem ser verificados e validados.
- Saídas do projeto e desenvolvimento: A saída necessária para as fases de projeto e desenvolvimento é satisfazer os requisitos de entrada das fases de projeto e desenvolvimento.
- Verificação de projeto e desenvolvimento: para verificação de projeto e desenvolvimento, o Guia ISO/IEC 9000-3, determina que a verificação deve ser executada de acordo com o planejado para assegurar que as saídas das fases de projeto e desenvolvimento satisfazem os requisitos de entradas.
- Validação de projeto e desenvolvimento: para validação de projeto e desenvolvimento, o Guia ISO/IEC 9000-3, determina que a validação deve ser executada de acordo com o planejado para assegurar que o produto resultante é capaz de satisfazer aos requisitos.
- Controle de alterações de projeto e desenvolvimento: alterações na especificação do software devem ser mantidas consistentes com os requisitos, projeto, codificação, especificação de teste e manuais do sistema.

- Identificação e rastreabilidade: quanto à produção e fornecimento de serviço, a ISO/IEC 9000-3, define que durante o ciclo de vida do produto, deve haver um processo para rastrear os componentes de um item de *software* ou produto.

5.4 Norma ISO 9004

A norma NBR ISO 9004 promove a adoção de uma abordagem de processo para o desenvolvimento, implementação e melhoria da eficácia e eficiência de um sistema de gestão de qualidade, para aumentar a satisfação das partes interessadas por meio do atendimento aos requisitos.

5.4.1 A questão requisitos e a norma ISO 9004

As práticas relacionadas a requisitos e relatadas pela norma NBR ISO 9001 (item 5.2.1) também são aplicadas à norma NBR ISO 9004. Portanto, serão descritas aqui, as práticas de requisitos ou orientações da NBR ISO 9004 que vão além das definidas na NBR ISO 9001.

- Gestão de processos: como a NBR ISO 9004 demonstra grande preocupação com a satisfação dos clientes e o atendimento aos seus requisitos, essa norma, orienta a direção da organização a identificar processos necessários de realização dos produtos para satisfazer aos requisitos dos clientes e de outras partes interessadas. Também orienta que um plano operacional seja definido para gestão dos processos, incluindo requisitos de entrada e de saída (por exemplo, especificações e recursos), verificação e validação dos processos e produtos.
- Entradas, saídas e análise crítica: Com relação a entradas, saídas e análise crítica de processo, a NBR ISO 9004 orienta que a abordagem de processo deve assegurar que as entradas do processo são definidas e registradas para fornecer uma base para a formulação dos requisitos a serem usados para a verificação das saídas. A resolução de requisitos de entrada ambíguos ou contraditórios pode envolver consulta às partes internas ou externas afetadas. É importante que as saídas dos processos que tenham sido verificadas em relação aos requisitos de entrada do processo,

incluindo critério de aceitação, considerem as necessidades e expectativas dos clientes e de outras partes interessadas.

- Validação de produto e processo de alterações: Com relação as possíveis alterações no produto e em seus requisitos, a orientação dada pela NBR ISO 9004, reflete a implementação de um processo de controle eficaz e eficiente de alterações que assegure que as alterações no produto ou processo beneficiem a organização e satisfaçam as necessidades e expectativas das partes interessadas. É recomendável que essas alterações sejam identificadas, registradas, avaliadas, analisadas criticamente (revisadas) e controladas para se compreender seu efeito sobre outros processos e as necessidades e expectativas dos clientes e de outras partes interessadas.
- Processos relacionados às partes interessadas: a NBR ISO 9004, para processos relacionados aos clientes (ou partes interessadas), recomenda que a organização implemente e mantenha processos para assegurar o entendimento adequado das necessidades e expectativas de suas partes interessadas, bem como convertê-las em requisitos (exemplos: requisitos do cliente ou de outras partes interessadas, requisito de contrato, processos devidos a requisitos estatutários ou regulamentares). É importante que obter uma compreensão completa dos requisitos.
- Entradas e saídas de projeto e desenvolvimento: é conveniente que as necessidades e expectativas externas à organização, em conjunto com as internas, sejam adequadas para se converter em requisitos de entrada para os processos de projeto e de desenvolvimento. É interessante que a saída inclua informações para permitir a verificação e validação com os requisitos planejados, na intenção de garantir qualidade.
- Processo de aquisição: a NBR ISO 9004 orienta a alta direção da organização a assegurar que os processos de aquisição sejam eficazes e eficientes e sejam definidos e implementados para a avaliação e controle de produtos adquiridos, para que os produtos adquiridos satisfaçam as necessidades e requisitos.
- Operações de produção e serviço: essa norma também orienta que a alta direção vá além do controle do processo de realização para alcançar tanto

a adequação com requisitos quanto a obtenção de benefícios para as partes interessadas. A organização pode estabelecer um processo para a identificação e rastreabilidade que supere os requisitos, a fim de coletar dados que possam ser usados para melhoria (requisito de contrato, requisito estatutários e regulamentares).

- Medição, análise e melhoria: a NBR ISO 9004 também orienta a direção da organização, a usar medições de satisfação do cliente como uma ferramenta vital. O processo de auditoria interna fornece uma ferramenta independente para demonstrar por meio da obtenção de evidências objetivas que os requisitos existentes estão sendo atendidos, visto que a auditoria interna avalia a eficácia e a eficiência da organização, o que é de grande importância e pode trazer grandes benefícios a organização.

5.5 Considerações finais

Devido a grande importância de sistemas de gestão da qualidade, foram tratadas neste capítulo as boas práticas, relacionadas à questão requisitos, das normas de qualidade da família ISO 9000.

Capítulo 6 Outros modelos e normas

Assim como os modelos **SW-CMM** e **CMMI**, as normas ISO da família 9000 também tratam questões relacionadas aos requisitos e relacionadas à satisfação dos clientes como visto no Capítulo 5 deste trabalho. E outras normas ISO que não fazem parte da família 9000, como a NBR ISO/IEC 9126-1 [ISO9126] que trata sobre modelo de qualidade e a NBR ISO/IEC 12207 [ISO12207] também apresentam questões relacionadas a requisitos, qualidade de software e a ao ciclo de vida de software. Neste capítulo são apresentadas as normas NBR ISO/IEC 9126-1 (item 6.1) e NBR ISO/IEC 12207 (item 6.2) com foco na questão requisito.

Com relação à especificação de requisitos de software, a IEEE 830 [IEEE84] (item 6.3) também é apresentada neste capítulo, pois trata desse assunto diretamente. E com relação a atividades de gestão de requisitos o item 6.4 apresenta o modelo PMBOK [PMBOK00].

6.1 Norma NBR ISO/IEC 9126-1

A ISO/IEC 9126 consiste nas seguintes partes, sob o título geral de Engenharia de Software - Qualidade de Produto: Parte 1: Modelo de qualidade, Parte 2: Métricas externas, Parte 3: Métricas internas e Parte 4: Métricas de qualidade em uso. A NBR ISO/IEC 9126-1, juntamente com suas partes (dois, três e quatro), cancela e substitui a NBR 13596:1996.

A NBR ISO/IEC 9126-1 permite que a qualidade do produto de software seja especificada e avaliada em diferentes perspectivas pelos envolvidos com as atividades de aquisição, requisitos, desenvolvimento, uso, avaliação, apoio, manutenção, garantia da qualidade e auditoria de software. O modelo de qualidade definido na NBR ISO/IEC 9126-1 é usado para validar a completitude da definição dos requisitos, identificar requisitos de software, identificar critérios de aceitação para produtos finais de software e outros.

A NBR ISO/IEC 9126-1 pode ser utilizada em conjunto com a norma NBR ISO/IEC 12207 (ciclo de vida de software) para fornecer uma estrutura para definição de requisitos de qualidade de software, nos processos fundamentais do ciclo de vida e fornecer apoio para revisão, verificação e validação nos processos de apoio do ciclo de

vida. A NBR ISO/IEC 9126-1 pode ser utilizada em conjunto com a norma NBR ISO 9001 (processos de garantia da qualidade) para fornecer apoio para estabelecer metas de qualidade e fornecer apoio para revisão, verificação e validação de projeto.

6.1.1 A questão requisitos e a norma NBR ISO/IEC 9126-1

A NBR ISO/IEC 9126-1 define que os requisitos para qualidade do produto de software incluem critérios para avaliação de qualidade interna, qualidade externa e qualidade em uso, para atingir as necessidades dos desenvolvedores, mantenedores, adquirentes e usuários finais.

Requisitos de qualidade interna são usados para especificar as propriedades dos produtos intermediários. Esses podem incluir modelos estáticos e dinâmicos, outros documentos e código fonte. Esses requisitos podem ser usados para definir estratégias de desenvolvimento. Qualidade interna é a totalidade das características do produto de software sob o ponto de vista interno. A qualidade interna é medida e avaliada com relação aos requisitos de qualidade interna.

Qualidade externa é a qualidade quando o software é executado, o qual é tipicamente medido e avaliado enquanto está sendo testado em ambiente simulado, com dados simulados. Durante os testes, a maioria dos defeitos devem ser descobertos e eliminados.

Qualidade em uso é a visão da qualidade do produto de software sob o ponto de vista do usuário, quando esse produto é usado em ambiente e contexto de uso específico.

As características definidas no modelo de qualidade para qualidade interna e externa são: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Estas características, quando observadas, podem auxiliar tarefas como extração e definição de requisitos, processo de alteração de requisitos e outras. A característica de funcionalidade define a capacidade do produto de software prover funções que satisfaçam os requisitos, e para isso, são relacionadas necessidades como adequação, acurácia, interoperabilidade, segurança de acesso e conformidade (requisitos definidos por leis ou regulamentos). Com relação à confiabilidade, o produto de software deve manter um nível de desempenho especificado. Quanto à característica de usabilidade, requisitos de interface com usuário podem ser especificados. Capacidade do produto de software de ser alterado implica em atividades relacionadas a controle de alterações de requisitos.

As características definidas no modelo de qualidade para qualidade em uso são: eficácia, produtividade, segurança e satisfação. Estas características, quando observadas, podem implicar na capacidade do produto de software de satisfazer usuários e de forma conjunta satisfazer os requisitos dos usuários.

As boas práticas, relacionadas a requisitos, definidas na NBR ISO/IEC 9126-1 são listadas e identificadas no ANEXO D: ISO 9126.

6.2 Norma NBR ISO/IEC 12207

A norma NBR ISO/IEC 12207 estabelece uma estrutura comum para os processos de ciclo de vida de software. Essa norma estabelece processos que podem ser utilizados para definir, controlar e melhorar os processos de ciclo de vida de software. A NBR ISO/IEC 12207 aplica-se à aquisição de sistemas, produtos e serviços de software, para o fornecimento, desenvolvimento, a operação e manutenção de produtos de software, software de *firmware*, quer sejam executados interna ou externamente a uma organização. A NBR ISO/IEC 12207 apresenta os processos de ciclo de vida de software que podem ser empregados para adquirir, fornecer, desenvolver, operar e manter produtos de software.

Os processos fundamentais do ciclo de vida são: Processo de aquisição, Processo de fornecimento, Processo de desenvolvimento, Processo de operação, Processo de manutenção.

Os processos de apoio de ciclo de vida são: Processo de documentação, Processo de gerência de configuração, Processo de garantia da qualidade, Processo de verificação, Processo de validação, Processo de revisão conjunto, Processo de auditoria, Processo de resolução de problema.

Processos organizacionais de ciclo de vida são: Processo de gerência, Processo de infra-estrutura, Processo de melhoria, Processo de treinamento.

6.2.1 A questão requisitos e a norma NBR ISO/IEC 12207

A NBR ISO/IEC 12207 descreve práticas relacionadas às atividades de requisitos durante os processos realizados no ciclo de vida de software. Como a NBR ISO/IEC 12207 define os processos do ciclo de vida de software, pode-se então iniciar o tratamento dos requisitos na fase de aquisição.

6.2.1.1 Processo de aquisição

O processo de aquisição contém as atividades e tarefas do adquirente. Inicia-se com a definição da necessidade de adquirir um sistema, um produto de software ou um serviço de software. Para essa etapa a norma NBR ISO/IEC 12207 indica uma série de atividades a serem executadas e estas, por sua vez, descrevem boas práticas relacionadas à questão requisitos que são descritas a seguir.

Item 5.1.1 Iniciação	Item 5.1.1.2	O adquirente deverá definir e analisar os requisitos do sistema, que inclui requisitos de negócio, organizacionais e requisitos de usuário, e também requisitos de segurança, proteção e outros requisitos críticos relacionados às atividades de projeto, testes e aderência a padrões e procedimentos.
	Item 5.1.1.3	O adquirente pode executar a definição e análise de requisitos do software por conta própria ou pode manter acordo com um
	Item 5.1.1.4	fornecedor para executar essa tarefa e se o adquirente mantiver acordo com um fornecedor para execução da análise dos requisitos de um sistema, o adquirente deverá aprovar estes requisitos.
	Item 5.1.1.7	Para aquisição de um produto de software de prateleira, o adquirente deverá assegurar que os requisitos do produto de software sejam satisfeitos.
	Item 5.1.1.8	O adquirente deverá preparar, documentar e executar um plano de aquisição que deve conter os requisitos para o sistema.
	Item 5.1.1.9	O adquirente deverá definir e documentar a estratégia e condições (critérios) de aceitação.
Item 5.1.2 Preparação de pedido de proposta	Item 5.1.2.1	O adquirente deve documentar os requisitos de aquisição, e essa documentação deve incluir, quando apropriado, os requisitos do sistema.
	Item 5.1.3.1	O adquirente deve estabelecer um procedimento para selecionar o fornecedor, incluindo critérios de avaliação de proposta e ponderação da aderência aos requisitos.
	Item 5.1.3.4	O adquirente deverá então, preparar e negociar um contrato com o fornecedor que trate dos requisitos de aquisição, incluindo custo e cronograma do produto ou serviço de software a ser entregue.

6.2.1.2 Processo de fornecimento

O processo de fornecimento contém as atividades e as tarefas do fornecedor. Para o processo de fornecimento, a NBR ISO/IEC 12207 indica uma série de atividades a serem executadas, e estas atividades, por sua vez, descrevem boas práticas relacionadas à questão requisitos que são descritas a seguir.

Item 5.2.1 Iniciação	Item 5.2.1.1	O fornecedor conduz uma revisão dos requisitos que constam no pedido de proposta, levando em consideração políticas e outros regulamentos da organização.
Item 5.2.4 Planejamento	Item 5.2.4.1	O fornecedor deve conduzir uma revisão dos requisitos de aquisição para definir a estrutura para gerenciar e garantir a qualidade do produto ou serviço de software a ser entregue.
	Item 5.2.4.3	O fornecedor deve estabelecer requisitos para os planos, e estes devem incluir necessidades de recursos e o envolvimento do adquirente.
	Item 5.2.4.5	O fornecedor deve desenvolver e documentar os planos de gerência do projeto de acordo com os requisitos de planejamento incluindo gerenciamento de proteção e segurança e outros requisitos críticos dos produtos ou serviços de software, e envolvimento do usuário através de exercícios de consolidação de requisitos, demonstrações de protótipos e avaliações.
Item 5.2.5 Execução e controle	Item 5.2.5.4	O fornecedor deve verificar todos os requisitos contratuais necessários, para assegurar que o produto ou serviço de software entregue ao adquirente foi desenvolvido ou executado de acordo com os requisitos do contrato original.
Item 5.2.6 Revisão e avaliação	Item 5.2.6.3	O fornecedor deve executar as atividades de verificação e validação para demonstrar que os produtos ou serviços de software e os processos satisfazem completamente os seus respectivos requisitos.

6.2.1.3 Processo de desenvolvimento

O processo de desenvolvimento contém as atividades e tarefas do desenvolvedor. A NBR ISO/IEC 12207 também descreve no processo de desenvolvimento um conjunto de atividades que relacionam boas práticas no que diz respeito às atividades de requisitos. Dessas práticas, as principais estão descritas a seguir.

Item 5.3.2 Análise dos requisitos do sistema	Item 5.3.2.1	A especificação dos requisitos do sistema deve descrever funções e capacidades do sistema, requisitos de negócio, requisitos organizacionais e de usuários, requisitos de proteção e segurança, requisitos de engenharia, requisitos de fatores humanos, requisitos de interface, requisitos de operação e manutenção, restrições do projeto e requisitos de qualificação. A especificação dos requisitos do sistema deve ser documentada.
	Item 5.3.2.2	Os requisitos do sistema devem ser avaliados, considerando rastreabilidade para as necessidades de aquisição, consistência com as necessidades de aquisição, testabilidade, viabilidade do projeto de arquitetura do sistema e viabilidade de operação e manutenção.
Item 5.3.3 Projeto da arquitetura do sistema	Item 5.3.3.1	Uma arquitetura de alto nível do sistema deve ser estabelecida para identificar itens de hardware, software e operações manuais e deve assegurar que todos os requisitos do sistema sejam alocados entre esses itens.
	Item 5.3.3.2	A arquitetura do sistema e os requisitos para os itens devem ser avaliados quanto a rastreabilidade para os requisitos do sistema, consistência com os requisitos do sistema, adequação aos métodos e padrões de projeto utilizados, viabilidade dos itens de software atenderem seus requisitos alocados e viabilidade de operação e manutenção. E os resultados das avaliações devem ser documentados.
Item 5.3.4 Análise de requisitos do software	Item 5.3.4.1	O desenvolvedor deve estabelecer e documentar os requisitos do software, incluindo as especificações das características de qualidade (NBR ISO/IEC 9126-1), como requisitos de qualificação, requisitos de base de dados, requisitos de instalação e aceitação do produto de software, requisitos do usuário para execução, operação e manutenção.
	Item 5.3.4.2	O desenvolvedor deve avaliar os requisitos de software considerando critérios como rastreabilidade para os requisitos do sistema e projeto do sistema, consistência externa com os requisitos do sistema.

	Item 5.3.4.3	O desenvolvedor deve conduzir revisões conjuntas e concluindo-as deve ser estabelecida uma <i>baseline</i> para os requisitos do item de software.
Item 5.3.5 Projeto da arquitetura do software	Item 5.3.5.1	O desenvolver deve transformar os requisitos para o item de software em uma arquitetura que descreve sua estrutura de alto nível e identifica os componentes de software. Deve ser garantido que todos os requisitos do item de software sejam alocados aos seus componentes de software, e sejam refinados para facilitar o projeto detalhado.
	Item 5.3.5.3	O desenvolvedor deve definir e documentar os requisitos preliminares de teste e o cronograma para a integração do software.
	Item 5.3.5.6	O desenvolvedor deve avaliar a arquitetura do item de software e os projetos de interface e base de dados, considerando critérios como rastreabilidade para os requisitos do item de software e consistência externa com os requisitos do item de software.
Item 5.3.6 Projeto detalhado de software	Item 5.3.6.1	O desenvolvedor deve desenvolver um projeto detalhado para cada componente de software do item de software. Os componentes de software devem ser refinados em níveis mais baixos, contendo unidades de software que possam ser codificadas, compiladas e testadas. Deve ser garantido que todos os requisitos do software sejam alocados para unidades de software a partir dos componentes de software. O projeto detalhado deve ser documentado.
	Item 5.3.6.5	O desenvolvedor deve definir e documentar os requisitos de teste que devem incluir testes de estresse de unidade de software, até o limite de seus requisitos.
	Item 5.3.6.7	O desenvolvedor deve avaliar o projeto detalhado do software e os requisitos de teste considerando critérios como rastreabilidade para os requisitos do item de software.
Item 5.3.7 Codificação e testes de software	Item 5.3.7.2	O desenvolvedor deve testar cada unidade do software e base de dados, garantindo que sejam atendidos seus requisitos.
	Item 5.3.7.4	O desenvolvedor deve atualizar os requisitos de teste e o cronograma para integração do software.
	Item 5.3.7.5	O desenvolvedor deve avaliar o código do software e os resultados dos testes, considerando critérios como rastreabilidade para os requisitos e projeto do item de software, consistência externa com os requisitos e projeto do item de software e consistência interna entre os requisitos da unidade.

Item 5.3.8 Integração do software	Item 5.3.8.2 5.3.8.4	Deve ser garantido que cada agregação atenda os requisitos do item de software e o desenvolvedor deve desenvolver e documentar, para cada requisito de qualificação do item de software, um conjunto de testes, casos de teste e procedimento de teste.
	Item 5.3.8.5	O desenvolvedor deve avaliar o plano de integração, projeto, código e testes considerando critérios como rastreabilidade para os requisitos do sistema, consistência externa com os requisitos do sistema e cobertura de teste dos requisitos do item de software.
Item 5.3.9 Teste de qualificação do software	Item 5.3.9.1	O desenvolvedor deve conduzir o teste de qualificação de acordo com os requisitos de qualificação para o item de software. Deve ser garantido que a implementação de cada requisito do software seja testada para conformidade.
	Item 5.3.9.3	O desenvolvedor deve avaliar o projeto, código, teste e os resultados dos testes, considerando critérios como cobertura de teste dos requisitos do item de software.
Item 5.3.10 Integração do sistema	Item 5.3.10.1	As agregações devem ser testadas quando integradas, de acordo com os seus requisitos.
	Item 5.3.10.3	O sistema integrado deve ser avaliado de acordo com critérios como cobertura de teste dos requisitos de sistema.
Item 5.3.11 Teste de qualificação do sistema	Item 5.3.11.1	O teste deve ser conduzido de acordo com os requisitos de qualificação e deve ser garantido que a implementação de cada requisito do sistema seja testada.
	Item 5.3.11.2	O sistema deve ser avaliado, considerando critérios como cobertura de teste dos requisitos do sistema.

6.2.1.4 Processo de manutenção

O processo de manutenção, definido na NBR ISO/IEC 12207, contém atividades e tarefas do mantenedor.

Item 5.5.3 Implementação da modificação	Item 5.5.3.2	Os requisitos do processo de desenvolvimento devem ser completados e deve ser garantida a implementação completa e correta dos requisitos novos e dos modificados. Também deve ser garantido que os requisitos originais não modificados não foram afetados.
---	-----------------	--

6.2.1.5 Processo de gerência de configuração

O processo de gerência de configuração é um processo de aplicação de procedimentos administrativos e técnicos, por todo o ciclo de vida do software, destinado a identificar e definir os itens de software em um sistema, e estabelecer *baselines*, controlar as modificações e liberações dos itens, registrar e apresentar as situações dos itens e dos pedidos de modificação, garantir a completeza, a consistência e a correção dos itens, e controlar o armazenamento, a manipulação e a distribuição dos itens.

Item 6.2.5 Avaliação da configuração	Item 6.2.5.1	Deve ser determinado e garantido a completeza funcional dos itens de software em relação aos seus requisitos e a completeza física dos itens de software (projeto e código refletem uma descrição técnica atualizada).
--	-----------------	--

6.2.1.6 Processo de garantia da qualidade

O processo de garantia da qualidade é o processo para fornecer garantia adequada de que os processos e produtos de software, no ciclo de vida do projeto, estejam em conformidade com os seus requisitos especificados e sejam aderentes aos planos estabelecidos.

Item 6.3.2 Garantia da qualidade do produto	Item 6.3.2.3	Na preparação da entrega dos produtos de software deve ser garantido que os produtos de software tenham seus requisitos contratuais inteiramente satisfeitos e sejam aceitáveis pelos adquirentes.
Item 6.3.3 Garantia da qualidade do processo	Item 6.3.3.3	Deve ser garantido que os requisitos aplicáveis ao contrato original sejam passados para os subcontratados e que os produtos de software do subcontratado satisfaçam os requisitos do contrato original.
	Item 6.3.3.6	Deve ser garantido que a equipe alocada tenha qualificação e o conhecimento necessário para atender os requisitos do projeto e recebam todo treinamento necessário.

6.2.1.7 Processo de verificação

O processo de verificação é um processo para determinar se os produtos de software de uma atividade atendem completamente os requisitos ou condições impostas a eles nas atividades anteriores.

Item 6.4.2 Verificação	Item 6.4.2.1	O contrato deve ser verificado segundo critérios como capacidade do fornecedor em atender aos requisitos, requisitos estão consistentes e cobrem as necessidades do usuário, procedimentos adequados para tratar alterações nos requisitos e priorização de problemas estão estipulados, critérios e procedimentos de aceitação estão estipulados de acordo com os requisitos.
	Item 6.4.2.3	Os requisitos devem ser verificados segundo critérios como requisitos do sistema são consistentes, viáveis e testáveis, os requisitos do sistema foram distribuídos apropriadamente para os itens de hardware, software, os requisitos de software são consistentes, viáveis e testáveis, os requisitos de software relacionados à proteção e segurança e aos fatores críticos estão corretos, conforme demonstrado por métodos adequadamente rigorosos.
	Item 6.4.2.4	O projeto é verificado segundo critérios como o projeto está correto e consistente com os requisitos e rastreável aos mesmos, o projeto implementa proteção, segurança e outros requisitos críticos corretamente, conforme demonstrado por métodos adequadamente rigorosos.
	Item 6.4.2.5	O código deve ser verificado segundo critérios como o código é rastreável para o projeto e para os requisitos, testável, correto e aderente aos requisitos e padrões de codificação, o código selecionado pode ser originado a partir do projeto ou dos requisitos, o código implementa proteção, segurança e outros requisitos críticos corretamente, conforme demonstrado por métodos adequadamente rigorosos.

6.2.1.8 Processo de validação

O processo de validação é um processo para determinar se os requisitos e o produto final, sistema ou produto de software construído, atendem ao uso específico pretendido.

Item 6.5.2 Validação	Item 6.5.2.1	Consiste nas tarefas de preparar os requisitos de teste e assegurar que esses requisitos de teste reflitam os requisitos particulares.
	6.5.2.2	

6.2.1.9 Processo de gerência

O processo de gerência contém as atividades e tarefas genéricas que podem ser empregadas por quaisquer das partes que tem que gerenciar seus respectivos processos.

Item 7.1.4 Revisão e avaliação	Item 7.1.4.1	O gerente deve garantir que o software e os planos sejam avaliados para satisfazer requisitos.
--------------------------------------	-----------------	--

As principais boas práticas relacionadas a requisitos definidas na NBR ISO/IEC 12207 são listadas e identificadas no ANEXO E: ISO 12207.

6.3 Norma IEEE 830

A norma IEEE 830 descreve uma abordagem para boas práticas de especificação de requisitos de software. Os requisitos podem ser explicitados pelo usuário ou podem ser alocados ao software pelo processo de análise de requisitos de sistema.

6.3.1 A questão requisitos e a norma IEEE 830

A norma IEEE 830 fornece informações necessárias para escrever uma boa especificação de requisitos de software. As suas principais características são: não ser ambíguo, estar completo, ser verificável, ser consistente, ser alterável, ser rastreável e ser útil durante as fases de operação e manutenção.

Cada uma dessas principais características descreve atributos e qualidades relacionadas a requisitos, como segue:

- Não ser ambíguo: uma Especificação de Requisitos de Software não é ambígua se e somente se, os requisitos descritos tem apenas uma interpretação.
- Estar completo: uma Especificação de Requisitos de Software é completa se possuir as seguintes qualidades: inclusão dos significados dos requisitos, definição das respostas do software, conformidade com a Especificação de Requisitos de Software, títulos e referências de figuras completas.
- Ser verificável: uma Especificação de Requisitos de Software é verificável se e somente se os requisitos forem verificáveis. Um requisito é verificável

se e somente se existe um processo finito com o qual a pessoa ou máquina pode verificar que o produto de software atende aos requisitos.

- Ser consistente: uma Especificação de Requisitos de Software é consistente se e somente se os requisitos não apresentarem conflitos entre si.
- Ser alterável: uma Especificação de Requisitos de Software é alterável se a estrutura ou estilo são apropriadas para que mudanças nos requisitos sejam feitas facilmente.
- Ser rastreável: uma Especificação de Requisitos de Software é rastreável se a origem de cada requisito é clara e se a referência para cada requisito é facilitada na documentação de desenvolvimento.
- Ser útil durante as fases de operação e manutenção: uma Especificação de Requisitos de Software indica as necessidades da fase de operação e manutenção, incluindo eventual recolocação de software.

As principais boas práticas relacionadas a requisitos definidas na IEEE 830 são listadas e identificadas no ANEXO F: IEEE 830.

6.4 Modelo PMBOK

PMBOK (*Project Management Body of Knowledge*) é um modelo desenvolvido pelo **PMI** (*Project Management Institute*). Esse modelo tem como principal objetivo conduzir gerentes de projeto e outros profissionais relacionados em uma linha de conduta sobre gerência de projetos e já está disseminado no mercado [PMBOK00].

Atualmente, o **PMI** possuiu 100.000 membros em 125 países em todo o mundo. Os membros de **PMI** são indivíduos que praticam e que estudam gerência de projetos em muitas áreas diferentes da indústria, incluindo projetos aeroespaciais, indústria automotiva, gerência de negócios, construção, engenharia, serviços financeiros, tecnologia de informação, indústria farmacêutica e telecomunicações.

No **PMBOK** as áreas de conhecimento da gerência de projetos descrevem os conhecimentos e práticas em gerência de projetos em termos dos processos que as compõem. Esses processos foram organizados em nove áreas de conhecimentos: Gerência da Integração do Projeto, Gerência do Escopo do Projeto, Gerência do Tempo do Projeto, Gerência do Custo do Projeto, Gerência da Qualidade do Projeto, Gerência

dos Recursos Humanos do Projeto, Gerência das Comunicações do Projeto, Gerência dos Riscos do Projeto, Gerência das Aquisições do Projeto.

6.4.1 A questão requisitos e o modelo PMBOK

A área de conhecimento Gerência do Escopo do Projeto inclui os processos requeridos para assegurar que o projeto inclua todo o trabalho necessário, e tão somente o trabalho necessário, para concluir de forma bem sucedida o projeto. A preocupação fundamental compreende em definir e controlar o que está ou não incluído no projeto. Os processos da gerência do escopo do projeto são:

- Iniciação: comprometimento da organização em iniciar a próxima fase do projeto.
- Planejamento do Escopo: desenvolver uma declaração escrita do escopo como base para decisões futuras do projeto.
- Detalhamento do Escopo: subdividir os principais subprodutos do projeto em componentes menores e mais manejáveis.
- Verificação do Escopo: formalizar a aprovação do escopo do projeto.
- Controle de Mudanças do Escopo: controlar as mudanças do escopo do projeto.

No contexto de projeto, o termo escopo deve se referir a escopo do produto (aspectos e funções que devam ser incluídos no produto ou serviço) ou escopo do projeto (o trabalho que deve ser feito com a finalidade de entregar um produto de acordo com os aspectos e as funções especificadas).

Uma das práticas relacionadas a requisitos definidas no PMBOK diz respeito ao planejamento do escopo que é o processo de desenvolvimento de uma declaração escrita do escopo como base para decisões futuras do projeto. Essa declaração escrita inclui, em particular, os critérios usados para determinar se o projeto ou fase foi completado com sucesso, ou seja, reflete a necessidade de uma documentação de requisitos e da necessidade de definição de critérios de aceitação. A declaração do escopo forma as bases para um acordo entre a equipe do projeto e o cliente através da identificação de objetivos do projeto bem como dos principais subprodutos do projeto. A declaração do escopo fornece a documentação que servirá de base para tomada de decisões futuras no projeto e para confirmar ou desenvolver um entendimento comum do escopo entre as

partes envolvidas. Com o progresso do projeto, a declaração do escopo pode necessitar ser revisada para refletir as mudanças do escopo do projeto, ou seja, manter-se consistente com os requisitos e alterações de requisitos. O plano de gerência de escopo é o documento que descreve como o escopo do projeto será gerenciado e como as mudanças no escopo serão integradas ao projeto.

Na fase de detalhamento do escopo, é elaborada uma EAP¹ (Estrutura Analítica do Projeto) que é um agrupamento orientado ao subproduto (*deliverable-oriented*) dos elementos do projeto que organiza e define o escopo total do projeto (o trabalho que não está na EAP está fora do escopo do projeto). Com relação à declaração do escopo, a EAP é freqüentemente usada para elaborar ou confirmar um entendimento comum do escopo do projeto.

O **PMBOK** apresenta a fase de verificação do escopo que é o processo de formalização do aceite do escopo do projeto pelas partes envolvidas (patrocinador, cliente e outros), ou seja, verificar se o produto atende aos requisitos estabelecidos. Isto exige uma revisão dos produtos e resultados do trabalho para garantir que tudo foi completado correta e satisfatoriamente. A verificação do escopo difere do controle da qualidade já que é fundamentalmente relacionada com a aceitação do resultado do trabalho enquanto o controle da qualidade é fundamentalmente relacionado com a exatidão dos resultados do trabalho.

Há uma notável preocupação em relação as alteração de requisitos e os impactos que elas podem causar. No **PMBOK**, o controle de mudanças do escopo consiste em influenciar os fatores que criam mudanças no escopo para garantir que as mudanças sejam benéficas, determinar que uma mudança no escopo ocorreu, e gerenciar as mudanças, quando e se elas ocorrem. O controle das mudanças de escopo deve ser completamente integrado com os outros processos de controle (controle de prazo, controle de custo, controle de qualidade). As mudanças podem exigir a expansão do escopo ou podem permitir que seja reduzido. No **PMBOK**, um sistema de controle de mudanças do escopo define os procedimentos pelos quais o escopo do projeto pode ser mudado. Uma parte importante do controle de mudanças do escopo é determinar o que está causando a variação e decidir se a variação exige ação corretiva. Nesse contexto, uma mudança do escopo é qualquer modificação no escopo combinado do projeto,

¹ WBS = *Work Breakdown Structure* (termo original em inglês)

conforme definido pela EAP aprovada. As mudanças do escopo, freqüentemente, exigem ajustes no custo, no prazo, na qualidade ou em outros objetivos do projeto.

As principais boas práticas relacionadas a requisitos definidas no **PMBOK** são listadas e identificadas no ANEXO G: PMBOK.

6.5 Considerações finais

Este capítulo apresentou as principais práticas relacionadas às atividades de requisitos e que estão definidas nas normas ISO/IEC 9126-1 e 12207 e nos modelos IEEE 830 e **PMBOK**.

Capítulo 7 Correlações entre normas e modelos

Os capítulos anteriores mostraram um resumo das boas práticas relacionadas às atividades de requisitos e que são aplicadas nos diversos modelos e normas pesquisados. Este capítulo analisa as correlações observadas entre essas boas práticas e confronta as terminologias utilizadas nos modelos e normas apresentados anteriormente.

Determinadas práticas ou determinados termos são utilizados por normas ou modelos de maneira semelhante e outras vezes alguns termos podem ter seu significado específico até mesmo pela questão de tradução. Certas divergências de tradução também são muito comuns na informática. Portanto, existe a necessidade de conhecer o significado das principais terminologias utilizadas em cada modelo ou norma a fim de buscar uma compreensão mais correta sobre o assunto requisitos no contexto dos modelos de processo e qualidade de software. Assim também é interessante identificar e relacionar as práticas que se referem a requisitos entre os modelos apresentados nos capítulos anteriores.

Neste capítulo serão tratadas as terminologias dos modelos **SW-CMM** (item 7.1.1), **CMMI** (item 7.1.2), PMBOK (item 7.1.7) e das normas da família NBR ISO 9000 (item 7.1.3) e NBR ISO 9126-1 (item 7.1.4), ISO/IEC 12207 (item 7.1.5), IEEE 830 (item 7.1.6).

O objetivo deste capítulo é mostrar o relacionamento de terminologias e práticas (item 7.2) entre um conjunto preestabelecido de normas, padrões e modelos, a fim de facilitar o entendimento dos diversos padrões de qualidade mundialmente aceitos, facilitar a aplicabilidade das práticas de requisitos envolvidas, bem como obter uma visualização global das boas práticas mais utilizadas e mais importantes para atividades de requisitos em todo o ciclo de vida do projeto de software e assim contribuir para os estudos aplicados à área de engenharia de requisitos.

7.1 Principais definições e terminologias

Nos itens a seguir são apresentadas as principais terminologias utilizadas nos modelos e normas estudados relacionados à engenharia de requisitos.

7.1.1 Terminologias SW-CMM

O modelo de maturidade de capacidade de software, o **SW-CMM**, utiliza termos como “requisito de sistema” e “requisito de software”. Nesse modelo, há uma utilização mais acentuada de um termo ou outro nos diferentes níveis de maturidade. Por exemplo, no nível 2, a utilização do termo “requisito alocado ao software” é mais acentuada, e no nível 3 é o uso do termo “requisito de software”. No **SW-CMM**, as definições de “requisito de sistema” e “requisito de software” são baseadas no padrão IEEE-STD-610 [IEEE90].

Os termos mais utilizados no **SW-CMM**, na área de requisitos, são listados a seguir:

- Requisito de sistema: uma condição ou capacidade que deve ser atribuída a um sistema ou a um componente de sistema para que este possa solucionar um problema de um usuário.
- Requisito de software: uma condição ou capacidade que um software necessita para solucionar um problema de um usuário ou alcançar um objetivo.
- Requisito técnico: são requisitos que descrevem o que o software deve fazer e suas regras operacionais (exemplos: requisitos funcionais, requisitos de desempenho, requisitos de interface e requisitos de qualidade).
- Requisito não técnico: é utilizado para descrever termos contratuais, condições e acordos que afetam e determinam as atividades de um projeto de software.
- Requisitos de sistema alocados ao software: são entendidos como sendo um subconjunto dos requisitos de sistema que são implementados em componentes de software do sistema (requisitos associados ao escopo do desenvolvimento).

- Análise dos requisitos de software: elabora e refina os requisitos alocados ao software e resulta em requisitos de software documentados.

7.1.2 Terminologias CMMI

O modelo de maturidade de capacidade de software integrado, o **CMMI**, utiliza termos relacionados às atividades de gestão e desenvolvimento de requisitos. Os principais termos utilizados por este modelo são:

- Requisito: uma condição ou capacidade necessárias para solucionar um problema de um usuário ou atingir um objetivo, e essa condição ou capacidade deve fazer parte do produto ou componente do produto para satisfazer um contrato, padrão, especificação ou outros documentos formais. Um requisito é uma representação documentada de uma condição ou capacidade [IEEE90].
- Requisito contratual: todos os requisitos técnicos e não técnicos relacionados na aquisição de contrato.
- Requisito técnico: propriedade (atributo) de produtos ou serviços que podem ser adquiridos e desenvolvidos.
- Requisito não técnico: regras contratuais, compromissos, condições e termos que afetam e determinam como os produtos e serviços serão adquiridos. Exemplos de requisitos não técnicos incluem produtos a serem entregues, data de entrega comercial, datas marco, requisitos de treinamento, requisitos locais e argumentos.
- Requisito alocado: requisito que abrange todo ou parte de um requisito de alto nível, como desempenho e funcionalidade, e também um elemento de arquitetura ou componente de projeto de nível baixo.
- Requisitos derivados¹: são requisitos que não estão explicitamente declarados nos requisitos do cliente, mas são deduzidos do contexto de requisitos, por exemplo, padrões aplicados, leis, políticas, práticas comuns, decisões gerenciais. Requisitos derivados também são os requisitos necessários para especificar componente de produto, e estes podem surgir nas fases de análise e projeto de componente de produto ou sistema.

¹ Também conhecidos como requisitos implícitos.

- Requisito de produto: um refinamento dos requisitos do cliente, e que utiliza uma linguagem de desenvolvedores, elaborando requisitos implícitos em requisitos derivados explícitos. O desenvolvedor usa requisitos de produto como guia para projetar e desenvolver o produto.
- Requisitos de componente de produto: provê uma especificação completa do componente de produto, incluindo funcionalidade, desempenho, e qualquer outro requisito.
- Análise de requisitos: determinação de características de desempenho e funcionalidade de um produto específico baseado na análise das necessidades dos clientes, expectativas, conceito operacional, utilização de ambiente projetado por pessoas, produtos ou processos e medidas de efetividade.
- Extração de requisitos: atividade que se refere ao uso sistemático de técnicas, como protótipos e pesquisa estruturada, a fim de pró-ativamente identificar e documentar as necessidades do cliente e dos usuários. Dentro deste assunto, na língua portuguesa, também pode ser encontrada a expressão elicitação de requisitos, embora a palavra elicitação não consta dos principais dicionários da língua portuguesa [FERRE93].
- Gerência de requisitos: refere-se à gerência de todos os requisitos do projeto, incluindo requisitos técnicos e não técnicos e todos os requisitos incorporados ao projeto.
- Rastreabilidade de requisitos: é a evidência de uma associação entre um requisito e a sua fonte, implementação e verificação.

7.1.3 Terminologias da família ISO 9000

A norma ISO 9000 descreve os fundamentos de sistemas de gestão da qualidade, que constituem o objetivo da família NBR ISO 9000, e define os termos a ela relacionados. Os termos relevantes, associados a requisitos, definidos na NBR ISO 9000, são os mesmos utilizados nas normas NBR ISO 9001, 9000-3 e 9004, com a ressalva de que nas normas NBR ISO 9001 e 9004 o termo “produto” também pode significar “serviço”. A seguir são mostrados os principais termos e seus significados, na área de requisitos, aplicados na norma ISO 9000.

Quanto aos termos relacionados à qualidade, a norma ISO 9000 define:

- Qualidade: o grau no qual um conjunto de características inerentes satisfaz aos requisitos.
- Requisito: é a necessidade ou expectativa expressa, geralmente de forma implícita ou obrigatória.

Quanto aos termos relacionados a processo e produto, a norma ISO 9000 define:

- Projeto e desenvolvimento: é um conjunto de processos que transformam requisitos em características especificadas ou na especificação de um produto, processo ou sistema.

Quanto aos termos relacionados com a documentação, a norma ISO 9000 define:

- Especificação: é o documento que estabelece requisitos.

E quanto aos termos relacionados com o exame, a norma ISO 9000 define:

- Verificação: é a comprovação através do fornecimento de evidência objetiva, de que os requisitos especificados foram atendidos.
- Validação: é a comprovação através do fornecimento de evidência objetiva, de que os requisitos para uma aplicação ou uso específicos pretendidos foram atendidos.
- Processo de qualificação: é o processo para demonstrar a capacidade de atender a requisitos especificados.

Nas normas ISO, para determinados casos, o termo em inglês “*review*” é traduzido na norma brasileira como “análise crítica” e não como “revisão”, porém é importante ressaltar que o entendimento para termo “análise crítica” se refere à atividade de revisão realizada pelo pessoal de desenvolvimento de software e não de análise técnica (fases de análise e projeto do ciclo de vida clássico).

7.1.4 Terminologias ISO/IEC 9126

Os principais termos e seus significados na área de requisitos aplicados na norma NBR ISO/IEC 9126-1 são similares aos termos definidos nas normas da família

ISO 9000, não apresentando terminologia adicional específica para a norma NBR ISO/IEC 9126-1.

7.1.5 Terminologias ISO/IEC 12207

Os principais termos e seus significados, na área de requisitos aplicados na norma NBR ISO/IEC 12207 são similares aos termos definidos nas normas da família ISO 9000. Algumas terminologias adicionais da norma NBR ISO/IEC 12207 são apresentadas abaixo:

- Modelo do ciclo de vida: é uma estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema desde a definição de seus requisitos até o término de seu uso.
- Cobertura de teste: é a extensão em que os casos de teste dos requisitos de um sistema ou produto de software são testados.
- Testabilidade: pode ser entendido como a extensão em que um teste objetivo e factível pode ser projetado para determinar se um requisito é atendido.

7.1.6 Terminologias IEEE 830

Os termos utilizados na norma IEEE 830 são definidos na IEEE 729-1983 [IEEE83], e revisados e atualizados na IEEE 610.12-1990 [IEEE90]. Os principais termos e seus significados, na área de requisitos, aplicados na norma IEEE 830 são apresentados a seguir.

- Requisito: é uma condição ou capacidade necessária para solucionar um problema do usuário ou atingir um objetivo.
- Análise de requisitos: é o processo de estudar as necessidades do usuário para chegar à definição do sistema, hardware ou requisitos de software.
- Fase de requisitos: é o período no ciclo de vida do software, no qual os requisitos do produto de software são definidos e documentados.
- Revisão de requisitos: é o processo no qual os requisitos de sistema, hardware ou software são apresentados aos envolvidos para seu acordo e aprovação.

- Especificação de requisitos: documento que especifica os requisitos de um sistema ou componente (incluindo requisitos funcionais, requisitos de desempenho, requisitos de interface e projeto, e padrões de desenvolvimento).
- Revisão dos requisitos de sistema: é a revisão conduzida para avaliar a completude e a adequação dos requisitos definidos para um sistema.
- Revisão dos requisitos de software: é a revisão dos requisitos especificados para um ou mais itens de configuração do software.
- Especificação de requisitos de software: é a documentação dos requisitos essenciais do software (requisitos funcionais, de desempenho, regras de projeto e atributos) e requisitos de interfaces externas.

7.1.7 Terminologias PMBOK

A seguir são apresentados os principais termos e seus significados, na área de requisitos aplicados no modelo **PMBOK**.

Planejamento de solicitações: refere-se ao ato de documentar os requisitos do produto e identificar suas fontes potenciais.

- Escopo do produto: as funcionalidades que caracterizam o produto ou serviço.
- Escopo do projeto: é o trabalho que deve ser feito para entregar um produto com as características e funcionalidades especificadas.
- Gerência de escopo de projeto: é um subconjunto da gerência de projeto que inclui os processos requisitados para assegurar que o projeto inclui todo o trabalho requisitado para concluí-lo com sucesso. Isto consiste em inicialização, planejamento de escopo, definição de escopo, verificação de escopo e controle de alteração de escopo.
- Controle de alteração de escopo: é entendido como o fato de controlar as mudanças do escopo do projeto.
- Definição de escopo: é subdividir os itens em partes menores a fim de prover melhor controle.

- Planejamento de escopo: é o processo progressivo de elaboração do trabalho de um projeto.
- Verificação de escopo: é a formalização da aceitação do escopo do projeto.

7.2 Correlações entre normas, padrões e modelos

Entre os modelos (ou normas e padrões) desenvolvidos pelo **SEI**, ISO, IEEE e **PMI** é possível identificar as semelhanças e diferenças entre as terminologias utilizadas para descrever cada um desses modelos e também as semelhanças e diferenças em relação as principais práticas definidas nesses modelos, normas e padrões. Como apresentado no item 7.1, é possível observar que os modelos da **SEI** utilizam definições da IEEE 830 e seus padrões relacionados. Por exemplo, a definição de requisito no modelo **CMMI** tem como base a definição descrita na IEEE 610.12-1990.

Assim como algumas terminologias se correlacionam entre os modelos apresentados, algumas das boas práticas definidas em cada um desses modelos também se relacionam. A seguir são apresentadas algumas dessas correlações, objetos de conclusão de um estudo realizado sobre os modelos que implementam processo e qualidade de software, possibilitando uma visão geral comparativa do enfoque de cada modelo ou norma.

As práticas e termos de cada modelo foram enquadrados em práticas globais que agrupam atividades da engenharia de requisitos encontradas nesses modelos e normas anteriormente apresentados. A Tabela 7-1 apresenta essas principais práticas globais da engenharia de requisitos e onde elas são descritas nos modelos e normas.

Prática	CMM			CMMI			ISO 9000-3	ISO 9001	ISO 9004	ISO 12297	IEEE 939	PMBOK
Extração de requisitos	N3 N3 N3	KPA5 KPA5 KPA5	Hab2 Ativ1 Ativ2	N3 N3 N3	KPA1 KPA1 KPA1	SG1SP11	Item 7.2.1	Item 7.2.1	Item 7.2.1	Item 5.1.1.2		
Documentação de requisitos	N2 N2 N3 N3	KPA1 KPA1 KPA5 KPA5	Com1S1 Hab2 Ativ2 Ativ2 S5	N3 N3 N3 N3	KPA1 KPA1 KPA1 KPA1	SG1SP12 SG2 SP2.1	Item 7.3.2	Item 7.3.2	Item 7.3.2	Item 5.1.2.1 Item 5.3.2.1 Item 5.3.4.1	Seção 4	Item 5.2.3.1
Especificação de requisitos	N3 N3	KPA5 KPA5	Ativ2 Ativ2 S3	N3 N3 N3	KPA1 KPA1 KPA1	SG2 SP2.1 SG2 SP2.2 SG2 SP2.3	Item 7.2.1			Item 5.3.2.1 Item 5.3.4.1	Seção 4	Item 5.3
Análise de requisitos	N2 N3 N3 N3	KPA1 KPA5 KPA5 KPA5	Hab1 Hab1 S1 Hab2 Ativ2 S2	N3 N3 N3 N3	KPA1 KPA1 KPA1 KPA1	SG3	Item 7.3.1	Item 7.3.1	Item 7.3.1	Item 5.3.2 Item 5.3.4		
Adequação e manutenção de artefatos do projeto em relação aos requisitos	N2 N2 N2 N2 N3 N3 N3 N3 N3 N3	KPA1 KPA1 KPA1 KPA1 KPA5 KPA5 KPA5 KPA5 KPA5 KPA5	Meta2 Com1S3 Ativ2 Ativ2 S2 Ativ3 S2 Ativ3 S11 Ativ4 Ativ4 S6 Ativ5 S8 Ativ10	N2 N2 N2 N2 N2 N2 N2 N2 N2 N2	KPA1 KPA1 KPA1 KPA1 KPA1 KPA1 KPA1 KPA1 KPA1 KPA1	SG1 SG1SP15	Item 7.2.1 Item 7.2.2 Item 7.3.7	Item 7.2.2	Item 7.2.2	Item 5.5.3.1		Item 5.2.3.1 Item 5.5.2
Revisão dos requisitos	N2 N2 N2 N3 N3 N3 N3	KPA1 KPA1 KPA1 KPA5 KPA5 KPA5 KPA5	Com1S2 Ativ1 Ativ3 Ativ2 S1,4,8,9,10 Ativ3 S2 Ativ4 S1 Verif3 S1	N3 N3 N3 N3 N3 N3 N3	KPA1 KPA1 KPA1 KPA1 KPA1 KPA1 KPA1	SG3	Item 7.2.2	Item 7.2.2	Item 7.2.2	Item 5.3.2.2 Item 5.3.3.2 Item 5.3.4.2 Item 5.3.4.3 Item 7.4.1.1	Item 4.3.3	Item 5.4.1
Controle de alterações dos requisitos	N3	KPA5	Ativ2 S12	N2 N2 N2	KPA1 KPA1 KPA1	SG1SP13 SG1SP15	Item 7.2.1 Item 7.2.2	Item 7.2.2	Item 7.1.3.3 Item 7.2.2	Item 5.5	Item 4.3.5	Item 5.5
Desenvolvimento de requisitos	N2 N3	KPA1 KPA5	Ativ2 S3 Ativ2	N3 N3 N3	KPA1 KPA1 KPA1	SG1 SG1SP12 SG2	Item 7.2.1 Item 7.3.2 Item 7.3.3	Item 7.3.2 Item 7.3.3	Item 7.3.2 Item 7.3.3	Item 5.3.6 Item 5.3.7		Item 5.3
Rastreabilidade dos requisitos	N3 N3 N3	KPA5 KPA5 KPA5	Com1S3 Ativ10 S2,3 Verif3 S9	N2 N2 N2	KPA1 KPA1 KPA1	SG1SP14	Item 7.2.1 Item 7.5.3	Item 7.5.3	Item 7.5.2	Item 5.3.2.2 Item 5.3.3.2 Item 5.3.4.2 Item 5.3.5.6 Item 5.3.6.7 Item 5.3.7.5 Item 5.3.8.5	Item 4.3.6	
Verificação	N3 N3	KPA5 KPA5	Ativ2 S7 Ativ6 S2,3	N3 N3	KPA1 KPA1	SG1SP12	Item 7.3.5	Item 7.3.5	Item 7.1.3.1 Item 7.3.5 Item 7.6	Item 5.4		Item 5.4
Validação	N3 N3	KPA5 KPA5	Ativ2 S7 Ativ6 S2,3	N3 N3	KPA1 KPA1	SG1SP12	Item 7.3.6	Item 7.3.6	Item 7.1.3.1 Item 7.3.6 Item 7.6	Item 5.5		
Auditoria e garantia da qualidade	N2 N2 N3	KPA1 KPA1 KPA5	Med1 Verif3 Verif3	N2 N3 N3	KPA1 KPA1 KPA1	GG2 GP2.9 GG3 GP2.9	8.2	8.2	8.2	Item 6.3 Item 6.7		
Critérios de aceitação e Aceitação formal do cliente	N2 N3 N3	KPA1 KPA5 KPA5	Hab2 S3 Ativ2 S6 Ativ7	N2 N2 N2	KPA1 KPA1 KPA1	SG1SP11	Item 7.1 Item 7.3.3	Item 7.1 Item 7.3.3	Item 7.1 Item 7.1.3.2	Item 5.3.13		Item 5.4.3
Gestão de configuração em requisitos	N2 N2 N3 N3	KPA1 KPA1 KPA5 KPA5	Meta1 Ativ2 S1 Ativ2 S11 Ativ7 S5	N2 N2 N3 N3	KPA1 KPA1 KPA1 KPA1	GG2 GP2.6 GG3 GP2.6	Item 7.5.3	Item 4.2.4 Item 7.5.3	Item 7.5.2	Item 8.2		Item 5.5.1 Item 5.5.3.1

Tabela 7-1 – Tabela de Correlação de Práticas de Requisitos

Tabela 7-2 – Legenda da Tabela 7-1

Legenda	CMM	CMMI
	N2 KPA1 = Requirements Management	N2 KPA1 = Requirements Management
	N3 KPA5 = Software Product Engineering	N3 KPA1 = Requirements Development
	Hab = Habilidade	GG = Generic Goal
	Ativ = Atividade	SG = Specific Goal
	Med = Medição	SP = Specific Practice
	Verif = Verificação	GP = Generic Practice
	S = Subprática	

Como pode também ser observado na Tabela 7-1, as normas ou modelos refletem preocupações diferentes quanto às práticas globais definidas nessa tabela.

Normas e modelos:

- O Guia ISO 9000-3 relaciona tais práticas ao software, descrevendo o processo relacionado ao cliente em parte do seu texto. A IEEE 830 descreve especificamente sobre especificação de requisitos de software. O modelo **PMBOK** reflete uma maior preocupação em gestão de requisitos (gerência de escopo), principalmente nos itens que se referem a planejamento, especificação e controle de alteração de requisitos.
- Comparando as normas ISO 9000-3 e ISO 9001 é observado que essas normas representam suas práticas de forma semelhante, sendo que o Guia ISO 9000-3 está relacionado diretamente a software e descreve algumas práticas adicionais, enquanto a ISO 9001 refere-se a produto de maneira genérica. A ISO 9004 também contém as práticas descritas na ISO 9001, sendo que há orientações adicionais descritas na ISO 9004. Por este motivo, são observados na Tabela 7-1 alguns itens adicionais em algumas práticas relacionadas a ISO 9000-3 e ISO 9004 .
- Como a norma ISO 12207 descreve o processo de ciclo de vida de software, praticamente podem ser encontradas todas as práticas relacionadas na Tabela 7-1.
- A norma ISO 9126-1 não descreve diretamente as práticas relacionadas à gestão e desenvolvimento de requisitos de software, pois seu foco é descrever quais características um software necessita ter, como manutenibilidade, segurança, confiabilidade, maturidade e outros, sendo que, conseqüentemente, essas características auxiliam os processos relacionados a requisitos. Por esse motivo não foi incluída na Tabela 7-1.

Analisando a Tabela 7-1, algumas observações comparativas podem ser obtidas, como mostram os itens a seguir:

- Análise de requisitos e revisão de requisitos: a prática **Análise de requisitos** se refere à análise técnica no nível de desenvolvimento de requisitos e tem como produto de trabalho o documento de especificação de requisitos em maior nível de detalhe. A prática de **Revisão dos requisitos** é genérica referenciando tanto a atividade aplicada aos requisitos do cliente ou requisitos do software e tem como produto de trabalho um relatório de revisão e o documento de requisitos corrigido.
- Controle de alteração de requisitos: Práticas de controle de alteração de requisitos e a manutenção dos artefatos do projeto são encontradas em vários modelos e normas, como mostram os itens **Adequação e manutenção de artefatos do projeto em relação aos requisitos** e **Controle de alterações de requisitos** da Tabela 7-1. Como é de grande importância permitir que requisitos sejam alterados em projetos de software, essas atividades são necessárias e precisam ser executadas de modo correto e efetivo, pois certamente traz grande benefício às organizações de software.
- Verificação, validação e critérios de aceite: Um dos grandes interesses das organizações é obter um alto grau de satisfação dos clientes. Para isso, a certificação de que os produtos de software satisfazem os requisitos do cliente e do produto tem sido uma atividade fundamental no processo de software. E então, pode ser observada a atenção dos modelos e normas apresentadas na Tabela 7-1, em se manter práticas de verificação, validação e práticas relacionadas a critérios de aceitação.
- Revisão de requisitos: Analisando os modelos **CMM** e **CMMI** através da Tabela 7-1 pode ser observada uma grande abrangência das práticas listadas e que se relacionam com requisitos. Também é possível observar uma grande preocupação desses modelos com a prática de revisão de requisitos, e isto se dá ao fato de que as pessoas com diferentes habilidades e diferentes grupos afetados estão envolvidas com atividades de revisão de requisitos. Isso reflete a cobertura que os modelos **CMM** e **CMMI** dão em relação à qualidade dos requisitos, ou seja, qualidade

naquilo que pode ser chamado de alicerce de um projeto de software. Com relação à prática de revisão de requisitos, essa mesma preocupação pode ser observada na norma ISO 12207. Outras normas e modelos também apresentam práticas de revisão de requisitos.

- Gestão de configuração: Práticas como gestão de configuração de requisitos são tratadas explicitamente pelos modelos **CMM** e **CMMI**. Entretanto, outros modelos e normas também citam atividades que podem ser relacionadas com gestão de configuração de requisitos, como exemplo, o modelo **PMBOK** também cita sobre *baseline* de requisitos, e as normas ISO 9001, 9000-3, 9004 e 12207 referem-se à gestão de configuração de maneira mais genérica.
- Documentação de requisitos: Boas práticas, como documentação de requisitos, são encontradas em todos os modelos e normas listadas na Tabela 7-1, e isso reflete a necessidade do controle e uso dos requisitos do projeto de software.

7.3 Considerações finais

Este capítulo descreveu as terminologias utilizadas pelos modelos e normas apresentados nos capítulos anteriores e também apresentou uma tabela correlacionando as boas práticas de requisitos desses modelos e normas.

A tabela de correlação de práticas de requisitos (Tabela 7-1) mostra grande abrangência das principais práticas de requisitos pelos modelos **CMM**, **CMMI**, normas da família ISO 9000 e ISO 12207, e também mostra que os modelos IEEE 830 e **PMBOK** apresentam práticas mais específicas em especificação de requisitos e gestão de requisitos, respectivamente, não abrangendo todas as principais práticas de requisitos.

Capítulo 8 Análise da relevância de práticas de requisitos a partir de dados experimentais

Como visto nos capítulos anteriores vários modelos de processo e qualidade de software descrevem boas práticas de requisitos e isso também se deve a importância do uso dessas práticas durante a produção de software. O fato de vários modelos apresentarem práticas relacionadas a requisitos motivou o estudo da correlação entre esses modelos, como foi apresentado no Capítulo 7.

Com o intuito de obter uma visão geral do que acontece em projetos de software reais são apresentados neste capítulo os resultados de um estudo que buscou analisar a relevância do uso de um conjunto de boas práticas de requisitos.

O item 8.1 apresenta detalhadamente a abordagem utilizada para obter os resultados almejados através de uma classificação e contabilização de solicitações¹ registradas em projetos de software. O item 8.2 apresenta passo a passo a execução do estudo experimental.

8.1 Metodologia

Este item apresenta a metodologia proposta para a realização da classificação e contabilização das solicitações relacionadas a requisitos (solicitações que poderiam ter sido evitados se as boas práticas de requisitos tivessem sido corretamente implementadas) e para a obtenção dos resultados almejados.

Os passos referentes à abordagem utilizada nesse estudo experimental são apresentados a seguir:

¹ Neste capítulo entende-se “solicitação” de software como sendo um registro de uma falha ou uma solicitação de novo requisito/alteração de requisitos ou uma solicitação que reflete uma necessidade por falta de requisitos ou problemas relacionados a requisitos.

Passo 1: Selecionar projetos de software de uma mesma empresa, que sejam da mesma área de domínio e que tenham as seguintes características:

- Processo de ciclo de vida semelhante.
- Plataforma Windows.
- Cliente Externo.
- Possuam registros das solicitações para o projeto de software.

Passo 2: Apresentar os cenários e características detalhadas de cada um dos projetos selecionados com relação a:

- Processo de ciclo de vida.
- Duração do projeto.
- Esforço.
- Quantidade de recursos humanos.
- Número de *baselines* de requisitos planejadas e realizadas.
- Características do cliente.
- Características das fases do ciclo de vida do projeto e produtos de trabalho.

Passo 3: Obter os registros de prazo e esforço (planejado e realizado) de cada um dos projetos de software selecionados.

Passo 4: Analisar o desempenho dos projetos selecionados com relação aos indicadores de prazo e esforço.

Passo 5: Obter os registros das solicitações dos projetos de software selecionados.

Passo 6: Classificar e contabilizar as solicitações dos projetos selecionados, separando-as em dois grandes grupos:

Grupo das solicitações relacionadas a requisitos (solicitações que poderiam ter sido evitados se as boas práticas de requisitos tivessem sido corretamente implementadas), subdividido em:

- **GrupoReq_ReqNov** = classifica todas as solicitações de novos requisitos para o projeto de software.
- **GrupoReq_AltReq** = classifica todas as solicitações registradas de alteração em requisitos do projeto de software.
- **GrupoReq_AltReqTec** = classifica todas as solicitações registradas de alteração de requisito em documentação técnica de domínio específico.
- **GrupoReq_DefimpReq** = classifica todas as solicitações registradas que relatam falhas no software por não estar implementado de acordo com os seus requisitos.
- **GrupoReq_ReqNaoEsp** = classifica todas as solicitações registradas que induzem o entendimento de que faltaram requisitos a serem especificados para o software.
- **GrupoReq_ArtAdeReq** = classifica todas as solicitações registradas que induzem o entendimento de que os artefatos do software (como documentos de especificação e procedimentos de teste) não foram criados ou mantidos coerentes com os requisitos do software.

Grupo das solicitações não relacionadas a requisitos, subdividido em:

- **GrupoNaoReq_ErrImp** = classifica todas as solicitações que retratam erros no software de maneira geral.
- **GrupoNaoReq_ErrIHC** = classifica todas as solicitações que retratam falhas de IHC (Interface Humano-Computacional) no software.

Passo 7: Analisar a classificação e contabilização das solicitações relacionando-as com os indicadores de prazo e esforço dos projetos.

Passo 8: Correlacionar as solicitações do “**Grupo das solicitações relacionadas a requisitos**” com as boas práticas de requisitos definidas na Tabela 7-1 do Capítulo 7 que forem possíveis de serem correlacionadas considerando as informações obtidas das solicitações.

A partir dos resultados obtidos gerar gráfico com esses resultados.

Passo 9: Analisar os resultados da correlação das solicitações do “**Grupo das solicitações relacionadas a requisitos**” com as boas práticas de requisitos.

Passo 10: Recomendar atividades que tendem a agregar valor na implementação das boas práticas de requisitos.

8.2 Execução

A seguir são apresentados pontos importantes que foram considerados na realização deste estudo e os passos realizados para chegar aos resultados.

8.2.1 Passo 1 – Seleção de projetos de software

Foram selecionados quatro projetos de software de uma mesma empresa:

- ✓ Projeto A
- ✓ Projeto B
- ✓ Projeto C
- ✓ Projeto D

Os projetos selecionados foram esses por serem projetos semelhantes no que diz respeito a:

- Cliente externo.
- Plataforma Windows.

-
- Utilização de requisitos da área de conhecimento específico (Telecomunicações).
 - Projetos conhecidos pelo autor deste estudo, pois possibilitou a contabilização e classificação das solicitações uma vez que os registros dessas solicitações não possuíam tais informações.

8.2.2 Passo 2 – Cenários e Características dos projetos

Os quatro projetos selecionados foram desenvolvidos utilizando processos de ciclo de vida semelhantes. Basicamente as fases do projeto foram estas:

1. Fase de proposta = elaboração e efetivação de contrato entre as partes (fornecedor e cliente).
2. Fase de requisitos = levantamento de requisitos, elaboração e revisão dos documentos de requisitos seguido de acompanhamento dos requisitos durante todo o ciclo.
3. Fase de planejamento = elaboração dos planos e cronograma do projeto incluindo planejamento de gestão de configuração com as seguintes *baselines*:
 - ✓ *Baseline* de proposta: representa um ponto de congelamento e de referência dos documentos de propostas finalizados.
 - ✓ *Baselines* de requisitos: representa um ponto de congelamento e de referência dos documentos de requisitos finalizados.
 - ✓ *Baselines* de integração e testes sistêmicos: representa um ponto de congelamento e de referência da codificação finalizada.
 - ✓ *Baseline* de *release*: representa um ponto de congelamento e de referência da liberação de uma versão do software.
4. Fase de análise = definição, elaboração e revisão dos casos do uso do projeto, especificação dos casos de testes e arquitetura do sistema.
5. Fase de projeto = elaboração dos documentos de núcleo (proximidade com a codificação), especificação de interface humano-computacional e procedimentos de testes.

6. Fase de codificação e teste unitário = codificação dos casos de uso e teste unitário.
7. Fase de teste de integração e teste sistêmico = testes realizados pela equipe de teste.
8. Fase de empacotamento = empacotamento do sistema incluindo manuais do sistema.

A Tabela 8-1 mostra um resumo dos cenários de cada um dos projetos de software selecionados e os itens a seguir apresentam os cenários mais detalhados aos quais se configuram os projetos A, B, C e D.

Tabela 8-1 – Resumo dos cenários dos projetos

Projeto	Duração (meses)	Esforço (homens.hora)	Cliente	Extração de Requisitos	Revisão de Requisitos	Obs
A	3,3	3.192	Externo	NOK	Superficial	Dificuldade alta no controle de requisitos
B	2,6	2.488	Externo	NOK	Superficial	Dificuldade média no controle de requisitos
C	3,5	2.672	Externo	Parcialmente OK	Média	Requisitos controlados
D	3,7	2.568	Externo	Parcialmente OK	Média	Requisitos controlados

8.2.2.1 Projeto A

O Projeto A se caracteriza por um sistema de software (plataforma Windows) de domínio específico, com um planejamento aproximado de 3,3 meses com sete recursos² (3.192 homens.hora) e mais um recurso de área específica de conhecimento para consultoria e definição de requisitos do domínio específico ao qual o sistema se baseia. O número de *baselines* de requisitos planejadas para o Projeto A foi de duas *baselines*, e,

² No item 8.2.2, entende-se “recursos” como “pessoas”, exemplo: sete recursos = sete pessoas.

no entanto, foram realizadas sete *baselines* de requisitos. A seguir são apresentados os fatores que caracterizaram esse projeto.

1. O cliente do Projeto A caracterizou-se por:
 - ✓ Cliente externo.
 - ✓ Levantamento de requisitos desordenado (através de entrevista) e sem processo de aprovação dos requisitos devidamente compreendido.
 - ✓ Desinteresse do cliente em entender e aprovar os requisitos do sistema e do software, resultando em aprovação de requisitos pelo gerente do projeto por decurso de prazo.
 - ✓ Não cumprimento de datas e tarefas acordadas.
 - ✓ Testes de campo e aceitação de sistema realizados com pouca ou nenhuma relação com os requisitos acordados.
2. Fases do projeto de software e produtos de trabalho do Projeto A caracterizaram-se por:
 - ✓ Levantamento de requisitos sem a objetividade adequada ao tipo de sistema de software requerido.
 - ✓ Especificação de requisitos elaborada a partir de casos de uso resultando em um documento muito extenso e em linguagem desconhecida do cliente.
 - ✓ Revisões incompletas e superficiais dos documentos de requisitos pelos grupos afetados.
 - ✓ Fase de projeto comprometida pelo cronograma irreal.
 - ✓ Implementação baseada na especificação de casos de uso e em outros documentos não controlados (documentos elaborados ou fornecidos de forma aleatória, sem padrão e sem controle de qualidade efetivo).
 - ✓ Testes baseados em especificação de casos de testes de nível superficial.
 - ✓ Dificuldade em controlar alterações de requisitos, pois os mesmos estavam no nível de caso de uso, resultando em muito esforço para

manter o controle de alterações e também resultando em descontrole de algumas alterações sofridas.

8.2.2.2 Projeto B

O Projeto B se caracteriza por um sistema de software (plataforma Windows) de domínio específico, com um planejamento aproximado de 2,6 meses com 3,3 recursos (2.488 homens.hora) e mais três recursos de área específica de conhecimento para consultoria e definição de requisitos do domínio específico ao qual o sistema se baseia. O número de *baselines* de requisitos planejadas para o Projeto B foi de uma *baseline*, e, no entanto, foram realizadas quatro *baselines* de requisitos. A seguir são apresentados os fatores que caracterizaram esse projeto.

1. O cliente do Projeto B caracterizou-se por:
 - ✓ Cliente externo.
 - ✓ Levantamento de requisitos pouco ordenado (através de entrevista) e sem processo de aprovação dos requisitos devidamente compreendido.
 - ✓ Desinteresse em aprovar os requisitos do sistema e do software.
 - ✓ Não cumprimento de datas e tarefas acordadas.
 - ✓ Testes de campo e aceitação de sistema realizados com pouca relação com os requisitos acordados.
2. Fases do projeto de software e produtos de trabalho do Projeto B caracterizaram-se por:
 - ✓ Levantamento de requisitos sem objetividade e baseado em requisições do cliente não devidamente contextualizadas.
 - ✓ Especificação de requisitos elaborada com base em funcionalidades do sistema resultando em um documento sucinto.
 - ✓ Revisões incompletas e superficiais dos documentos de requisitos pelos grupos afetados.
 - ✓ Fase de projeto comprometida pelo cronograma irreal.
 - ✓ Implementação baseada na especificação de casos de uso e em outros documentos não controlados (documentos elaborados ou fornecidos de forma aleatória, sem padrão e sem controle de qualidade efetivo).

- ✓ Testes baseados em especificação de casos de testes de nível superficial.
- ✓ Dificuldade em controlar alterações de requisitos.

8.2.2.3 Projeto C

O Projeto C se caracteriza por um sistema de software (plataforma Windows) de domínio específico, com um planejamento aproximado de 3,5 meses com 3,4 recursos (2.672 homens.hora) e mais dois recursos de área específica de conhecimento para consultoria e definição de requisitos do domínio específico ao qual o sistema se baseia. O número de *baselines* de requisitos planejadas para o Projeto C foi de duas *baselines*, e, no entanto, foram realizadas três *baselines* de requisitos. A seguir são apresentados os fatores que caracterizaram esse projeto.

1. O cliente do Projeto C caracterizou-se por:
 - ✓ Cliente externo.
 - ✓ Levantamento de requisitos ordenado (através de entrevista) e com processo de aprovação dos requisitos devidamente compreendido.
 - ✓ Interesse em aprovar os requisitos do sistema e do software, porém com algumas poucas deficiências na compreensão dos mesmos.
 - ✓ Cumprimento de datas e tarefas acordadas.
 - ✓ Testes de campo e aceitação de sistema realizados com bom nível de relação com os requisitos acordados.
2. Fases do projeto de software e produtos de trabalho do Projeto C caracterizaram-se por:
 - ✓ Levantamento de requisitos com objetividade.
 - ✓ Especificação de requisitos elaborada com base em funcionalidades do sistema resultando em um documento sucinto.
 - ✓ Revisões completas dos documentos de requisitos e com bom nível de compreensão pelos grupos afetados.
 - ✓ Fase de projeto pouco comprometida pelo cronograma irreal.

- ✓ Implementação baseada na especificação de casos de uso e em outros documentos controlados (documentos elaborados ou fornecidos de forma padronizada e com algum tipo de controle de qualidade, como por exemplo, especificação de banco de dados, especificação de núcleo, especificação de Interface Homem-Máquina e outros).
- ✓ Testes baseados em especificação de casos de testes e procedimento de testes de nível superficial.
- ✓ Requisitos rigorosamente controlados, entretanto houve deficiências na manutenção e adequação dos artefatos (outros documentos impactados por tais alterações).

8.2.2.4 Projeto D

O Projeto D se caracteriza por um sistema de software com interface WEB (plataforma Windows) de domínio específico, com um planejamento aproximado de 3,7 meses com 3,8 recursos (2.568 homens.hora) e mais três recursos de área específica de conhecimento para consultoria e definição de requisitos do domínio específico ao qual o sistema se baseia. O número de *baselines* de requisitos planejadas para o Projeto D foi de duas *baselines*, e, no entanto, foram realizadas seis *baselines* de requisitos. A seguir são apresentados os fatores que caracterizaram esse projeto.

1. O cliente do Projeto D caracterizou-se por:

- ✓ Cliente externo.
- ✓ Levantamento de requisitos ordenado (através de entrevista e questionário) e com processo de aprovação dos requisitos devidamente compreendido, porém com burocratização extrema por parte do cliente.
- ✓ Desinteresse em entender e aprovar os requisitos do sistema e do software, inclusive com desentendimento entre as diferentes áreas envolvidas da empresa cliente. Entretanto os requisitos foram aprovados pelo cliente com sua plena ciência.
- ✓ Não cumprimento de datas e tarefas acordadas.
- ✓ Testes de campo e aceitação de sistema realizados com bom nível de relação com os requisitos acordados.

2. Fases do projeto de software e produtos de trabalho do Projeto D caracterizaram-se por:

- ✓ Levantamento de requisitos com objetividade através de questionários e cartas.
- ✓ Especificação de requisitos elaborada com base em funcionalidades do sistema resultando em um documento sucinto.
- ✓ Revisões completas dos documentos de requisitos e com bom nível de compreensão pelos grupos afetados.
- ✓ Fase de projeto pouco comprometida pelo cronograma irreal.
- ✓ Implementação baseada na especificação de casos de uso e em outros documentos controlados (documentos elaborados ou fornecidos de forma padronizada e com algum tipo de controle de qualidade, como por exemplo, especificação de banco de dados, especificação de núcleo, especificação de Interface Homem-Máquina e outros).
- ✓ Testes baseados em especificação de casos de testes e procedimento de testes de bom nível de detalhamento.
- ✓ Requisitos rigorosamente controlados, entretanto houve poucas deficiências na manutenção e adequação dos artefatos (outros documentos impactados por tais alterações).

8.2.3 Passo 3 – Desempenho dos projetos

Este item apresenta uma visão geral do desempenho de cada um dos projetos selecionados e para isso foram utilizadas as medidas de esforço e prazo (planejado e realizado) que estavam registradas e disponíveis de cada projeto, como mostra a Tabela 8-2.

Tabela 8-2 – Esforço/Prazo dos Projetos

Projetos	EsforçoPlanejado (pd)	EsforçoReal (pd)	PrazoPlanejado (dias)	PrazoReal (dias)
Projeto A	399	685	100	91
Projeto B	311	347	107,5	76
Projeto C	334	436	70	87
Projeto D	321	400	107	131

8.2.4 Passo 4 – Análise comparativa do desempenho dos projetos

Por meio das medidas de esforço, prazo e tamanho registradas foi possível verificar as variações de esforço e prazo de cada projeto (Tabela 8-3) e a produtividade relativa de cada projeto (Tabela 8-4).

As medidas de esforço foram disponibilizadas em pd (“pessoas.dia”). As medidas de prazo foram disponibilizadas em “dias úteis”. E as medidas de tamanho foram disponibilizadas em “ut” que significa “unidade de trabalho” que é uma unidade criada e utilizada especificamente pela empresa que desenvolveu os quatro projetos selecionados e é calculada pela multiplicação de fatores relacionados à complexidade, número de objetos, tipo de interface e porcentagem de retrabalho por macro caso de uso. O projeto A disponibilizou medida de tamanho apenas em número de casos de uso. Para fins de análise o número de ut relacionado na Tabela 8-4 para o projeto A foi criado por estimativa simplificada.

Tabela 8-3 – Variação de Esforço/Prazo dos Projetos

Projetos	Qtde Replanejamentos	EsforçoPlan (pd)	EsforçoReal (pd)	Variação %	PrazoPlan (dias)	PrazoReal (dias)	Variação %
Projeto A	n/a	399	685	71,7%	100	91	-9,0%
Projeto B	4	311	347	11,6%	107,5	76	-29,3%
Projeto C	3	334	436	30,5%	70	87	24,3%
Projeto D	3	321	400	24,6%	107	131	22,4%

O quadro a seguir (Tabela 8-4) mostra um fator de produtividade de cada projeto calculado através da divisão da medida de esforço real por tamanho real. Quanto menor o número que representa o fator de produtividade melhor produtividade teve o projeto.

Tabela 8-4 – Tamanho dos Projetos

Projetos	ut_Plan	ut_Real	Variação %	Produtividade EsforçoReal/ut_Real
Projeto A	340	340	0,0%	2,01
Projeto B	337,4	337,4	0,0%	1,03
Projeto C	199,6	174	-12,8%	2,51
Projeto D	198	212	7,1%	1,89

A partir dos números apresentados na Tabela 8-3 e Tabela 8-4 pode-se verificar que o projeto A teve a maior variação de esforço seguido respectivamente dos projetos C, D e B (nesta ordem). Os projetos A e B apresentaram variação de prazo negativo, ou seja, a duração real do projeto que foi medido durante o projeto foi menor que a duração planejada. Entretanto o projeto D teve menor variação de prazo em comparação com o projeto C. Combinando os números apresentados na Tabela 8-3 e Tabela 8-4 e os cenários dos projetos não é possível concluir diretamente qual projeto teve melhor desempenho no geral pois vários fatores influenciam neste caso, como por exemplo, o nível de controle do projeto referente ao número de replanejamentos, a eficiência na coleta das medidas utilizadas e também os resultados obtidos neste estudo que são apresentados nos itens 8.2.6, 8.2.8 e 8.2.9.

8.2.5 Passo 5 – Obtenção dos registros de solicitações

Durante a execução dos projetos selecionados para esse estudo foram registradas as solicitações referentes ao software em uma ferramenta que mantém esses registros em base de dados. E através da base de dados dessa ferramenta foram obtidos os registros de cada solicitação de maneira geral relacionada ao software. O registro dessas solicitações obtidas foram feitos durante as fases de testes e testes com o cliente, salvo as exceções relacionadas a solicitações de novos requisitos ou alterações de requisitos ocorridas durante o desenvolvimento do sistema de software.

8.2.6 Passo 6 – Classificação e Contabilização das solicitações

O conteúdo descritivo, de cada solicitação registrada, foi interpretado e julgado e assim foi classificado como sendo de um determinado tipo de solicitação. Como a classificação foi feita com base em interpretação humana e em alguns casos foi necessário arbitrar, uma margem de erro pode ser considerada.

Os tipos de solicitações registradas foram classificados em dois grupos: o grupo das solicitações relacionadas a requisitos por deficiência no uso de boas práticas de requisitos (denominado **GrupoReq**) e o grupo das solicitações sem relação direta com requisitos ou boas práticas de requisitos (denominado **GrupoNaoReq**). Algumas solicitações não foram possíveis de serem classificadas devido à insuficiência de informações.

Cada um desses dois grupos (**GrupoReq** e **GrupoNaoReq**) foi dividido em subgrupos como segue:

- **GrupoReq**
 - ✓ GrupoReq_ReqNov
 - ✓ GrupoReq_AltReq
 - ✓ GrupoReq_AltReqTec
 - ✓ GrupoReq_DefImpReq (o nível de detalhe para entendimento de requisitos nesse caso é ao nível de caso de uso)
 - ✓ GrupoReq_ReqNaoEsp
 - ✓ GrupoReq_ArtAdeReq
- **GrupoNaoReq**
 - ✓ GrupoNaoReq_ErrImp
 - ✓ GrupoNaoReq_ErrIHC

Uma planilha foi utilizada para classificar as solicitações registradas nos grupos e subgrupos e também para contabilizá-las. Os resultados obtidos estão representados na Tabela 8-5 e Tabela 8-6.

Tabela 8-5 – Solicitações por Projeto

	Projeto A		Projeto B		Projeto C		Projeto D	
Total de Solicitações	517		331		255		198	
	Qtde	%	Qtde	%	Qtde	%	Qtde	%
GrupoReq	244	47,2%	164	49,5%	80	31,3%	109	55,0%
GrupoNaoReq	239	46,2%	128	38,6%	140	54,9%	77	38,8%

Pode-se notar que uma parte significativa (de 31% a 55%) das solicitações é resultante das deficiências relacionadas às práticas de requisitos.

Tabela 8-6 – Ocorrência de Solicitação por Tipo de Solicitações e por Projeto

	Projeto A		Projeto B		Projeto C		Projeto D	
Total GrupoReq	244		164		80		109	
	n	%	n	%	n	%	n	%
GrupoReq_ReqNov	14	5,74	5	3,05	0	0	6	5,50
GrupoReq_AltReq	28	11,48	58	35,37	1	1,25	10	9,17
GrupoReq_AltReqTec	7	2,87	7	4,27	3	3,75	13	11,93
GrupoReq_DefImpReq	127	52,05	44	26,83	33	41,25	44	40,37
GrupoReq_ReqNaoEsp	58	23,77	37	22,56	29	36,25	29	26,61
GrupoReq_ArtAdeReq	10	4,10	13	7,93	14	17,5	7	6,42
Total	244	100	164	100	80	100	109	100

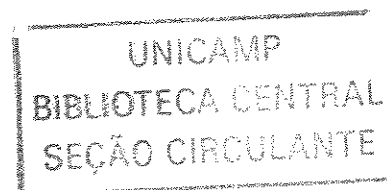
8.2.7 Passo 7 – Análise comparativa entre as solicitações e o desempenho dos projetos selecionados

A Tabela 8-7 apresenta uma normalização para que seja possível uma visão geral referente ao número de solicitações registradas por tamanho de projeto de cada projeto selecionado. Quanto maior o número que representa esse fator (tot/ut_Real) pior qualidade teve o projeto, pois indica um alto índice de solicitações registradas.

Tabela 8-7 – Normalização

Projetos	ut_Real	Solicitações		SolicitaçõesReq	
		tot	tot/ut_Real	tot	tot/ut_Real
Projeto A	340	517	1,52	244	0,72
Projeto B	337,4	331	0,98	164	0,49
Projeto C	174	255	1,47	80	0,46
Projeto D	212	198	0,93	109	0,51

O Gráfico 8.1 apresenta o desvio de esforço por projeto versus a normalização resultante do número de solicitações por tamanho de projeto.



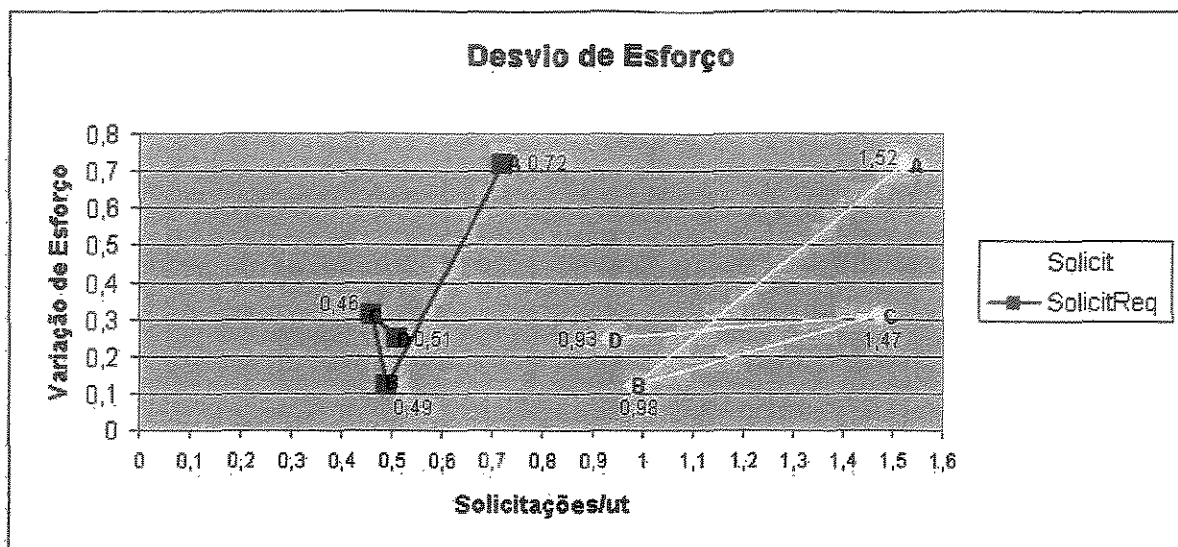


Gráfico 8.1 – Desvio de Esforço

O Gráfico 8.2 apresenta o desvio de prazo por projeto versus a normalização resultante do número de solicitações por tamanho de projeto.

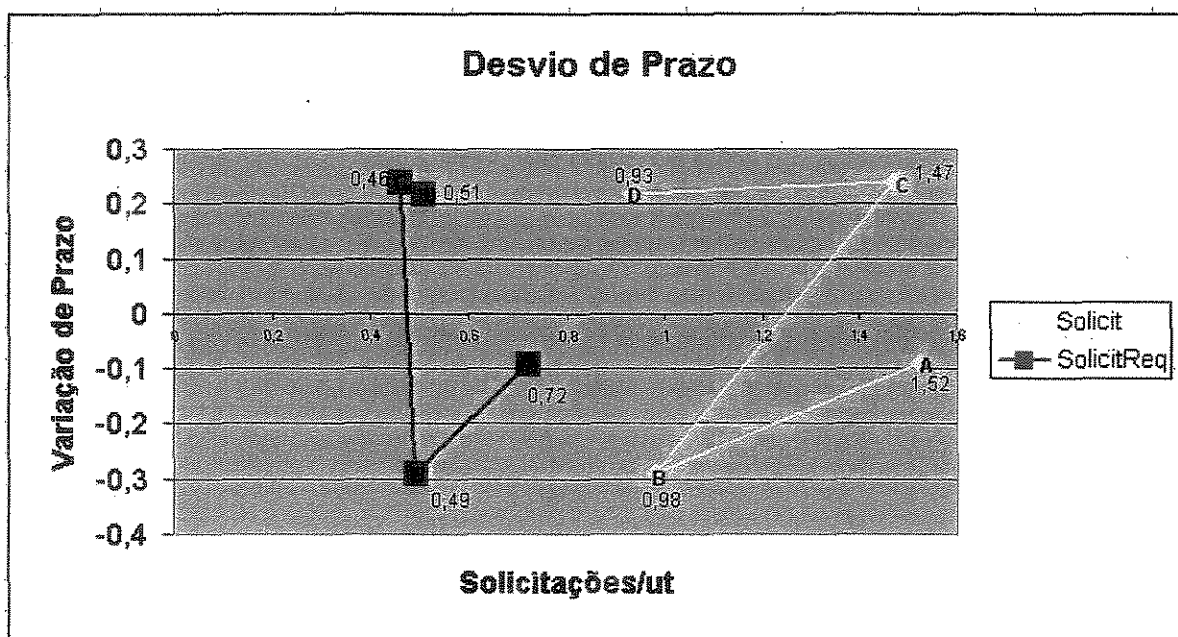


Gráfico 8.2 – Desvio de Prazo

Diante do que foi ilustrado pelo Gráfico 8.1 e Gráfico 8.2, pode-se analisar que não foi possível determinar qual dos projetos teve melhor desempenho ou melhor qualidade apenas analisando os indicadores de número de solicitações e variações de

esforço e prazo. Os cenários, as características e as solicitações relacionadas a requisitos que caracterizam cada projeto são fundamentais para se analisar desempenho e qualidade dos projetos e por isso o item 8.2.8 apresenta mais um nível de detalhamento na análise dos resultados encontrados correlacionando as solicitações relacionadas a requisitos com as boas práticas de requisitos.

8.2.8 Passo 8 – Correlação com boas práticas de requisitos

A seguir são apresentadas algumas análises sobre os resultados apresentados com referências as principais boas práticas de requisitos apresentadas na Tabela 7-1 do Capítulo 7. Para elaborar a Tabela 8-8 cada solicitação relacionada a requisitos foi classificada como sendo consequência da deficiência na implementação das práticas relacionadas nessa tabela.

A partir desse ponto, todas as solicitações classificadas dentro do grupo **GrupoReq**, foram classificadas, de acordo com peso de relevância, como sendo impactadas por algumas das boas práticas genéricas extraídas da tabela de correlação (Tabela 7-1) do Capítulo 7, e que são elas:

- ✓ Extração de Requisitos
- ✓ Documentação
- ✓ Manutenção e Adequação de Artefatos
- ✓ Revisão
- ✓ Controle de Alterações

Obs: Outras boas práticas, como rastreabilidade, verificação & validação, critérios de aceitação e aceitação formal do cliente, gestão de configuração em requisitos não foram utilizadas na classificação e contabilização por não se dispor de informações suficientes. As possíveis causas dessa ausência de informações são deduzidas da apresentação dos cenários dos projetos descrita no item 8.2.2.

Para obter os números e conseqüentemente a porcentagem relacionada a cada prática, as solicitações, que poderiam ser consequência da deficiência de mais de uma prática (superposição), foram contabilizadas em todas as práticas relacionadas a elas, porém com peso diferenciado a fim de somar o valor um para cada solicitação,

independentemente de quantas práticas apresentaram deficiências resultando na geração daquela solicitação. Para melhor ilustrar, segue um exemplo: uma solicitação registrada que poderia ter sido evitada se as práticas de extração de requisitos, documentação e revisão fossem bem utilizadas a classificação dessa solicitação ficou com peso 0,33 para cada prática, somando o valor um, pois 0,33 vezes três práticas é igual a um. Se uma outra solicitação poderia ter sido evitada se as práticas de revisão e controle de alterações fossem bem utilizadas a classificação ficou com peso 0,5 para cada prática, somando um, pois 0,5 vezes duas práticas é igual a um.

Tabela 8-8 – Boas Práticas de Requisitos e Solicitações

	Projeto A		Projeto B		Projeto C		Projeto D	
Total Solicitações	517		331		255		198	
	n	% do Total	n	% do Total	n	% do Total	n	% do Total
Total GrupoReq	244	47,20	164	49,55	80	31,37	109	55,05
	n	% GrupoReq	n	% GrupoReq	n	% GrupoReq	n	% GrupoReq
Extração de requisitos	47,57	19,50	66,42	40,50	12,75	15,94	18,77	17,22
Documentação	88,99	36,47	38	23,17	20,92	26,15	42,08	38,61
Adequação de artefatos	72,81	29,84	40,42	24,65	29,42	36,78	29,74	27,28
Revisão	25,64	10,51	9,17	5,59	13,41	16,76	17,41	15,97
Controle de alterações	8,98	3,68	5,99	3,65	0,5	0,63	1	0,92
Não Classificados	0	0,00	4	2,44	3	3,75	0	0,00
Total	244	100,00	164	100,00	80	100,00	109	100,00

A seguir podemos ver a Tabela 8-9 e o Gráfico 8.3 que retrata os números resultantes dessa classificação.

Tabela 8-9 – Classificação por boas práticas

	REQ		Extração		Documentação		Adequação de artefatos		Revisão		Controle de alterações	
	n	%	n	%	n	%	n	%	n	%	n	%
PROJ A	244	47,19%	47,56	19,49%	88,98	36,46%	72,81	29,84%	25,64	10,51%	8,98	3,68%
PROJ B	164	49,54%	66,41	40,49%	37,99	23,17%	40,41	24,64%	9,16	5,58%	5,99	3,65%
PROJ C	80	31,37%	12,75	15,93%	20,92	26,14%	29,42	36,77%	13,41	16,77%	0,50	0,62%
PROJ D	109	55,05%	18,75	17,20%	42,08	38,60%	29,74	27,29%	17,41	15,97%	1,00	0,91%

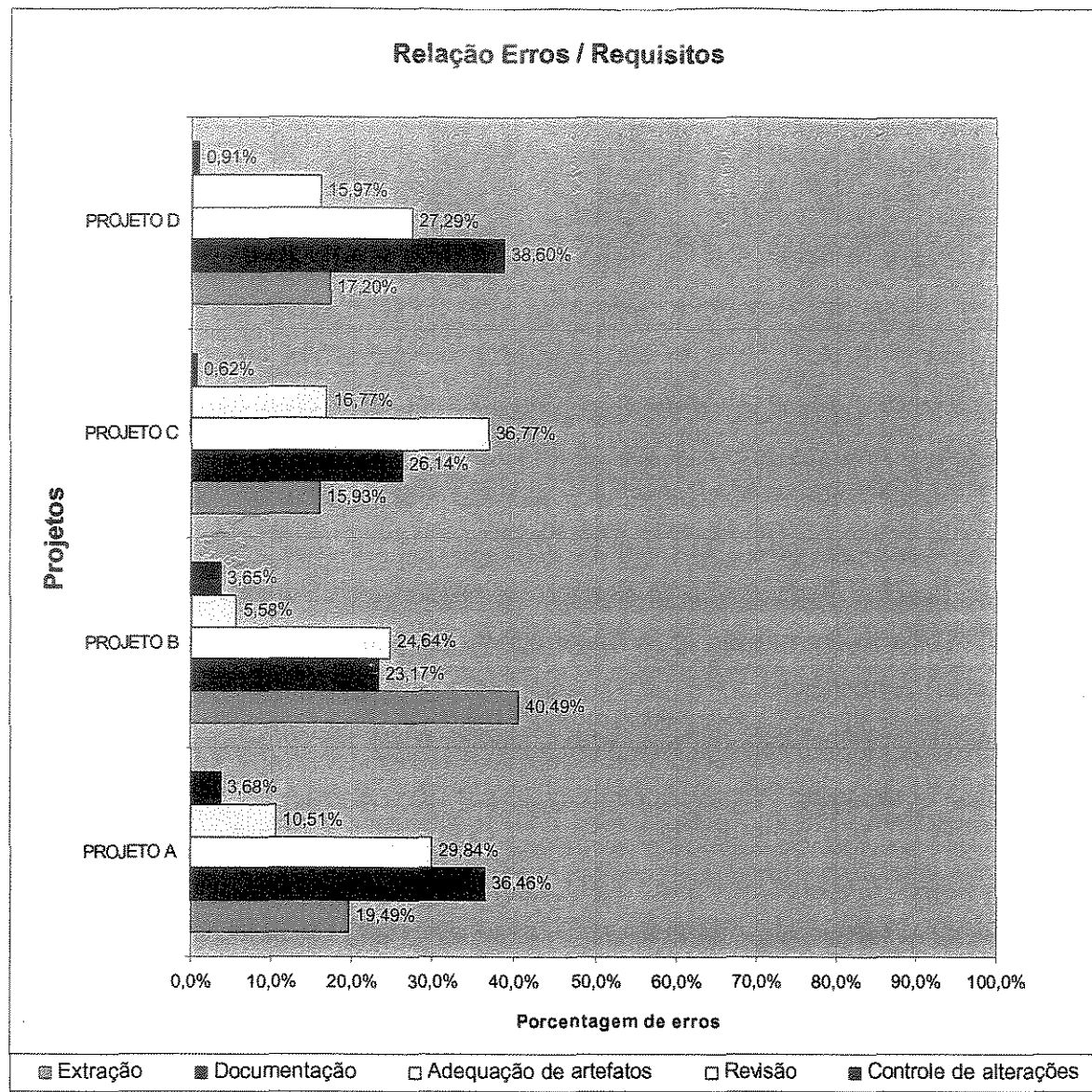


Gráfico 8.3 – Relação de Erros por Práticas de Requisitos

8.2.9 Passo 9 – Análise dos resultados

Por intermédio dos números dos resultados apresentados pode-se concluir, de maneira geral, que as boas práticas de requisitos são fundamentais para um projeto de software, pois o índice de incidência de solicitações relacionadas a requisitos é alto obtendo-se uma porcentagem média de 45% de solicitações relacionadas a requisitos. Observando essa porcentagem, pode-se ter uma visão da relevância que as boas práticas de requisitos tem sobre projetos de software.

Nos resultados apresentados pode-se notar um maior índice de solicitações provocadas por deficiências na extração de requisitos e documentação de requisitos. Esses altos índices configuram-se pelo tipo de software em questão, que exige grande esforço e atenção na documentação dos requisitos por se tratar de software de domínio específico, e também pelo cenário de cada projeto (item 8.2.2).

O Projeto A teve seu pior índice em documentação, 36% das solicitações relacionadas a requisitos foram configuradas por problemas em documentação dos requisitos. Isso se deve ao fato de que esse projeto necessitava que os requisitos de domínio específico fossem documentados adequadamente. Como este conhecimento específico estava fora do alcance da equipe de software, dependendo da interação com a equipe de especialistas no assunto em questão, deficiências na comunicação e deficiências no entendimento das necessidades para se construir o software foram os principais agravantes. Esse projeto apresentou, de maneira geral, um nível muito baixo de satisfação³ do cliente em relação ao software que lhe foi entregue comparando com suas necessidades e expectativas.

O Projeto B teve seu pior índice em extração de requisitos, 40% das solicitações relacionadas a requisitos foram configuradas por deficiências no levantamento de requisitos. Isso se deve ao fato de que esse projeto necessitava de um maior esforço na fase de extração de requisitos. Como houve deficiências no entendimento por parte do cliente e as técnicas utilizadas para levantar os requisitos não foram eficientes para esse caso, muitos requisitos não foram registrados elevando assim esse índice. Esse projeto apresentou, de maneira geral, um nível baixo de satisfação do cliente em relação ao software que lhe foi entregue comparando com suas necessidades e expectativas.

O Projeto C teve seu pior índice em adequação de artefatos, 36% das solicitações relacionadas a requisitos foram configuradas por falhas em manutenção e adequação de artefatos do projeto de software aos seus requisitos. Problemas relacionados a questões de domínio específico que não tiveram documentação de requisitos apropriada e conseqüentemente as documentações durante as fases do projeto não refletiam esses requisitos de domínio específico resultaram nesse índice. Esse projeto apresentou, de maneira geral, um bom nível de satisfação do cliente em relação ao software que lhe foi entregue comparando com suas necessidades e expectativas.

³ Para esses quatro projetos de software, entende-se que o nível de satisfação foi obtido observando-se o retorno demonstrado pelo cliente após realização dos testes de aceitação e o próprio resultado dos testes de aceitação pelos clientes desses projetos.

O projeto D teve seu pior índice em documentação de requisitos, 38% das solicitações relacionadas a requisitos foram configuradas por deficiências na documentação dos requisitos. Nesse caso houve um maior comprometimento da equipe de especialista com a equipe de software, pois os requisitos de domínio específico foram fornecidos pela equipe de especialistas. Porém não houve entendimento por parte da equipe de especialistas em relação à manutenção desses requisitos, uma vez que, foram fornecidos requisitos incompletos que resultou em muitas alterações de requisitos durante o desenvolvimento do software. Esse cenário também se configurou pela falta de treinamento da equipe de especialistas para que esses pudessem conhecer o ciclo de desenvolvimento de software. Esse projeto apresentou, de maneira geral, um bom nível de satisfação do cliente em relação ao software que lhe foi entregue comparando com suas necessidades e expectativas.

Em uma visão geral dos índices apresentados, concluímos que os projetos A e B apresentaram maior índice de problemas em documentação e extração de requisitos, respectivamente, elevando assim o índice geral de problemas gerados por requisitos. Devido ao fato do uso de especificação de requisitos inadequada o índice de controle de alterações também ficou comprometido. Esses fatores certamente colaboraram para o baixo nível de satisfação de seus clientes. Já os projetos C e D apresentaram maior índice de problemas em adequação de artefatos do projeto e documentação de requisitos, respectivamente, porém com um controle de alteração de requisitos mais rigoroso o índice de problemas apresentados por controle de alterações de requisitos foi reduzido e isso certamente colaborou para o bom nível de satisfação de seus clientes.

8.2.10 Passo 10 – Recomendações

Dessa análise realizada pode-se retirar alguns pontos importantes e que podem ser entendidos como recomendações que podem ser seguidas por projetos de software que tenham características semelhantes às características da amostra utilizada:

- Necessidade de adequar o tipo de ciclo de vida para projetos de software que tratam questões de domínio específico.
- Adequação de tempo e esforço necessários na fase de extração de requisitos.

- Utilização de técnicas eficientes para extração de requisitos de projetos de software que tratam questões de domínio específico.
- Adequação de tempo e esforço necessários na fase de documentação dos requisitos.
- Documentar requisitos de maneira completa e coerente e com bom nível de detalhe, principalmente requisitos referentes a questões de domínio específico.
- Promover treinamento para as outras equipes envolvidas na solução utilizada pelo software e que não seja a equipe de software.
- Promover o papel de facilitador para melhorar a comunicação e integração entre equipes de especialistas e equipes de software.
- Elaborar matriz de rastreabilidade de requisitos a fim de minimizar as deficiências na adequação e manutenção dos artefatos do software (outros documentos).
- Promover a melhoria da comunicação e integração entre equipes de testes e equipes de software a fim de obter-se documentação de teste mais completa e coerente com os requisitos do sistema de software.

8.3 Considerações Finais

Vimos neste capítulo a quantificação e qualificação das solicitações registradas para quatro projetos de software reais desenvolvidos em uma empresa. A partir desse levantamento e qualificação, vimos como as boas práticas de requisitos definidas nas normas e modelos de processo e qualidade de software podem ser aplicadas com o objetivo de reduzir problemas e conseqüentemente reduzir o nível de retrabalho que podem provocar aumento do custo do projeto e atrasos no cronograma resultando na insatisfação do cliente e na perda de resultado para a empresa que desenvolve software.

Visando a questão de satisfação do cliente e o sucesso de projetos de software, é notável a relevância do uso de boas práticas de requisitos.

Capítulo 9 Conclusão

Por meio da elaboração desta dissertação, vários aspectos envolvidos na área de requisitos foram discutidos, pesquisados e analisados. Assim, foi possível obter uma visão geral das boas práticas de requisitos descritas nas normas e modelos de qualidade de software selecionados e pesquisados para este trabalho. No eixo teórico foram correlacionadas as boas práticas de requisitos entre as normas e modelos selecionados. No eixo prático foram contabilizados e classificados problemas encontrados em projetos de software reais a fim de obter uma visão geral da relevância do uso das boas práticas de requisitos.

Com o entendimento e análise das pesquisas realizadas para desenvolver esta dissertação foi possível:

- Destacar alguns dos problemas que cercam a área de requisitos;
- Verificar na literatura artigos relacionados aos objetivos deste trabalho. Analisá-los e concluir que alguns cenários descritos nos artigos selecionados (Capítulo 3) poderiam ser desenhados de forma diferente se fossem utilizadas boas práticas de requisitos;
- Identificar nos modelos de processo e qualidade de software as boas práticas de requisitos que auxiliam no desenvolvimento de sistemas de software no intuito de minimizar esses problemas que cercam a área de requisitos;
- Correlacionar as boas práticas de requisitos encontradas nos modelos de processo e qualidade de software pesquisados;
- Analisar na prática a relevância do uso dessas boas práticas de requisitos em projetos de software reais;
- Destacar recomendações que possam auxiliar no desenvolvimento de sistemas de software de forma positiva e construtiva a fim de melhorar os aspectos envolvidos na obtenção dos resultados almejados.

9.1 Perspectivas futuras

Novos estudos podem ser desenvolvidos relacionados à área de engenharia de requisitos, pois ainda há grande necessidade de aprimorar o trabalho realizado nessa área. A partir desse trabalho outros trabalhos e pesquisas podem ser desenvolvidos nas áreas de engenharia de requisitos e garantia da qualidade, como se pode citar:

- Estudos relacionados à adaptação das práticas de requisitos para diferentes tipos de projetos de software com o objetivo de analisar, simplificar ou detalhar a implementação de determinadas práticas de requisitos de acordo com o tipo de projeto de software a ser desenvolvido.
- Estudos relacionados à medição de esforço realizado para implementação de correções de problemas relacionados a requisitos, pois permite avaliar o retrabalho possibilitando obter uma visão geral de quanto esforço é despendido por causa dos problemas em requisitos.
- Estudos relacionados à utilização de ferramentas que possibilitem classificação mais detalhada dos problemas de software relacionados a requisitos a fim de obter novos indicadores com o objetivo de verificar novos pontos de problemas e buscar ações corretivas para esses problemas.

A aplicação do assunto tratado neste trabalho visa também contribuir com as organizações envolvidas em desenvolvimento de sistemas de software a fim de que essas organizações possam obter sucesso em seus empreendimentos. Ao termo sucesso aqui citado se refere à produção de sistemas de software de alta qualidade, muito bem gerenciados com alto controle de custo, esforço e prazo, e que atendem as expectativas do cliente e aos seus requisitos preestabelecidos.

9.2 Conclusões finais

Em resumo, problemas relacionados a requisitos que podem ocorrer desde a fase inicial de um projeto de software até a sua conclusão trazendo consequências negativas e resultando em um projeto mal sucedido merecem atenção especial, pois podem custar muito as organizações.

A partir dessa visão, os seguintes pontos podem ser ressaltados:

- Dificuldades são quase sempre encontradas em relação à questão requisitos, tanto por parte do cliente como também do fornecedor (quem desenvolve sistemas de software) e também por parte de quem controla os requisitos de um projeto de software.
- Atividades básicas na área de requisitos não devem ser negligenciadas, como por exemplo, atividades de extração, documentação e controle de requisitos do cliente e requisitos do software.
- Modelos de processo de software devem ser considerados pelas organizações que desejam satisfazer seus clientes e obter sucesso no desenvolvimento de sistemas de software.
- Modelos de processo e de qualidade que auxiliam no desenvolvimento de sistemas de software a fim de reduzir prazo, esforço e custo de projetos de software e melhorar as atividades relacionadas a requisitos, devem ser bem especificados e colocados em prática.
- Boas práticas de requisitos devem ser utilizadas corretamente a fim de reduzir o número de problemas no software e assim diminuir o retrabalho.
- Medidas de qualidade de software podem auxiliar na visibilidade da qualidade mantida durante o desenvolvimento do software e também ajudar no bom planejamento de futuros projetos de software.

Assim, é fundamental ressaltar a importância da engenharia de requisitos e a aplicabilidade dos modelos de processos e de qualidade em relação à questão requisitos.

Capítulo 10 Referências

- [BAUME92] BAUMERT, John H. e McWHINNEY, Mark S. "Software Measures and the Capability Maturity Model", CMU/SEI-92-TR-25, 1982.
- [BOEHM89] BOEHM, B. W. "Software Risk Management", IEEE Computer Society Press, Washington, 1989.
- [BUREN98] BUREN, Jim Van e COOK, David A. "Experiences in the Adoption of Requirements Engineering Technologies", Software Technology Support Center, Crosstalk The Journal of Defense Software Engineering, 1998.
- [FERRE93] FERREIRA, A. B. H. "Dicionário Aurélio da língua portuguesa", Nova Fronteira, 1993.
- [FLORA92] FLORAC, William A. "Software Quality Measurement: A framework for Counting Problems and Defect", Technical Report, CMU/SEI-92-TR-022, 1992.
- [HERBS94] HERBSLEB, James e CARLETON, Anita e ROZUM, James e SIEGEL, Jane e ZUBROW, David - "Benefits of CMM-Based Software Process Improvement: Initial Results", Technical Report, CMU/SEI-94-TR-013 – 1994.
- [HUMP87a] W.S. Humphrey, Characterizing the Software Process: A Maturity Framework, Software Engineering Institute, CMU/SEI-87-TR-11, ADA182895, June 1987.

- [HUMP87b] W.S. Humphrey and W.L. Sweet, A Method for Assessing the Software Engineering Capability of Contractors, Software Engineering Institute, CMU/SEI-87-TR-23, ADA187320, September 1987.
- [HUMPH88] W.S. Humphrey, "Characterizing the Software Process," IEEE Software, Vol. 5, No. 2, March, 1988, pp. 73-79.
- [HUMPH89] W.S. Humphrey, Managing the Software Process, Addison-Wesley, Reading, MA, 1989.
- [IEEE83] IEEE Standard Glossary of Software Engineering Terminology. ANSI/IEEE Standard 729-1983. New York: Institute of Electrical and Electronic Engineers, Inc.
- [IEEE84] IEEE Std. 830 "IEEE Guide to Software Requirements Specifications", The Institute of Electrical and Electronics Engineers. New York, 1984.
- [IEEE90] IEEE Std. 610.12 "IEEE Standard Glossary of Software Engineering Terminology", The Institute of Electrical and Electronics Engineers. New York, 1990.
- [ISO12207] NBR ISO/IEC 12207 "Tecnologia de informação – Processos de ciclo de vida de software", ABNT – Associação Brasileira de Normas Técnicas, 1998.
- [ISO9000] NBR ISO 9000 "Sistemas de gestão da qualidade – Fundamentos e vocabulário", ABNT – Associação Brasileira de Normas Técnicas, 2000.
- [ISO9000-3] ISO 9000-3 "Diretrizes para aplicação da ISO 9001 ao desenvolvimento, fornecimento e manutenção de software", 2001.

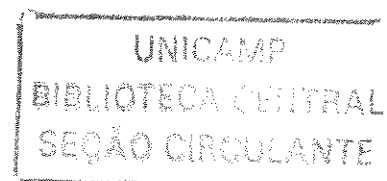
-
- [ISO9001] NBR ISO 9001:2000 "Sistemas de gestão da qualidade – Requisitos", ABNT – Associação Brasileira de Normas Técnicas, 2000.
- [ISO9004] NBR ISO 9004:2000 "Sistemas de gestão da qualidade – Diretrizes para melhoria de desempenho", ABNT – Associação Brasileira de Normas Técnicas, 2000.
- [ISO9126] NBR ISO/IEC 9126-1:2001 "Engenharia de software – Qualidade de produto – Parte 1: Modelo de Qualidade", ABNT – Associação Brasileira de Normas Técnicas, 2002.
- [PAULK91] M.C. Paulk, B. Curtis, M.B. Chrissis, et al, Capability Maturity Model for Software, Software Engineering Institute, CMU/SEI-91-TR-24, ADA240603, August 1991.
- [PMBOK00] PMBOK - Project Management Body of Knowledge, PMI – Project Management Institute, tradução para português de Antônio José Soares e Márcio Tibo, PMI MG, 2000.
- [PRESS95] PRESSMAN, ROGER S. "Engenharia de Software", MAKRON Books, 1995.
- [SEMAT95] SEMATECH - "Software Quality Engineering for Managers— Training Materials" - Technology Transfer # 92111389B-TRG, material de treinamento da SEMATECH, 31 de Maio de 1995.
- [SOMME03] SOMMERVILLE, Ian. "Engenharia de Software" – 6o.Edição, Addison Wesley, 2003.
- [STARK98] STARK, George – "An Examination of the Effects of Requirements Changes on Software Releases", Crosstalk, 1998.

- [THAYER97] Thayer, R. & Dorfman, M. Software Requirements Engineering, 2nd ed. Los Alamitos, CA: IEEE Computer Society Press, 1997 (ISBN 0-8186-7738-4).
- [TR02493] PAULK, Mark C. e CURTIS, Bill e CHRISSIS, Mary Beth e WEBER, Charles V. - CMU/SEI-93-TR-024 - "Capability Maturity Model for Software", version 1.1, Software Engineering Institute, Carnegie Mellon University, 1993.
- [TR02593] PAULK, Mark C. e WEBER, Charles V. e GARCIA, Suzanne M. e CHRISSIS, Mary Beth e BUSH, Marilyn - CMU/SEI-93-TR-025 - "Key Practices of the Capability Maturity Model for Software", version 1.1, Software Engineering Institute, Carnegie Mellon University, 1993.
- [TR02802] CMU/SEI-2002-TR-028 "Capability Maturity Model Integration - Continuous Representation" version 1.1, Software Engineering Institute, Carnegie Mellon University, 2002.
- [TR02902] CMU/SEI-2002-TR-029 "Capability Maturity Model Integration - Staged Representation" version 1.1, Software Engineering Institute, Carnegie Mellon University, 2002.
- [VINCE 04] VINCENZI, Auri Marcelo Rizzo - "Orientação a Objeto: Definição, Implementação e Análise de Recursos de Teste e Validação", Tese de Doutorado / USP - São Carlos, 2004.

10.1 Outras publicações

Abaixo são apresentadas publicações que foram pesquisadas e analisadas durante a elaboração do trabalho, mas não foram referenciadas no texto.

- [BOEHM81] BOEHM, B. W. "Software Engineering Economics", Prentice-Hall, 1981.
- [CARVA01] CARVALHO, Ariadne M. B. R. e CHIOSSI, Thelma C. S. "Introdução à Engenharia de Software", Editora da UNICAMP, 2001.
- [CASTR95] CASTRO, Jaelson F. B. "Introdução à Engenharia de Requisitos", Instituto de Computação da Universidade Federal do Rio Grande do Sul, 1995.
- [CORTE01] CÔRTEZ, Mario Lúcio e CHIOSSI, Thelma C. S. "Modelos de Qualidade de Software", Editora da UNICAMP, 2001.
- [HILBU99] HILBURN, Thomas B. - A Software Engineering Body of Knowledge, CMU/SEI-99-TR-004, 1999.
- [IEEE98] IEEE Std. 1233 "IEEE Guide for Developing System Requirements Specifications", The Institute of Electrical and Electronics Engineers, 1998.
- [MAYRH90] MAYRHAUSER, Annelise von e BASILI, Victor R. "Software Engineering Methods and Management", 1990.
- [SHULL00] SHULL, Forrest e RUS, Ioana e BASILI, Victor - "How Perspective-Based Reading Can Improve Requirements Inspections" - Fraunhofer Center for Experimental Software Engineering, Maryland, 2000.



- [SIMPO92] VI Simpósio Brasileiro de Engenharia de Software, Gramado - Rio Grande do Sul, 1992, "Apoio Automatizado à Elicitação de Requisitos: Proposta e Protótipo de Ferramenta".
- [SIMPO97] XI Simpósio Brasileiro de Engenharia de Software, Fortaleza - Ceará, 1997, "Seção Técnica 2 Engenharia de Requisitos".
- [SUBRA99] SUBRAMANIAM, Bala – "Effective Software Defect Tracking – Reducing Project Costs and Enhancing Quality", Crosstalk, 1999.
- [THAYE90] THAYER, Richard H. e DORFMAN, Merlin e YEH, Raymond T. "System and Software Requirements Engineering", IEEE Computer Society Press Tutorial, 1990.
- [VINCE98] VINCENZI, Auri Marcelo Rizzo - "Subsídios para o Estabelecimento de Estratégias de Teste Baseadas na Técnica de Mutação", Dissertação de Mestrado / USP - São Carlos, 1998.
- [YOUNG02] YOUNG, Dr. Ralph R. "Recommended Requirements Gathering Practices", Northrop Grumman Information Technology, Crosstalk, 2002.

10.2 Referências WEB

- [WEBSEI] <http://www.sei.cmu.edu/>
- [WEBISO] <http://www.iso.org/>
- [WEBIEEE] <http://www.ieee.org/>
- [WEBPMI] <http://www.pmi.org/>
- [WEBCTALK] <http://www.stsc.hill.af.mil/crosstalk>
- [WEBERPUC] <http://www.er.les.inf.puc-rio.br/>
"Engenharia de Requisitos".
- [WEBERIEEE] <http://www.shu.ac.uk/tfre/>
"IEEE Task Force on Requirements Engineering". (ver no ANEXO H: Bibliografia a lista de bibliografia recomendada por este site).
- [WEBGERPUC] <http://www.er.les.inf.puc-rio.br/GrupoER/>
"Grupo de Engenharia de Requisitos na PUC-Rio".

Capítulo 11 Anexos

11.1 ANEXO A: SW-CMM

A seguir são apresentadas as principais práticas de requisitos do modelo **SW-CMM** (*Capability Maturity Model*).

11.1.1 CMM Nível 2

A tabela a seguir é composta pelas colunas:

- Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.
- Coluna 2 - "Nível" : Identifica o nível de maturidade a qual a prática se refere.
- Coluna 3 - "KPA" : Identifica a KPA que contém tal prática.
- Coluna 4 - "Tipo" : Identifica o tipo da prática descrita.
- Coluna 5 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO			
	Nível	KPA	Tipo	Item
Os requisitos de sistema alocados ao software são controlados para estabelecer uma <i>baseline</i> para o desenvolvimento de software e para uso gerencial.	2	GR	Meta	Meta 1
Os planos, os produtos e as atividades de software são mantidos consistentes com os requisitos de sistema alocados ao software.	2	GR	Meta	Meta 2
O projeto segue uma política estabelecida pela organização para o gerenciamento de requisitos de sistema alocados ao software.	2	GR	Prática	Comprometimento 1
Para cada projeto é definida a responsabilidade pelos requisitos de sistema e alocação desses requisitos ao hardware, ao software e aos outros componentes do sistema.	2	GR	Prática	Habilidade 1
Os requisitos alocados são documentados.	2	GR	Prática	Habilidade 2
São providos os recursos e orçamento adequado para a atividade de gerência dos requisitos alocados.	2	GR	Prática	Habilidade 3
As equipes de engenharia de software e outros grupos relacionados são treinados para realizar suas atividades de gerência de requisitos.	2	GR	Prática	Habilidade 4
A equipe de engenharia de software revisa os requisitos alocados antes deles serem alocados ao projeto de software.	2	GR	Prática	Atividade 1
A equipe de engenharia de software deve utilizar os requisitos alocados como base para os planos, para os produtos e para as atividades de software.	2	GR	Prática	Atividade 2
As modificações nos requisitos alocados são revisadas e incorporadas ao projeto de software.	2	GR	Prática	Atividade 3

PRÁTICAS	LOCALIZAÇÃO			
	Nível	KPA	Tipo	Item
As medições são realizadas e utilizadas para determinar a situação das atividades de gerência dos requisitos alocados.	2	GR	Prática	Medição 1
As atividades de gerência de requisitos alocados são revisadas periodicamente por um gerente superior.	2	GR	Prática	Verificação 1
As atividades de gerência de requisitos alocados são revisadas pelo gerente de projeto, periodicamente ou motivo por um evento.	2	GR	Prática	Verificação 2
A equipe de garantia de qualidade de software executa auditoria das atividades e dos produtos de gerência de requisitos alocados e reporta os resultados.	2	GR	Prática	Verificação 3
Os requisitos alocados são documentados.	2	GR	Subprática	Comprometimento 1 Subprática 1
Os requisitos alocados são revisados.	2	GR	Subprática	Comprometimento 1 Subprática 2
Os planos, os produtos e as atividades de software são modificados para se tornarem consistentes com as mudanças nos requisitos alocados.	2	GR	Subprática	Comprometimento 1 Subprática 3
Durante a vida do projeto, há gerenciamento, documentação e alocação dos requisitos.	2	GR	Subprática	Habilidade 1 Subprática 1
Efetivação das alterações dos requisitos de sistema e a alocação desses requisitos.	2	GR	Subprática	Habilidade 1 Subprática 2
Os requisitos alocados incluem os requisitos não técnicos que afetam e determinam as atividades de projeto de software.	2	GR	Subprática	Habilidade 2 Subprática 1
Os requisitos alocados incluem os requisitos técnicos para o software.	2	GR	Subprática	Habilidade 2 Subprática 2
Os requisitos alocados incluem o critério de aceitação que será utilizado para verificar se os produtos de software satisfazem os requisitos alocados.	2	GR	Subprática	Habilidade 2 Subprática 3
São designadas pessoas que têm experiência e perícia no domínio da aplicação e em engenharia de software para gerenciar os requisitos alocados.	2	GR	Subprática	Habilidade 3 Subprática 1
São disponibilizadas ferramentas para suportar as atividades de gerência de requisitos	2	GR	Subprática	Habilidade 3 Subprática 2
São identificados os requisitos alocados que estão incompletos.	2	GR	Subprática	Atividade 1 Subprática 1
Os requisitos alocados são revisados para verificar se são factíveis, claros, consistentes e testáveis.	2	GR	Subprática	Atividade 1 Subprática 2
Requisitos que apresentarem problemas são revisados e são realizadas as alterações necessárias.	2	GR	Subprática	Atividade 1 Subprática 3
O comprometimento resultante dos requisitos alocados é negociado com os grupos afetados.	2	GR	Subprática	Atividade 1 Subprática 4
Os requisitos alocados são gerenciados e controlados.	2	GR	Subprática	Atividade 2 Subprática 1
Os requisitos alocados são a base para o plano de desenvolvimento de software.	2	GR	Subprática	Atividade 2 Subprática 2
Os requisitos alocados são a base para o desenvolvimento dos requisitos de software.	2	GR	Subprática	Atividade 2 Subprática 3
O impacto nos compromissos existentes é avaliado e as alterações são negociadas apropriadamente.	2	GR	Subprática	Atividade 3 Subprática 1

PRÁTICAS	LOCALIZAÇÃO			
	Nível	KPA	Tipo	Item
As alterações necessárias nos planos, nos produtos e nas atividades de software resultantes das modificações dos requisitos alocados são: identificadas, avaliadas, documentadas, planejadas, comunicadas aos grupos afetados, acompanhadas até a conclusão.	2	GR	Subprática	Atividade 3 Subprática 2
A auditoria deve verificar se os requisitos alocados foram revisados e os problemas foram resolvidos antes do comprometimento da equipe de engenharia de software.	2	GR	Subprática	Verificação 3 Subprática 1
A auditoria deve verificar se os planos, produtos e as atividades de software foram apropriadamente revisados quando os requisitos alocados sofreram alterações.	2	GR	Subprática	Verificação 3 Subprática 2
A auditoria deve verificar se as alterações em compromissos, resultantes de modificações nos requisitos alocados, foram negociadas com os grupos afetados.	2	GR	Subprática	Verificação 3 Subprática 3
Os requisitos de sistema alocados ao software são usados como base para o planejamento de projeto de software.	2	PPS	Subprática	Comprometimento 2 Subprática 1
O plano de desenvolvimento de software é baseado nos requisitos alocados ao software e é conforme com estes requisitos.	2	PPS	Subprática	Atividade 6 Subprática 1
A situação e os resultados das atividades de planejamento de projeto de software são revisados e devem estar de acordo com os requisitos alocados.	2	PPS	Subprática	Verificação 2 Subprática 2
Os produtos e atividades de software para serem subcontratados, são selecionados baseado na avaliação das características técnicas e não técnicas do projeto. As atividades e produtos de especificação de software para serem subcontratados, são determinados baseado em análise sistemática e na divisão apropriada dos requisitos de sistema e de software.	2	GSS	Subprática	Atividade 1 Subprática 1
A especificação do trabalho a ser subcontratado e os padrões e procedimentos a serem seguidos, são derivados do projeto: requisitos do sistema alocados ao software, requisitos de software.	2	GSS	Subprática	Atividade 1 Subprática 2
Os requisitos do produto devem ser desenvolvidos.	2	GSS	Subprática	Atividade 3 Subprática 3
Verificar que a interpretação e a implementação dos requisitos técnicos pelos subcontratados estão conforme os requisitos contratuais.	2	GSS	Subprática	Atividade 8 Subprática 3
Aderência dos produtos e atividades de software aplicáveis a padrões, procedimentos e requisitos é verificada objetivamente.	2	GQS	Meta	Meta 2
Os produtos de trabalho de software são avaliados contra os padrões, procedimentos de software e requisitos contratuais.	2	GQS	Subprática	Atividade 5 Subprática 2
Os itens de configuração são selecionados baseados nos critérios documentados. Como exemplo os requisitos de software podem ser identificados como itens de configuração.	2	GCS	Subprática	Atividade 4 Subprática 1

LEGENDA**KPA**

Meta
Comprometimento
Habilidade
Atividade
Medição
Verificação

GR = Gestão de Requisitos
PPS = Planejamento de Projeto de Software
APS = Acompanhamento de Projeto de Software
GSS = Gestão de Subcontratação de Software
GQS = Garantia da Qualidade de Software
GCS = Gestão de Configuração de Software

Key Process Area

Goal
Commitment
Ability
Activity
Measurement
Verification

Requirement Management
Software Project Planning
Software Project Tracking and Oversight
Software Subcontract Management
Software Quality Assurance
Software Configuration Management

11.1.2 CMM Nível 3

A tabela a seguir é composta pelas colunas:

- Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.
 Coluna 2 - "Nível" : Identifica o nível de maturidade a qual a prática se refere.
 Coluna 3 - "KPA" : Identifica a KPA que contém tal prática.
 Coluna 4 - "Tipo" : Identifica o tipo da prática descrita.
 Coluna 5 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO			
	Nível	KPA	Tipo	Item
A equipe técnica de engenharia de requisitos recebe treinamento para desempenhar tarefas, como elicitar requisitos e analisar requisitos.	3	EPS	Prática	Habilidade 2
Os requisitos de software são desenvolvidos, gerenciados, documentados e verificados sistematicamente de acordo com o processo de software definido.	3	EPS	Prática	Atividade 2
O projeto do software é desenvolvido, mantido, documentado e verificado de acordo com o processo definido para atender aos requisitos de software e ao modelo de codificação.	3	EPS	Prática	Atividade 3
O código do software é desenvolvido, mantido, documentado e verificado de acordo com o processo definido para implementar os requisitos de software.	3	EPS	Prática	Atividade 4
Testes de sistema e teste de aceite do software são executados para demonstrar que o software satisfaz os requisitos.	3	EPS	Prática	Atividade 7
Consistência é mantida nos produtos de trabalho do software, incluindo planos, definição de processo, requisitos alocados, requisitos de software, projeto do software, código, planos e procedimentos de teste.	3	EPS	Prática	Atividade 10
Os planos, tarefas e produtos são rastreáveis para os requisitos de sistema alocados ao software.	3	EPS	Subprática	Comprometimento 1 Subprática 3
As habilidades individuais são disponibilizadas para executar diferentes tarefas em engenharia de software, inclusive análise de requisitos de software.	3	EPS	Subprática	Habilidade 1 Subprática 1
As pessoas envolvidas no desenvolvimento dos requisitos de software revisam os requisitos alocados.	3	EPS	Subprática	Atividade 2 Subprática 1
Os métodos efetivos de análise de requisitos são usados para identificar e derivar os requisitos de software.	3	EPS	Subprática	Atividade 2 Subprática 2
O resultado da análise de requisitos de software é documentado.	3	EPS	Subprática	Atividade 2 Subprática 3
Os requisitos de software são analisados para assegurar que estão completos, são implementáveis e testáveis.	3	EPS	Subprática	Atividade 2 Subprática 4
Os requisitos de software são documentados.	3	EPS	Subprática	Atividade 2 Subprática 5

PRÁTICAS	LOCALIZAÇÃO			
	Nível	KPA	Tipo	Item
O grupo responsável pelo teste de aceite e teste de sistema verifica se os requisitos de software são testáveis.	3	EPS	Subprática	Atividade 2 Subprática 6
Métodos para verificação e validação de cada requisitos de software são identificados e documentados	3	EPS	Subprática	Atividade 2 Subprática 7
O documento de requisitos de software deve passar por revisão par-a-par antes de ser considerado completo.	3	EPS	Subprática	Atividade 2 Subprática 8
O documento de requisitos de software é revisado e aprovado.	3	EPS	Subprática	Atividade 2 Subprática 9
O documento de requisitos de software é revisado pelo cliente e usuários finais.	3	EPS	Subprática	Atividade 2 Subprática 10
O documento de requisitos de software é colocado sob gestão de configuração.	3	EPS	Subprática	Atividade 2 Subprática 11
Os requisitos de software são modificados, quando um requisito alocado sofre alteração.	3	EPS	Subprática	Atividade 2 Subprática 12
As pessoas envolvidas no projeto de software revisam os requisitos de software.	3	EPS	Subprática	Atividade 3 Subprática 2
O documento de projeto de software é modificado quando um requisito de software é alterado.	3	EPS	Subprática	Atividade 3 Subprática 11
As pessoas envolvidas na codificação revisam os requisitos de software.	3	EPS	Subprática	Atividade 4 Subprática 1
O código é modificado quando um requisito de software é alterado.	3	EPS	Subprática	Atividade 4 Subprática 6
Os planos, procedimentos de teste e casos de teste são modificados quando um requisito de software ou requisito alocado é alterado.	3	EPS	Subprática	Atividade 5 Subprática 8
Os casos de teste e procedimento de teste são revisados pelas pessoas responsáveis pelos requisitos de software.	3	EPS	Subprática	Atividade 6 Subprática 2
O teste de integração de software é executado e verificado com o documento de requisitos de software.	3	EPS	Subprática	Atividade 6 Subprática 3
Teste do software é executado a partir da baseline do software e dos documentos de requisitos de software e requisitos alocados.	3	EPS	Subprática	Atividade 7 Subprática 5
O resultado do teste é documento e usado como base para determinar se o software satisfaz os requisitos.	3	EPS	Subprática	Atividade 7 Subprática 7
Os requisitos de software, projeto, código e casos de teste são traçados nas atividades de engenharia de requisitos.	3	EPS	Subprática	Atividade 10 Subprática 2
A documentação de rastreabilidade dos requisitos alocados através dos requisitos de software, projeto, código e teste são gerenciados e controlados.	3	EPS	Subprática	Atividade 10 Subprática 3
A auditoria deve verificar que os requisitos de software são revisados.	3	EPS	Subprática	Verificação 3 Subprática 1
A auditoria deve verificar se os produtos de software estão de acordo com os padrões e os requisitos especificados.	3	EPS	Subprática	Verificação 3 Subprática 3
A auditoria deve verificar se é executada a rastreabilidade dos requisitos alocados através dos requisitos de software, projeto, código e teste.	3	EPS	Subprática	Verificação 3 Subprática 9

PRÁTICAS	LOCALIZAÇÃO			
	Nível	KPA	Tipo	Item
A auditoria deve verificar se a documentação usada para operar e manter o software é obtida na baseline do software e aplicável aos requisitos alocados antes da liberação do produto de software para o cliente.	3	EPS	Subprática	Verificação 3 Subprática 10
Reuso de requisitos, projeto, código, planos de teste e procedimentos de teste, etc.	3	GIS	Subprática	Atividade 6 Subprática 3
Os valores dos parâmetros dos modelos usados para estimar esforço e os custos do software são atualizados sempre que as principais alterações são feitas nos requisitos do software.	3	GIS	Subprática	Atividade 7 Subprática 4
O planejamento dos recursos computacionais, requisitos de sistemas alocados ao software, requisitos de software, e projeto de software são ajustados para alcançar os requisitos computacionais críticos do projeto.	3	GIS	Subprática	Atividade 8 Subprática 2
Os requisitos do cliente são acordados com todos os grupos afetados.	3	CI	Meta	Meta 1
Os requisitos de sistema para o projeto são definidos e revisados pelos grupos afetados.	3	CI	Subprática	Comprometimento 1 Subprática 1
O grupo de engenharia de software com o cliente e usuários finais, estabelecem os requisitos do sistema.	3	CI	Prática	Atividade 1
Definir os requisitos do cliente com características críticas.	3	CI	Subprática	Atividade 1 Subprática 1
Coordenar atividades de revisão técnica e aprovação dos requisitos do sistema. Prover revisão técnica e análise necessária para gerenciar e controlar alterações nos requisitos do sistema.	3	CI	Subprática	Atividade 2 Subprática 1
O grupo técnico soluciona conflitos e esclarece requisitos do sistema.	3	CI	Subprática	Atividade 2 Subprática 2
Assegurar que a interpretação e implementação dos requisitos técnicos estão de acordo com os requisitos do sistema.	3	CI	Subprática	Atividade 7 Subprática 3

LEGENDA

KPA

Key Process Area

Meta

Goal

Comprometimento

Commitment

Habilidade

Ability

Atividade

Activity

Medição

Measurement

Verificação

Verification

FSPO = Foco Sobre os Processos da Organização

Organization Process Focus

DPO = Definição do Processo da Organização

Organization Process Definition

PT = Programa de Treinamento

Training Program

GIS = Gestão Integrada de Software

Integrated Software Management

EPS = Engenharia de Produto de Software

Software Product Engineering

CI = Coordenação Intergrupos

Intergroup Coordination

RP = Revisão por Pares

Peer Reviews

11.1.3 CMM - Outros Níveis

A tabela a seguir é composta pelas colunas:

- Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.
 Coluna 2 - "Nível" : Identifica o nível de maturidade a qual a prática se refere.
 Coluna 3 - "KPA" : Identifica a KPA que contém tal prática.
 Coluna 4 - "Tipo" : Identifica o tipo da prática descrita.
 Coluna 5 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO			
	Nível	KPA	Tipo	Item
Para a qualidade do software é necessário que as necessidades da organização e dos usuários sejam rastreáveis para os requisitos de sistema alocados ao software e as metas de qualidade de software.	4	GQS	Subprática	Atividade 1 Subprática 2
O plano de qualidade do software é atualizado no início do projeto e quando os requisitos alocados são alterados significativamente.	4	GQS	Subprática	Atividade 1 Subprática 6

LEGENDA

KPA

Key Process Area

Meta

Goal

Comprometimento

Commitment

Habilidade

Ability

Atividade

Activity

Medição

Measurement

Verificação

Verification

GQS = Gestão da Qualidade de Software

Software Quality Management

11.2 ANEXO B: CMMI

A seguir são apresentadas as principais práticas de requisitos do modelo **CMMI** (*Capability Maturity Model Integration*).

11.2.1 CMMI Nível 2

A tabela a seguir é composta pelas colunas:

- Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.
- Coluna 2 - "Nível" : Identifica o nível de maturidade a qual a prática se refere.
- Coluna 3 - "PA" : Identifica a PA que contém tal prática.
- Coluna 4 - "Tipo" : Identifica o tipo da prática descrita.
- Coluna 5 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO			
	Nível	PA	Tipo	Item
Os requisitos são gerenciados e inconsistências com os planos do projeto e produtos de trabalho são identificados.	2	GR	Meta	ME 1
Obter o entendimento dos requisitos.	2	GR	Prática	PE 1
Obter o comprometimento dos participantes do projeto com os requisitos.	2	GR	Prática	PE 2
Gerenciar mudanças nos requisitos.	2	GR	Prática	PE 3
Manter rastreabilidade bidirecional dos requisitos.	2	GR	Prática	PE 4
Identificar inconsistências entre os planos do projeto, produtos de trabalho e requisitos.	2	GR	Prática	PE 5
Estabelecer critérios para diferenciar apropriadamente os fornecedores de requisitos.	2	GR	Subprática	PE 1 S1
Estabelecer critérios objetivos para aceitação dos requisitos.	2	GR	Subprática	PE 1 S2
Analisar os requisitos para assegurar que os critérios estabelecidos estão corretos.	2	GR	Subprática	PE 1 S3
Buscar o entendimento dos requisitos com seus fornecedores para que os participantes do projeto possam comprometer-se com eles.	2	GR	Subprática	PE 1 S4
Avaliar o impacto dos requisitos nos compromissos existentes.	2	GR	Subprática	PE 2 S1
Negociar e registrar os compromissos.	2	GR	Subprática	PE 2 S2
Capturar todos os requisitos e mudanças de requisitos que são gerados pelo projeto.	2	GR	Subprática	PE 3 S1
Manter o histórico de mudanças de requisitos e as diferenças entre essas mudanças.	2	GR	Subprática	PE 3 S2
Avaliar o impacto das mudanças de requisitos da mesma maneira que os grupos afetados mais relevantes avaliariam.	2	GR	Subprática	PE 3 S3
Elaborar os requisitos e alterar os dados do projeto.	2	GR	Subprática	PE 3 S4

PRÁTICAS	LOCALIZAÇÃO			
	Nível	PA	Tipo	Item
Manter rastreabilidade de requisitos para assegurar que os requisitos de mais baixo nível estejam documentados.	2	GR	Subprática	PE 4 S1
Manter rastreabilidade de requisitos, de um requisito para seus derivados.	2	GR	Subprática	PE 4 S2
Manter rastreabilidade horizontal entre funções e interfaces.	2	GR	Subprática	PE 4 S3
Gerar a matriz de rastreabilidade de requisitos.	2	GR	Subprática	PE 4 S4
Rever os planos do projeto, atividades e produtos de trabalho afim de mantê-los consistentes com os requisitos e as mudanças que eles sofreram.	2	GR	Subprática	PE 5 S1
Identificar as fontes de inconsistências.	2	GR	Subprática	PE 5 S2
Identificar as mudanças que precisam ser feitas nos planos do projeto e produtos de trabalho a partir de uma <i>baseline</i> de requisitos.	2	GR	Subprática	PE 5 S3
Dar início as ações corretivas.	2	GR	Subprática	PE 5 S4
Fatores que tipicamente são considerados quando se estimam parâmetros de planejamento, são: requisitos de projeto, incluindo requisitos de produto, requisitos impostos pela organização, requisitos impostos pelo cliente e outros requisitos que impactam o projeto.	2	PP	Prática	ME 1
As estimativas devem ser consistentes com os requisitos do projeto, para determinar esforço, custo e cronograma.	2	PP	Prática	ME 1 PE 2
O ciclo de vida de projeto consiste em fases que precisam ser definidas dependendo do escopo dos requisitos, das estimativas para recursos de projeto, e a natureza do projeto.	2	PP	Prática	ME 1 PE 3
Estimar esforço e custo que usando históricos. Requisitos de produto e componente de produto são usados para calcular esforço e custo.	2	PP	Subprática	ME 1 PE 4 S3
Um plano de projeto é um documento formal e aprovado, usado para gerenciar e controlar a execução do projeto. Está baseado nos requisitos de projeto e nas estimativas estabelecidas.	2	PP	Meta	ME 2
Selecionar fornecedores baseado em uma avaliação de suas habilidades para atender os requisitos especificados e critérios estabelecidos.	2	GCF	Prática	ME 1 PE 2
Identificar o potencial dos fornecedores e distribuir as solicitações e requisitos para eles.	2	GCF	Subprática	ME 1 PE 2 S2
Revise os requisitos a serem cumpridos pelo fornecedor para refletir negociações com o fornecedor quando necessário.	2	GCF	Subprática	ME 1 PE 3 S1
Assegure que todas as partes do contrato estão entendidas e acordadas para todos os requisitos antes de implementar o contrato.	2	GCF	Subprática	ME 1 PE 3 S4
Revisar o COTS do produto para assegurar que eles satisfazem os requisitos especificados e que são cobertos no contrato do fornecedor.	2	GCF	Prática	ME 2 PE 1

PRÁTICAS	LOCALIZAÇÃO			
	Nível	PA	Tipo	Item
Avaliar o COTS do produto contra os requisitos e critérios associados.	2	GCF	Subprática	ME 2 PE 1 S2
Verificar que os produtos adquiridos satisfazem os requisitos.	2	GCF	Subprática	ME 2 PE 3 S3
Itens que devem estar sob gestão de configuração incluem especificações e documentos de interface que definem requisitos do produto.	2	GC	Prática	ME 1 PE 1

LEGENDA

PA

Process Area

ME = Meta Específica

Specific Goal

PE = Prática Específica

Specific Practice

S = Subprática

Sub practices

GR = Gestão de Requisitos

Requirements Management

PP = Planejamento de Projeto

Project Planning

MPC = Monitoramento de Projeto e Controle

Project Monitoring and Control

GCF = Gestão de Contrato de Fornecedor

Supplier Agreement Management

MA = Medida e Análise

Measurement and Analysis

GQPP = Garantia de Qualidade de Processo e Produto

Process and Product Quality Assurance

GC = Gestão de Configuração

Configuration Management

11.2.2 CMMI Nível 3

A tabela a seguir é composta pelas colunas:

- Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.
 Coluna 2 - "Nível" : Identifica o nível de maturidade a qual a prática se refere.
 Coluna 3 - "PA" : Identifica a PA que contém tal prática.
 Coluna 4 - "Tipo" : Identifica o tipo da prática descrita.
 Coluna 5 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO			
	Nível	PA	Tipo	Item
Desenvolver requisitos do cliente.	3	DR	Meta	ME 1
Desenvolver requisitos do produto.	3	DR	Meta	ME 2
Analisar e validar requisitos.	3	DR	Meta	ME 3
Fazer o levantamento das necessidades (elicitar requisitos).	3	DR	Prática	ME 1 PE 1
Desenvolver os requisitos do cliente.	3	DR	Prática	ME 1 PE 2
Estabelecer requisitos de produto e de componente de produto.	3	DR	Prática	ME 2 PE 1
Alocar requisitos de componente de produto.	3	DR	Prática	ME 2 PE 2
Identificar requisitos de interface.	3	DR	Prática	ME 2 PE 3
Estabelecer conceitos e cenários operacionais.	3	DR	Prática	ME 3 PE 1
Estabelecer uma definição de uma funcionalidade requisitada.	3	DR	Prática	ME 3 PE 2
Analisar os requisitos.	3	DR	Prática	ME 3 PE 3
Analisar os requisitos para balanceamento.	3	DR	Prática	ME 3 PE 4
Validar requisitos com métodos compreensivos.	3	DR	Prática	ME 3 PE 5
Envolver os grupos afetados usando métodos para levantamento de necessidades e expectativas (elicitar requisitos).	3	DR	Subprática	ME 1 PE 1 S1
Elaborar o documento de requisitos do cliente utilizando o levantamento das necessidades e expectativas dos envolvidos.	3	DR	Subprática	ME 1 PE 2 S1
Definir regras para verificação e validação.	3	DR	Subprática	ME 1 PE 2 S2
Desenvolver os requisitos em termos técnicos para atender as necessidades de projeto de componente de produto e produto.	3	DR	Subprática	ME 2 PE 1 S1
Derivar requisitos que resultem de decisões de projeto.	3	DR	Subprática	ME 2 PE 1 S2
Estabelecer e manter relacionamentos entre requisitos e alocação de requisitos.	3	DR	Subprática	ME 2 PE 1 S3
Alocar requisitos às funcionalidades.	3	DR	Subprática	ME 2 PE 2 S1

PRÁTICAS	LOCALIZAÇÃO			
	Nível	PA	Tipo	Item
Alocar requisitos para componentes de produto.	3	DR	Subprática	ME 2 PE 2 S2
Alocar regras de projeto para componentes de produto.	3	DR	Subprática	ME 2 PE 2 S3
Documentar relacionamentos entre os requisitos alocados.	3	DR	Subprática	ME 2 PE 2 S4
Identificar interfaces externas e internas ao produto.	3	DR	Subprática	ME 2 PE 3 S1
Desenvolver requisitos para identificar interfaces.	3	DR	Subprática	ME 2 PE 3 S2
Desenvolver conceitos e cenários que incluam funcionalidade, desempenho, manutenibilidade e suporte.	3	DR	Subprática	ME 3 PE 1 S1
Definir o ambiente em que o produto irá operar.	3	DR	Subprática	ME 3 PE 1 S2
Rever os conceitos e cenários para refinar os requisitos.	3	DR	Subprática	ME 3 PE 1 S3
Desenvolver conceitos detalhados para definir a interação entre o usuário final e o produto e que satisfaça as necessidades operacionais, de manutenção e suporte.	3	DR	Subprática	ME 3 PE 1 S4
Analisar e quantificar usuários finais e requisições de funcionalidades.	3	DR	Subprática	ME 3 PE 2 S1
Analisar requisitos para identificar partições funcionais e lógicas.	3	DR	Subprática	ME 3 PE 2 S2
Dividir os requisitos em grupos, baseado nos critérios estabelecidos para facilitar a análise de requisitos.	3	DR	Subprática	ME 3 PE 2 S3
Considerar a seqüência de funções críticas.	3	DR	Subprática	ME 3 PE 2 S4
Alocar requisitos do cliente.	3	DR	Subprática	ME 3 PE 2 S5
Alocar requisitos funcionais e requisitos de desempenho.	3	DR	Subprática	ME 3 PE 2 S6
Analisar as necessidades e expectativas do envolvidos para exterminar conflitos e organizar dentro dos assuntos relacionados.	3	DR	Subprática	ME 3 PE 3 S1
Analisar os requisitos para avaliar se satisfazem os requisitos de mais alto nível.	3	DR	Subprática	ME 3 PE 3 S2
Analisar os requisitos para assegurar que são completos, viáveis e verificáveis.	3	DR	Subprática	ME 3 PE 3 S3
Identificar requisitos chaves que tenham influências em custos, cronograma, riscos ou desempenho.	3	DR	Subprática	ME 3 PE 3 S4
Identificar medidas de desempenho técnico.	3	DR	Subprática	ME 3 PE 3 S5
Analisar os conceitos e cenários para refinar as necessidades do cliente e descobrir novos requisitos.	3	DR	Subprática	ME 3 PE 3 S6
Usar modelos, protótipos e simulações para analisar as necessidades dos grupos afetados e as regras.	3	DR	Subprática	ME 3 PE 4 S1
Executar a avaliação de riscos nos requisitos e na arquitetura funcional.	3	DR	Subprática	ME 3 PE 4 S2
Examinar o ciclo de vida do produto.	3	DR	Subprática	ME 3 PE 4 S3

PRÁTICAS	LOCALIZAÇÃO			
	Nível	PA	Tipo	Item
Analisar os requisitos para determinar o risco resultante do produto não ser executado no ambiente apropriado.	3	DR	Subprática	ME 3 PE 5 S1
Explorar a adequação e a completude dos requisitos.	3	DR	Subprática	ME 3 PE 5 S2
Avaliar o projeto dentro do contexto de validação de requisitos.	3	DR	Subprática	ME 3 PE 5 S3
Obter uma alocação completa dos requisitos para cada alternativa.	3	ST	Subprática	ME 1 PE 1 S4
Documentar relacionamentos entre requisitos e componentes de produto.	3	ST	Subprática	ME 1 PE 3 S2
Identificar e solucionar itens com soluções alternativas e requisitos.	3	ST	Subprática	ME 1 PE 3 S3
Estabelecer os requisitos associados com um conjunto de alternativas selecionado como um conjunto de requisitos alocados para componentes de produto.	3	ST	Subprática	ME 1 PE 3 S5
Assegurar que o projeto é aderente aos requisitos alocados.	3	ST	Subprática	ME 2 PE 1 S4
Base detalhada de descrições nos requisitos alocados aos componentes de produto, arquitetura e projeto de alto nível.	3	ST	Subprática	ME 2 PE 2 S2
Revisar requisitos, projeto, produto, e teste para assegurar que assuntos que afetam a instalação, operação, manutenção e documentação são identificados e solucionados.	3	ST	Subprática	ME 3 PE 2 S1
Identificar requisitos para o ambiente de integração do produto.	3	IP	Subprática	ME 1 PE 2 S1
Revisar requisitos, projeto, produto, resultados da verificação, e documentação para assegurar que itens que afetam o empacotando e entrega do produto são identificados e solucionados.	3	IP	Subprática	ME 3 PE 4 S1
Satisfazer os requisitos aplicáveis e padrões para empacotar e entregar o produto.	3	IP	Subprática	ME 3 PE 4 S3
Identificar requisitos para satisfazer cada produto de trabalho selecionado.	3	VER	Subprática	ME 1 PE 1 S2
Submeter a integração com o plano de projeto, a identificação de produtos de trabalho serem verificados, os requisitos devem ser satisfeitos, e os métodos serem usados.	3	VER	Subprática	ME 1 PE 1 S5
Identifique requisitos no ambiente de verificação.	3	VER	Subprática	ME 1 PE 2 S1
Identificar os resultados esperados e outros critérios por satisfazer os requisitos.	3	VER	Subprática	ME 1 PE 3 S3
Definir requisitos para coletar dados durante a revisão por pares.	3	VER	Subprática	ME 2 PE 1 S2
São verificados produtos de trabalho selecionados contra seus requisitos especificados.	3	VER	Meta	ME 3
Executar verificação de produtos de trabalho selecionados contra seus requisitos.	3	VER	Meta	ME 3 PE 1 S1
Baseado nos critérios de verificação estabelecidos, identifique produtos que não satisfazem os requisitos.	3	VER	Meta	ME 3 PE 2 S2

PRÁTICAS	LOCALIZAÇÃO			
	Nível	PA	Tipo	Item
Identificar requisitos de ambiente de validação.	3	VAL	Subprática	ME 1 PE 2 S1
Revisar os requisitos de produto para assegurar que itens que afetam a validação do produto ou componente de produto são identificados e solucionados.	3	VAL	Subprática	ME 1 PE 3 S1
Revisar requisitos, planos e compromissos.	3	GIP	Subprática	ME 1 PE 4 S3
Periodicamente revisar e alinhar o desempenho do projeto com as necessidades atuais e antecipadas, objetivos, e requisitos da organização, cliente, e usuários finais, se apropriado.	3	GIP	Subprática	ME 1 PE 4 S5
Assegurar que produtos de trabalho que são produzidos para satisfazer compromissos satisfazem os requisitos do projeto.	3	GIP	Subprática	ME 2 PE 1 S2
Desenvolver recomendações e coordenar as ações para solucionar problemas com os requisitos do produto e componente de produto, arquitetura do produto e componente de produto, e projeto do produto e componente de produto.	3	GIP	Subprática	ME 2 PE 1 S3

LEGENDA

PA

Process Area

ME = Meta Específica

Specific Goal

PE = Prática Específica

Specific Practice

S = Subprática

Sub practices

DR = Desenvolvimento de Requisitos

Requirements Development

ST = Solução Técnica

Technical Solution

IP = Integração de Produto

Product Integration

VER = Verificação

Verification

VAL = Validação

Validation

FSPO = Foco Sobre os Processos da Organização

Organizational Process Focus

DPO = Definição do Processo da Organização

Organizational Process Definition

TO = Treinamento Organizacional

Organizational Training

GIP = Gestão Integrada de Projeto

Integrated Project Management

GRi = Gestão de Riscos

Risk Management

ADR = Análise de Decisão e Resolução

Decision Analysis and Resolution

11.3 ANEXO C: Normas da família ISO 9000

A seguir são apresentadas as principais práticas de requisitos das normas da família ISO 9000.

11.3.1 ISO 9000

Vide terminologias descritas no Capítulo 7.

11.3.2 ISO 9000-3

A tabela a seguir é composta pelas colunas:

Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.

Coluna 2 - "Área" : Identifica a área da norma onde se encontra tal prática.

Coluna 3 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
A organização deve determinar os requisitos especificados pelo cliente.	Processos relacionados a clientes (Determinação de requisitos relacionados ao produto)	7.2.1
A organização deve determinar os requisitos não declarados pelo cliente, mas que se fazem necessários.	Processos relacionados a clientes (Determinação de requisitos relacionados ao produto)	7.2.1
A organização deve determinar os requisitos estatutários e regulamentares relacionados ao produto.	Processos relacionados a clientes (Determinação de requisitos relacionados ao produto)	7.2.1
A organização deve determinar qualquer requisito adicional.	Processos relacionados a clientes (Determinação de requisitos relacionados ao produto)	7.2.1
Os requisitos do produto estão definidos.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
Os requisitos de contrato que difiram daqueles previamente manifestados estão resolvidos.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
A organização tem a capacidade de atender aos requisitos definidos.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
Devem ser mantidos registros dos resultados da análise crítica dos requisitos.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
Quando o cliente não fornecer uma declaração documentada dos requisitos, a organização deve confirmar os requisitos do cliente antes da aceitação.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Quando os requisitos de produto forem alterados, a organização deve assegurar que os documentos pertinentes são complementados e que o pessoal envolvido é alertado sobre os requisitos alterados.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
Entradas relativas a requisitos de produto devem ser determinadas e registros devem ser mantidos e devem incluir: requisitos de funcionamento e de desempenho, requisitos estatutários e regulamentares aplicáveis e outros requisitos essenciais para o projeto e desenvolvimento.	Projeto e Desenvolvimento (Entradas do Projeto e Desenvolvimento)	7.3.2
Requisitos devem ser completos, sem ambigüidades e não conflitantes entre si.	Projeto e Desenvolvimento (Entradas de Projeto e Desenvolvimento)	7.3.2
As saídas de projeto e desenvolvimento devem atender aos requisitos de entrada para o projeto e desenvolvimento e devem conter ou referenciar critérios de aceitação do produto.	Projeto e Desenvolvimento (Saídas de Projeto e Desenvolvimento)	7.3.3
Devem ser realizadas, em fases apropriadas, análises críticas sistemáticas de projeto e desenvolvimento, de acordo com as disposições planejadas para avaliar a capacidade dos resultados do projeto e desenvolvimento em atender aos requisitos.	Projeto e Desenvolvimento (Análise crítica de Projeto e Desenvolvimento)	7.3.4
A verificação deve ser executada conforme disposições planejadas, para assegurar que as saídas do projeto e desenvolvimento estejam atendendo aos requisitos de entrada do projeto e desenvolvimento.	Projeto e Desenvolvimento (Verificação de Projeto e Desenvolvimento)	7.3.5
A validação do projeto e desenvolvimento deve ser executada conforme disposições planejadas, para assegurar que o produto resultante é capaz de atender os requisitos para a aplicação especificada.	Projeto e Desenvolvimento (Validação de Projeto e Desenvolvimento)	7.3.6
A organização deve assegurar que o produto adquirido está conforme com os requisitos especificados de aquisição.	Aquisição (Processo de Aquisição)	7.4.1
As informações de aquisição devem descrever o produto a ser adquirido e incluir, onde apropriado, requisitos para aprovação de produto, procedimentos, processos e equipamento, requisitos para qualificação de pessoal e requisitos para sistema de gestão de qualidade.	Aquisição (Informações de Aquisição)	7.4.2
A organização deve estabelecer e implementar inspeção ou outras atividades necessárias para assegurar que o produto adquirido atende aos requisitos de aquisição especificados.	Aquisição (Verificação do Produto Adquirido)	7.4.3
A organização deve identificar a situação do produto no que se refere aos requisitos de monitoramento e de medição	Produção e Fornecimento de Serviço (Identificação e rastreabilidade)	7.5.3
Quando a rastreabilidade é um requisito, a organização deve controlar e registrar a identificação única do produto.	Produção e Fornecimento de Serviço (Identificação e rastreabilidade)	7.5.3
A organização deve determinar as medições e monitoramentos a serem realizados e os dispositivos de medição e monitoramento necessários para evidenciar a conformidade do produto com os requisitos determinados.	Controle de dispositivos de Medição e Monitoramento	7.6

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Como uma das medições de desempenho do sistema de gestão de qualidade, a organização deve monitorar informações relativas à percepção do cliente sobre se a organização atendeu aos requisitos do cliente.	Medição e Monitoramento (Satisfação dos clientes)	8.2.1
A organização deve medir e monitorar as características do produto para verificar se os requisitos do produto tem sido atendidos.	Medição e Monitoramento (Medição e monitoramento de produto)	8.2.4
A organização deve assegurar que produtos que não estejam conformes com os requisitos do produto sejam identificados e controlados para evitar seu uso ou entrega não intencional.	Controle de Produto Não-Conforme	8.3
Quando o produto não conforme for corrigido, esse deve ser reverificado para demonstrar a conformidade com os requisitos.	Controle de Produto Não-Conforme	8.3

11.3.3 ISO 9001

A tabela a seguir é composta pelas colunas:

Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.

Coluna 2 - "Área" : Identifica a área da norma onde se encontra tal prática.

Coluna 3 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
A alta direção deve comunicar a organização da importância em atender aos requisitos do cliente.	Responsabilidade da direção	5.1
A alta direção deve assegurar que os requisitos do cliente são determinados e atendidos com o propósito de aumentar a satisfação do cliente.	Responsabilidade da direção	5.2
A alta direção deve indicar um membro da organização que deve Ter responsabilidade e autoridade para assegurar a promoção da conscientização sobre os requisitos do cliente em toda a organização.	Responsabilidade da direção	5.4.1
As saídas da análise crítica pela direção devem incluir quaisquer decisões e ações relacionadas à melhoria do produto em relação aos requisitos do cliente.	Responsabilidade da direção	5.6.3
A organização deve determinar e prover recursos necessários para aumentar a satisfação de cliente mediante o atendimento aos seus requisitos.	Gestão de recursos	6.1
A organização deve determinar, prover e manter a infra-estrutura necessária para alcançar a conformidade com os requisitos do produto.	Gestão de recursos	6.3
A organização deve determinar os requisitos especificados pelo cliente.	Processos relacionados a clientes (Determinação de requisitos relacionados ao produto)	7.2.1
A organização deve determinar os requisitos não declarados pelo cliente, mas que se fazem necessários.	Processos relacionados a clientes (Determinação de requisitos relacionados ao produto)	7.2.1
A organização deve determinar os requisitos estatutários e regulamentares relacionados ao produto.	Processos relacionados a clientes (Determinação de requisitos relacionados ao produto)	7.2.1
A organização deve determinar qualquer requisito adicional.	Processos relacionados a clientes (Determinação de requisitos relacionados ao produto)	7.2.1
Os requisitos do produto estão definidos.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
Os requisitos de contrato que difiram daqueles previamente manifestados estão resolvidos.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
A organização tem a capacidade de atender aos requisitos definidos.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
Devem ser mantidos registros dos resultados da análise crítica dos requisitos.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Quando o cliente não fornecer uma declaração documentada dos requisitos, a organização deve confirmar os requisitos do cliente antes da aceitação.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
Quando os requisitos de produto forem alterados, a organização deve assegurar que os documentos pertinentes são complementados e que o pessoal envolvido é alertado sobre os requisitos alterados.	Processos relacionados a clientes (Análise crítica dos requisitos relacionados ao produto)	7.2.2
Entradas relativas a requisitos de produto devem ser determinadas e registros devem ser mantidos e devem incluir: requisitos de funcionamento e de desempenho, requisitos estatutários e regulamentares aplicáveis e outros requisitos essenciais para o projeto e desenvolvimento.	Projeto e Desenvolvimento (Entradas do Projeto e Desenvolvimento)	7.3.2
Requisitos devem ser completos, sem ambigüidades e não conflitantes entre si.	Projeto e Desenvolvimento (Entradas de Projeto e Desenvolvimento)	7.3.2
As saídas de projeto e desenvolvimento devem atender aos requisitos de entrada para o projeto e desenvolvimento e devem conter ou referenciar critérios de aceitação do produto.	Projeto e Desenvolvimento (Saídas de Projeto e Desenvolvimento)	7.3.3
Devem ser realizadas, em fases apropriadas, análises críticas sistemáticas de projeto e desenvolvimento, de acordo com as disposições planejadas para avaliar a capacidade dos resultados do projeto e desenvolvimento em atender aos requisitos.	Projeto e Desenvolvimento (Análise crítica de Projeto e Desenvolvimento)	7.3.4
A verificação deve ser executada conforme disposições planejadas, para assegurar que as saídas do projeto e desenvolvimento estejam atendendo aos requisitos de entrada do projeto e desenvolvimento.	Projeto e Desenvolvimento (Verificação de Projeto e Desenvolvimento)	7.3.5
A validação do projeto e desenvolvimento deve ser executada conforme disposições planejadas, para assegurar que o produto resultante é capaz de atender os requisitos para a aplicação especificada.	Projeto e Desenvolvimento (Validação de Projeto e Desenvolvimento)	7.3.6
A organização deve assegurar que o produto adquirido está conforme com os requisitos especificados de aquisição.	Aquisição (Processo de Aquisição)	7.4.1
As informações de aquisição devem descrever o produto a ser adquirido e incluir, onde apropriado, requisitos para aprovação de produto, procedimentos, processos e equipamento, requisitos para qualificação de pessoal e requisitos para sistema de gestão de qualidade.	Aquisição (Informações de Aquisição)	7.4.2
A organização deve estabelecer e implementar inspeção ou outras atividades necessárias para assegurar que o produto adquirido atende aos requisitos de aquisição especificados.	Aquisição (Verificação do Produto Adquirido)	7.4.3
A organização deve identificar a situação do produto no que se refere aos requisitos de monitoramento e de medição	Produção e Fornecimento de Serviço (Identificação e rastreabilidade)	7.5.3
Quando a rastreabilidade é um requisito, a organização deve controlar e registrar a identificação única do produto.	Produção e Fornecimento de Serviço (Identificação e rastreabilidade)	7.5.3

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
A organização deve determinar as medições e monitoramentos a serem realizados e os dispositivos de medição e monitoramento necessários para evidenciar a conformidade do produto com os requisitos determinados.	Controle de dispositivos de Medição e Monitoramento	7.6
Quando usado na medição e monitoramento de requisitos especificados, deve ser confirmada a capacidade do software de computador para satisfazer a aplicação pretendida.	Controle de dispositivos de Medição e Monitoramento	7.6
Como uma das medições de desempenho do sistema de gestão de qualidade, a organização deve monitorar informações relativas à percepção do cliente sobre se a organização atendeu aos requisitos do cliente.	Medição e Monitoramento (Satisfação dos clientes)	8.2.1
A organização deve medir e monitorar as características do produto para verificar se os requisitos do produto tem sido atendidos.	Medição e Monitoramento (Medição e monitoramento de produto)	8.2.4
A organização deve assegurar que produtos que não estejam conformes com os requisitos do produto sejam identificados e controlados para evitar seu uso ou entrega não intencional.	Controle de Produto Não-Conforme	8.3
Quando o produto não conforme for corrigido, esse deve ser reverificado para demonstrar a conformidade com os requisitos.	Controle de Produto Não-Conforme	8.3

11.3.4 ISO 9004

A tabela a seguir é composta pelas colunas:

Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.

Coluna 2 - "Área" : Identifica a área da norma onde se encontra tal prática.

Coluna 3 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Convém que a direção identifique os processos necessários de realização dos produtos para satisfazer aos requisitos dos clientes e de outras partes interessadas.	Recomendações Gerais (Gestão de Processos)	7.1.3.1
Convém que um plano operacional seja definido para gestão dos processos, incluindo requisitos de entrada e de saída (por exemplo, especificações e recursos), verificação e validação dos processos e produtos.	Recomendações Gerais (Gestão de Processos)	7.1.3.1
Abordagem de processo assegura que as entradas do processo são definidas e registradas para fornecer uma base para a formulação dos requisitos a serem usados para a verificação das saídas.	Recomendações Gerais (Gestão de Processos)	7.1.3.2
A resolução de requisitos de entrada ambíguos ou contraditórios pode envolver consulta às partes internas ou externas afetadas.	Recomendações Gerais (Gestão de Processos)	7.1.3.2
Convém que as saídas dos processos que tenham sido verificadas em relação aos requisitos de entrada do processo, incluindo critério de aceitação, considerem as necessidades e expectativas dos clientes e de outras partes interessadas.	Recomendações Gerais (Gestão de Processos)	7.1.3.2
Convém que a organização implemente um processo de controle eficaz e eficiente de alterações para assegurar que as alterações no produto o processo beneficiam a organização e satisfazem as necessidades e expectativas das partes interessadas.	Recomendações Gerais (Gestão de Processos)	7.1.3.3
É recomendável que as mudanças sejam identificadas, registradas, avaliadas, analisadas criticamente e controladas para se compreender seu efeito sobre outros processos e as necessidades e expectativas dos clientes e de outras partes interessadas.	Recomendações Gerais (Gestão de Processos)	7.1.3.3
É recomendado que a organização implemente e mantenha processos para assegurar o entendimento adequado das necessidades e expectativas de suas partes interessadas, bem como convertê-las em requisitos para a organização (exemplos: requisitos do cliente ou de outras partes interessadas, requisito de contrato, processos devidos a requisitos estatutários ou regulamentares).	Processos relacionados às partes interessadas	7.2

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Convém que a organização tenha uma compreensão completa dos requisitos de processo do cliente.	Processos relacionados às partes interessadas	7.2
Convém que as necessidades e expectativas externas a organização, em conjunto com as internas, sejam adequadas para se converter em requisitos de entrada para os processos de projeto e de desenvolvimento.	Projeto e Desenvolvimento (Entrada e Saída de Projeto e Desenvolvimento)	7.3.2
Convém que a saída inclua informações para permitir a verificação e validação com os requisitos planejados.	Projeto e Desenvolvimento (Entrada e Saída de Projeto e Desenvolvimento)	7.3.2
Convém que a alta direção da organização assegure que processos de aquisição eficazes e eficientes sejam definidos e implementados para a avaliação e controle de produtos adquiridos, para que os produtos adquiridos satisfaçam as necessidades e requisitos da organização ou partes interessadas.	Aquisição (Processo de aquisição)	7.4.1
Convém que a alta direção vá além do controle do processo de realização para alcançar tanto a adequação com requisitos quanto a obtenção de benefícios para as partes interessadas.	Operações de Produção e Serviço (Operação e Realização)	7.5.1
A organização pode estabelecer um processo para a identificação e rastreabilidade que supere os requisitos, a fim de coletar dados que possam ser usados para melhoria (requisito de contrato, requisito estatutários e regulamentares).	Operações de Produção e Serviço (Identificação e Rastreabilidade)	7.5.2
Convém que a direção considere a necessidade de qualquer requisito especial resultante da natureza do produto. Requisitos especiais podem ser associados com programas de computador, meio de comunicação eletrônico, materiais perigosos, produtos que requeiram pessoas especializadas para o serviço, instalação ou aplicação, e produtos ou materiais que são únicos.	Operações de Produção e Serviço (Preservação de Produto)	7.5.4
Convém que a direção da organização use medições de satisfação do cliente como uma ferramenta vital. É conveniente que o processo da organização para solicitar, medir e monitorar a realimentação da satisfação dos clientes forneça informações freqüentes.	Medição e Monitoramento (Medição e Monitoramento do desempenho do sistema)	8.2.1.2
Convém que este processo considere a conformidade em relação aos requisitos, atendimento as necessidades e expectativas dos clientes, bem como ao preço e a entrega do produto.	Medição e Monitoramento (Medição e Monitoramento do desempenho do sistema)	8.2.1.2
O processo de auditoria interna fornece uma ferramenta independente para demonstrar por meio da obtenção de evidências objetivas que os requisitos existentes estão sendo atendidos, visto que a auditoria interna avalia a eficácia e a eficiência da organização.	Medição e Monitoramento (Medição e Monitoramento do desempenho do sistema)	8.2.1.3

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Convém que a organização estabeleça e especifique os requisitos de medição (incluindo critérios de aceitação) para os seus produtos. É conveniente que a medição do produto seja planejada e executada para verificar que os requisitos das partes interessadas foram atendidos e usados para melhorar os processos de realização.	Medição e Monitoramento (Medição e Monitoramento do produto)	8.2.3
Convém que a autoridade, para tratar de não-conformidades, seja definida a fim de manter o atendimento aos requisitos do processo e do produto.	Controle de Não-Conformidade (Generalidades)	8.3.1
A aceitação da correção de não conformidades pode ser um requisito contratual do cliente ou um requisito de outras partes interessadas.	Controle de Não-Conformidade (Controle de produto não-conforme)	8.3.2

11.4 ANEXO D: ISO 9126

A seguir são apresentadas as principais práticas de requisitos da ISO 9126.

A tabela a seguir é composta pelas colunas:

Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.

Coluna 2 - "Área" : Identifica a área da norma onde se encontra tal prática.

Coluna 3 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
As características de qualidade de produto de software definidas na ISO/IEC 9126-1 podem ser usadas para especificar requisitos funcionais e não-funcionais do cliente e do usuário.	Introdução	0
A ISO/IEC 9126-1 permite que a qualidade do produto de software seja especificada e avaliada em diferentes perspectivas pelos envolvidos com aquisição, requisitos, desenvolvimento, uso, avaliação, apoio, manutenção, garantia da qualidade e auditoria de software.	Objetivo	1
O modelo de qualidade definido na ISO/IEC 9126-1 é usado para validar a completitude da definição dos requisitos, identificar requisitos de software, identificar critérios de aceitação para produtos finais de software, e outros.	Objetivo	1
Qualquer requisito, especificação ou avaliação de qualidade do produto de software que esteja em conformidade com a ISO/IEC 9126-1 deve usar as características e subcaracterísticas das seções 6 e 7 desta norma.	Conformidade	2

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Uma especificação de requisitos de qualidade do produto de software que contenha métricas usadas para comparação deve declarar se as métricas possuem as propriedades especificadas no item A.4 desta norma.	Conformidade	2
As necessidades de qualidade do usuário incluem requisitos de qualidade em uso em contextos de uso específicos.	Abordagens para qualidade	5.1
Os requisitos para a qualidade do produto de software incluirão, geralmente, critérios para avaliação da qualidade interna, qualidade externa e qualidade em uso, para atingir as necessidades dos desenvolvedores, mantenedores, adquirentes e usuários finais.	Abordagens para qualidade	5.1
A qualidade especificada como requisito no início do ciclo de vida é uma visão, principalmente, do ponto de vista de qualidade externa e do usuário e, difere da qualidade do produto intermediário, tal como a qualidade na fase de projeto, que é uma visão principalmente do ponto de vista de qualidade interna e do desenvolvedor.	Qualidade de produto e o ciclo de vida de software	5.2
As necessidades do usuário podem mudar após terem sido explicitadas.	Qualidade de produto e o ciclo de vida de software	5.2
Requisitos de qualidade não podem ser completamente definidos antes do início do projeto.	Qualidade de produto e o ciclo de vida de software	5.2
É necessário entender as necessidades reais do usuário tão detalhadas quanto possível para representá-las nos requisitos.	Qualidade de produto e o ciclo de vida de software	5.2
Escalas de medições das métricas usadas nos requisitos de qualidade podem ser divididas em categorias correspondentes a diferentes graus de satisfação dos requisitos.	Qualidade de produto e o ciclo de vida de software	5.2
As necessidades de qualidade do usuário podem ser especificadas como requisitos de qualidade pelas métricas de qualidade em uso, pelas métricas externas e às vezes por métricas internas.	Qualidade de produto e o ciclo de vida de software	5.2
Convém que requisitos especificados por meio das métricas sejam utilizados como critérios quando o produto for validado.	Qualidade de produto e o ciclo de vida de software	5.2
Os requisitos de qualidade externa especificam o nível de qualidade requerido sob o ponto de vista externo e incluem requisitos derivados das necessidades de qualidade dos usuários e requisitos de qualidade em uso.	Qualidade de produto e o ciclo de vida de software	5.2
Os requisitos de qualidade externa são usados como meta para validação em várias fases (estágios) do desenvolvimento.	Qualidade de produto e o ciclo de vida de software	5.2
Convém que os requisitos de qualidade externa sejam declarados na especificação de requisitos de qualidade usando métricas externas, bem como convém que sejam traduzidos em requisitos de qualidade interna, e que também sejam usados como critérios quando da avaliação do produto.	Qualidade de produto e o ciclo de vida de software	5.2

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Os requisitos de qualidade interna especificam o nível de qualidade requerido sob o ponto de vista interno do produto e são usados para especificar as propriedades dos produtos intermediários (estes podem incluir modelos estáticos e dinâmicos, outros documentos e código fonte).	Qualidade de produto e o ciclo de vida de software	5.2
Requisitos de qualidade interna podem ser usados como meta para verificação em várias fases (estágios) de desenvolvimento.	Qualidade de produto e o ciclo de vida de software	5.2
Requisitos de qualidade interna podem ser usados para definir estratégia de desenvolvimento e critérios de avaliação e verificação durante o desenvolvimento.	Qualidade de produto e o ciclo de vida de software	5.2
A qualidade interna é medida e avaliada com relação aos requisitos de qualidade interna.	Qualidade de produto e o ciclo de vida de software	5.2
Capacidade do produto de software de prover funções que atendam necessidades explícitas e implícitas quando o software estiver sendo utilizado sob condições especificadas.	Funcionalidade	6.1
Capacidade do produto de software de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados.	Acurácia	6.1.2
Capacidade do produto de software estar de acordo com normas, convenções ou regulamentações previstas em leis e prescrições similares relacionadas às funcionalidades.	Conformidade relacionada à funcionalidade	6.1.5
Capacidade do produto de software de manter um nível de desempenho especificado quando usado em condições especificadas.	Confiabilidade	6.2
Falhas na especificação de requisitos podem limitar a confiabilidade do produto de software.	Confiabilidade	6.2
Capacidade do produto de software estar de acordo com normas, convenções ou regulamentações relacionadas à confiabilidade.	Conformidade relacionada à confiabilidade	6.2.4
Capacidade do produto de software estar de acordo com normas, convenções, guias de estilos ou regulamentações relacionadas à usabilidade.	Conformidade relacionada à usabilidade	6.3.5
Capacidade do produto de software estar de acordo com normas e convenções relacionadas à eficiência.	Conformidade relacionada à eficiência	6.4.3
Capacidade do produto de software de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais.	Manutenibilidade	6.5
Atendimento aos requisitos especificados	Anexos	B.22
Verificação e Validação (capacidade do produto de software estar de acordo com os requisitos especificados)	Anexos	B.33
Verificação e Validação (capacidade do produto de software estar de acordo com os requisitos especificados)	Anexos	B.34

11.5 ANEXO E: ISO 12207

A seguir são apresentadas as principais práticas de requisitos da ISO 12207.

A tabela a seguir é composta pelas colunas:

Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.

Coluna 2 - "Área" : Identifica a área da norma onde se encontra tal prática.

Coluna 3 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
O processo de aquisição contém as atividades e tarefas do adquirente. Inicia-se com a definição da necessidade de adquirir um sistema, um produto de software ou um serviço de software. O processo continua com a preparação e emissão de pedido de proposta, seleção de fornecer e gerência do processo de aquisição através da aceitação do sistema, produto de software ou serviço de software.	Processo de Aquisição	5.1
O adquirente deverá definir e analisar os requisitos do sistema, que inclui requisitos de negócio, organizacionais e requisitos de usuário, e também requisitos de segurança, proteção e outros requisitos críticos relacionados às atividades de projeto, testes e aderência a padrões e procedimentos.	Processo de Aquisição	5.1.1.2
Se o adquirente mantiver acordo com um fornecedor para execução da análise dos requisitos de um sistema, o adquirente deverá aprovar estes requisitos.	Processo de Aquisição	5.1.1.3
O adquirente pode executar a definição e análise de requisitos do software por conta própria ou pode manter acordo com um fornecedor para executar essa tarefa.	Processo de Aquisição	5.1.1.4
Para aquisição de um produto de software de prateleira, o adquirente deverá assegurar que os requisitos do produto de software sejam satisfeitos.	Processo de Aquisição	5.1.1.7
O adquirente deverá preparar, documentar e executar um plano de aquisição que deve conter os requisitos para o sistema.	Processo de Aquisição	5.1.1.8
O adquirente deverá definir e documentar a estratégia e condições (critérios) de aceitação.	Processo de Aquisição	5.1.1.9
Na atividade de preparação de pedido de proposta, o adquirente deve documentar os requisitos de aquisição, e esta documentação deve incluir, quando apropriado, os requisitos do sistema.	Processo de Aquisição	5.1.2.1
Na atividade de preparação e atualização do contrato, o adquirente deve estabelecer um procedimento para selecionar o fornecedor, incluindo critérios de avaliação de proposta e ponderação da aderência aos requisitos.	Processo de Aquisição	5.1.3.1

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
O adquirente deverá então, preparar e negociar um contrato com o fornecedor que trate dos requisitos de aquisição, incluindo custo e cronograma do produto ou serviço de software a ser entregue.	Processo de Aquisição	5.1.3.4
O processo de fornecimento contém as atividades e as tarefas do fornecedor. Na atividade de iniciação, o fornecedor conduz uma revisão dos requisitos que constam no pedido de proposta, levando em consideração políticas e outros regulamentos da organização.	Processo de Fornecimento	5.2
O fornecedor deve conduzir uma revisão dos requisitos que constam no pedido de proposta, levando em consideração políticas e outros regulamentos da organização.	Processo de Fornecimento	5.2.1.1
O fornecedor deve conduzir uma revisão dos requisitos de aquisição, para definir a estrutura para gerenciar e garantir a qualidade do produto ou serviço de software a ser entregue.	Processo de Fornecimento	5.2.4.1
O fornecedor deve estabelecer requisitos para os planos, e estes devem incluir necessidades de recursos e o envolvimento do adquirente.	Processo de Fornecimento	5.2.4.3
O fornecedor deve desenvolver e documentar os planos de gerência do projeto de acordo com os requisitos de planejamento, incluindo gerenciamento de proteção e segurança e outros requisitos críticos dos produtos ou serviços de software, e envolvimento do usuário através de exercícios de consolidação de requisitos, demonstrações de protótipos e avaliações.	Processo de Fornecimento	5.2.4.5
O fornecedor deve verificar todos os requisitos contratuais necessários, para assegurar que o produto ou serviço de software entregue ao adquirente foi desenvolvido ou executado, pelos subcontratados, de acordo com os requisitos do contrato original.	Processo de Fornecimento	5.2.5.4
O fornecedor deve executar as atividades de verificação e validação para demonstrar que os produtos ou serviços de software e os processos satisfazem completamente os seus respectivos requisitos.	Processo de Fornecimento	5.2.6.2
O processo de desenvolvimento contém as atividades e tarefas do desenvolvedor.	Processo de Desenvolvimento	5.3
O desenvolvedor deve desenvolver planos para conduzir as atividades do processo de desenvolvimento, e estes devem incluir padrões, métodos, ferramentas, ações e responsabilidades associadas com o desenvolvimento e qualificação de todos os requisitos, inclusive proteção e segurança.	Processo de Desenvolvimento	5.3.1.4

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
A especificação dos requisitos do sistema deve descrever funções e capacidades do sistema, requisitos de negócio, requisitos organizacionais e de usuários, requisitos de proteção e segurança, requisitos de engenharia, requisitos de fatores humanos, requisitos de interface, requisitos de operação e manutenção, restrições do projeto e requisitos de qualificação.	Processo de Desenvolvimento	5.3.2.1
A especificação dos requisitos do sistema deve ser documentada.	Processo de Desenvolvimento	5.3.2.1
Os requisitos do sistema devem ser avaliados, considerando rastreabilidade para as necessidades de aquisição, consistência com as necessidades de aquisição, testabilidade, viabilidade do projeto de arquitetura do sistema e viabilidade de operação e manutenção.	Processo de Desenvolvimento	5.3.2.2
Uma arquitetura de alto nível do sistema deve ser estabelecida para identificar itens de hardware, software e operações manuais e deve assegurar que todos os requisitos do sistema sejam alocados entre estes itens.	Processo de Desenvolvimento	5.3.3.1
A arquitetura do sistema e os requisitos para os itens devem ser avaliados quanto a rastreabilidade para os requisitos do sistema, consistência com os requisitos do sistema, adequação aos métodos e padrões de projeto utilizados, viabilidade dos itens de software atenderem seus requisitos alocados e viabilidade de operação e manutenção. E os resultados das avaliações devem ser documentados.	Processo de Desenvolvimento	5.3.3.2
O desenvolvedor deve estabelecer e documentar os requisitos do software, incluindo as especificações das características de qualidade (ISO/IEC 9126), como requisitos de qualificação, requisitos de base de dados, requisitos de instalação e aceitação do produto de software, requisitos do usuário para execução, operação e manutenção.	Processo de Desenvolvimento	5.3.4.1
O desenvolvedor deve avaliar os requisitos de software considerando critérios como rastreabilidade para os requisitos do sistema e projeto do sistema, consistência externa com os requisitos do sistema.	Processo de Desenvolvimento	5.3.4.2
O desenvolvedor deve conduzir revisões conjuntas e concluindo-as, uma baseline para os requisitos do item de software deve ser estabelecida.	Processo de Desenvolvimento	5.3.4.3
O desenvolver deve transformar os requisitos para o item de software em uma arquitetura que descreve sua estrutura de alto nível e identifica os componentes de software. Deve ser garantido que todos os requisitos do item de software sejam alocados aos seus componentes de software, e sejam refinados para facilitar o projeto detalhado.	Processo de Desenvolvimento	5.3.5.1
O desenvolvedor deve definir e documentar os requisitos preliminares de teste e o cronograma para a integração do software.	Processo de Desenvolvimento	5.3.5.5

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
O desenvolvedor deve avaliar a arquitetura do item de software e os projetos de interface e base de dados, considerando critérios como rastreabilidade para os requisitos do item de software e consistência externa com os requisitos do item de software.	Processo de Desenvolvimento	5.3.5.6
O desenvolvedor deve desenvolver um projeto detalhado para cada componente de software do item de software. Os componentes de software devem ser refinados em níveis mais baixos, contendo unidades de software que possam ser codificadas, compiladas e testadas. Deve ser garantido que todos os requisitos do software sejam alocados para unidades de software a partir dos componentes de software. O projeto detalhado deve ser documentado.	Processo de Desenvolvimento	5.3.6.1
O desenvolvedor deve definir e documentar os requisitos de teste que devem incluir testes de estresse de unidade de software, até o limite de seus requisitos.	Processo de Desenvolvimento	5.3.6.5
O desenvolvedor deve atualizar os requisitos de teste.	Processo de Desenvolvimento	5.3.6.6
O desenvolvedor deve avaliar o projeto detalhado do software e os requisitos de teste considerando critérios como rastreabilidade para os requisitos do item de software.	Processo de Desenvolvimento	5.3.6.7
O desenvolvedor deve testar cada unidade do software e base de dados, garantindo que sejam atendidos seus requisitos.	Processo de Desenvolvimento	5.3.7.2
O desenvolvedor deve atualizar os requisitos de teste e o cronograma para integração do software.	Processo de Desenvolvimento	5.3.7.4
O desenvolvedor deve avaliar o código do software e os resultados dos testes, considerando critérios como rastreabilidade para os requisitos e projeto do item de software, consistência externa com os requisitos e projeto do item de software e consistência interna entre os requisitos da unidade.	Processo de Desenvolvimento	5.3.7.5
Na atividade de integração do software, deve ser garantido que cada agregação atenda os requisitos do item de software.	Processo de Desenvolvimento	5.3.8.2
O desenvolvedor deve desenvolver e documentar, para cada requisito de qualificação do item de software, um conjunto de testes, casos de teste e procedimento de teste.	Processo de Desenvolvimento	5.3.8.4
O desenvolvedor deve avaliar o plano de integração, projeto, código e testes considerando critérios como rastreabilidade para os requisitos do sistema, consistência externa com os requisitos do sistema e cobertura de teste dos requisitos do item de software.	Processo de Desenvolvimento	5.3.8.5
O desenvolvedor deve conduzir o teste de qualificação de acordo com os requisitos de qualificação para o item de software. Deve ser garantido que a implementação de cada requisito do software seja testada para conformidade.	Processo de Desenvolvimento	5.3.9.1

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
O desenvolvedor deve avaliar o projeto, código, teste e os resultados dos testes, considerando critérios como cobertura de teste dos requisitos do item de software.	Processo de Desenvolvimento	5.3.9.3
As agregações devem ser testadas quando integradas, de acordo com os seus requisitos.	Processo de Desenvolvimento	5.3.10.1
O sistema integrado deve ser avaliado de acordo com critérios como cobertura de teste dos requisitos de sistema.	Processo de Desenvolvimento	5.3.10.3
O teste deve ser conduzido de acordo com os requisitos de qualificação e deve ser garantido que a implementação de cada requisito do sistema seja testada.	Processo de Desenvolvimento	5.3.11.1
O sistema deve ser avaliado, considerando critérios como cobertura de teste dos requisitos do sistema.	Processo de Desenvolvimento	5.3.11.2
O mantenedor deve conduzir a análise e determinar que documentação, unidades de software e versões destas necessitam ser modificadas. Estas devem ser documentadas.	Processo de Manutenção	5.5.3.1
Os requisitos do processo de desenvolvimento devem ser completados e deve ser garantida a implementação completa e correta dos requisitos novos e dos modificados. Deve ser garantido que os requisitos originais não modificados não foram afetados.	Processo de Manutenção	5.5.3.2
Deve ser determinado e garantido a completeza funcional dos itens de software em relação aos seus requisitos e a completeza física dos itens de software (projeto e código refletem uma descrição técnica atualizada).	Processo de Gerência de Configuração	6.2.5.1
O processo de garantia da qualidade é o processo para fornecer garantia adequada de que os processos e produtos de software, no ciclo de vida do projeto, estejam em conformidade com os seus requisitos especificados e sejam aderentes aos planos estabelecidos.	Processo de garantia da qualidade	6.3
As atividades e tarefas de garantia da qualidade agendadas e em andamento devem ser executadas e quando problemas ou não-conformidades aos requisitos do contrato são detectados, devem ser documentados e servem de entrada ao processo de resolução de problema.	Processo de garantia da qualidade	6.3.1.4
Na preparação da entrega dos produtos de software deve ser garantido que os produtos de software tenham seus requisitos contratuais inteiramente satisfeitos e sejam aceitáveis pelos adquirentes.	Processo de garantia da qualidade	6.3.2.3
Deve ser garantido que os requisitos aplicáveis ao contrato original sejam passados para os subcontratados e que os produtos de software do subcontratado satisfaçam os requisitos do contrato original.	Processo de garantia da qualidade	6.3.3.3

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Deve ser garantido que a equipe alocada tenha qualificação e o conhecimento necessário para atender os requisitos do projeto e recebam todo treinamento necessário.	Processo de garantia da qualidade	6.3.3.6
O processo de verificação é um processo para determinar se os produtos de software de uma atividade atendem completamente os requisitos ou condições impostas a eles nas atividades anteriores.	Processo de Verificação	6.4
Deve ser determinado se o projeto justifica um esforço de verificação e os requisitos devem ser analisados em função dos fatores críticos.	Processo de Verificação	6.4.1.1
O contrato deve ser verificado segundo critérios como capacidade do fornecedor de atender os requisitos, requisitos estão consistentes e cobrem as necessidades do usuário, procedimentos adequados para tratar alterações nos requisitos e priorização de problemas estão estipulados, critérios e procedimentos de aceitação estão estipulados de acordo com os requisitos.	Processo de Verificação	6.4.2.1
Os requisitos devem ser verificados segundo critérios como requisitos do sistema são consistentes, viáveis e testáveis, os requisitos do sistema foram distribuídos apropriadamente para os itens de hardware, software, os requisitos de software são consistentes, viáveis e testáveis, os requisitos de software relacionados à proteção e segurança e aos fatores críticos estão corretos, conforme demonstrado por métodos adequadamente rigorosos.	Processo de Verificação	6.4.2.3
O projeto é verificado segundo critérios como: o projeto está correto e consistente com os requisitos e rastreável aos mesmos, o projeto implementa proteção, segurança e outros requisitos críticos corretamente, conforme demonstrado por métodos adequadamente rigorosos.	Processo de Verificação	6.4.2.4
O código deve ser verificado segundo critérios como o código é rastreável para o projeto e para os requisitos, testável, correto e aderente aos requisitos e padrões de codificação, o código selecionado pode ser originado a partir do projeto ou dos requisitos, o código implementa proteção, segurança e outros requisitos críticos corretamente, conforme demonstrado por métodos adequadamente rigorosos.	Processo de Verificação	6.4.2.5
O processo de validação é um processo para determinar se os requisitos e o produto final, sistema ou produto de software construído, atendem ao uso específico pretendido. A validação pode ser conduzida nos estágios iniciais.	Processo de Validação	6.5
A atividade de validação consiste nas seguintes tarefas: preparar os requisitos de teste e assegurar que esses requisitos de teste reflitam os requisitos particulares.	Processo de Validação	6.5.2
O processo de auditoria é um processo para determinar adequação aos requisitos, planos e contratos, quando apropriado.	Processo de Auditoria	6.7

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
As atividades de auditoria devem assegurar que a revisão de aceitação e requisitos de teste prescritos pela documentação estejam adequados para aceitação dos produtos de software.	Processo de Auditoria	6.7.2
O gerente deve garantir que o software e os planos sejam avaliados para satisfazer requisitos.	Processo de Gerência	7.1.4.1

11.6 ANEXO F: IEEE 830

A seguir são apresentadas as principais práticas de requisitos da IEEE 830.

A tabela a seguir é composta pelas colunas:

Coluna 1 - "PRÁTICAS" : Apresenta uma descrição da prática.

Coluna 2 - "Área" : Identifica a área da norma onde se encontra tal prática.

Coluna 3 - "Item" : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Reduzir o esforço de desenvolvimento. A preparação de um SRS força os grupos interessados nos clientes da organização a considerar rigorosamente todos os requisitos de iniciar a fase de projeto e reduzir o retrabalho.	Meta	Foreword
Revisões cuidadosas dos requisitos no SRS podem revelar problemas no início do ciclo de vida de modo a tornar sua correção mais fácil.	Meta	Foreword
Os requisitos juntamente com o plano de desenvolvimento podem ser usados para medir o processo.	Meta	Foreword
Requisitos devem ter apenas uma interpretação.	Características de um bom SRS (Não ambíguos)	4.3.1
Quem escreve SRS em linguagem natural devem ser especificamente cuidadosos para que os requisitos não apresentem ambigüidades.	Características de um bom SRS (Não ambíguos)	4.3.1.1
Para evitar ambigüidade, escrever SRS em linguagem formal de especificação de requisitos.	Características de um bom SRS (Não ambíguos)	4.3.1.2
Um SRS deve ser completo e deve incluir todos os requisitos significantes.	Características de um bom SRS (Completo)	4.3.2
Um SRS pode ser verificado se e somente todos os requisitos declarados são verificáveis.	Características de um bom SRS (Verificável)	4.3.3
Se um método não pode ser criado para determinar se o software possui um requisito particular, então aquele requisito deve ser removido ou revisado.	Características de um bom SRS (Verificável)	4.3.3
Se um requisito não é expresso na verificação dos termos quando o SRS é preparado, então um ponto no ciclo de desenvolvimento (revisão, planos de testes) deve ser verificado qual requisito deve ser colocado no formulário de verificação.	Características de um bom SRS (Verificável)	4.3.3
Um SRS é consistente se e somente se os requisitos descritos não apresentarem conflitos entre si.	Características de um bom SRS (Consistente)	4.3.4
Um SRS é alterável se sua estrutura ou estilo são tal que, qualquer alteração necessária nos requisitos, podem ser feitas de maneira fácil, completa e consistente.	Características de um bom SRS (Alterável)	4.3.5
Para um SRS ser alterável, ele não pode ser redundante, ou seja, um mesmo requisito não pode aparecer em dois ou mais lugares diferentes no SRS.	Características de um bom SRS (Alterável)	4.3.5

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Um SRS é rastreável se a origem de cada um dos requisitos está clara e se a referência de cada requisito é facilitada na documentação de desenvolvimento.	Características de um bom SRS (Rastreabilidade)	4.3.6
Como o código ou documentação de projeto é alterável, é essencial ser capaz de averiguar o conjunto completo de requisitos que podem ser afetados por essas alterações.	Características de um bom SRS (Rastreabilidade)	4.3.6
O projeto e a documentação de requisitos são essenciais quando alterações são feitas em larga escala.	Características de um bom SRS (Usabilidade durante as fases de operação e manutenção)	4.3.7
O processo de desenvolvimento inicia com o acordo entre fornecedor e cliente do que o software deve fazer.	Preparação em conjunto da especificação	4.4
Fornecedores geralmente têm dificuldade de entender o problema do cliente e precisam de grande esforço para especificar requisitos para um sistema satisfatório.	Preparação em conjunto da especificação	4.4
Os requisitos devem ser especificados por completo.	Evolução SRS	4.5.1
As alterações aprovadas devem ser incorporadas no SRS.	Evolução SRS	4.5.2
Requisitos podem ser expressos por especificação de entradas e saídas, pelo uso de um conjunto de exemplos representativos e pela especificação de modelos.	Métodos usados para expressar requisitos de software	5.1
Expressar os requisitos na forma de um modelo (requisitos complexos).	Métodos usados para expressar requisitos de software (Modelos)	5.1.3
Manter uma definição de modelo dentro do domínio dos requisitos.	Métodos usados para expressar requisitos de software (Modelos)	5.1.3.5
Sempre que um SRS usar um modelo, isso significa que o modelo prove um meio eficiente para especificar requisitos.	Métodos usados para expressar requisitos de software (Modelos)	5.1.3.5
Um modelo que funciona efetivamente para explicar requisitos em um documento escrito pode não ser bom para a atual implementação de software.	Métodos usados para expressar requisitos de software (Modelos)	5.1.3.5
Todos os requisitos relacionados ao produto de software não são igualmente importantes (alguns requisitos podem ser essenciais).	Anotação dos requisitos de software	5.2
Clientes dão mais consideração cuidadosa para cada requisito.	Anotação dos requisitos de software	5.2
Um requisito pode ser considerado estável quando não será alterado durante a vida do software.	Anotação dos requisitos de software (Estabilidade)	5.2.1
Distinguir classes de requisitos como obrigatório, desejado e opcional.	Anotação dos requisitos de software (Grau de necessidade)	5.2.2
Antes de anotar requisitos, um entendimento completo das implicações contratuais de tais anotações, devem ser obtidas.	Anotação dos requisitos de software (Precaução)	5.2.3
Os itens básicos que os requisitos escritos devem tratar são: funcionalidade, desempenho, regras de projeto impostas a implementação, atributos, e interfaces externas.	Itens expressos em requisitos	5.3

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
O escritor de requisitos deve evitar colocar requisitos de projeto no SRS.	Itens expressos em requisitos	5.3
Considerações de segurança podem impor requisitos que refletem diretamente nas regras do projeto.	Itens expressos em requisitos (Projeto no SRS)	5.3.1
Requisitos de projeto representam um entendimento entre um cliente e um fornecedor dos assuntos contratuais pertinentes na produção de software.	Itens expressos em requisitos (Projeto no SRS)	5.3.2
Requisitos de projeto são especificados em outros documentos, como plano de desenvolvimento de programas de computador. Os requisitos apenas para produto de software são descritos no SRS.	Itens expressos em requisitos (Projeto no SRS)	5.3.2

LEGENDA

SRS = *Software Requirements Specifications*

11.7 ANEXO G: PMBOK

A seguir são apresentadas as principais práticas de requisitos do modelo PMBOK.

A tabela a seguir é composta pelas colunas:

Coluna 1 - “PRÁTICAS” : Apresenta uma descrição da prática.

Coluna 2 - “Área” : Identifica a área do modelo onde se encontra tal prática.

Coluna 3 - “Item” : Identifica o item ao qual a prática se refere.

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
O <i>project charter</i> é um documento que reconhece formalmente os requisitos do projeto.	Saídas da Iniciação	5.1.3
Planejamento do escopo: desenvolver uma declaração escrita do escopo como base para decisões futuras do projeto.	Planejamento do Escopo	5.2
A declaração de escopo fornece a documentação que servirá de base para tomada de decisões futuras no projeto e para confirmar ou desenvolver um entendimento comum do escopo entre as partes envolvidas.	Saídas do Planejamento do Escopo	5.2.3
Os detalhes de suporte para a declaração do escopo devem ser documentados e organizados.	Saídas do Planejamento do Escopo	5.2.3
O plano de gerência de escopo descreve como o escopo do projeto será gerenciado.	Saídas do Planejamento do Escopo	5.2.3
Detalhamento do escopo: subdividir os principais subprodutos do projeto em componentes menores e mais manejáveis.	Detalhamento do Escopo	5.3
Uma estrutura analítica do projeto (EAP) é um agrupamento orientado ao subproduto dos elementos do projeto que organiza e define o escopo do projeto.	Saídas do Detalhamento do Escopo	5.3.3
Verificação do escopo: formalizar a aprovação do escopo do projeto.	Verificação do Escopo	5.4
Documentação deve estar disponível para revisão.	Entradas para Verificação do Escopo	5.4.1
A inspeção inclui atividades tais como medição, exames e testes incumbidos de determinar se os resultados estão de acordo com os requisitos.	Ferramentas e Técnicas para Verificação do Escopo	5.4.2
Preparar a documentação de aceitação formal do cliente.	Saídas da Verificação do Escopo	5.4.3
Controlar as mudanças do escopo do projeto.	Controle de Mudanças do Escopo	5.5
Garantir que as mudanças no escopo sejam benéficas.	Controle de Mudanças do Escopo	5.5
Detectar mudança no escopo.	Controle de Mudanças do Escopo	5.5
Gerenciar as mudanças.	Controle de Mudanças do Escopo	5.5
A estrutura analítica do projeto defini a <i>baseline</i> do escopo do projeto.	Entradas para o Controle de Mudanças do Escopo	5.5.1
No projeto sob contrato, o sistema de controle de mudança de escopo deve estar em conformidade com as cláusulas relevantes do contrato.	Ferramentas e Técnicas para o Controle de Mudanças do Escopo	5.5.2

PRÁTICAS	LOCALIZAÇÃO	
	Área	Item
Adequação dos planos às mudanças de escopo.	Ferramentas e Técnicas para o Controle de Mudanças do Escopo	5.5.2
Manter banco de dados com histórico do controle de mudanças de escopo.	Saídas do Controle de Mudanças do Escopo	5.5.3

11.8 ANEXO H: Bibliografia

Bibliografia em engenharia de requisitos recomendada pelo IEEE *Task Force on Requirements Engineering*:

Abbott, R. J. and Moorhead, D. K. (1981). Software requirements and specifications: A survey of needs and languages. *The Journal of Systems and Software*, 2:297-316.

Akao, Y. (1990). An introduction to quality function deployment. In Akao, Y., editor, *Quality Function Deployment. Integrating Customer Requirements into Product Design*, chapter 1, pages 3-24. Productivity Press.

Alford, M. (1992). Strengthening the systems engineering process. *Engineering Management Journal*, 4(1):7-14.

Antón, A.I. (1996). Goal-Based Requirements Analysis. I *2nd IEEE International Conference on Requirements Engineering*.

Bailin, S. C. (1989). An object-oriented requirements specification method. *Communications of the ACM*, 32(5):608-623.

Balzer, R. and Goldman, N. (1981). Principles of good software specification and their implications for specification languages. In *National Computer Conference*, pages 393-400.

Barnard, J. and Price, A. (1994). Managing Code Inspection Information. *IEEE Software*, 11(2):59-69.

Basili, V.R. and Perricone, B.T. (1984). Software errors and complexity: An empirical investigation. *Communications of the ACM*, 27(1):42-52.

Basili, V.R. and Rombach, D. (1988). The TAME Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions on Software Engineering*, SE-14(6):758-773.

Basili, V.R. and Weiss, D.M. (1981). Evaluation of a software requirements document by analysis of change data. In *Proceedings of the 5th International Conference on Software Engineering*, pages 314-321.

Berry, D.M. (1995). The importance of ignorance in requirements engineering. *The Journal of Systems and Software*, 28(2):179-184.

Boehm, B.W. and In, H. (1996). Identifying Quality-Requirements Conflicts. *IEEE Software*, 13(2):25-35.

Boehm, B.W. and Ross, R. (1989). Theory-W Software Project Management: Principles and Examples. *IEEE Transactions on Software Engineering*, 15(7):902-916.

Boehm, B. W., Gray, T. E., and Seewaldt, T. (1984). Prototyping versus specifying: A multiproject experiment. *IEEE Transactions on Software Engineering*, 10(3):290-302.

Boehm, B.W., Bose, P., Horowitz, E. and Lee, M. (1994). Software Requirements As Negotiated Win Conditions. I *1st IEEE International Conference On Requirements Engineering*, pp. 74-83.

Boehm, B.W. (1980). *Software Engineering Economics*. Prentice-Hall International, Englewood Cliffs, New Jersey.

Boehm, B.W. (1984). Verifying and Validating Software Requirements and Design Specifications. *IEEE Software*, 1(1):75-88.

Boehm, B.W. (1987). Industrial software metrics top 10 list. *IEEE Software*, 4(5):84-85.

- Boad, G. (1994). *Object-Oriented Analysis and Design with Applications*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 2nd edition.
- Bowen, J.P. and Hinchey, M.G. (1994). Seven more myths of formal methods. Technical report PRG-TR-7-94, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD.
- Brackett, J.W. (1990). Software Requirements. Technical Report SEI-CM-19-1.2, Software Engineering Institute.
- Brooks Jr., F.P. (1987). No Silver Bullet - Essence and Accidents of Software Engineering. *IEEE Computer*, 20(4):10-19.
- Bubenko Jr., J.A. (1995). Challenges in requirements engineering. In *Second IEEE International Symposium on Requirements Engineering*, pages 160-162. IEEE Computer Society Press.
- Byrne, J.G. and Barlow, T. (1993). Structured brainstorming: A method for collecting user requirements. In *Human Factors and Ergonomics Society 37th Annual Meeting*, pages 427-432.
- Carlshamre, P. (1994). *A Collaborative Approach to Usability Engineering: Technical Communicators and Systems Developers in Usability-Oriented Systems Development*. Licentiatavhandling 455, Institutionen för Datavetenskap, Linköpings universitet.
- Carlshamre, P. and Karlsson, J. (1996). A Usability-Oriented Approach to Requirements Engineering. I *2nd IEEE International Conference On Requirements Engineering*, pp. 145-152.
- Ceri, S. (1986). Requirements collection and analysis in information systems design. In Kugler, H.-J., editor, *Proceedings of the IFIP 10th World Computer Congress*, pages 205-214. North-Holland.
- Christel, M.G. and Kang, K.C. (1992). Issues in Requirements Elicitation. Technical Report CMU/SEI-92-TR-12, Software Engineering Institute, Carnegie Mellon University.
- Chung, L. (1991). Representation and utilization of non-functional requirements for information system design. In Andersen, R., Bubenko Jr., J., and Sölvberg, editors, *Advanced Information Systems Engineering. Third International Conference CAiSE '91*, pages 5-30. Springer-Verlag.
- Coad, P. and Yourdon, E. (1991). *Object-Oriented Analysis*. Prentice-Hall International, Inc.
- Cohen, B., Harwood, W., and Jackson, M. (1986). *The Specification of Complex Systems*. Addison-Wesley Publishing Company, Inc.
- Cohen, L. (1995). *Quality Function Deployment: How to Make QFD Work for You*. Addison-Wesley Publishing Company.
- Cordes, D.W. and Carver, D.L. (1988). Knowledge base applications with software engineering: A tool for requirements specifications. In *Proceedings of the First International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. 2-3 June.
- Cordes, D.W. and Carver, D.L. (1989). Evaluation method for user requirements documents. *Information Software and Technology*, 31(4):181-188.
- Cordes, D.W. and Carver, D.L. (1992). An object-based requirements modeling method. *Journal of the American Society for Information Science*, 43(1):62-71.
- Costello, R.J. and Liu, D.-B. (1995). Metrics for requirements engineering. *The Journal of Systems and Software*, 29(1):39-63.
- Crosby, P.B. (1979). *Quality Is Free. The Art of Making Quality Certain*. McGraw-Hill, Inc., New York.
- Curtis, B. Krasner, H. and Iscoe, N. (1988). A Field Study of the Software Design Process for Large Systems. *Communications of the ACM*, 31(11):1268-1287.
- Dalal, S.R., Horgan, J.R., and Kettenring, J.R. (1993). Reliable software and communication: Software quality, reliability, and safety. In *15th International Conference on Software Engineering*, pages 425-435. IEEE Computer Society Press.

- Dasarathy, B. (1985). Timing constraints of real-time systems: Constructs for expressing them, methods of validating them. *IEEE Transactions on Software Engineering*, SE-11(1):80-86.
- Davis, A. (1988a). A taxonomy for the early stages of the software development life cycle. *The Journal of Systems and Software*, 8(4).
- Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A. and Theofanos, M. (1993). Identifying and Measuring Quality in a Software Requirements Specification. I *1st International Symposium on Software Metrics*, pp. 141-152.
- Davis, A.M. (1988b). A comparison of techniques for the specification of external system behaviour. *Communications of the ACM*, 31(9):1098-1115.
- Davis, A.M. (1992). Operational Prototyping: A New Development Approach. *IEEE Software*, 9(5):70-78.
- Davis, A. (1993). *Software Requirements: Objects, Functions and States*. Prentice-Hall International, Englewood Cliffs, New Jersey.
- Davis, G.B. (1982). Strategies for information requirements determination. *IBM Systems Journal*, 21(1):4-30.
- de Boer, R.R.A (1993) Computer Aided Requirements - Vehicle for Software System Development, Vakgroep Informatica, Faculteit der Economische Wetenschappen, Erasmus Universiteit Rotterdam.
- Delta (1994). *Delta Method Handbook*. Ericsson Infocom Consultants AB.
- Dion, R. (1993). Process improvement and the corporate balance sheet. *IEEE Software*, 10(4):28-35.
- Dobson, J., Blyth, A., Chudge, J., and Strens, M. (1992). The ORDIT approach to requirements identification. In *International Computer Software and Applications Conference*, pages 356-361.
- DOD-STD-2167A (1988). Military Standard. Defense Systems Software Development. Department of Defense, Washington, D.C. 20301.
- Doolan, E.P. (1992). Experience with Fagan's Inspection Method. *Software-Practice and Experience*, 22(2):173-182.
- Easterbrook, S. and Nuseibeh, B. (1995). Managing inconsistencies in an evolving specification. In *Second IEEE International Symposium on Requirements Engineering*, pages 48-55. IEEE Computer Society Press.
- El Emam, K. and Madhavji, N.H. (1995a). A field study of requirements engineering practices in information systems development. In *Second IEEE International Symposium on Requirements Engineering*, pages 68-80. IEEE Computer Society Press.
- El Emam, K. and Madhavji, N.H. (1995b). Measuring the success of requirements engineering processes. In *Second IEEE International Symposium on Requirements Engineering*, pages 204-211. IEEE Computer Society Press.
- Fagan, M. (1976). Design and Code Inspections To Reduce Errors in Program Development. *IBM Systems Journal*, 12(3):182-211.
- Fagan, M.E. (1986). Advances in Software Inspections. *IEEE Transactions on Software Engineering*, SE-12(7):744-751.
- Fickas, S. and Nagarajan, P. (1988). Critiquing software specifications. *IEEE Software*, 5(6):37-47.
- Finkelstein, A. (1988). Re-use of formatted requirements specifications. *Software Engineering Journal*, 3(5):186-197.
- Finkelstein, A. (1991). Tracing back from requirements. In Colloquium On "Tools and Techniques For Maintaining Traceability During Design". The institution of electrical engineers.

- Finkelstein, A. and Potts, C. (1986). Structured common sense: The elicitation and formalization of system requirements. In Barnes and Brown, editors, *Software Engineering'86*, pages 237-250. Peter Peregrinus Ltd., London.
- Finkelstein, A., Easterbrook, S., Kramer, J., and Nuseibeh, B. (1992?). Requirements engineering through viewpoints. Technical report, Imperial College, Department of Computing, 180 Queen's Gate, London SW7 2BZ.
- Fraser, M.D., Kumar, K., and Vaishnavi, V.K. (1991). Informal and formal requirements specification languages: Bridging the gap. *IEEE Transactions on Software Engineering*, 17(5):454-466.
- Gause, D.C. and Weinberg, G.W. (1989). *Exploring Requirements: Quality Before Design*. Dorset House Publishing, New York.
- Goldsack, S. and Finkelstein, A. (1991). Requirements engineering for real-time systems. *Software Engineering Journal*, 6(3):101-115.
- Gomaa, H. (1990). The Impact of Prototyping on Software System Engineering. I Thayer, R.H. and Dorfman, M. (eds.), *System and Software Requirements Engineering*, pp. 543-552.
- Gomaa, H. and Scott, D. B.H. (1981). Prototyping as a tool in the specification of user requirements. In *Proceeding of the 5th International Conference on Software Engineering*, pages 333-339.
- Goodrich, V. and Offman, L. (1990). An Experimental Evaluation of Task and Methodology Variables for Requirements Definition Phase Success. I *23rd Annual Hawaii International Conference on Systems Sciences*, pp. 201-209.
- Grady, R. B. and Van Slack, T. (1994). Key Lessons in Achieving Widespread Inspection Use. *IEEE Software*, 11(4):46-57.
- Guttag, J. and Horning, J. (1993). *LARCH: languages and Tools for Formal Specification*. Springer-Verlag.
- Haag, S., Raja, M.K. and Schkade, L.L (1996). Quality Function Deployment: Usage in Software Development. *Communications of the ACM*, 39(1):41-49.
- Hall, A. (1990). Seven Myths of Formal Methods. *IEEE Software*, 7(5):11-19.
- Hammer, M. (1990). Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review*, July-Aug., pp. 104-112.
- Harker, S. D.P. and Eason, K.D. (1993). The change and evolution of requirements as a challenge to the practice of software engineering. In *IEEE International Symposium on Requirements Engineering*, pages 266-272. IEEE Computer Society.
- Hauser, J.R. and Clausing, D. (1988). The House of Quality. *Harvard Business Review*, May-June issue, pp. 63-73.
- Herman, J. S., editor (1991). Requirements Engineering and Analysis Workshop Proceeding, number CMU/SEI-91-TR-30, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA.
- Hoffmann, H. F. (1993). Requirements Engineering - A Survey of Methods and Tools, Institut für Informatik der Universität Zürich.
- Holbrook III, H. (1990). A scenario-based methodology for conducting requirements elicitation. *ACM SIGSOFT*, 18(1):95-104.
- Horowitz, E. (1995). WinWin Reference Manual: A System for Collaboration and Negotiation of Requirements.
- Hsia, P., Davis, A., and Kung, D. (1993). Status report: Requirements engineering. *IEEE Software*, 10(6):75-79.
- Hughes, J., O'Brien, J., Rodden, T., Rouncefield, M., and Sommerville, I. (1995). Presenting ethnography in the requirements process. In *Second IEEE International Symposium on Requirements Engineering*, pages 27-34. IEEE Computer Society Press.

- Humphrey, W.S. (1989). *Managing the Software Process*. Addison-Wesley Publishing Company, Reading, MA.
- Hunter, M.R. and Van Landingham, R.D. (1994). Listening to the Customer Using QFD. *Quality Progress*, 27(4):55-59.
- Ibanez, M. and Rempp, H. (1996). European User Survey Analysis. Report USV_EUR, version 2.1, 30 januari.
- IEEE Standard 729 (1983). *IEEE Standard Glossary of Software Engineering Terminology*. Institute of Electrical and Electronic Engineers, New York.
- IEEE Std. 830 (1984). *IEEE Guide to Software Requirements Specifications*. Institute of Electrical and Electronic Engineers, New York.
- Jackson, M. (1995). *Software Requirements & Specifications*. Addison-Wesley Publishing Company. [40]
- Jacobson, I., Christersson, M., Jonsson, P. and Overgaard, G. (1992). *Object-Oriented Software Engineering-A Use Case Driven Approach*. Addison-Wesley Publishing Company.
- Jacobson, I., Ericsson, M. and Jacobson, A. (1995). *The Object Advantage-Business Process Reengineering with Object Technology*. Addison-Wesley Publishing Company.
- Jarke, M. and Pohl, K. (1994). Requirements engineering in 2001: (virtually) managing a changing reality. *Software Engineering Journal*, November issue, pages 257-266.
- Jeffrey, H.J. and Putman, A.O. (1994). Relationship definition and management: Tools for requirements analysis. *The Journal of Systems and Software*, 24(3):277-294.
- Johnson, R.M. (1974). Trade-off analysis of consumer values. *Journal of Marketing Research*, XI:121-127.
- Jones, C.B. (1986). *Systematic Software Development using VDM*. Prentice-Hall International, London.
- Kabodian, A. (1996). *The Customer Is Always Right!* McGraw-Hill, Inc. Se även <http://www.customersatisfaction.com/book.html>
- Kaindl, H. (1993). The missing link in requirements engineering. *ACM SIGSOFT*, 18(2):30-39.
- Karlsson, J. and Lindvall, M. (1994). Experiences with requirements specification evolution. Memo 95-02, ASLAB, Linköping University.
- Karlsson, J. (1995). Towards a strategy for software requirements selection. Licentiate Thesis 513, Department of Computer and Information Science, Linköping University, Sweden. ([Abstract .ps](#)) ([Abstract .html](#))
- Karlsson, J. (1996). Software Requirements Prioritizing. In *2nd IEEE International Conference on Requirements Engineering*, pp. 100-116.
- Karlsson, J. and Ryan, K. (1996). Supporting the Selection of Software Requirements. In *8th International Workshop on Software Specification and Design*, pp. 146-149.
- Karlsson, J., Olsson, S. and Ryan, K. (1997). Improved Practical Support for Large-scale Requirements Prioritizing. To appear in *Requirements Engineering Journal*
- Karlsson, J. and Ryan, K. Prioritizing Requirements Using a Cost-Value Approach.
- Karlsson, J. Managing Software Requirements Using Quality Function Deployment.
- Karlsson, J., Wohlin, C. and Regnell, B. An Evaluation of Methods for Prioritizing Software Requirements.
- Keller, S.E., Kahn, L.G., and Panara, R.B. (1990). Specifying software quality requirements with metrics. In Thayer, R.H. and Dorfman, M., editors, *System and Software Requirements Engineering*, pages 145-163. IEEE Computer Society Press.

- King, B. (1989). *Better Designs in Half the Time. Implementing QFD in America*. GOAL/QPC, 3rd edition.
- Kotonya, G. and Sommerville, I. (1992). Viewpoints for requirements definition. *Software Engineering Journal*, 7(6):375-387.
- Kramer, J., Ng, K., Potts, C., and Whitehead, K. (1988). Tool support for requirements analysis. *Software Engineering Journal*, 3(3):86-96.
- Kuwana, E. (1993). Representing knowledge in requirements engineering: An empirical study of what software engineers need to know. In *IEEE International Symposium on Requirements Engineering*, pages 273-276. IEEE Computer Society.
- LaBudde, E.V. (1994). Why is requirements engineering underused? *IEEE Software*, 11(2):6-8.
- Lano, R.J. (1990). A structured approach for operational concept formulation. In Thayer, R.H. and Dorfman, M., editors, *System and Software Requirements Engineering*, pages 48-59. IEEE Computer Society Press.
- Lee, M. (1996). *Foundations of the WinWin Requirements Negotiation System*. Ph.D. Dissertation. Technical Report USC/CSE-96-501, Center for Software Engineering, University of Southern California, Los Angeles, CA 90089.
- Lee, M. and Boehm, B.W. (1996). *The WinWin Requirements Negotiation System: A Model-Driven Approach*. Technical Report USC/CSE-96-502, Center for Software Engineering, University of Southern California, Los Angeles, CA 90089.
- Leite, J.C.S.P. (1987). A survey on requirements analysis. Technical Report RTP-071, Department of Information and Computer Science, University of California at Irvine, Irvine, CA 92717.
- Lindstrom, D.R. (1993). Five Ways to Destroy a Development Project. *IEEE Software*, 10(5):55-58.
- Lindvall, M. (1994). *A study of traceability in object-oriented systems development*. Licentiate Thesis 462, Department of Computer and Information Science, Linköping University, Sweden.
- LORAL (1994). Software process improvements and maturity growth. Seminar notes, LORAL Space Information Systems, Houston, Texas.
- Loucopoulos, P. and Champion, R. (1989). Knowledge-based support for requirements engineering. *Information and Software Technology*, 31(3):123-135.
- Loucopoulos, P. and Champion, R. (1990). Concept acquisition and analysis for requirements specification. *Software Engineering Journal*, 5(2):116-124.
- Loucopoulos, P. and Karakostas, V. (1995). *System Requirements Engineering*. McGraw-Hill.
- Lubars, M., Potts, C. and Richter, C. (1993). A Review of the State of the Practice in Requirements Modeling. I *IEEE International Symposium On Requirements Engineering*, pp. 2-14.
- Lutz, R.R. (1993). Analyzing Software Requirements Errors in Safety-Critical, Embedded Systems. I *IEEE International Symposium on Requirements Engineering*, pp. 126-133.
- Löwgren, J. (1993). *Human-Computer Interaction. What every system developer should know*. Studentlitteratur, Lund, Sweden.
- Macaulay, L.A. (1996). *Requirements Engineering*. Springer, London.
- MacLean, A., Young, R.M., and Moran, T.P. (1989). Design rationale: The argument behind the artifact. In *CHI '89 Proceedings*, pages 247-252.
- Maiden, N.A.M. and Rugg, G. (1996). ACRE: Selecting Methods for Requirements Acquisition. *Software Engineering Journal*, 11(3):183-192.
- Maiden, N. A. M., Mistry, P., and Sutcliffe, A. G. (1995). How people categorise requirements for reuse: A natural approach. In *Second IEEE International Symposium on Requirements Engineering*, pages 148-155. IEEE Computer Society Press.

- Martin, J. and Tsai, W. T. (1990). N-Fold Inspection: A Requirements Analysis Technique. *Communications of the ACM*, 33(2):225-232.
- McDermid, J.A., editor (1991). *Software Engineer's Reference Book*. Butterworth-Heinemann, Ltd., Halley Court, Jordan Hill, Oxford OX2 8EJ.
- Meyer, B. (1985). On Formalism in Specifications. *IEEE Software*, 2(1):6-26.
- Mizuno, Y. (1983). Software quality improvement. *IEEE Computer*, 16(3):66-72.
- Mullery, G.P. (1979). CORE - A method for controlled requirements specification. In *Proceeding of the Fourth International Conference on Software Engineering*, pages 126-135. The Institute of Electrical and Electronic Engineers.
- Mylopoulos, J., Chung, L., and Nixon, B. (1992). Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on Software Engineering*, 18(6):483-497.
- Nelsen, E.D. (1990). Systems engineering and requirement allocation. In Thayer, R.H. and Dorfman, M., editors, *System and Software Requirements Engineering*, pages 60-76. IEEE Computer Society Press.
- Ohnishi, A. and Agusa, K. (1993). CARD: A software requirement definition environment. In *IEEE International Symposium On Requirements Engineering*, pages 90-93. IEEE Computer Society Press.
- Palmer, J.D. and Fields, N.A. (1992). An integrated environment for requirements engineering. *IEEE Software*, 9(3):80-85.
- Parnas, D.L. (1977). The Use of Precise Specifications in the Development of Software. I *IFIP Congress 77*, pp. 861-867.
- Pohl, K. (1993). The three dimensions of requirements engineering. In Rolland, C., Bodart, F., and Cauvet, C., editors, *5th International Conference on Advanced Information Systems Engineering*, pages 275-292. Springer-Verlag.
- Porter, A. and Votta, L. (1994). An experiment to assess different defect detection methods for software requirements inspections. In *16th International Conference on Software Engineering*, pages 103-112. IEEE Computer Society Press.
- Poston, R.M. (1987). Preventing most-probable errors in requirements. *IEEE Software*, 4(5):81-83.
- Potts, C. (1991). Seven (plus or minus two) challenges for requirements research. In *Proceedings of the 6th International Workshop on Software Specification and Design*, pages 256-259. IEEE Computer Society Press.
- Potts, C. and Bruns, G. (1988). Recording the reasons for design decisions. In *Proceedings of the 10th International Conference on Software Engineering*, pages 418-426.
- Potts, C., Takahashi, K. and Antón, A. (1994). Inquiry-Based Scenario Analysis of System Requirements, Georgia Institute of Technology
- Prehn, S. and Toetenel, W., editors (1991). VDM'91: formal software development methods: 4th international symposium of VDM Europe. Lecture notes in computer science: 551-552. Springer-Verlag, New York.
- Ramamoorthy, C., Usuda, Y., Prakash, A., and Tsai, W.T. (1990). The evolution support environment system. *IEEE Transactions on Software Engineering*, 16(11):1225-1234.
- Ramesh, B. and Dhar, V. (1992). Supporting systems development by capturing deliberations during requirements engineering. *IEEE Transactions on Software Engineering*, 18(6):498-510.
- Ramesh, B. and Luqi (1993). Process knowledge based rapid prototyping for requirements engineering. In *IEEE International Symposium on Requirements Engineering*, pages 248-255. IEEE Computer Society.

- Regnell, B., Kimbler, K. and Wesslén, A. (1995). Improving the Use Case Driven Approach to Requirements Engineering. I *2nd IEEE International Symposium On Requirements Engineering*, pp. 40-47.
- Robinson, W.N. (1990). Negotiation behaviour during requirements specification. In *12th International Conference on Software Engineering*, pages 268-276. IEEE Computer Society Press.
- Roman, G.-C. (1985). A taxonomy of current issues in requirements engineering. *IEEE Computer*, 18(4):14-21.
- Ross, D.T. and Schoman Jr., K.E. (1977). Structured analysis for requirements definition. *IEEE Transactions on Software Engineering*, SE-3(1):6-15.
- Royce, W. W. (1975). Software requirements analysis: Sizing and costing. In Horowitz, E., editor, *Practical Strategies for Developing Large Software Systems*, pages 57-71. Addison-Wesley Publishing Company.
- Royce, W.W. (1970). Managing the development of large software systems: concepts and techniques. In *Proceedings IEEE WESCON*, pages 1-9.
- Rumbaugh, J. (1994). Getting started: Using use cases to capture requirements. *Journal of Object-Oriented Programming*, September issue, pp. 8- 12, 23.
- Russell, G. W. (1991). Experience with Inspection in Ultralarge-scale Developments. *IEEE Software*, 8(1):25-31.
- Ryan, K. (1993). The role of natural language in requirements engineering. In *IEEE International Symposium On Requirements Engineering*, pages 240-242. IEEE Computer Society Press.
- Ryan, K. and Karlsson, J. (1997). Prioritizing Software Requirements in an Industrial Setting. To appear in *International Conference on Software Engineering (ICSE'97)*.
- Rzepka, W. and Ohno, Y. (1985). Requirements engineering environments: Software tools for modeling user needs. *IEEE Computer*, 18(4):9-12.
- Sacha, K.M. (1992). Transnet approach to requirements specification and prototyping. Source unknown.
- Saeki, M., Horai, H., and Enomoto, H. (1989). Software development process from natural language specification. In *11th International Conference on Software Engineering*. IEEE Computer Society Press.
- Sandahl, K., Karlsson, J., Krysanter, C., Lindvall, M. and Ohlsson, N. A Cross-cultural Replication of an Experiment For Assessing Methods for Software Requirements Inspections.
- Scharer, L. (1990). Pinpointing requirements. In Thayer, R.H. and Dorfman, M., editors, *System and Software Requirements Engineering*, pages 17-22. IEEE Computer Society Press.
- Schneider, G. M., Martin, J. and Tsai, W. T. (1992). An Experimental Study of Fault Detection in User Requirements Documents. *ACM Transactions on Software Engineering and Methodology*, 1(2):188-204.
- Schulmeyer, G.G. (1990). *Zero Defect Software*. McGraw-Hill.
- Senge, P.M. (1990). *The Fifth Discipline: The Art & Practice of The Learning Organization*. Doubleday.
- Shumate, K. and Keller, M. (1992). *Software specification and design*. John Wiley & Sons Ltd.
- Sibley, E.H., Wexelblat, R.L., Michael, J.B., Tanner, M.C., and Littman, D.C. (1993). The role of policy in requirements definition. In *IEEE International Symposium on Requirements Engineering*, pages 277-280. IEEE Computer Society.
- Ppdiqui, J. (1994). Challenging universal truths of requirements engineering. *IEEE Software*, 11(2):18-19.

- Ppdiqi, J. and Shekaran, M.C. (1996). Requirements Engineering: The Emerging Wisdom. *IEEE Software*, 13(2):15-19.
- Singer, C.A. (1992). A Requirements Tutorial. Quality Systems and Software Requirements. Special Report SR-NWT-002159, Bellcore.
- Sommerville, J. (1996). *Software Engineering*. Addison-Wesley Publishing Company, 5th edition.
- Spence, I. and Carey, B. (1991). Customers do not want frozen specifications. *Software Engineering Journal*, 6(4):175-180.
- Spivey, J. (1992). *The Z Notation: A Reference Manual*. Prentice-Hall International, Inc., New York, 2nd edition.
- Strauss, S.H. and Ebenau, R.G. (1994). *Software Inspection Process*. McGraw-Hill, Inc.
- Sullivan, L. (1986). Quality Function Deployment. *Quality Progress*, June issue, pp. 39-50.
- Swartout, W. and Balzer, R. (1982). On the Inevitable Intertwining of Specification and Implementation. *Communications of the ACM*, 25(7):438-440.
- Tamai, T. and Itou, A. (1993). Requirements and design change in large-scale software development: Analysis from the viewpoint of process backtracking. In 15th International Conference on Software Engineering, pages 167-176, Los Alamitos, California. IEEE Computer Society Press.
- Thayer, R.T., Pyster, A., and Wood, R.C. (1980). The challenge of software engineering management. *IEEE Computer*, 13(8):51-59.
- van Vliet, H. (1993). *Software Engineering: Principles and Practice*. John Wiley & Sons Ltd.
- Wang, W., Hufnagel, S., Hsia, P., and Yang, S.M. (1992). Scenario driven requirements analysis method. In 2nd International conference on systems integration, pages 127-136.
- Wangler, B. (1993). Contributions to Functional Requirements Modelling. PhD thesis, Stockholm University.
- Wardle, P. (1991). Methodology and tools for requirements capture, traceability and verification. In International conference on software engineering for real-time systems, pages 46-50.
- Watkins, R. and Neal, M. (1994). Why and how of requirements tracing. *IEEE Software*, 11(4):104-106.
- VI (1995a). *Kontinuerlig förbättring av programvaruprocessen via mognadsmodeller - erfarenheter från industrin*. Sveriges Verkstadsindustrier, Rapport Informationsteknologi.
- VI (1995b). *Krav påKrav*. Sveriges Verkstadsindustrier, Rapport Informationsteknologi.
- Wiktorin, L. (1994). Programvarusystem. Kortade ledtider and ökad produktivitet. [software systems. Decreased lead times and increased productivity]. Technical Report V040016, Sveriges Verkstadsindustrier, Box 5510, 114 85 Stockholm, Sweden.
- Wing, J.M. (1990). A specifier's introduction to formal methods. *IEEE Computer*, 23(9):8-24.
- Wright, S. (1991). Requirements traceability - what? why? and how? In Colloquium On "Tools and Techniques For Maintaining Traceability During Design". The institution of electrical engineers.
- Yadav, S.B., Bravoco, R.R., Chatfield, A.T., and Rajkumar, T.M. (1988). Comparison of analysis techniques for information requirement determination. *Communications of the ACM*, 31(9):1090-1097.
- Yeh, R.T. and Ng, P.A. (1990). Software requirements - a management perspective. In Thayer, R.H. and Dorfman, M., editors, *System and Software Requirements Engineering*, pages 450-461. IEEE Computer Society Press.
- Yeh, R.T. and Zave, P. (1980). Specifying Software Requirements. *Proceedings of the IEEE*, 68(9):1077-1085.

Yourdon, E. (1989). Modern Structured Analysis. Prentice-Hall International, Inc., Englewood Cliffs, New Jersey.

Yue, K. (1987). What does it mean to say that a specification is complete? In 4th International workshop on Software Specification and Design, pages 42-49. IEEE Computer Society Press.

Zave, P. (1979). A comprehensive approach to requirements problems. In International computer software and applications conference, COMPSAC 3, pages 117-122, Chicago, IL.

Zave, P. (1995). Classification of research efforts in requirements engineering. In Second IEEE International Symposium on Requirements Engineering, pages 214-216. IEEE Computer Society Press.

Zultner, R. E. (1990). Objectives: The missing piece of the requirements puzzle. In Structured Development Forum XI, pages 1-11.

Journals and newsletters

Requirements Engineering Journal. www.mac.co.umist.ac.uk/RE/Journal.html

Requirements Engineering Newsletter. web.cs.city.ac.uk/homes/acwf/rehome.html

Requirements Engineering Student Newsletter.
www.cc.gatech.edu/computing/SW_Eng/resnews.html

Requiraonautics Quarterly. research.ivv.nasa.gov/~steve/resq/

3rd IEEE International Symposium on Requirements Engineering.
www.itd.nrl.navy.mil/conf/ISRE97/

Requirements Engineering and Analysis Workshop Proceedings.
ftp.sei.cmu.edu/pub/documents/91.reports/tr30.91.ps

Requirements engineering and the CMM
rbse.isc.nasa.gov:80/process_maturity/CMM/TR25/tr25_l2a.html

Quality Function Deployment
www.nauticom.net/www/qfdi/
mijuno.larc.nasa.gov/dfc/qfd.html

Research
www.cc.gatech.edu/computing/SW_Eng/reqts.html
www.tts.lth.se/Personal/bjornr/
www.ida.liu.se/labs/aslab/people/loaka/
www.ifi.unizh.ch/groups/req/index.html
www.comlab.ox.ac.uk/oucl/crf.html

Bibliographies
www.comp.mpce.mq.edu.au/~didar/seweb/bibliography.html
mozart.uccs.edu/adavis/reqbib.html

Discussion group
<http://www-comp.mpce.mq.edu.au/~didar/seweb/groups.html>