



**Estudo sobre a aderência da disciplina de Requisitos da metodologia RUP Rational Unified Process ao modelo CMMI Capability Maturity Model Integration**

SÃO PAULO

2011



**Estudo sobre a aderência da disciplina de Requisitos da metodologia RUP Rational Unified Process ao modelo CMMI Capability Maturity Model Integration**

Rodrigo Marques de Messias

Monografia apresentada à Faculdade de Tecnologia de São Paulo para a obtenção de Grau de Tecnólogo em Processamento de Dados

Orientador: Carlos Hideo Arima

SÃO PAULO

2011

Eu dedico este trabalho aos meus pais, que  
sempre me apoiaram,

a minha irmã, que vem descobrindo a arte  
de ser adulta,

a minha namorada, que me dá forças  
quando mais preciso,

e a todos que sempre torceram por mim.

*“O que é errado é errado, mesmo que todos  
o façam. O que é certo é certo, mesmo que  
ninguém o faça”*

*(Sri Sathya Sai Baba)*

## Sumário

<b>LISTA DE FIGURAS</b> .....	v
<b>LISTA DE QUADROS</b> .....	vi
<b>LISTA DE ABREVIATURAS</b> .....	vii
<b>CURRICULUM VITAE</b> .....	viii
<b>RESUMO</b> .....	ix
<b>ABSTRACT</b> .....	x
<b>1. INTRODUÇÃO</b> .....	1
<b>1.1. Identificação do Problema</b> .....	2
<b>1.2. Objetivo do Trabalho</b> .....	2
<b>1.3. Justificativa</b> .....	3
<b>1.4. Metodologia de Pesquisa Aplicada</b> .....	3
<b>1.5. Estrutura do Trabalho</b> .....	3
<b>2. A METODOLOGIA RUP E O MODELO CMMI</b> .....	5
<b>2.1. RUP</b> .....	5
<b>2.2. CMMI</b> .....	9
<b>2.2.1. Gerenciamento de Requisitos</b> .....	13
<b>2.2.2. Desenvolvimento de Requisitos</b> .....	19
<b>3. ADERÊNCIA DA METODOLOGIA RUP AO MODELO CMMI</b> .....	37
<b>3.1. Técnicas de Verificação de Aderência</b> .....	37
<b>3.2. Práticas de Gerenciamento de Requisitos</b> .....	37
<b>3.3. Práticas de Desenvolvimento de Requisitos</b> .....	40
<b>3.4. Avaliação de Aderência</b> .....	46
<b>4. CONCLUSÃO</b> .....	48
<b>REFERÊNCIAS</b> .....	49
<b>BIBLIOGRAFIA CONSULTADA</b> .....	49

## LISTA DE FIGURAS

<b>Figura 1: RUP - Gráfico de Baleias .....</b>	<b>6</b>
<b>Figura 2: RUP – Fluxo de Trabalho da Disciplina de Teste .....</b>	<b>7</b>
<b>Figura 3: CMMI – Níveis de Maturidade .....</b>	<b>10</b>

**LISTA DE QUADROS**

<b>Quadro 1: CMMI – Níveis de Maturidade x Áreas de Processo .....</b>	<b>12</b>
<b>Quadro 2: CMMI – Gerenciamento de Requisitos .....</b>	<b>14</b>
<b>Quadro 3: CMMI – Desenvolvimento de Requisitos .....</b>	<b>22</b>
<b>Quadro 4: Modelo CMMI x Metodologia RUP.....</b>	<b>46</b>

**LISTA DE ABREVIATURAS**

<b>RUP</b>	<b>Rational Unified Process</b>
<b>CMMI</b>	<b>Capability Maturity Model Integration</b>
<b>SEI</b>	<b>Software Engineering Institute</b>
<b>REQM</b>	<b>Requirements Management</b>
<b>RD</b>	<b>Requirements Development</b>
<b>CUA</b>	<b>Common User Access</b>
<b>GUI</b>	<b>Graphical User Interface</b>
<b>SRS</b>	<b>Software Requirements Specification</b>

## **CURRICULUM VITAE**

Rodrigo Marques, nascido em 20 (vinte) de outubro de 1988 (mil novecentos e oitenta e oito), natural de São Paulo, Residente na Rua Alto do Parnaíba, número 623 (seiscentos e vinte e três) A, São Paulo, está cursando o último semestre do curso de Processamento de Dados, período noturno, da Faculdade de Tecnologia de São Paulo.

Atualmente ocupa o cargo de Analista de Requisitos na IBM Brasil. Possui conhecimentos em modelagem de negócio, gerenciamento e elicitação de requisitos e análise de sistemas. Certificado em RUP - Rational Unified Process v7.

## RESUMO

Este trabalho tem como objetivo verificar o nível de aderência da metodologia RUP em relação ao modelo CMMI com foco na área de Requisitos. São apresentadas as práticas das áreas de processo Gerenciamento de Requisitos e Desenvolvimento de Requisitos do modelo CMMI e a partir destas é avaliado o quanto as tarefas contidas na disciplina de Requisitos da metodologia RUP cumprem as exigências de maturidade propostas. Por meio de métodos indutivos, com uma abordagem qualitativa do problema, o objetivo é realizar uma pesquisa bibliográfica de natureza descritiva, a fim de chegar a um consenso sobre a relação aos dois produtos avaliados. Como resultado do estudo, pode-se considerar que a metodologia RUP, por ser adaptável a maioria das situações estabelecidas pelo modelo CMMI, pode ser considerada uma metodologia aderente ao modelo, pelo menos no que diz respeito à área de Requisitos.

## **ABSTRACT**

This work aims to verify the RUP methodology adherence level in relation to CMMI model, focusing on Requirements Management and Requirements Development practices of CMMI model are presented and assessed considering how much the tasks in the Requirements discipline of RUP methodology meet the proposed matureness requirements. By inductive methods, with a qualitative approach to the problem, the objective is to perform a bibliographic research with a descriptive nature, in order to reach a consensus about the relationship of the evaluated products. As result of the study, it can be considered that the RUP methodology can be adaptable to most situations established by the CMMI model and may be considered an adherent methodology to the model, at least regarding Requirements area.

## 1. INTRODUÇÃO

Com o cenário de transformações constantes do mercado mundial, a temática prioritária no campo empresarial passou a ser a competitividade. A necessidade de se impor em um mercado globalizado, sem fronteiras, faz com que empresas vivam um processo contínuo de transformação. Diante desse contexto, os grandes desafios estão na busca da garantia da qualidade e produtividade. No âmbito da tecnologia da informação essa realidade não é diferente.

O objetivo do CMMI – *Capability Maturity Model Integration* é fornecer direcionamentos para melhorar os processos da organização e a capacidade de gerenciar o desenvolvimento, aquisição e manutenção de produtos e serviços. O CMMI coloca abordagens comprovadas em uma estrutura que ajuda a organização a avaliar a maturidade organizacional ou a capacitação da área de processo, estabelecer prioridades de melhoria e implementá-las.

A conformidade com esse modelo, no entanto, exige um grande esforço na definição de procedimentos e padrões, e envolve a implantação de uma metodologia de desenvolvimento que descreva as atividades, produtos e responsáveis pelas diversas etapas do ciclo de vida do *software*. Neste sentido, metodologias orientadas a objetos, que estimulem um desenvolvimento iterativo e incremental, com foco em análise de riscos, têm se mostrado convenientes e úteis.

O RUP – *Rational Unified Process* é um processo de engenharia de *software* cujas principais características são um desenvolvimento iterativo e incremental, orientado a objetos, com foco na criação de uma arquitetura robusta, análise de riscos e utilização de casos de uso para o desenvolvimento. O RUP foi desenvolvido para ser

aplicável a uma grande classe de projetos diferentes e pode ser considerado como um *framework* genérico para processos de desenvolvimento.

Isso significa que ele deve ser configurado para ser usado eficientemente. A configuração pode ser feita para empresas (para definir o processo padrão de desenvolvimento da empresa) ou mesmo para projetos específicos e normalmente envolve remoção e/ou modificação de atividades do *framework*.

### **1.1. Identificação do Problema**

Num ambiente corporativo com variadas técnicas e metodologias de desenvolvimento de *software*, a busca de uma forma de trabalho que deixe a organização confortável e seus produtos confiáveis se torna um dos pontos chaves na estratégia organizacional. Nesse sentido, o RUP vem como umas das opções mais plausíveis, pois é uma metodologia comprovadamente eficaz para projetos de qualquer grandeza. Mas será que as técnicas aplicadas na metodologia RUP são consideradas maduras o suficiente para propiciar a organização o selo CMMI?

### **1.2. Objetivo do Trabalho**

Este trabalho tem como objetivo analisar a metodologia de desenvolvimento de *software* RUP e verificar se o conteúdo da disciplina de requisitos esta em conformidade com as práticas encontradas nas áreas de processo relacionadas a requisitos do modelo CMMI.

### **1.3. Justificativa**

Este estudo pode auxiliar as empresas a buscarem uma metodologia de desenvolvimento mais adequada, que seja capaz de garantir a qualidade na entrega do produto final, além de estar de acordo com um dos modelos de maturidade mais reconhecido no mercado na atualidade.

### **1.4. Metodologia de Pesquisa Aplicada**

De acordo com Diehl (2004), pode-se fazer a classificação de pesquisas em algumas categorias. No que diz respeito ao método de investigação, foi utilizado o método indutivo, onde as constatações particulares levam à elaboração de generalizações. Em relação aos objetivos, considera-se como sendo uma pesquisa descritiva, onde se faz um levantamento da relação entre os dois produtos. Quanto à abordagem do problema, utiliza-se a qualitativa, pois não há medições nem métodos estatísticos. O propósito é gerar novos conhecimentos sobre a utilização do RUP, sem aplicação prática, portanto pode-se classificar como sendo uma pesquisa básica, onde são analisados os materiais já existentes sobre o assunto, aplicando procedimentos técnicos de uma pesquisa bibliográfica.

### **1.5. Estrutura do Trabalho**

Este trabalho está dividido em três partes: revisão bibliográfica do RUP e CMMI, uma comparação entre os dois produtos e a conclusão da pesquisa.

No capítulo 1 apresenta uma contextualização do tema escolhido e o problema a ser explorado. A partir do problema, estabelece o objetivo do trabalho bem como a motivação para fazê-lo. A metodologia aplicada também é apresentada para um melhor entendimento das técnicas utilizadas.

No capítulo 2 é apresentado um *overview* do RUP e como estão estruturadas as disciplinas dessa metodologia. O modelo CMMI também é apresentado e as metas e práticas das áreas de processo exploradas são detalhadas.

No capítulo 3 descreve a avaliação de conformidade entre o RUP e CMMI. A disciplina de Requisitos do RUP é analisada para medir o nível de aderência das tarefas em relação às práticas das áreas de processo de Gerenciamento de Requisitos e Desenvolvimento de Requisitos do modelo CMMI.

No capítulo 4 encerra-se o estudo com os resultados obtidos através da pesquisa realizada.

## 2. A METODOLOGIA RUP E O MODELO CMMI

Para que se possa fazer a comparação entre o conteúdo de dois produtos, é necessário primeiro entendê-los. Traçaremos um overview da metodologia de desenvolvimento de *software* RUP para conhecer as disciplinas que fazem parte deste *framework* e como estas estão estruturadas. Também iremos expor a estrutura do modelo de maturidade CMMI e como são representadas as áreas de processo.

### 2.1. RUP – *Rational Unified Process*

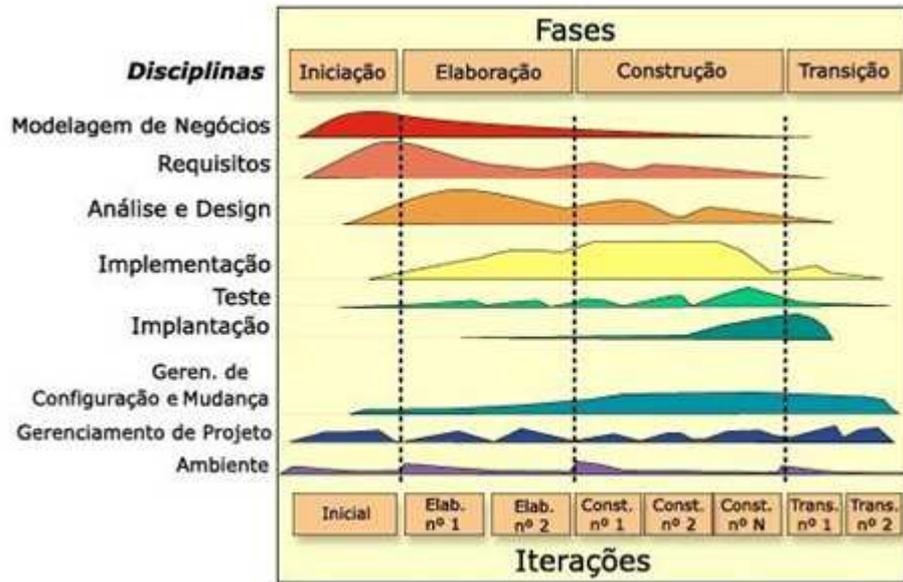
O RUP - *Rational Unified Process* é um processo de engenharia de *software* desenvolvido pela Rational Software Corporation, cujas principais características são um desenvolvimento iterativo e incremental, orientado a objetos, com foco na criação de uma arquitetura robusta, análise de riscos e utilização de casos de uso para o desenvolvimento.

O RUP foi desenvolvido para ser aplicável a classes de projetos diferentes e pode ser considerado um *framework* genérico para processos de desenvolvimento. Isso significa que ele deve ser configurado para ser usado eficientemente. A configuração pode ser feita para empresas (para definir o processo padrão de desenvolvimento da empresa) ou para projetos específicos, onde normalmente envolve remoção e/ou modificação de atividades do *framework*.

O RUP tem duas dimensões, como mostra a Figura 1:

- o eixo horizontal representa o tempo e mostra as fases do ciclo de vida do processo à medida que se desenvolve;

- o eixo vertical representa as disciplinas, onde são agrupadas as atividades de maneira lógica.



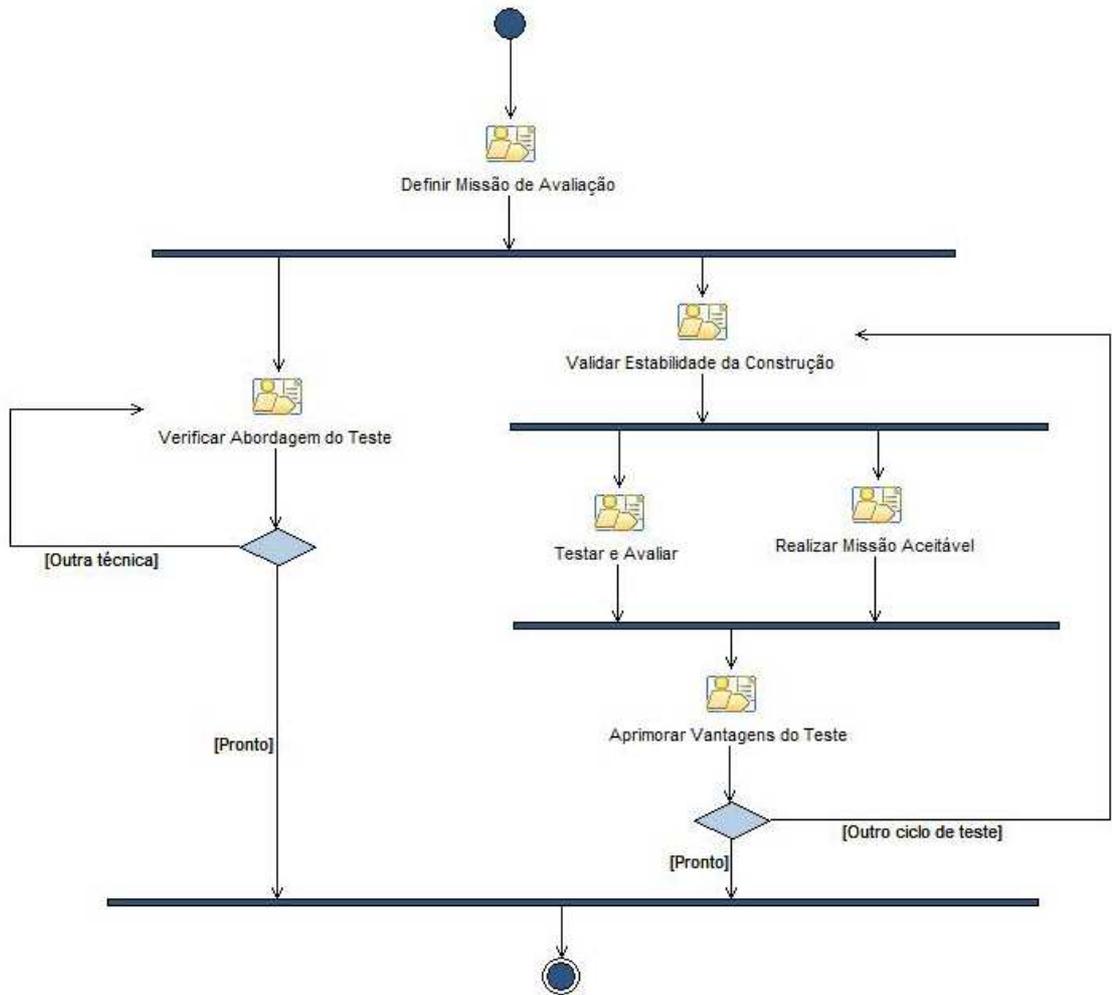
**Figura 1:** RUP – Gráfico de Baleias

*Fonte: IBM – Metodologia RUP*

Cada disciplina possui atividades específicas, formadas por um conjunto de tarefas executadas por recursos, sendo estas tarefas geradoras de artefatos ou não. Para as atividades, temos quatro seções:

- Descrição – contém o objetivo e o contexto geral da atividade, com relacionamentos e definição dos recursos que executarão as tarefas;
- Estrutura de Decomposição de Trabalho – contém uma lista das tarefas relacionadas à atividade e suas informações principais;
- Alocação da Equipe – contém os recursos que executarão as tarefas e com quais produtos de trabalho eles se relacionam;
- Uso de Produto de Trabalho – contém todos os produtos de trabalho relacionados a alguma tarefa referente à atividade.

O Fluxo de Trabalho de uma disciplina é representado por atividades a serem realizadas, conforme Figura 2.



**Figura 2:** RUP – Fluxo de Trabalho da Disciplina de Teste

*Fonte: IBM – Metodologia RUP*

Para cada atividade, temos tarefas específicas, com recursos, etapas e produtos de trabalho relacionados. A atividade Definir Missão de Avaliação, por exemplo, possui as seguintes tarefas a serem executadas:

- Identificar Motivadores de Teste
- Concordar com a Missão
- Identificar Objetivos do Teste
- Definir Necessidade de Avaliação e Rastreabilidade
- Identificar Idéias de Teste
- Definir Abordagem do Teste

Dentro de cada tarefa, existem etapas, que descrevem como e por quem as tarefas devem ser executadas. Ainda na atividade Definir Missão de Avaliação, a tarefa Identificar Motivadores de Teste contém as seguintes etapas:

- Identificar Itens de Destino de Iteração
- Reunir e Examinar Informações Relacionadas
- Identificar Sugestões de Motivadores
- Determinar Riscos de Qualidade
- Definir a Lista de Motivadores
- Manter Relacionamentos de Rastreabilidade
- Avaliar e Verificar os Resultados

No detalhe das etapas de trabalho, o RUP descreve como aquela etapa deverá ser executada para atingir os objetivos da atividade em que ela se encontra. Além disso, ainda é contemplada a visão de quais produtos de trabalho serão necessários para executar a tarefa, e quais serão obrigatórios serem desenvolvidos como saída da execução da tarefa. Os recursos responsáveis por executar as tarefas e desenvolver os artefatos de saída também são relacionados na visão de Alocação da Equipe.

Para este trabalho, analisa-se somente a disciplina de Requisitos da metodologia RUP.

## **2.2. CMMI – *Capability Maturity Model Integration***

O CMMI - *Capability Maturity Model Integration* é um guia de melhoria de processos desenvolvido pela SEI (*Software Engineering Institute*) em parceria com a Carnegie Mellon University.

O objetivo do CMMI é fornecer direcionamentos para melhorar os processos da organização e sua capacidade de gerenciar o desenvolvimento, aquisição e manutenção de produtos e serviços. O CMMI coloca abordagens comprovadas em uma estrutura que ajuda a organização a avaliar a maturidade organizacional ou a capacitação da área de processo, estabelecer prioridades de melhoria e implementá-las.

Para a Carnegie Mellon University e a Software Engineering Institute (2010), existem duas formas de representação para o CMMI: contínua e por estágios. Na representação contínua, os níveis de maturidade sugerem uma seqüência para melhoria de processos em cada área. Ao mesmo tempo, a representação contínua permite alguma flexibilidade em relação à ordem em que as áreas de processo são abordadas. A representação em estágios agrupa as áreas de processos por nível de maturidade, indicando quais áreas de processos deverão ser implementadas para atingir cada um dos cinco níveis de maturidade. Os níveis de maturidade representam um caminho de melhoria de processos ilustrando a evolução da melhoria para a organização como um todo.

Um nível de maturidade é uma etapa evolucionária definida da melhoria de processos. Cada nível de maturidade estabiliza uma parte importante dos processos da organização. Nos modelos CMMI com representação em estágios, existem cinco níveis de maturidade, cada um representando uma camada da base da melhoria de processos, definidos pelos números de 1 a 5, como podemos ver na Figura 3.



**Figura 3:** CMMI – Níveis de Maturidade

*Fonte: ISD Brasil*

No nível de maturidade 1, os processos são informais e caóticos. As organizações normalmente não possuem um ambiente estável e são caracterizadas por uma tendência a não cumprir compromissos, abandonar processos em momentos de crises e não serem capazes de repetir sucessos do passado.

No nível de maturidade 2, a organização atingiu todas as metas específicas e genéricas das áreas de processos do nível 2 de maturidade. Os projetos da organização asseguraram que os requisitos são gerenciados e que os processos são planejados, executados, medidos e controlados.

No nível de maturidade 3, uma organização atingiu todas as metas específicas e genéricas das áreas de processos definidas para os níveis de maturidade 2 e 3. Os processos são bem caracterizados e entendidos e estão descritos em padrões, procedimentos, ferramentas e métodos.

No nível de maturidade 4, uma organização atingiu todas as metas específicas das áreas de processos atribuídas aos níveis de maturidade 2, 3 e 4 e as metas genéricas atribuídas aos níveis de maturidade 2 e 3. São selecionados os subprocessos que contribuem significativamente para o desempenho geral do processo. Estes subprocessos selecionados são controlados utilizando estatísticas e outras técnicas quantitativas.

No nível de maturidade 5, uma organização atingiu todas as metas específicas das áreas de processos atribuídas aos níveis de maturidade 2, 3, 4 e 5 e as metas genéricas atribuídas aos níveis de maturidade 2 e 3. Os processos são continuamente melhorados com base em um entendimento quantitativo das causas comuns de variações inerentes aos processos.

A partir do nível 2, para todos os níveis de maturidade, existem metas e práticas genéricas, as quais todas as áreas de processos precisam estar em conformidade, e metas e práticas específicas de cada área de processo. Cada nível de maturidade possui áreas de processos diferentes, relacionadas aos objetivos daquele nível. Podemos visualizar os níveis de maturidade e suas áreas de processo através do Quadro 1.

**Quadro 1: CMMI – Níveis de Maturidade x Áreas de Processo**

Nível		Área de Processo
5	Otimizado	Análise de Causas e Resoluções
		Inovação e Desenvolvimento Organizacional
4	Gerenciado Quantitativamente	Desempenho do Processo Organizacional
		Gerenciamento Quantitativo de Projeto
3	Definido	Desenvolvimento de Requisitos
		Soluções Técnicas
		Integração de Produtos
		Verificação
		Validação
		Foco no Processo Organizacional
		Definição do Processo Organizacional
		Treinamento Organizacional
		Gerenciamento Integrado do Projeto
		Gerenciamento de Riscos
2	Gerenciado	Análises de Decisões e Resoluções
		Gerenciamento de Requisitos
		Planejamento do Projeto
		Monitoramento e Controle do Projeto
		Gerenciamento de Acordos com Fornecedores
		Medições e Análises
		Garantia da Qualidade do Processo e do Produto
Gerenciamento de Configurações		
1	Inicial	

*Fonte: Elaborado pelo pesquisador*

Para este trabalho, analisaremos somente as seguintes áreas de processos:

- Gerenciamento de Requisitos (nível 2 de maturidade)
- Desenvolvimento de Requisitos (nível 3 de maturidade)

### 2.2.1. Gerenciamento de Requisitos

O objetivo da área de processo Gerenciamento de Requisitos (REQM - *Requirements Management*) é gerenciar os requisitos dos produtos e componentes de produtos do projeto e identificar as inconsistências entre estes requisitos e os planos e os produtos de trabalho do projeto.

Os processos de gerenciamento de requisitos gerenciam todos os requisitos recebidos ou gerados pelo projeto, incluindo requisitos técnicos e não técnicos, bem como os requisitos impostos no projeto pela organização. Em particular, se a área de processo de Desenvolvimento de Requisitos estiver implementada, seus processos gerarão os requisitos de produtos e componentes de produtos que também serão gerenciados pelos processos de gerenciamento de requisitos.

O projeto executa os passos apropriados para assegurar que o conjunto de requisitos acordados é gerenciado para suportar as necessidades de planejamento e execução do projeto. Quando um projeto recebe requisitos de um gerador aprovado de requisitos, estes são revisados com o fornecedor de requisitos para resolver questões e prevenir o mau entendimento, antes que os requisitos sejam incorporados aos planos do projeto. Uma vez que o fornecedor e o receptor dos requisitos cheguem a um acordo, é obtido um compromisso dos participantes do projeto sobre os requisitos. O projeto gerencia as mudanças feitas nos requisitos conforme eles evoluem e identifica quaisquer inconsistências que ocorram entre planos, produtos de trabalho e requisitos.

Parte do gerenciamento de requisitos é documentar as mudanças nos requisitos e suas justificativas, e manter a rastreabilidade bidirecional entre os requisitos fonte e todos os requisitos de produtos e componentes de produtos.

Para a área de processo Gerenciamento de Requisitos, apresenta as metas e práticas específicas de acordo com o Quadro 2.

**Quadro 2: CMMI – Práticas de Gerenciamento de Requisitos**

Gerenciamento de Requisitos	
Metas	Práticas
Gerenciar Requisitos	Obter um Entendimento dos Requisitos
	Obter Compromisso sobre os Requisitos
	Gerenciar Mudanças nos Requisitos
	Manter a Rastreabilidade Bidirecional dos Requisitos
	Identificar Inconsistências entre o Trabalho do Projeto e os Requisitos

*Fonte: Elaborado pelo pesquisador*

### **Meta: Gerenciar Requisitos**

A meta Gerenciar Requisitos tem por objetivo gerenciar os requisitos e encontrar as inconsistências entre os requisitos e os planos do projeto e ou produtos de trabalho. O projeto mantém um conjunto atualizado e aprovado de requisitos durante toda a vida do projeto através de:

- Gerenciamento de todas as mudanças nos requisitos
- Manutenção dos relacionamentos entre os requisitos, planos de projeto e produtos de trabalho
- Identificação de inconsistências entre os requisitos, os planos de projetos e os produtos de trabalho
- Tomada de ações corretivas

### **Prática: Obter um Entendimento dos Requisitos**

A prática Obter um Entendimento dos Requisitos tem por objetivo desenvolver um entendimento com os fornecedores dos requisitos sobre o significado dos requisitos.

Para evitar que os requisitos cresçam indistintamente, são estabelecidos critérios para determinar os canais apropriados ou fontes oficiais, a partir dos quais deve-se receber os requisitos. As atividades de recebimento executam análises dos requisitos junto ao fornecedor dos requisitos para assegurar que um entendimento compartilhado e compatível do significado dos requisitos foi conseguido. O resultado desta análise e diálogo é um conjunto de requisitos acordados.

As subpráticas encontradas nesta prática são:

1. Estabelecer critérios para distinguir os fornecedores apropriados de requisitos.
2. Estabelecer critérios objetivos para a aceitação de requisitos.

A falta de critérios de aceitação muitas vezes resulta numa verificação inadequada, um caro retrabalho ou a rejeição do cliente. Alguns critérios de aceitação de requisitos são:

- Clara e apropriadamente declarados
  - Completos
  - Consistentes uns com os outros
  - Identificados de forma única
  - Apropriados para serem implementados
  - Verificáveis (podem ser testados)
  - Rastreáveis
3. Analisar os requisitos para assegurar que os critérios estabelecidos estão sendo atendido.
  4. Chegar a um entendimento dos requisitos com o fornecedor dos requisitos, de forma que os participantes do projeto possam estabelecer compromissos com relação a eles.

**Prática: Obter Compromissos com os Requisitos**

A prática Obter Compromissos com os Requisitos tem por objetivo obter dos participantes do projeto compromissos com os requisitos. Enquanto a prática Obter um Entendimento dos Requisitos trata de se chegar a um entendimento com os fornecedores de requisitos, esta prática específica trata de acordos e compromissos entre aqueles que terão que executar as atividades necessárias para implementar os requisitos. Conforme os requisitos evoluem, esta prática específica assegura que os participantes do projeto se comprometem com os requisitos atuais aprovados e com as mudanças resultantes nos planos de projeto, atividades e produtos de trabalho.

As subpráticas encontradas nesta prática são:

1. Analisar o impacto dos requisitos nos compromissos existentes.

O impacto nos participantes do projeto deverá ser avaliado quando os requisitos mudarem ou no início de um novo requisito.

2. Negociar e registrar compromissos.

As mudanças nos compromissos existentes deverão ser negociadas antes dos participantes do projeto assumir compromissos com o requisito ou com a mudança de requisitos.

**Prática: Gerenciar as Mudanças de Requisitos**

A prática Gerenciar as Mudanças de Requisitos tem por objetivo gerenciar as mudanças nos requisitos durante o projeto. Durante o projeto, os requisitos mudam por uma série de motivos. Conforme as necessidades mudam e o trabalho prossegue, requisitos adicionais são derivados e mudanças podem ter que ser feitas nos requisitos já existentes. É essencial gerenciar esses acréscimos e mudanças de forma eficiente e

eficaz. Para analisar de forma eficaz o impacto das mudanças, é necessário que a fonte de cada requisito seja conhecida e que a justificativa para qualquer mudança seja documentada. O gerente do projeto pode, entretanto, desejar rastrear as medidas adequadas de volatilidade de requisitos para julgar se são necessários novos controles ou a revisão dos já existentes.

As subpráticas encontradas nesta prática são:

1. Capturar todos os requisitos e mudanças de requisitos que foram recebidas ou geradas pelo projeto.
2. Manter o histórico das mudanças nos requisitos juntamente com as justificativas para as mudanças.

Manter o histórico de mudanças auxilia a rastrear a volatilidade dos requisitos.

3. Avaliar o impacto das mudanças nos requisitos a partir do ponto de vista dos *stakeholders* relevantes.
4. Tornar disponíveis para o projeto os dados de requisitos e de mudanças.

### **Prática: Manter a Rastreabilidade Bidirecional de Requisitos**

A prática Manter a Rastreabilidade Bidirecional de Requisitos tem por objetivo manter a rastreabilidade bidirecional entre os requisitos e os planos do projeto e produtos de trabalho. A intenção desta prática específica é manter a rastreabilidade bidirecional dos requisitos para cada nível de decomposição do produto. Quando os requisitos são bem gerenciados, a rastreabilidade pode ser estabelecida, desde um requisito fonte até seus requisitos de mais baixo nível e destes de volta para o seu requisito fonte. Tal rastreabilidade bidirecional auxilia a determinar se todos os requisitos fonte foram completamente tratados e se todos os requisitos de mais baixo

nível podem ser rastreados para uma fonte válida. A rastreabilidade de requisitos pode também cobrir os relacionamentos com outras entidades, como produtos de trabalho intermediários e finais, mudanças na documentação de *design*, planos de testes e tarefas de trabalho. A rastreabilidade deverá cobrir os relacionamentos horizontais e verticais, como as *interfaces*. A rastreabilidade é particularmente necessária na condução da análise do impacto de mudanças de requisitos nos planos do projeto, atividades e produtos de trabalho.

As subpráticas encontradas nesta prática são:

1. Manter a rastreabilidade dos requisitos para assegurar que a fonte dos requisitos de mais baixo nível (derivados) está documentada.
2. Manter a rastreabilidade dos requisitos até seu requisito derivado, bem como a sua alocação de funções, objetos, pessoas, processos e produtos de trabalho.
3. Manter a rastreabilidade horizontal de função para função e entre *interfaces*.
4. Gerar a matriz de rastreabilidade de requisitos.

### **Prática: Identificar Inconsistências entre o Trabalho do Projeto e os Requisitos**

A prática Manter Identificar Inconsistências entre o Trabalho do Projeto e os Requisitos tem por objetivo identificar inconsistências entre os planos do projeto e produtos de trabalho do projeto e os requisitos. Esta prática específica descobre as inconsistências entre os requisitos e os planos do projeto e produtos de trabalho e inicia a ação corretiva para corrigi-las.

As subpráticas encontradas nesta prática são:

1. Revisar os planos do projeto, atividades e produtos de trabalho com relação à consistência com os requisitos e com as mudanças que foram feitas.

2. Identificar a fonte da inconsistência e sua justificativa.
3. Identificar as mudanças que precisam ser feitas nos planos e produtos de trabalho resultantes das mudanças na *baseline* de requisitos.
4. Iniciar as ações corretivas.

### 2.2.2. Desenvolvimento de Requisitos

O objetivo da área de processo Desenvolvimento de Requisitos (RD - *Requirements Development*) é produzir e analisar requisitos de clientes, produtos e componentes de produtos.

Esta área de processo descreve três tipos de requisitos: requisitos de clientes, requisitos de produtos e requisitos de componentes de produtos. Juntos, estes requisitos tratam as necessidades dos *stakeholders* relevantes, incluindo aquelas pertinentes às diversas fases do ciclo de vida do produto (por exemplo, critérios de testes de aceitação) e atributos do produto (por exemplo, segurança, confiabilidade e possibilidade de manutenção). Os requisitos também tratam as restrições causadas pela seleção de soluções de *design* (por exemplo, integração de produtos comerciais de prateleira).

Os requisitos são à base do *design*. O desenvolvimento de requisitos inclui as seguintes atividades:

- Elicitação, análise, validação e comunicação das necessidades dos clientes, expectativas e restrições para obter requisitos de clientes que constituem um entendimento do que satisfará os *stakeholders*
- Coleta e coordenação das necessidades dos *stakeholders*
- Desenvolvimento dos requisitos do ciclo de vida do produto
- Estabelecimento dos requisitos de clientes

- Estabelecimento dos requisitos iniciais de produtos e de componentes de produtos consistentes com os requisitos de clientes

Esta área de processo trata todos os requisitos de clientes, e não somente os requisitos de produto, porque o cliente pode também fornecer requisitos específicos de *design*.

Os requisitos de clientes são mais bem refinados em requisitos de produtos e de componentes de produtos. Além dos requisitos de clientes, requisitos de produtos e de componentes de produtos são derivados a partir das soluções de *design* selecionadas.

Os requisitos são identificados e refinados em todas as fases do ciclo de vida do produto. As decisões de *design*, ações corretivas subsequentes e *feedback* durante cada fase do ciclo de vida do produto são analisados com relação ao impacto nos requisitos derivados e alocados.

A área de processo de Desenvolvimento de Requisitos inclui três metas específicas. A meta específica Desenvolver Requisitos de Clientes trata da definição de um conjunto de requisitos do cliente para uso no desenvolvimento dos requisitos de produtos. A meta específica Desenvolver Requisitos de Produtos trata da definição de um conjunto de requisitos de produtos ou de componentes de produtos para uso no *design* de produtos e componentes de produtos. A meta específica Analisar e Validar Requisitos trata da necessária análise dos requisitos de clientes, produtos e componentes de produtos para definir, derivar e entender os requisitos. As práticas específicas da terceira meta específica pretendem auxiliar as práticas específicas das duas primeiras metas específicas.

As análises são utilizadas para entender, definir e selecionar os requisitos em todos os níveis a partir de alternativas. Estas análises incluem:

- Análise das necessidades e requisitos para cada fase do ciclo de vida do produto, incluindo as necessidades dos *stakeholders* relevantes, o ambiente operacional e fatores que refletem as expectativas e satisfação geral do cliente e dos usuários finais, como segurança, proteção e preço
- Desenvolvimento de um conceito operacional
- Definição da funcionalidade exigida

A definição da funcionalidade, também chamada de “análise funcional”, não é a mesma análise estruturada do desenvolvimento de *software* e não presume um *design* de *software* orientado a funcionalidades. No *design* de *software* orientado a objetos, ela se relaciona com a definição dos serviços. A definição de funções, seus agrupamentos lógicos e sua associação com requisitos são chamados de “arquitetura funcional”.

As análises ocorrem recursivamente em camadas sucessivas, cada vez mais detalhadas, da arquitetura de um produto até que detalhes suficientes estejam disponíveis para possibilitar o *design* detalhado, aquisição e teste do produto a ser executado. Como resultado da análise de requisitos e do conceito operacional (incluindo funcionalidade, suporte, manutenção e descarte), o conceito de manufatura ou produção produz requisitos mais derivados, incluindo as seguintes considerações:

- Restrições de diversos tipos
- Limitações tecnológicas
- Custo e direcionadores de custo
- Restrições de tempo e direcionadores de cronograma
- Riscos
- Consideração de questões citadas, mas não explicitamente declaradas pelo cliente ou usuário final

- Fatores introduzidos por considerações únicas de negócios do desenvolvedor, regulamentações e leis

Uma hierarquia de entidades lógicas (funções e subfunções, classes e subclasses de objetos) é estabelecida através de iterações com o conceito operacional em evolução. Os requisitos são refinados, derivados e alocados a estas entidades lógicas. Os requisitos e entidades lógicas são alocados a produtos, componentes de produtos, pessoas, processos associados ou serviços.

O envolvimento dos *stakeholders* relevantes no desenvolvimento e análise de requisitos lhes dá a visibilidade da evolução dos requisitos. Esta atividade continuamente assegura a eles que os requisitos estão sendo apropriadamente definidos.

Para a área de processo Desenvolvimento de Requisitos, tem as metas e práticas específicas de acordo com o Quadro 3.

**Quadro 3: CMMI – Práticas de Desenvolvimento de Requisitos**

Desenvolvimento de Requisitos	
Metas	Práticas
Desenvolver Requisitos do Cliente	Elicitar Necessidades
	Desenvolver os Requisitos do Cliente
Desenvolver Requisitos do Produto	Estabelecer os Requisitos de Produtos e Componentes de Produtos
	Alocar Requisitos de Componentes de Produtos
	Identificar Requisitos de Interfaces
Analisar e Validar os Requisitos	Estabelecer Conceitos e Cenários Operacionais
	Estabelecer uma Definição da Funcionalidade Necessária
	Analisar Requisitos
	Analisar Requisitos para Alcançar o Equilíbrio
	Validar Requisitos com Métodos Abrangentes

Fonte: Elaborado pelo pesquisador

**Meta: Desenvolver Requisitos do Cliente**

A meta Desenvolver Requisitos do Cliente tem por objetivo coletar e traduzir as necessidades, expectativas, restrições e *interfaces* dos *stakeholders* em requisitos de clientes.

As necessidades dos *stakeholders* (por exemplo, clientes, usuários finais, fornecedores, construtores e testadores) são a base para determinar os requisitos de clientes. As necessidades dos *stakeholders*, expectativas, restrições, *interfaces*, conceitos operacionais e conceitos do produto são analisados, harmonizados, refinados e elaborados para serem traduzidos em um conjunto de requisitos de clientes.

Muitas vezes, as necessidades, expectativas, restrições e *interfaces* dos *stakeholders* são pobremente identificadas ou são conflitantes. Uma vez que as necessidades, expectativas, restrições e limitações dos *stakeholders* deverão estar claramente identificadas e entendidas, um processo iterativo é usado durante toda a vida do projeto para atingir este objetivo. Para facilitar a necessária interação, um substituto do usuário final ou cliente é freqüentemente envolvido para representar suas necessidades e ajudar a resolver conflitos. A área de relações com o cliente ou de *marketing* da organização, assim como membros da equipe de desenvolvimento de disciplinas como a de engenharia humana ou suporte podem ser utilizados como substitutos. Restrições ambientais, legais e outras deverão ser consideradas quando da criação e resolução do conjunto de requisitos do cliente.

**Prática: Elicitar Necessidades**

A prática Elicitar Necessidades tem por objetivo elicitare as necessidades, expectativas, restrições e *interfaces* dos *stakeholders*. Elicitar vai além de coletar requisitos, envolvendo identificar de forma pró-ativa requisitos adicionais que não

foram explicitamente fornecidos pelos clientes. Os requisitos adicionais devem tratar as diversas atividades do ciclo de vida do produto e seu impacto no produto. Alguns exemplos de técnicas para elicitare necessidades incluem:

- Demonstrações tecnológicas
- Grupos de trabalho de controle de *interfaces*
- Grupos de trabalho de controle técnico
- Revisões intermediárias do projeto
- Questionários, entrevistas e cenários operacionais obtidos dos usuários finais
- *Walkthroughs* operacionais e análise de tarefas de usuários finais
- Protótipos e modelos
- *Brainstorming*
- Implementação de Funções de Qualidade (*Quality Function Deployment*)
- Pesquisas de mercado
- Testes betas
- Extração de fontes como documentos, padrões ou especificações
- Observação de produtos existentes, ambientes e padrões de fluxo de trabalho
- *Use cases*
- Análises de casos de negócios
- Engenharia reversa (para produtos legados)

As subpráticas encontradas nesta prática são:

1. Engajar os *stakeholders* relevantes utilizando métodos para elicitare as necessidades, expectativas, restrições e *interfaces* externas.

**Prática: Desenvolver os Requisitos do Cliente**

A prática Desenvolver os Requisitos do Cliente tem por objetivo transformar as necessidades, expectativas, restrições e *interfaces* dos *stakeholders* em requisitos de clientes. As várias entradas vindas do cliente devem ser consolidadas, as informações que faltarem devem ser obtidas e os conflitos devem ser resolvidos na documentação do conjunto reconhecido de requisitos de clientes. Os requisitos de clientes podem incluir necessidades, expectativas e restrições com relação à verificação e validação.

As subpráticas encontradas nesta prática são:

1. Traduzir as necessidades, expectativas, restrições e *interfaces* dos *stakeholders* em requisitos de clientes documentados.
2. Definir restrições para verificação e validação.

**Meta: Desenvolver Requisitos de Produtos**

A meta Desenvolver Requisitos de Produtos tem por objetivo refinar e elaborar os requisitos dos clientes para desenvolver requisitos de produtos e de componentes de produtos.

Os requisitos de clientes são analisados em conjunto com o desenvolvimento do conceito operacional para derivar conjuntos de requisitos mais detalhados e precisos chamados de “requisitos de produtos e de componentes de produtos”. Os requisitos de produtos e de componentes de produtos tratam as necessidades associadas com cada fase do ciclo de vida do produto. Requisitos derivados provêm das restrições, consideração das questões citadas, mas não explicitamente declaradas na *baseline* de requisitos de clientes e fatores introduzidos pela arquitetura selecionada, o *design* e as considerações únicas de negócios dos desenvolvedores. Os requisitos são reexaminados

com cada nível sucessivo mais baixo do conjunto de requisitos e arquitetura funcional e o conceito preferencial do produto é refinado.

Os requisitos são alocados a funções de produtos e componentes de produtos incluindo objetos, pessoas e processos. A rastreabilidade dos requisitos a funções, objetos, testes, questões e outras entidades é documentada. Os requisitos e funções alocados são a base para a síntese da solução técnica. Conforme os componentes internos são desenvolvidos, *interfaces* adicionais são definidas e requisitos de *interfaces* são estabelecidos.

### **Prática: Estabelecer Requisitos de Produtos e de Componentes de Produtos**

A prática Estabelecer Requisitos de Produtos e de Componentes de Produtos tem por objetivo estabelecer e manter os requisitos de produtos e componentes de produtos, que são baseados nos requisitos de clientes.

Os requisitos de clientes podem ser expressos nos termos do cliente e podem ser descrições não técnicas. Os requisitos de produtos são a expressão destes requisitos em termos técnicos que podem ser utilizados para decisões de *design*. Um exemplo desta tradução é encontrado na primeira Casa de Implementação de Qualidade Funcional (*The first House of Quality Functional Deployment*), que mapeia os desejos do cliente em parâmetros técnicos. Por exemplo, “uma porta sólida e sonora” pode ser mapeada com o tamanho, peso, formato, umidade e frequências de ressonância.

Requisitos de produtos e de componentes de produtos tratam a satisfação dos clientes, negócios e objetivos do projeto e atributos associados, como eficiência e custo.

Restrições de *design* incluem especificações sobre componentes de produtos que são derivados de decisões de *design*, e não de requisitos de nível mais elevado.

Requisitos derivados também tratam o custo e o desempenho de outras fases do ciclo de vida (por exemplo, produção, operação e descarte) na medida compatível com os objetivos de negócios.

A modificação dos requisitos devido a mudanças de requisitos é coberta pela função de “manutenção” desta prática específica, enquanto a administração de mudanças de requisitos é coberta pela área de processo de Gerenciamento de Requisitos.

As subpráticas encontradas nesta prática são:

1. Desenvolver requisitos nos termos técnicos necessários para o *design* dos produtos e componentes de produtos.

Desenvolver os requisitos de arquitetura trata das qualidades críticas do produto e do desempenho necessário para o *design* da arquitetura do produto.

2. Derivar requisitos que resultam de decisões de *design*.

A seleção de uma tecnologia traz requisitos adicionais. Por exemplo, o uso de eletrônica exige requisitos tecnológicos específicos adicionais, tais como os limites da interferência eletromagnética.

3. Estabelecer e manter os relacionamentos entre os requisitos para consideração durante o gerenciamento de mudanças e alocação de requisitos.

Relacionamentos entre os requisitos podem auxiliar na avaliação do impacto das mudanças.

### **Prática: Alocar Requisitos de Componentes de Produtos**

A prática Alocar Requisitos de Componentes de Produtos tem por objetivo alocar os requisitos para todos os componentes de produto. Os requisitos para os componentes de produtos da solução definida incluem a alocação do desempenho do

produto; restrições de *design* e encaixe, formato e funções para atender os requisitos e facilitar a produção. Em casos onde um nível mais elevado de requisitos especifica um desempenho que é de responsabilidade de dois ou mais componentes de produtos, o desempenho deve ser particionado para alocação única a cada componente de produto como um requisito derivado.

As subpráticas encontradas nesta prática são:

1. Alocar requisitos a funções.
2. Alocar requisitos a componentes de produtos.
3. Alocar restrições de *design* a componentes de produtos.
4. Documentar relacionamentos entre os requisitos alocados.

Relacionamentos incluem as dependências nas quais uma mudança em um requisito pode afetar outros requisitos.

### **Prática: Identificar Requisitos de *Interface***

A prática Identificar Requisitos de *Interface* tem por objetivo identificar requisitos de *interface*. *Interfaces* entre funções (ou entre objetos) são identificadas. As *interfaces* funcionais podem direcionar o desenvolvimento de soluções alternativas.

Requisitos de *interfaces* entre produtos e componentes de produtos identificados na arquitetura do produto são definidos. Eles são controlados como parte da integração do produto e dos componentes do produto e são parte integral da definição da arquitetura.

As subpráticas encontradas nesta prática são:

1. Identificar *interfaces* externas e internas ao produto (isto é, entre partes funcionais ou objetos).

Conforme o *design* progride, a arquitetura do produto será alterada pelos processos de soluções técnicas, criando novas *interfaces* entre os componentes de produtos e componentes externos ao produto.

As *interfaces* com os processos do ciclo de vida relacionado ao produto deverão também ser identificadas.

Exemplos destas *interfaces* incluem *interfaces* com equipamento de teste, sistemas de transporte, sistemas de suporte e recursos de fabricação.

2. Desenvolver os requisitos para as *interfaces* identificadas.

Os requisitos das *interfaces* são definidos em termos de características de origem, destino, estímulo e de dados para o *software* e características elétricas e mecânicas para o *hardware*.

### **Meta: Analisar e Validar os Requisitos**

A meta Analisar e Validar os Requisitos tem por objetivo analisar e validar os requisitos e desenvolver uma definição das funcionalidades necessárias.

As práticas específicas desta meta suportam o desenvolvimento dos requisitos nas metas específicas Desenvolver Requisitos do Cliente e Desenvolver Requisitos de Produtos. As práticas específicas associadas com esta meta específica cobrem a análise e validação dos requisitos com relação ao ambiente pretendido para o usuário.

As análises são executadas para determinar o impacto que o ambiente operacional pretendido terá na capacidade de satisfazer as necessidades, expectativas, restrições e *interfaces* dos *stakeholders*. Considerações sobre a facilidade de construção, necessidades da missão, restrições de custo, tamanho potencial do mercado e estratégia de aquisição devem ser levadas em conta, dependendo do contexto do produto. Uma definição da funcionalidade exigida é também estabelecida. Todos os modos de utilização especificados para o produto são considerados e uma análise de tempo é gerada para o seqüenciamento de funções criticamente dependentes do tempo.

Os objetivos das análises são determinar os requisitos candidatos para os conceitos de produtos que satisfarão as necessidades, expectativas e restrições dos *stakeholders* e, então, traduzir estes conceitos em requisitos. Em paralelo com esta atividade, os parâmetros que serão utilizados para avaliar a eficiência do produto são determinados, com base nas entradas do cliente e no conceito preliminar do produto.

Os requisitos são validados para aumentar a probabilidade de o produto resultante funcionar como previsto no ambiente de uso.

### **Prática: Estabelecer Conceitos e Cenários Operacionais**

A prática Estabelecer Conceitos e Cenários Operacionais tem por objetivo estabelecer e manter conceitos e cenários operacionais associados.

Um cenário é uma seqüência de eventos que podem ocorrer no uso do produto, que é utilizado para tornar explícitas algumas das necessidades dos *stakeholders*. Em contraste, um conceito operacional para um produto normalmente depende da solução de *design* e do cenário. Por exemplo, o conceito operacional para um produto de comunicação por satélite é bastante diferente de outro baseado em meridianos. Uma vez que as soluções alternativas normalmente ainda não foram definidas quando se está preparando os conceitos operacionais iniciais, as soluções conceituais são desenvolvidas para serem utilizadas quando da análise dos requisitos. Os conceitos operacionais são refinados conforme as decisões de soluções são tomadas e requisitos detalhados de nível mais baixo são desenvolvidos.

Da mesma forma que uma decisão de *design* para um produto pode se tornar um requisito para os componentes do produto, o conceito operacional pode se tornar o cenário (requisitos) para os componentes do produto.

Os cenários podem incluir seqüências operacionais, desde que essas seqüências sejam uma expressão dos requisitos do cliente e não conceitos operacionais.

As subpráticas encontradas nesta prática são:

1. Desenvolver conceitos operacionais e cenários que incluem a funcionalidade, desempenho, manutenção, suporte e descarte, conforme apropriado.  
Identificar e desenvolver cenários, consistentes com o nível de detalhes nas necessidades, expectativas e restrições dos *stakeholders*, sob os quais o produto proposto deverá operar.
2. Definir o ambiente no qual o produto irá operar, incluindo limites e restrições.

3. Revisar os conceitos operacionais e cenários para refinar e descobrir requisitos.

O desenvolvimento de conceitos operacionais e cenários é um processo iterativo. As revisões deverão ocorrer periodicamente para assegurar que estes atendem os requisitos. A revisão pode tomar a forma de um *walkthrough*.

4. Desenvolver um detalhado conceito operacional, conforme os produtos e componentes de produtos são selecionados, que define a interação do produto, usuário final e ambiente e que satisfaz as necessidades operacionais, de manutenção, de suporte e de descarte.

### **Prática: Estabelecer uma Definição da Funcionalidade Exigida**

A prática Estabelecer uma Definição da Funcionalidade Exigida tem por objetivo estabelecer e manter uma definição da funcionalidade exigida.

A definição da funcionalidade, também chamada de “análise funcional”, é a descrição do que o produto deve fazer. A descrição da funcionalidade pode incluir ações, seqüência, entradas, saídas e outras informações que transmitem a maneira pela qual o produto será utilizado.

A análise funcional não é a mesma coisa que a análise estruturada do desenvolvimento de *software* e não presume um *design* de *software* orientado a funcionalidades. No *design* de *software* orientado a objetos, ela se relaciona com a definição dos serviços. A definição das funções, seus agrupamentos lógicos e suas associações com os requisitos são referenciados como uma arquitetura funcional.

As subpráticas encontradas nesta prática são:

1. Analisar e quantificar as funcionalidades exigidas pelos usuários finais.

2. Analisar os requisitos para identificar particionamentos lógicos ou funcionais (por exemplo, subfunções).
3. Particionar os requisitos em grupos com base em critérios estabelecidos (por exemplo, funcionalidades semelhantes, desempenho ou acoplamento), para facilitar e concentrar a análise de requisitos.
4. Considerar a seqüência das funções críticas de tempo, tanto no momento inicial como durante o desenvolvimento dos componentes do produto.
5. Alocar os requisitos de clientes às partições funcionais, objetos, pessoas ou elementos de suporte para suportar a síntese de soluções.
6. Alocar requisitos funcionais e de desempenho a funções e subfunções.

### **Prática: Analisar os Requisitos**

A prática Analisar os Requisitos tem por objetivo analisar os requisitos para assegurar que eles são necessários e suficientes.

À luz do conceito operacional e dos cenários, os requisitos para um nível de hierarquia de produto são analisados para determinar se eles são necessários e suficientes para atender os objetivos dos níveis mais elevados da hierarquia do produto. Os requisitos analisados, então, fornecem a base para requisitos mais precisos e detalhados para os níveis mais baixos da hierarquia do produto.

Conforme os requisitos são definidos, seus relacionamentos com requisitos e funcionalidades de mais alto nível devem ser entendidos. Outra ação é a determinação dos requisitos chave que serão utilizados para rastrear o progresso técnico. Por exemplo, o peso de um produto ou tamanho de um produto de *software* podem ser monitorados através do desenvolvimento baseado no seu risco.

As subpráticas encontradas nesta prática são:

1. Analisar as necessidades, expectativas e *interfaces* externas dos *stakeholders* para remover conflitos e organizá-las em assuntos relacionados.
2. Analisar requisitos para determinar se eles satisfazem os objetivos de requisitos de nível mais alto.
3. Analisar requisitos para assegurar que eles são completos, possíveis de serem executados, percebidos e verificados.

Embora o *design* determine a possibilidade de se construir uma solução específica, esta subprática trata da descoberta de quais requisitos afetam esta possibilidade.

4. Identificar requisitos chave que têm uma forte influência no custo, cronograma, funcionalidade, risco ou desempenho.
5. Identificar medidas de desempenho técnico que serão rastreadas durante o esforço de desenvolvimento.
6. Analisar os conceitos operacionais e cenários para refinar as necessidades, restrições e *interfaces* dos clientes e descobrir novos requisitos.

Esta análise pode resultar em conceitos operacionais e cenários mais detalhados, bem como suportar a derivação de novos requisitos.

### **Prática: Analisar os Requisitos para Alcançar o Equilíbrio**

A prática Analisar os Requisitos para Alcançar o Equilíbrio tem por objetivo analisar os requisitos para equilibrar as necessidades e as restrições dos *stakeholders*. As necessidades dos *stakeholders* e as restrições podem tratar de custos, cronograma, desempenho, funcionalidade, componentes reusáveis, possibilidade de manutenção e riscos.

As subpráticas encontradas nesta prática são:

1. Usar modelos comprovados, simulações e protótipos para analisar o equilíbrio entre as necessidades dos *stakeholders* e as restrições.  
Os resultados das análises podem ser usados para reduzir o custo do produto e o risco do desenvolvimento do produto.
2. Executar uma análise dos riscos sobre os requisitos e a arquitetura funcional.
3. Examinar os conceitos do ciclo de vida do produto com relação ao impacto dos requisitos nos riscos.

### **Prática: Validar os Requisitos com Métodos Abrangentes**

A prática Validar os Requisitos com Métodos Abrangentes tem por objetivo validar os requisitos para assegurar que o produto resultante funcionará como previsto no ambiente do usuário utilizando diversas técnicas, conforme necessário.

A validação dos requisitos é executada antecipadamente ao esforço de desenvolvimento para se adquirir a confiança de que os requisitos são capazes de guiar um desenvolvimento que resultará numa validação final bem sucedida. Esta atividade deverá estar integrada com as atividades de gerenciamento de riscos. Organizações maduras normalmente farão a validação dos requisitos de forma mais sofisticada e irão ampliar a base de validação para incluir outras necessidades e expectativas dos *stakeholders*. Estas organizações geralmente farão análises, simulações ou protótipos para assegurar que os requisitos satisfarão as necessidades e expectativas dos *stakeholders*.

As subpráticas encontradas nesta prática são:

1. Analisar os requisitos para determinar o risco do produto resultante não funcionar apropriadamente no seu ambiente de uso previsto.
2. Explorar a adequação e a completitude dos requisitos através do desenvolvimento de representações do produto (por exemplo, protótipos, simulações, modelos, cenários e *storyboards*) e da obtenção de *feedback* sobre eles dos *stakeholders* relevantes.
3. Analisar o *design*, conforme ele amadurece, no contexto do ambiente de validação dos requisitos para identificar questões de validação e expor necessidades e requisitos de clientes não declarados.

### **3. ADERÊNCIA DA METODOLOGIA RUP AO MODELO CMMI**

Para uma organização obter a certificação CMMI, é preciso que todas as práticas das áreas de processo pertencentes ao nível de maturidade que se deseja conseguir estejam implementadas. A seguir, temos a comparação das práticas das áreas de processos Gerenciamento de Requisitos e Desenvolvimento de Requisitos que são atendidas pela disciplina de Requisitos da metodologia de RUP.

#### **3.1. Técnicas de Verificação de Aderência**

Para a verificação de aderência da metodologia RUP ao modelo CMMI, serão analisadas as metas e práticas específicas pertencentes às áreas de processo Gerenciamento de Requisitos e Desenvolvimento de Requisitos do modelo CMMI, a fim de encontrar uma equivalência dos itens nas atividades e tarefas relacionadas à disciplina de Requisitos do RUP.

#### **3.2. Práticas de Gerenciamento de Requisitos**

A área de processo de Gerenciamento de Requisitos visa manter um conjunto de requisitos atualizados e aprovados por meio do gerenciamento de mudanças, manutenção dos relacionamentos, identificação de inconsistências e correção dos defeitos.

Analisando o RUP, temos as seguintes tarefas que se encaixam nos procedimentos descritos no modelo CMMI:

- Identificar Pedidos dos Investidores
- Gerenciar Dependências
- Revisar Requisitos

### **Tarefa: Identificar Pedidos dos Investidores**

Prática Atendida: Obter um Entendimento dos Requisitos

A finalidade dessa tarefa é:

- Entender quem são os investidores do projeto.
- Coletar pedidos de quais necessidades o sistema deve preencher.
- Priorizar pedidos dos envolvidos.

Pedidos dos Investidores são obtidos e reunidos ativamente a partir de origens existentes para obter uma "lista de desejos" do que os diferentes investidores do projeto (clientes, usuários, patrocinadores do produto) esperam ou desejam que o sistema inclua, juntamente com informações sobre como cada pedido foi considerado pelo projeto.

Depois de selecionar os fornecedores de requisitos, a coleta das informações é feita via questionários, entrevistas ou outro método, com critérios e perguntas já estruturadas. São realizadas reuniões e *workshops* entre os fornecedores de requisitos e os analistas para um entendimento e aceitação dos requisitos. No fim da tarefa de identificação das necessidades dos investidores, é feita uma avaliação para comprovar se toda a necessidade contém informações suficientes e origem relacionada. Se houver inconsistências entre os pedidos, estas devem ser esclarecidas.

### **Tarefa: Gerenciar Dependências**

Práticas Atendidas: Obter Compromissos com os Requisitos, Gerenciar Mudanças nos Requisitos e Manter a Rastreabilidade Bidirecional dos Requisitos.

A finalidade dessa tarefa é utilizar os atributos e a rastreabilidade dos requisitos do projeto para auxiliar no gerenciamento do escopo do projeto e gerenciar os requisitos variáveis.

Esta tarefa é responsável por gerenciar os requisitos de acordo com o Plano de Gerenciamento de Requisitos. Os atributos de requisitos são rastreados de acordo com o tipo de requisitos de forma que as mudanças ocorridas possam ser analisadas e o impacto nos compromissos existentes possa ser avaliado.

Junto com a tarefa Desenvolver Visão, as necessidades são registradas de forma que os envolvidos possam estar de acordo com as definições estabelecidas.

### **Tarefa: Revisar Requisitos**

Prática Atendida: Identificar Inconsistências entre o Trabalho do Projeto e os Requisitos.

A finalidade dessa tarefa é garantir formalmente que os resultados das tarefas dos requisitos estejam em conformidade com a visão que o cliente tem do sistema.

Para a revisão dos requisitos, deverão ser marcadas reuniões entre Analistas de Sistemas, Investidores, Arquiteto de *Software* e qualquer outro envolvido que venha trabalhar com os requisitos revisados. Deve-se verificar continuamente o que está sendo produzido para certificar-se de que a qualidade do produto seja a melhor possível. As revisões podem ser divididas em algumas reuniões:

- Uma revisão de controles de mudanças que afeta o conjunto de requisitos existentes.

- Uma revisão para cada um dos produtos de trabalho dos requisitos-chave. Se o sistema for grande, divida essa revisão em várias reuniões, possivelmente uma por área-chave funcional.

Após as revisões, os resultados são documentados em um registro de revisão. Se houver inconsistências, estas serão documentadas de acordo com o processo de gerenciamento de mudanças.

### **3.3. Práticas de Desenvolvimento de Requisitos**

A área de processo de Desenvolvimento de Requisitos visa produzir e analisar requisitos de clientes, produtos e de componentes de produtos.

Analisando o RUP, temos as seguintes tarefas que se encaixam nos procedimentos descritos no modelo CMMI:

- Identificar Pedidos dos Investidores
- Localizar Agentes e Casos de Uso
- Detalhar um Caso de Uso
- Desenvolver Especificações Suplementares
- Detalhar os Requisitos de *Software*
- Revisar Requisitos

#### **Tarefa: Identificar Pedidos dos Investidores**

Prática Atendida: Elicitar Necessidades

A descrição desta tarefa está na seção 3.2 Práticas de Gerenciamento de Requisitos deste trabalho.

**Tarefa: Localizar Agentes e Casos de Uso**

Práticas Atendidas: Desenvolver os Requisitos do Cliente e Estabelecer Conceitos e Cenários Operacionais

A finalidade dessa tarefa é:

- Definir o escopo do sistema - o que será tratado pelo sistema e o que será tratado fora dele.
- Definir quem e o que interagirá com o sistema.
- Delinear a funcionalidade do sistema.

Qualquer indivíduo, grupo ou fenômeno que o sistema deverá interagir é chamado de agente. Os agentes são localizados e uma breve descrição de cada um é descrita. Após essa etapa são desenvolvidos os casos de uso, ou seja, as funcionalidades que o sistema disponibilizará. Com os casos de uso já criados, o fluxo entre os casos de uso e os agentes são desenvolvidos a fim de termos as interações entre eles.

Ainda nessa tarefa, são identificadas as funcionalidades não tratadas pelo modelo de caso de uso. Todas as funcionalidades são avaliadas, e ao final o relatório sintético do modelo de caso de uso deverá estar claramente especificado.

**Tarefa: Detalhar um Caso de Uso**

Prática Atendida: Estabelecer os Requisitos de Produtos e Componentes de Produtos

A finalidade dessa tarefa é:

- Descrever um ou mais dos fluxos de eventos do caso de uso em detalhes suficientes para permitir que o desenvolvimento do *software* seja iniciado nele.

- Detalhar o caso de uso para a compreensão e satisfação do representante agente ou cliente.

O ponto de partida desta tarefa é a revisão e refinamento dos cenários identificados na tarefa Localizar Agentes e Casos de Uso. Os fluxos de eventos são detalhados por meio de esboços seqüenciais e, se houver, protótipos de *interface* com o usuário. Na descrição dos casos de uso, considerar os seguintes pontos:

- Como o caso de uso começa?
- Como o caso de uso termina?
- Como o caso de uso troca dados com os agentes?
- Como o caso de uso repete o mesmo comportamento?
- Há alguma situação excepcional em um fluxo de eventos de caso de uso?
- Como o caso de uso deve ser descrito para que o cliente e os usuários possam compreendê-lo?

O fluxo de eventos deverá ser estruturado, podendo ser dividido em vários subfluxos. Neste caso, quando o caso de uso é ativado, os subfluxos podem ser combinados de várias maneiras, desde que as seguintes questões sejam verdadeiras:

- O caso de uso pode prosseguir a partir de um dos diversos caminhos possíveis, dependendo das informações fornecidas por determinado agente ou dos valores de algum atributo ou objeto.
- O caso de uso pode executar alguns subfluxos em seqüências opcionais.
- O caso de uso pode executar vários subfluxos ao mesmo tempo.

A representação da interação entre casos de uso e agentes é representada via diagramas de caso de uso. Os requisitos não funcionais são descritos como sendo requisitos especiais do caso de uso.

Nesta tarefa também deverão ser descritos os protocolos de comunicação com agentes externos, pré-condições para o início do caso de uso e pós-condições para o término do caso de uso.

### **Tarefa: Desenvolver Especificações Suplementares**

Práticas Atendidas: Alocar Requisitos de Componentes de Produtos e Identificar Requisitos de Interfaces.

A finalidade dessa tarefa é capturar os requisitos que não são prontamente capturados em Casos de Uso.

É importante certificar-se de que todos os requisitos estejam especificados no nível de detalhe necessário, para que sejam distribuídos aos *designers*, testadores e escritores de documentação. Se os requisitos forem rastreados, certifique-se de que cada requisito esteja claramente identificado.

Requisitos funcionais descrevem o comportamento (funções ou serviços) do sistema que suporta objetivos do usuário, tarefas ou atividades, e são documentados em casos de uso específicos. Porém há requisitos funcionais que não se encaixam em casos de uso, devendo ser documentados em especificações suplementares.

Os requisitos não funcionais incluem qualidades e restrições que o sistema deve obedecer. Qualidades do sistema são propriedades ou características do sistema com as quais seus envolvidos se preocupam e que, portanto, afetarão o seu grau de satisfação com o sistema. As preocupações em relação à qualidade do sistema são:

- Usabilidade: tempo de treinamento para utilização do sistema, tempo de tarefas e requisitos para a conformidade dos padrões de utilização (CUA, GUI, etc.).

- Confiabilidade: disponibilidade (a quantidade do sistema "funcionando apropriadamente"), exatidão dos cálculos do sistema e a capacidade de recuperação do sistema após falhas.
- Desempenho: rendimento do processamento, tempo de resposta, tempo de recuperação, tempo de inicialização e tempo de encerramento
- Suportabilidade: possibilidade de teste, adaptabilidade, sustentabilidade, compatibilidade, configurabilidade, instalabilidade, escalabilidade e possibilidade de localização.

Podemos considerar restrição como uma limitação no grau de liberdade que temos em fornecer uma solução. As categorias de restrições são:

- Restrição de *Design*: especifica ou restringe as opções para a arquitetura e/ou o *design* de um sistema. Por exemplo, requerer que um banco de dados relacional seja utilizado para persistência.
- Restrição de Implementação: especifica ou restringe a codificação ou construção de um sistema. Por exemplo, padrões requeridos, processos, ferramentas, idiomas de implementação, plataformas de *hardware*, sistemas operacionais, componentes de terceiras partes, bibliotecas de classes e limites de recursos (uso da memória ou do espaço em disco).
- Restrição de *Interface*: especifica um item externo com o qual um sistema deve interagir ou restrições em formatos ou outros fatores utilizados dentro de tal interação. Por exemplo, efetuar *interface* com um sistema externo através de filas de mensagens.
- Restrição Física: especifica a restrição física imposta no *hardware* utilizado para hospedar o sistema — *shape*, tamanho ou peso, por exemplo.

**Tarefa: Detalhar os Requisitos de *Software***

Práticas Atendidas: Estabelecer uma Definição da Funcionalidade Necessária.

A finalidade dessa tarefa é coletar, detalhar e organizar o conjunto (pacote) de produtos de trabalho que descrevem completamente os requisitos de *software* do sistema.

Detalhar os requisitos de *software* envolve detalhamento dos casos de uso e da especificação suplementar para os requisitos que devem ser entregues no *release* atual. Depois de certificar-se que os requisitos estão especificados em um bom nível de detalhes, é preciso gerar a documentação das ferramentas de modelagem gráfica e rastreabilidade e gerenciamento de requisitos a fim de enviar esta documentação para a revisão. Todos os produtos de trabalho dos requisitos são montados e qualquer informação adicional é fornecida para que seja produzida uma definição completa dos requisitos de *software*.

Em projetos menos formais, esta etapa consiste em empacotar os relatórios relevantes e a documentação gerada manualmente, com materiais de suporte suficientes. Em projetos mais formais, uma ou mais SRS (Especificações de Requisitos de *Software*) podem ser produzidas. Por exemplo, uma SRS separada pode descrever todos os requisitos de *software* para cada recurso em uma determinada liberação do produto. Isso pode incluir os requisitos que são documentados em outros produtos de trabalho dos requisitos. Em tais casos, o SRS pode se referir aos requisitos documentados nos outros produtos de trabalho. Isso permite que você tenha uma SRS formal, sem a necessidade de duplicar as informações nos outros produtos de trabalho.

**Tarefa: Revisar Requisitos**

Práticas Atendidas: Analisar Requisitos, Analisar Requisitos para Alcançar o Equilíbrio, Validar Requisitos com Métodos Abrangentes.

A descrição desta tarefa está na seção 3.2 Práticas de Gerenciamento de Requisitos deste trabalho.

### 3.4. Avaliação de Aderência

Como resultado da comparação entre o modelo CMMI e a metodologia RUP, o Quadro 4 faz a associação de equivalência para os dois produtos.

**Quadro 4:** Modelo CMMI x Metodologia RUP

	CMMI	RUP	
Gerenciamento de Requisitos	Obter um Entendimento dos Requisitos	Identificar Pedidos dos Investidores	Requisitos
	Obter Compromisso com os Requisitos	Gerenciar Dependências	
	Gerenciar Mudanças nos Requisitos	Gerenciar Dependências	
	Manter a Rastreabilidade Bidirecional dos Requisitos	Gerenciar Dependências	
	Identificar Inconsistências entre o Trabalho do Projeto e os Requisitos	Revisar Requisitos	
Desenvolvimento de Requisitos	Elicitar Necessidades	Identificar Pedidos dos Investidores	Requisitos
	Desenvolver os Requisitos do Cliente	Localizar Agentes e Casos de Uso	
	Estabelecer os Requisitos de Produtos e Componentes de Produtos	Localizar Agentes e Casos de Uso	
	Alocar Requisitos de Componentes de Produtos	Desenvolver Especificações Suplementares	
	Identificar Requisitos de Interfaces	Desenvolver Especificações Suplementares	
	Estabelecer Conceitos e Cenários Operacionais	Localizar Agentes e Casos de Uso	
	Estabelecer uma Definição da Funcionalidade Necessária	Detalhar os Requisitos de Software	
	Analisar Requisitos	Revisar Requisitos	
	Analisar Requisitos para Alcançar o Equilíbrio	Revisar Requisitos	
	Validar Requisitos com Métodos Abrangentes	Revisar Requisitos	

*Fonte: Elaborado pelo pesquisador*

O quadro representa qual tarefa da disciplina de Requisitos do RUP detêm no seu conteúdo as práticas das áreas de processo de Gerenciamento de Requisitos e Desenvolvimento de Requisitos do CMMI. Observar-se que uma tarefa pode suprir mais de uma prática, como é o caso da tarefa Gerenciar Dependências, que suporta as

práticas Obter Compromisso com os Requisitos, Gerenciar Mudanças nos Requisitos e Manter a Rastreabilidade Bidirecional dos Requisitos.

Para as áreas de processo de Gerenciamento de Requisitos e Desenvolvimento de Requisitos do modelo CMMI, pode-se considerar o **nível de aderência de 100%** da metodologia RUP, pois todas as práticas específicas destas áreas de processo foram atendidas.

#### 4. CONCLUSÃO

Para conseguir fazer a comparação entre o RUP e o CMMI foi preciso primeiro achar um nível de conteúdo compatível. O RUP possui na sua estrutura a quebra por disciplina, que por sua vez tem atividades que formam o fluxo de trabalho. Todas as atividades são compostas por uma seqüência de tarefas, realizadas em várias etapas. Deste produto obtém-se a árvore Disciplina > Atividade > Tarefa > Etapa. O modelo CMMI tem a quebra por área de processo, onde estas possuem metas a serem alcançadas. Para que se atinja a meta, é preciso antes concluir todas as práticas estabelecidas, que por sua vez são formadas por subpráticas. Dessa forma, obtém-se a árvore Área de Processo > Meta > Prática > Subprática. Ao estudar os dois produtos, foi determinado que a comparação fosse feita entre o conteúdo das Práticas de CMMI contra as Tarefas de RUP, pois ambos são níveis de estrutura onde não encontra-se tantas quebras de atribuições, o que facilita a análise de conteúdo. O fato de restringir a pesquisa apenas no âmbito de Requisitos também foi um dos fatores que a tornou possível de ser realizada, pois ambos os produtos possuem vasta área de atuação, fato que deixa em aberto a possibilidade de futuras pesquisas relacionadas a novos relacionamentos, seja entre estes produtos, ou entre produtos diferentes.

## REFERÊNCIAS

Carnegie Mellon University – Software Engineering Institute - **CMMI for Development, Version 1.3**. Pittsburgh, 2010.

DIEHL, A. A.; TATIM, D. C. **Pesquisa em ciências sociais aplicadas. Métodos e técnicas**. São Paulo. Pearson, 2004.

IBM - [http://www.ibm.com/developerworks/downloads/r/rup/?S\\_CMP=rnav](http://www.ibm.com/developerworks/downloads/r/rup/?S_CMP=rnav)  
Acesso em 03/09/2011

ISD Brasil - [http://www.isdbrasil.com.br/artigos/artigo\\_excelencia.php](http://www.isdbrasil.com.br/artigos/artigo_excelencia.php)  
Acesso em 04/11/2011

## BIBLIOGRAFIA CONSULTADA

Blog CMMI - <http://www.blogcmmi.com.br/>  
Acesso em 04/11/2011

GIL, A. C. – **Como elaborar Projetos de Pesquisa**. São Paulo. Atlas, 1994

KOSCIANSKI, A.; SOARES, M.S.S. - **Qualidade de Software**. São Paulo. Editora Novatec, 2006.

Wibas GmbH - <http://www.cmmi.de/#el=CMMI/0/HEAD/folder/folder.CMMI>  
Acesso em 07/11/2011