

UNIVERSIDADE SÃO JUDAS TADEU  
CURSO DE PÓS-GRADUAÇÃO – LATO SENSU  
ENGENHARIA DE SOFTWARE

PABLO FERNANDO FELIPE

**C.M.M.I. E SPICE: UM ESTUDO COMPARATIVO NA  
ABORDAGEM DA ENGENHARIA DE REQUISITOS**

SÃO PAULO  
2006

UNIVERSIDADE SÃO JUDAS TADEU  
CURSO DE PÓS-GRADUAÇÃO – LATO SENSU  
ENGENHARIA DE SOFTWARE

PABLO FERNANDO FELIPE

**C.M.M.I. E SPICE: UM ESTUDO COMPARATIVO NA  
ABORDAGEM DA ENGENHARIA DE REQUISITOS**

Monografia apresentada ao curso de Pós-graduação Lato Sensu da Universidade São Judas Tadeu, como requisito parcial para conclusão do curso de Especialização em Engenharia de Software.

ORIENTADOR: Professor Ms.<sup>o</sup> Aluizio Saiter

SÃO PAULO

2006

UNIVERSIDADE SÃO JUDAS TADEU  
CURSO DE PÓS-GRADUAÇÃO – LATO SENSU  
ENGENHARIA DE SOFTWARE

PABLO FERNANDO FELIPE

**C.M.M.I. E SPICE: UM ESTUDO COMPARATIVO NA  
ABORDAGEM DA ENGENHARIA DE REQUISITOS**

Monografia apresentada ao curso de Pós-graduação Lato Sensu da Universidade São Judas Tadeu, como requisito parcial para conclusão do curso de Especialização em Engenharia de Software.

Aprovada em agosto de 2006

---

ORIENTADOR: Professor Ms. ° Aluizio Saiter

*Aos meus pais, amigos e a Alcileide do Carmo de Godoy pelo apoio e incentivo recebido durante a elaboração deste trabalho.*

## AGRADECIMENTOS

Ao Professor Ms. <sup>o</sup> Aluizio Saiter, por sua dedicação e colaboração no decorrer desta pesquisa, e apresentação de observações importantes em seus comentários.

*“A sabedoria da vida não consiste em fazer aquilo que se gosta, mas em gostar do que se faz”.*

Leonardo da Vinci

## RESUMO

FELIPE, Pablo Fernando. CMMI e Spice: Um estudo comparativo na abordagem da Engenharia de Requisitos. Monografia. Curso de Pós-Graduação Lato Sensu, da Universidade São Judas Tadeu. São Paulo. P. 94 páginas.

Este trabalho tem por objetivo levantar as principais diferenças na abordagem dos tópicos inerentes a Engenharia de Requisitos entre as metodologias mais utilizadas atualmente na melhoria de processos de produção de software, C.M.M.I. e Spice.

Hoje existe um entendimento entre profissionais da área tecnológica e entre pesquisadores sobre o papel da Engenharia de Requisitos no processo de produção de software. Levantamentos recentes da comunidade europeia bem como de organizações norte-americanas apontam para problemas relacionados com Engenharia de Requisitos como os que trazem maiores problemas para os profissionais. Um questionário distribuído em 3.805 empresas na Europa apontou os maiores problemas na produção de um software: especificação de requisitos (53%), gerência de requisitos (43%), documentação (36%) e teste (35%).

Desta forma, torna-se clara a importância do entendimento da Engenharia de Requisitos e a visão comparativa e sistêmica das diferentes abordagens desta disciplina entre as metodologias mais usadas atualmente por empresas do mundo inteiro.

Durante a discussão deste trabalho procuro levantar imparcialmente as diferenças, as possíveis vantagens, às desvantagens e observações sobre as duas metodologias no tocante a Engenharia de Requisitos, e na melhor utilização de suas práticas e conceitos.

Palavras-chave: Engenharia de Software, C.M.M.I., SPICE, Engenharia de Requisitos.

## ABSTRACT

This work has for objective to more raise the main differences in the boarding of the inherent topics the engineering of requirements between the methodologies used currently in the improvement of processes of software production, CMMI and Spice.

Today an agreement between professionals of the technological area and researchers exists on the paper of the engineering of requirements in the process of software production. Recent surveys of the Europe community as well as of North American organizations point with respect to problems related with Engineering of Requirements as the ones that bring greater problems for the professionals. A questionnaire distributed in 3.805 companies in the Europe pointed the biggest problems in the production of a software: specification of requirements (53%), management of requirements (43%), documentation (36%) and has tested (35%).

Of this form, the importance of the agreement of the Engineering of Requirements becomes clear and the comparative and systemic vision of the different boarding's of this disciplines currently enters the methodologies most used for companies of the entire world.

During the quarrel of this work I look for to impartially raise the differences, the possible advantages, the disadvantages and comments on the two methodologies in regards to Engineering of Requirements, and in the best practical use of its and concepts.

## SUMÁRIO

<b>1 – INTRODUÇÃO</b> .....	10
<b>2 – JUSTIFICATIVA</b> .....	12
<b>3 – MODELOS DE QUALIDADE</b> .....	13
<b>4 – CONCEITUAÇÃO DE ENGENHARIA DE REQUISITOS</b> .....	16
<b>4.1 – ANÁLISE DE NEGÓCIO</b> .....	21
<b>4.1.1 – ANÁLISE DO PERFIL DOS STAKEHOLDERS</b> .....	21
<b>4.1.2 – ANÁLISE DO CLIENTE</b> .....	22
<b>4.1.3 – ANÁLISE DO CONCORRENTE</b> .....	23
<b>4.1.4 – ANÁLISE DE MERCADO</b> .....	23
<b>4.1.5 – ANÁLISE DA TECNOLOGIA</b> .....	23
<b>4.1.6 – ANÁLISE DO USUÁRIO</b> .....	24
<b>4.2 – VISÃO</b> .....	25
<b>4.2.1 – VISÃO DE NEGÓCIO</b> .....	25
<b>4.2.2 – VISÃO DE SISTEMA</b> .....	25
<b>4.2.3 – VISÃO DE APLICAÇÃO</b> .....	26
<b>4.2.4 – VISÃO DE COMPONENTE</b> .....	26
<b>4.3 – DESENVOLVIMENTO DE REQUISITOS</b> .....	27
<b>4.3.1 – IDENTIFICAÇÃO DOS REQUISITOS</b> .....	27

4.3.2 – REUSO DOS REQUISITOS .....	27
4.3.3 – ANÁLISE DOS REQUISITOS.....	28
4.3.4 – PROTOTIPAGEM DOS REQUISITOS .....	28
4.3.5 – ESPECIFICAÇÃO DOS REQUISITOS.....	29
4.3.6 – VALIDAÇÃO DOS REQUISITOS .....	29
4.4 – GERENCIAMENTO DE REQUISITOS .....	29
5 – C.M.M.I. ....	30
5.1 – A ORIGEM DO C.M.M.I. ....	30
5.2 – O QUE É O C.M.M.I.....	32
5.3 – OS NÍVEIS DO C.M.M.I. ....	34
5.4 – CONCEITUAÇÃO DA ENGENHARIA DE REQUISITOS NO C.M.M.I. ....	37
5.4.1 – A ENGENHARIA DE REQUISITOS NAS ÁREAS DE PROCESSO.....	41
5.4.2 – REQUISITOS NAS VISÕES CONTÍNUA E POR ESTÁGIO .....	43
6 – SPICE .....	46
6.1 – A ORIGEM DO SPICE .....	46
6.2 – O QUE É O SPICE.....	48
6.3 – CONCEITUAÇÃO DA ENGENHARIA DE REQUISITOS NO SPICE .....	60
7 – COMPARAÇÃO ENTRE C.M.M.I. E SPICE .....	63
7.1 – COMPARAÇÃO COM FOCO EM ENGENHARIA DE REQUISITOS .....	71

<b>8 – ESTUDOS DE CASO .....</b>	<b>83</b>
<b>8.1 – IMPLEMENTAÇÃO DA ISO/IEC 15504 (SPICE).....</b>	<b>83</b>
<b>8.2 – IMPLEMENTAÇÃO DO C.M.M.I.....</b>	<b>86</b>
<b>9 – CONCLUSÕES.....</b>	<b>89</b>
<b>10 – CONSIDERAÇÕES FINAIS.....</b>	<b>91</b>
<b>11 – BIBLIOGRAFIA .....</b>	<b>92</b>

## 1 – INTRODUÇÃO

O processo de produção de software constitui-se de uma série de etapas que visam atender os requisitos de um cliente levando-se em consideração prazos e custos da entrega de um produto de qualidade aceitável. Porém, ao longo das décadas verificou-se que é impraticável atender a todos estes objetivos sem a devida organização dos processos e conseqüente maturidade de uma organização, principalmente quanto à disciplina e gerenciamento em nível básico visando à melhoria continuada de todas as etapas da produção de software. Buscando estas melhorias, inúmeras propostas surgiram, entre elas o C.M.M.I. e o Spice.

CMMI significa Capability Maturity Model Integration, isto é, Modelo Integrado de Maturidade e Capacitação e, conforme o S.E.I.<sup>1</sup>, é um framework detalhado que descreve as melhores práticas para o desenvolvimento e manutenção de produtos e serviços incluindo todo o ciclo de vida de um produto em sua concepção, entrega e manutenção. O CMMI foi desenvolvido para definir um ponto inicial para modelos integrados aprimorando as melhores práticas na criação de modelos com o estabelecimento de um framework que possibilita a integração para novos futuros modelos juntamente com a criação de uma forma associada de avaliação de desempenho de produtos.

---

<sup>1</sup> SEI significa Software Engineering Institute (Instituto de Engenharia de Software) – sediado na CMU – Carnegie Mellon University em Pittsburgh, Pennsylvania, Estados Unidos e é um centro de pesquisa e desenvolvimento criado em 1.984 pelo Departamento de Defesa dos Estados Unidos (DoD – Department of Defense).

Spice é um framework regulamento pela ISO<sup>2</sup> que provê a avaliação dos processos de software, sendo que este framework é usado pelas organizações para planejamento, gerenciamento, monitoração, controle e melhoria das aquisições, desenvolvimentos, operações, evoluções e suporte de software.

---

<sup>2</sup> ISO significa International Organization for Standardization (Organização Internacional para Normalização) – sediada em Genebra, Suíça, foi fundada em 1.947 e tem o propósito de desenvolver e promover normas e padrões mundiais que traduzam o consenso dos diferentes países de forma a facilitar o comércio internacional.

## 2 – JUSTIFICATIVA

Uma série de fatores vem ameaçando o sucesso dos processos de produção de software como: os constantes atrasos na entrega dos produtos, os custos fora dos padrões estabelecidos no início do projeto ou mesmo o cancelamento dos projetos antes da sua conclusão.

Segundo estudo feito pelo The Standish Group (2000 CHAOS Report) apenas 16% dos projetos de desenvolvimento de software terminam dentro do cronograma e do custo estimados inicialmente, enquanto 72% falham. Destes 72%, 53% obtiveram estouros no orçamento de cerca de 90%, enquanto 31% destes projetos foram cancelados antes mesmo do seu término. Mesmo observando apenas os projetos que obtiveram êxito, apenas 61% atendeu todas as características solicitadas pelo cliente, o que denota claramente a falta de uma observação mais detalhada na engenharia de requisitos nestas empresas.

Entre os problemas enfrentados pelas equipes de produção de software estão, a falta de participação do usuário, a fraca gerência, a obstrução dos requisitos do usuário, o planejamento inapropriado e as mudanças das especificações.

A pesquisa de frameworks de produção de software com foco em Engenharia de Requisitos torna-se desafiador no que tange a importância de padrões de processo para produção de softwares reconhecidos internacionalmente, obtendo melhor qualidade do produto software, com prazos previstos e custos mensuráveis.

### **3 – MODELOS DE QUALIDADE**

Um modelo é uma abstração de uma situação do mundo real, um ponto de partida, um comparativo que tem por objetivo apresentar uma possível saída podendo utilizar conceitos oriundos de um framework. Um framework é um conjunto de funcionalidades com atribuições específicas e que podem ser usadas em conjunto ou separadamente com o objetivo de alcançar uma meta bem delineada. Um modelo de qualidade é uma possível alternativa de utilização de um framework focando a qualidade de um processo e por conseqüência uma melhoria significativa de um produto.

O CMMI é um framework por disponibilizar inúmeras soluções e práticas para determinados tipos de problema. O CMMI possui em sua concepção quatro modelos previamente definidos que são o SW-CMM que tem por objetivo o desenvolvimento de sistemas exclusivamente de software, o SS-CMM que possui foco em modificações em funções críticas, o SE-CM que se preocupa com o desenvolvimento de sistemas que incluem ou não softwares e o IPD-CMM que visa à integração de produtos, portanto estes são modelos de qualidade por utilizarem as práticas sugeridas no CMMI com o objetivo de melhoria da qualidade de determinada processo.

O CMMI contempla também uma visão de evolução contínua de processos o que possibilita a adequação e geração de novos modelos que se adaptem melhor às organizações. Os modelos do CMMI serão explicados de forma mais detalhada no tópico 5.2 (O que é o CMMI)

O Spice, por sua vez, é um modelo que possui alguns conceitos extremamente semelhantes com a representação contínua do CMMI, tendo diferenças peculiares no tipo de abordagem do problema da melhoria de processos.

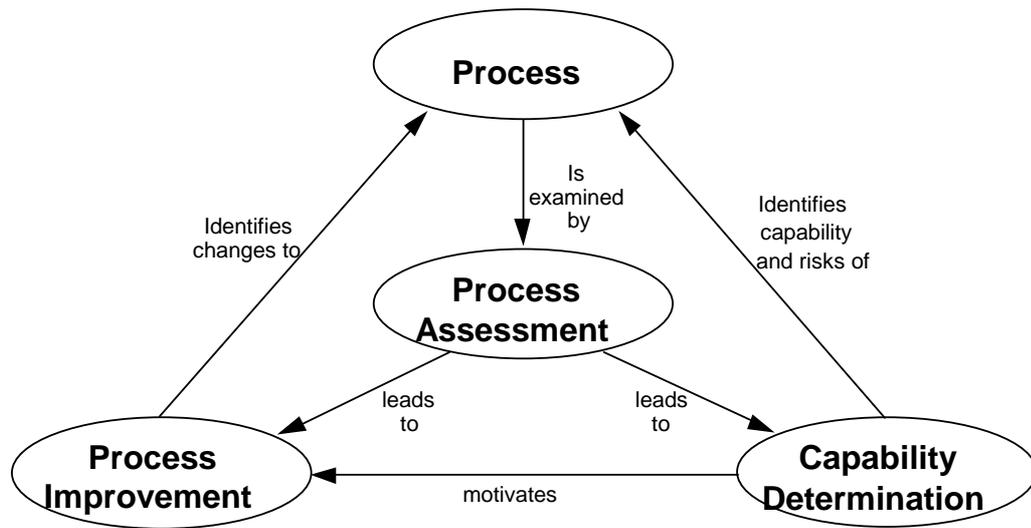


Figura 1.0 – Abordagem Spice

A abordagem do Spice determina que o reconhecimento da maturidade de uma organização é a motivação pela qual a leva à procura da melhoria dos seus processos, isto faz com que identifique as mudanças necessárias para tal melhoria. Para tal reconhecimento de maturidade de uma organização é necessário antes que a mesma identifique o que deve melhorar em seus processos e os riscos de tais mudanças.

Depois de ter seus processos melhorados, estes deverão passar por uma avaliação que conduz à organização a novos caminhos para novos níveis de maturidade e para novas melhorias de processo. Desta forma, a organização é levada a um processo de melhoria contínua de seus processos sem um fim pré-determinado.

Uma das maiores diferenças entre o Spice e o CMMI é exatamente essa noção de fim pré-determinado, isto é, enquanto uma organização certificada CMMI nível 5 está no ápice, isto é, teoricamente deve melhorar mas não há mecanismos para atestar que estas melhorias realmente irão acontecer, ou pior, não há mecanismos para prever que tais melhorias implantadas até o momento não deixaram de ser utilizadas pela empresa no decorrer do tempo, uma organização certificada Spice nível 5, tem a necessidade de ter seus processos sempre aperfeiçoados, uma vez que a certificação ISO tem uma validade

que o CMMI não possui. Muitos profissionais da área vêm contestando as empresas certificadoras por essa razão, uma certificação coerente deve ter um prazo de validade, para que seja um mecanismo que verifique coerentemente se a empresa está realmente melhorando seus processos de forma aplicável e condizente com a realidade.

## 4 – CONCEITUAÇÃO DE ENGENHARIA DE REQUISITOS

A engenharia de requisitos é uma subárea da engenharia de software que visa compreender as necessidades do cliente estudando os processos de definição de requisitos.

Conforme a O.P.F.<sup>3</sup> um engenheiro de requisitos deve possuir perícia, experiência e conhecimento, e suas principais responsabilidades são:

- Projetar as exigências para o sistema;
- Projetar as exigências para a aplicação;
- Projetar exigências reusáveis para um domínio de aplicação;
- Projetar as exigências para um componente reusável.

Algumas características peculiares para o engenheiro de requisitos são a compreensão profunda da teoria e da prática da modelagem de casos de uso, habilidades verbais e escritas incluindo uma excelente comunicação, que deve ser utilizada para esclarecimento de dúvidas relativas às exigências do cliente, alta capacidade de abstração, conhecimento básico do negócio do cliente e do domínio da aplicação, capacidade de controlar ambigüidades e possíveis contradições nas especificações e bons conhecimentos na área de engenharia de sistemas, tecnologia e programação.

---

<sup>3</sup> O.F.P. significa Open Process Framework (Processo de Framework Aberto) e é um meta-modelo de processo que pode ser gerado para uma organização específica formando uma base de conhecimento aberta de exemplos, podendo assim ser utilizado escolhendo tarefas e técnicas específicas para otimização focando um problema.

Conforme a O.F.P. uma série de tarefas deve ser executada de forma iterativa, incremental ou até mesmo paralela durante o processo de engenharia de requisitos.

Podemos classificar as tarefas do processo de engenharia de requisitos como:

- Análise de Negócio;
- Visão;
- Desenvolvimento de Requisitos;
- Gerenciamento de Requisitos.

Na atividade de análise de negócio temos as seguintes tarefas:

- Análise do Perfil dos Stakeholders;
- Análise do Cliente;
- Análise do Concorrente;
- Análise de Mercado;
- Análise da Tecnologia;
- Análise do Usuário;

Na atividade de Visão temos as tarefas:

- Visão de Negócio;
- Visão de Sistema;
- Visão de Aplicação;
- Visão de Componente;

Na atividade de desenvolvimento de requisitos temos:

- Identificação dos Requisitos;
- Reuso dos Requisitos;
- Análise dos Requisitos;
- Prototipagem dos Requisitos;
- Especificação dos Requisitos;
- Validação dos Requisitos.

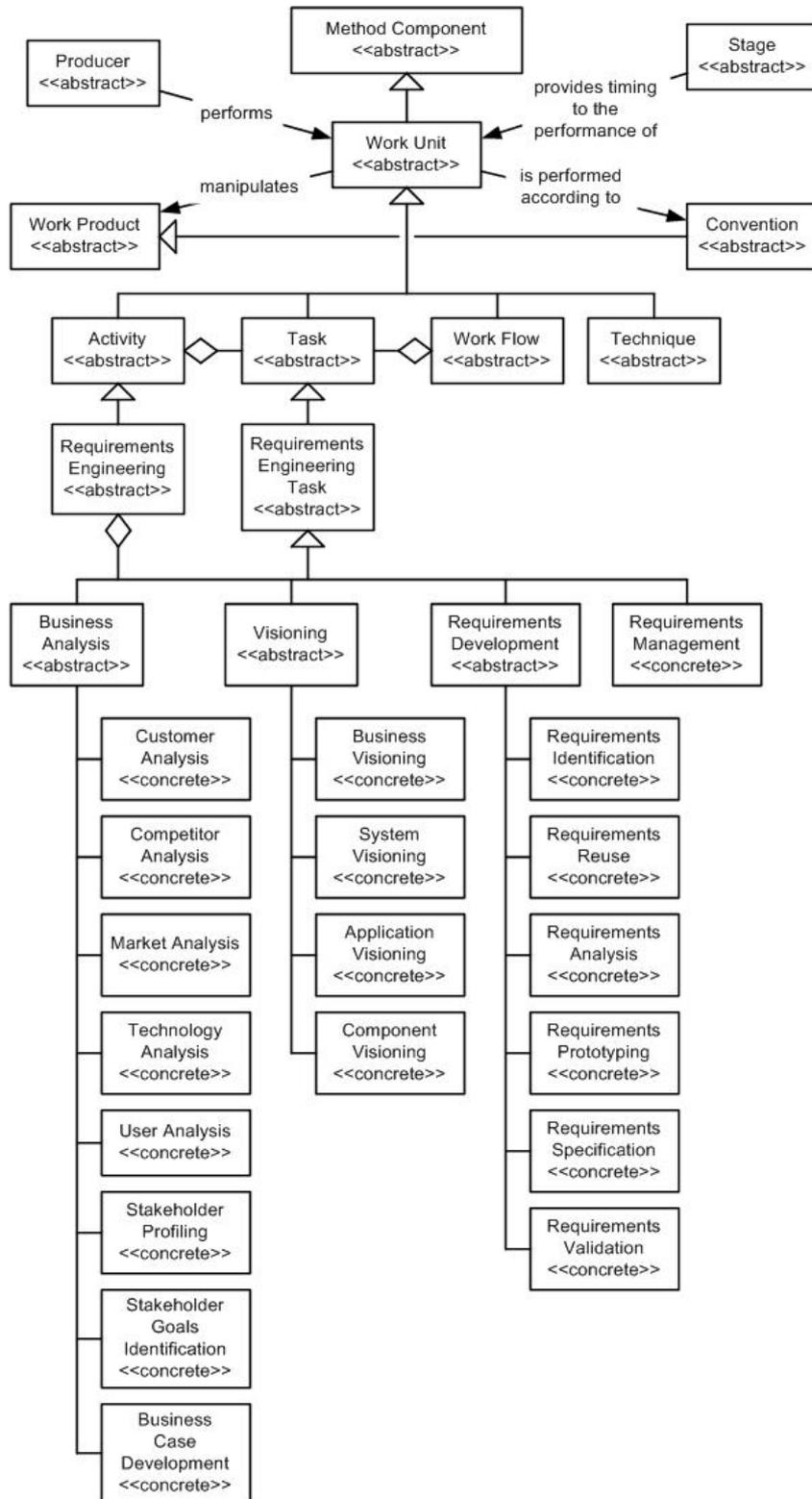


Figura 2 – Modelo de Etapas da Engenharia de Requisitos

## **4.1 – ANÁLISE DE NEGÓCIO**

A análise do negócio tem por objetivo identificar o que é mais importante para o cliente com base em seu negócio.

### **4.1.1 – ANÁLISE DO PERFIL DOS STAKEHOLDERS**

O objetivo da análise do perfil dos stakeholders é entender a organização do cliente por meio do estudo, modelagem e análise de seus sistemas e aplicações. As principais responsabilidades neste momento são: a identificação dos stakeholders, a sua categorização em grupos, a compreensão das suas necessidades, fornecimento de base para as tarefas de análise de cliente, análise do concorrente, identificação de requisitos e análise do usuário.

Dentre as etapas dessa análise estão:

- Identificação das diferentes classes de stakeholders verificando os papéis de cada um na organização e identificando pessoas-chave tais como patrocinadores e organizações.
- Desenvolvimento de um overview sobre esta identificação incluindo informações sobre o contato, descrição do trabalho, relacionamento, esforço, objetivos, necessidades e interesses, responsabilidades e tarefas.
- Verificações de critérios de sucesso para o esforço.
- Verificação da porcentagem do grupo de stakeholders.

- Nas reuniões com os stakeholders uma série de atribuições com relação aos usuários deve ser verificada como, por exemplo, idade média, nível de instrução, linguagem nativa, expectativas, pedidos de mudança, experiência geral e frequência de uso do computador, tipos de computadores utilizados, browsers usados, redes usadas, experiência com o produto a ser desenvolvido, plataforma de uso do produto, conhecimentos e familiaridades com o produto, etc.

#### **4.1.2 – ANÁLISE DO CLIENTE**

A análise do cliente trata-se da primeira tarefa na análise de negócio, ou seja, a primeira tarefa no desenvolvimento dos requisitos na produção de um sistema e para tanto, é necessário que a organização tenha traçado uma estratégia de negócio para o desenvolvimento do produto.

O conhecimento do cliente, por meio de uma análise minuciosa de suas características é fundamental para a organização obter meios de negociação, entender o que deve ser prioritário, o que pode ser postergado, o que o cliente considera de maior e de menor importância, fazendo com que a organização possa se adequar ao cliente.

As típicas responsabilidades na fase de análise do cliente são:

- Análise do modelo de negócio do cliente;
- Análise do relacionamento da organização com o cliente;
- Análise da tecnologia usada pelo cliente;
- Identificação das melhorias de planejamento relevantes ao cliente;
- Produção de um relatório completo da análise do cliente.

### **4.1.3 – ANÁLISE DO CONCORRENTE**

A análise do concorrente define algumas responsabilidades com base nos concorrentes da organização do cliente:

- Identificação de concorrentes com base no tipo de negócio do cliente;
- Verificação e análise do perfil destes possíveis concorrentes;
- Abertura da possibilidade de mudanças de negócio para seu cliente.

### **4.1.4 – ANÁLISE DE MERCADO**

A análise de mercado pressupõe a verificação dos tipos de mercado em que o cliente participa, os tipos de soluções adotadas pelo cliente para obter competitividade neste mercado e a produção de uma análise exata da atual situação do segmento de mercado.

### **4.1.5 – ANÁLISE DA TECNOLOGIA**

A análise tecnológica tenta entender principalmente as estratégias adotadas pelo cliente para utilização da tecnologia. Esta fase procura o entendimento das exigências e limitações inerentes ao cliente e o tipo de controle de qualidade que o mesmo faz em seus produtos, incluindo tecnologias potenciais para uso, identificação de tecnologias

críticas, levantamento breve de um histórico sobre as tecnologias já utilizadas, maturidade tecnológica, produtividade, custos, nível de risco adotado, desenvolvendo, por fim, um relatório de análise da tecnologia.

#### **4.1.6 – ANÁLISE DO USUÁRIO**

A análise do usuário preocupa-se em identificar os diferentes tipos de usuário da organização do cliente compreendendo suas necessidades assim como as suas tarefas relevantes produzindo maneiras diferentes e criativas de melhorar a execução de suas tarefas inclusive atuando no design de interfaces com o mesmo.

É identificado por uma série de etapas a seguir:

- Identificar o usuário;
- Obter informações sobre o usuário;
- Análise das tarefas do usuário;
- Produção de um documento de análise do usuário.

## **4.2 – VISÃO**

O principal objetivo de uma visão da organização do cliente em seu foco de negócio ou da visão da aplicação no seu ambiente sistêmico é absorver exatamente qual o tipo de cliente com que estamos trabalhando e com isso entender o que ele espera com o desenvolvimento da nova aplicação.

### **4.2.1 – VISÃO DE NEGÓCIO**

A visão de negócio tem como propósito obter um consenso a respeito do negócio do cliente, além de determinar e documentar os desafios e oportunidades futuros, os novos valores, os objetivos, critérios de sucesso, mercado alvo, produtos, serviços, as novas iniciativas, tecnologias e aplicações de negócio da organização do cliente.

### **4.2.2 – VISÃO DE SISTEMA**

A visão de sistema tem como objetivo obter um consenso entre os stakeholders sobre o contexto da aplicação a ser desenvolvida no contexto do sistema referenciando possíveis integrações, comunicações e possibilidades de problemas e oportunidades com relação à definição do sistema como um todo.

### **4.2.3 – VISÃO DE APLICAÇÃO**

A visão de aplicação provê um consenso entre os stakeholders com relação à aplicação a ser desenvolvida e tem como objetivo a determinação e documentação das funcionalidades, dos objetivos, da qualidade da aplicação mencionando também problemas e oportunidades com relação à definição da aplicação.

### **4.2.4 – VISÃO DE COMPONENTE**

Em um ambiente componentizado a visão de componente provê um consenso sobre como serão desenvolvidos os componentes relacionando possíveis problemas e oportunidades para a reutilização dos mesmos.

## **4.3 – DESENVOLVIMENTO DE REQUISITOS**

Através do desenvolvimento de requisitos é possível, criando diversas versões de um mesmo software, criar uma maior aproximação do que o cliente realmente deseja e o desenvolvedor deve fazer. Um dos melhores caminhos para isto é através de protótipos simples que demonstrem cada área funcional do software ao cliente.

### **4.3.1 – IDENTIFICAÇÃO DOS REQUISITOS**

Uma das principais funções da fase de identificação de requisitos é a verificação das exigências 'cruas' por parte do cliente, isto é, a verificação das funcionalidades que precisam estar inclusas no software, aquelas que possuem maior importância e são lembradas rapidamente pelo cliente. Esta fase também se caracteriza fundamentalmente pelas entrevistas com os representantes do cliente, não se esquecendo de verificar também possíveis exigências potenciais como qualidade e praticidade.

### **4.3.2 – REUSO DOS REQUISITOS**

A fase de reuso de requisitos atenta para prováveis requisitos que já tenham sido definidos em projetos e/ou etapas anteriores, que possam ser reutilizados neste momento. No tocante a outros projetos, podem-se verificar relevâncias referentes a projetos on-line, por exemplo, ou projetos que foram feitos para o mesmo tipo de mercado do cliente. Quanto às etapas anteriores, análises do cliente, análises da visão de negócio e do

usuário podem ser muito úteis nesta fase do projeto. É útil também que seja feita uma verificação de quais requisitos podem ser reusáveis e, se necessário, adaptar requisitos anteriores para que estes possam então ser reutilizados.

### **4.3.3 – ANÁLISE DOS REQUISITOS**

A análise de requisitos preocupa-se principalmente com o estudo, categorização, decomposição, modelagem e refinamento dos requisitos. Nesta fase, os textos informais devem servir de base para a criação de novos documentos. A negociação de prioridades deve ser considerada e todas as suposições relacionadas devem ser verificadas. Verifique possíveis ambigüidades e a praticabilidade, exatidão e consistências dos requisitos recolhidos. É necessária a certeza de que os requisitos foram totalmente entendidos e amplamente analisados.

Os diagramas de contexto, de dados, de função, de objeto, de processo, de qualidade, de estado e de caso de uso também são construídos nesta fase.

### **4.3.4 – PROTOTIPAGEM DOS REQUISITOS**

A prototipagem dos requisitos ajuda na verificação de possíveis erros em requisitos existentes, no entendimento de novos requisitos identificando também erros em requisitos existentes. A análise de custo / benefício é identificada e com isso as decisões com respeito à prioridade são tomadas.

### **4.3.5 – ESPECIFICAÇÃO DOS REQUISITOS**

A especificação dos requisitos tem por objetivo gerar e / ou atualizar e publicar os requisitos aprovados pelo cliente. Esta fase também se preocupa com a revalidação das documentações do projeto em busca de possíveis falhas e itens não preenchidos que possam estar nas documentações geradas anteriormente.

### **4.3.6 – VALIDAÇÃO DOS REQUISITOS**

A validação dos requisitos se preocupa com a exatidão das exigências analisadas, determinando se as mesmas estão corretamente identificadas, analisadas e especificadas. Nesta fase, verifica-se se os requisitos são uma descrição aceitável do sistema, do software ou do componente a ser desenvolvido.

## **4.4 – GERENCIAMENTO DE REQUISITOS**

O gerenciamento de requisitos se preocupa fundamentalmente com o armazenamento dos requisitos e seus atributos em repositórios com controle total de acesso aos mesmos, a negociação com os stakeholders para a eliminação de possíveis inconsistências nos requisitos e as prioridades entre possíveis alterações e o relato do status do desenvolvimento de cada requisito.

## **5 – C.M.M.I.**

### **5.1 – A ORIGEM DO C.M.M.I.**

A Engenharia de Software tem como principal preocupação o estudo das disciplinas envolvidas na especificação, desenvolvimento, gerenciamento e evolução de sistemas de softwares visando atender as necessidades das organizações. Sommerville (1995) define o processo de software como um conjunto de atividades e resultados associados que produzem um produto de software.

O CMMI é um dos últimos resultados de uma série de evoluções referente á área de qualidade de software. A evolução dos conceitos de qualidade iniciou-se nos anos trinta com Walter Shewhart desenvolvendo alguns princípios para controle estatístico de processos. Shewhart empregava técnicas de probabilidade e gráficos de controle para detectar variações no processo de fabricação que poderia gerar produtos defeituosos. No início dos anos oitenta, Philip Crosby desenvolveu a chamada grade de maturidade da qualidade que consistia no desenvolvimento do pensamento da qualidade, desde a incerteza passando pelo despertar, esclarecimento e sabedoria até a plena certeza. Em 1986, Walter Humphrey, alterou a idéia da grade de maturidade da qualidade de Crosby incluindo o conceito de níveis de maturidade. (Radice 85).

Em 1987 o S.E.I. lançou uma breve descrição da estrutura da maturidade de processo [Humphrey 87a] e um questionário de maturidade [Humphrey 87b] juntamente com dois métodos de avaliação da maturidade do processo de software: Avaliação do Processo de Software (Software Process Assessment [Dunaway 96]) e Avaliação da Capacitação de Software (Software Capability Evaluation [Byrnes 96]).

Após quatro anos de experiência com a estrutura de maturidade de processo, o S.E.I. evoluiu a estrutura de maturidade de processo para o chamado Capability Maturity Model for Software – C.M.M. – [Paulk 91]. Em fevereiro de 1993 foi lançada a versão 1.1 do C.M.M. como resultado de recomendações da comunidade de software. Em 1994 foi lançado o livro The Capability Maturity Model – Guidelines for Improving the Software Process [Paulk 95].

Basicamente o C.M.M.I. é uma evolução do SW-C.M.M. (o C.M.M. voltado para o desenvolvimento de software) e entre outras pequenas modificações incluiu um novo conceito que se refere à representação contínua.

A idéia da representação contínua é a de oferecer maior flexibilidade à organização. Uma organização deve escolher as áreas de processo que trabalhará desde que estas estejam perfeitamente alinhadas com os objetivos de negócio da empresa. Basicamente quando uma organização alcança uma melhora em uma área de processo pode escolher se ajusta ao próximo nível de capacidade ou se aumenta o escopo do nível atual em que se encontra, incluindo as áreas de processo que possui melhor capacidade.

## 5.2 – O QUE É O C.M.M.I.

Conforme o Sei, o CMMI consiste de um framework de produtos que provêm a habilidade de gerar múltiplos modelos, treinamentos associados e materiais para avaliação. Muito embora o CMMI esteja fortemente fundamentado em software, contempla também os desenvolvimentos multidisciplinares, cobrindo outras áreas do desenvolvimento de sistemas. Até o momento, são quatro as disciplinas incorporadas ao CMMI:

- SW-CMM (The Capability Maturity Model for Software) – A Engenharia de Software cobre o desenvolvimento de sistemas de software focando em aproximações sistemáticas e disciplinadas determinando a aplicação do desenvolvimento, da operação e da manutenção do software;
- SS-CMM (The Supplier Sourcing Capability Model) – Visa o atendimento de modificações e a execução de funções específicas críticas que podem vir a ser necessária em um produto;
- SE-CM (The Systems Engineering Capability Model) – A Engenharia de Sistema cobre o desenvolvimento total de sistemas podendo ou não incluir um software. Os coordenadores dos sistemas focalizam as necessidades e as expectativas de seus clientes na elaboração dos produtos e na durabilidade dos mesmos;
- IPD-CMM (The Integrated Product Development Capability Maturity Model) – O desenvolvimento de produtos integrados é uma aproximação sistemática na tentativa de uma colaboração das partes relevantes durante todo o ciclo de vida de um produto buscando satisfazer as necessidades, expectativas e exigências dos clientes.

O CMMI contempla também as chamadas áreas-chave de processo. Uma área-chave é um conjunto de melhores práticas relacionadas a uma determinada área específica, que quando executada com outros procedimentos satisfaz os objetivos considerados importantes para uma melhoria significativa da área.

### 5.3 – OS NÍVEIS DO C.M.M.I.

Diferentemente da abordagem seqüência, a qual é necessária um estudo da regra de negócio da organização para a determinação de quais áreas-chave serão abordadas inicialmente, a abordagem por estágios do CMMI possui cinco níveis diferentes e distintos de maturidade: O nível um que é considerado um nível caótico, aonde a empresa não possui um ambiente estável, em que geralmente depende de atos heróicos dos seus colaboradores, o nível dois se caracteriza por ter os requisitos, os processos, os produtos e os serviços controlados e o status dos produtos e entrega visíveis à gerência em alguns pontos definidos, o nível três é caracterizado por processos bem caracterizados e compreendidos aonde são descritos padrões nos procedimentos, nas ferramentas e nos métodos, o nível quatro possui instrumentos para verificação estatística dos processos e é compreendido durante toda a vida do processo, baseando-se nas necessidades dos clientes, dos usuários, da organização e dos implementadores dos processos e o nível cinco que possui como objetivo a otimização, ou seja, a melhoria contínua dos processos, aonde essas melhorias são selecionadas baseando-se na compreensão do retorno previsto pela organização e pelo impacto que a mudança acarretará.

Segundo a I.S.D. <sup>4</sup>, o maior problema com os profissionais brasileiros é o fato de subestimarem as mudanças e a mentalidade de que as teorias geralmente não dão certo na prática.

O nível dois do C.M.M.I. é geralmente identificado como um dos mais difíceis de ser alcançado pelo fato de consistir de uma série de mudanças comportamentais dos envolvidos, da padronização e da disciplina em seguir uma série de modelos e conceitos.

---

<sup>4</sup> I.S.D. – Integrated System Diagnostics Brasil é uma consultoria com atuação na América do Sul focada com qualidade de processo baseada em modelos reconhecidos internacionalmente.

O nível dois do C.M.M.I. aborda principalmente a questão gerencial e suas áreas-chave são as seguintes:

- Gerenciamentos de Requisitos;
- Planejamento de Projetos;
- Monitoração e Controle de Projetos;
- Gerenciamento de Acordos com Fornecedores;
- Medição e Análise;
- Garantia de Qualidade do Processo e Produto;
- Gerência de Configuração.

O nível três possui as seguintes áreas-chave:

- Desenvolvimento dos Requisitos;
- Soluções Técnicas;
- Integrações do Produto;
- Verificações e Validações;
- Processo e treinamento organizacional;
- Gerência de projeto integrada para IPPD;
- Gerência de risco;
- Integração de equipe;
- Gerência de fornecedores;

- Análise e resolução de decisões;
- Ambiente organizacional de integração.

O nível quatro do C.M.M.I. possui as seguintes áreas-chave:

- Performance do processo organizacional;
- Gerenciamento quantitativo de projetos.

O último nível C.M.M.I. foca as seguintes áreas:

- Inovação e distribuição organizacional;
- Estudos de caso e definições.

## **5.4 – CONCEITUAÇÃO DA ENGENHARIA DE REQUISITOS NO C.M.M.I.**

Na abordagem por estágios, o C.M.M.I. sugere que a engenharia de requisitos tenha seu gerenciamento implantado no nível dois e o seu desenvolvimento no nível três.

Na definição de gerenciamento de requisitos no nível dois, o C.M.M.I. define alguns objetivos específicos para a verificação das inconsistências com os planos do projeto e os produtos de trabalho definidos.

- Obter um entendimento dos requisitos: Este objetivo tem como preocupação a averiguação dos canais de comunicação para condução dos requisitos visando a compreensão adequada dos requisitos. Seus principais produtos de trabalho são uma lista de critérios para distinção dos requisitos apropriados, critérios para avaliação e aceitação de requisitos, resultado das análises dos critérios, concordância com os requisitos.
- Obter comprometimento com os requisitos: Este objetivo assegura um comprometimento de todos os participantes do projeto com relação aos requisitos atuais, com as evoluções dos mesmos e com as mudanças resultantes destas evoluções. Os principais produtos de trabalho são: o documento de avaliações de impacto e o documento de comprometimento com novos requisitos e com suas mudanças.
- Gerenciar as mudanças dos requisitos: Este objetivo verifica os impactos de novos requisitos ou da mudança deles nos requisitos já existentes, sendo necessário contudo, possuir fontes confiáveis. Os produtos deste trabalho são os documentos de status dos requisitos, a base de dados de requisitos e a base de decisão de requisitos.
- Manter a traceabilidade bidirecional dos requisitos: Este objetivo tem a intenção de manter a traceabilidade dos requisitos para a necessidade de uma fonte de requisitos

anterior a alguma mudança, esta rastreabilidade é particularmente importante na condução da avaliação de impacto de mudanças.

- Identificar inconsistências entre requisitos e produtos de trabalho: Este objetivo se preocupa com possíveis inconsistências entre os requisitos existentes, os produtos de trabalho gerados e as suas correções.

Na definição do desenvolvimento dos requisitos, abordado no nível três do C.M.M.I., os objetivos específicos são detalhados em três práticas genéricas que são desenvolvimento de requisitos do cliente, desenvolvimento de requisitos do produto e análise e validação dos requisitos.

A base do desenvolvimento de requisitos de cliente está nos requisitos tanto do cliente, quanto dos usuários finais, dos fornecedores, construtores e dos testadores. Frequentemente são usadas representações para ajudar a resolver conflitos. Seus objetivos específicos são:

- Coletar necessidades: Coletar necessidades vai além de verificar requisitos explícitos, é necessário um trabalho de verificação de requisitos implícitos e suas inter-relações com requisitos já existentes. Uma série de técnicas pode ser utilizada para a coleta destes dados tais como entrevistas, questionários técnicos, 'brainstorming', observações de padrões já existentes, análise do negócio do cliente, engenharia reversa, etc.

- Desenvolver os requisitos do cliente: Este objetivo tenta consolidar as informações transmitidas pelo cliente, resolver os conflitos e documentar os requisitos resolvidos. Os produtos de trabalho são os requisitos do cliente, verificação e validação do escopo.

No desenvolvimento de requisitos de produto leva-se em conta também o seu ciclo de vida. Tal desenvolvimento depende de fatores tais como o 'baseline' de requisitos, a arquitetura selecionada para o projeto e pelas considerações de regra de negócio dos colaboradores. Os requisitos são reexaminados e a arquitetura funcional revalidada tendo

o refinamento do desenvolvimento dos requisitos do produto. Seus objetivos específicos são:

- Estabelecer requisitos de produto e de produto-componente: Os requisitos de produto devem ser expressos em termos técnicos mesmo que os requisitos de cliente não sejam expressos desta forma, pois os requisitos de produto podem ser utilizados em decisões de projeto. Os requisitos de produto são direcionados à satisfação do cliente, do negócio e dos objetivos do projeto.

- Alocação de requisitos de produto-componente: Este objetivo preocupa-se com o desempenho do produto e o escopo do projeto definindo, inclusive, a limitação de responsabilidades no caso de uma maior complexidade seja necessária no desenvolvimento de um ou mais componentes. Seus produtos de trabalho são as alocações e deslocamentos de requisitos, requisitos e escopo de projeto e relações entre requisitos derivados.

- Requisitos de interface: Este objetivo controla as relações identificadas entre componentes da arquitetura do produto, sendo também parte integral da definição da arquitetura.

A análise e validação dos requisitos pressupõem a validação dos requisitos no ambiente pretendido. Tais análises são determinadas para verificar o impacto dos requisitos no ambiente operacional do cliente. A praticabilidade, o custo, a estratégia de aquisição e o tamanho potencial de mercado são alguns dos pontos que devem ser validados nesta etapa.

- Estabelecer conceitos operacionais e cenários: Um cenário pode ser definido como uma seqüência de eventos que ocorre no uso de um produto, no entanto, um conceito operacional depende geralmente da solução do projeto e do cenário. Conceitos operacionais são desenvolvidos com base na análise dos requisitos de um projeto. Tais conceitos são refinados quando as decisões da solução já foram feitas e os requisitos foram bem detalhados. Os produtos de trabalho são: conceito operacional, a instalação,

operação, manutenção e suporte de um produto, conceitos de eliminação, casos de uso, cenários de linha de tempo, novos requisitos.

- Estabelecer uma definição de funcionalidade requerida: Também conhecida como análise funcional é a descrição do que o produto deve fazer. Esta definição pode incluir entradas, saídas, ações ou qualquer outra informação de comunicação. Em projetos orientados a objetos, análise funcional relaciona-se com a definição dos serviços. As definições de funções, agrupamentos lógicos e associações com requisitos são também chamados de arquitetura funcional. Produtos de trabalho: Arquitetura funcional, diagramas de atividade e casos de uso, análise orientada a objetos com identificação dos serviços.

- Analisar requisitos: Os requisitos analisados fornecem a base para requisitos com riqueza de detalhes, melhorando a compreensão das funcionalidades de níveis mais elevados. Produtos de trabalho: Relatório de defeitos, propostas de mudanças para resolução dos defeitos, chave de requisitos, medidas de performance técnica.

- Analisar os requisitos com foco em equilíbrio: Este objetivo tem como foco auxiliar os stakeholders a confrontar custos, performances, funcionalidades, reuso de componentes, manutibilidade e riscos do sistema. Produtos de trabalho: Avaliação dos riscos relacionados aos requisitos.

- Validar os requisitos com métodos compreensíveis: Este objetivo procura validar os requisitos afim de ganhar a confiança de que os requisitos satisfaçam as necessidades e expectativas dos stakeholders e o seu desenvolvimento será bem sucedido. Esta atividade deve ser integrada com atividade de gerência de riscos. Tais validações podem ser concretizadas com simulações e/ou protótipos. Produtos de trabalho: Arquivo dos métodos de análise e seus resultados.

## 5.4.1 – A ENGENHARIA DE REQUISITOS NAS ÁREAS DE PROCESSO

As duas áreas de processo que estão diretamente ligadas à engenharia de requisitos foram abordadas no CMMI na categoria “Engenharia” e são: “Desenvolvimento de Requisitos” e “Gerenciamento de Requisitos”. A categoria Engenharia se preocupa ainda com as áreas “Soluções Técnicas”, “Integração de Produtos”, “Verificação” e “Validação”.

A categoria Engenharia no CMMI é responsável por cobrir as atividades de desenvolvimento e manutenção compartilhadas através das disciplinas da engenharia. As áreas do processo de Engenharia foram desenvolvidas usando-se a terminologia geral da engenharia de modo que toda a disciplina técnica envolvida no processo de desenvolvimento do produto possui suporte a estratégias de melhorias orientadas a produto.

As áreas de processo da Engenharia se aplicam para o desenvolvimento de qualquer produto ou serviço no domínio de desenvolvimento da engenharia, como produtos de software, de hardware, serviços ou processos.

O Desenvolvimento de Requisitos na Engenharia identifica as necessidades do cliente e traduz essas necessidades em requisitos de produto. A gama de requisitos para um produto é analisada produzindo-se soluções conceituais de alto nível. Esta gama de requisitos é então alocada para estabelecer um conjunto inicial de requisitos de componentes para o produto. Outros requisitos podem ser definidos através da derivação dos requisitos originais sendo incluídos também no conjunto dos requisitos iniciais. Estes requisitos de produto e de componente são, dessa forma, claramente descritos verificando-se a visão da performance do produto, as características do design, a verificação dos requisitos e o bom entendimento dos desenvolvedores.

O Desenvolvimento de Requisitos aborda os requisitos com foco na solução técnica do problema, aonde os requisitos são convertidos em uma arquitetura de produtos, design de

componentes de produto e a produção dos componentes em si. Os requisitos são fornecidos também para a área de processo de Integração de Produto, onde os componentes são combinados e as interfaces verificadas para assegurar que estão de acordo com os requisitos especificados.

Já a área de processos Gerenciamento dos Requisitos mantém os requisitos, isto é, descrevendo atividades com foco na obtenção e controle das mudanças dos requisitos e levando em consideração planos relevantes, a manutenção e a atualização das informações provendo traceabilidade de requisitos do cliente para o produto e para os componentes do produto.

O Gerenciamento de Requisitos se preocupa com os impactos das mudanças de requisitos nos planos do projeto, nas atividades e nos produtos de trabalho. Este ciclo de mudanças geralmente reflete em outras áreas de processo da Engenharia, pois os requisitos são dinâmicos e oferecem uma seqüência recursiva de eventos. O gerenciamento de requisitos é fundamental para o controle disciplinado do processo de engenharia de projetos.

## 5.4.2 – REQUISITOS NAS VISÕES CONTÍNUA E POR ESTÁGIO

O propósito da área de processo Desenvolvimento de Requisitos é produzir e analisar requisitos de cliente, requisitos de produto e requisitos de componentes de produtos, incluindo as exigências dos stakeholders incluindo as fases pertinentes do ciclo de vida (aceitação, critérios de teste) e atributos de produto (segurança, confiabilidade, manuteabilidade).

O desenvolvimento de requisitos inclui as seguintes atividades:

- Definição, análise, validação e comunicação das necessidades do cliente, expectativas, e variáveis para obtenção dos requisitos do cliente que constituem no entendimento do que satisfará os stakeholders;
- Coleção e coordenação das necessidades dos stakeholders;
- Desenvolvimento do ciclo de vida dos requisitos de produto;
- Estabelecimento dos requisitos do cliente;
- Estabelecimento do produto inicial e consistência dos requisitos de componentes do produto com os requisitos do cliente.

A identificação de todos os requisitos de cliente é satisfatória para nivelar o produto de software com os requisitos específicos do projeto, refinando os requisitos do produto e dos componentes do produto.

Abaixo tabela de comparação entre a Representação Contínua e a Representação por Estágio com foco na área de Requisitos.

<b>Representação Contínua</b>	<b>Representação por Estágio</b>
<u>SG 1 – Desenvolvimento dos Requisitos de Cliente</u>	<u>SG 1 – Desenvolvimento dos Requisitos do Cliente</u>
SP 1.1-1 – Coletar Necessidades dos Stakeholders	
SP 1.1-2 – Elucidar Necessidades	SP 1.1-2 – Elucidar Necessidades
SP 1.2-1 – Desenvolver Requisitos do Cliente	SP 1.2-1 – Desenvolver Requisitos do Cliente
<u>SG 2 – Desenvolvimento dos Requisitos de Produto</u>	<u>SG 2 – Desenvolvimento dos Requisitos de Produto</u>
SP 2.1-1 – Estabelecer Requisitos de Produtos e Componentes de Produto	SP 2.1-1 – Estabelecer Requisitos de Produtos e Componentes de Produto
SP 2.2-1 – Alocar Requisitos de Componentes de Produto	SP 2.2-1 – Alocar Requisitos de Componentes de Produto
SP 2.3-1 – Identificar Requisitos de Interface	SP 2.3-1 – Identificar Requisitos de Interface
<u>SG 3 – Análise e Validação de Requisitos</u>	<u>SG 3 – Análise e Validação de Requisitos</u>
SP 3.1-1 – Estabelecer conceitos operacionais e cenários	SP 3.1-1 – Estabelecer conceitos operacionais e cenários
SP 3.2-1 – Estabelecer uma definição de funcionalidades requeridas	SP 3.2-1 – Estabelecer uma definição de funcionalidades requeridas
SP 3.3-1 – Analisar Requisitos	SP 3.3-1 – Analisar Requisitos
SP 3.4-3 – Analisar Requisitos para contrapeso	SP 3.4-3 – Analisar Requisitos para contrapeso
SP 3.5-1 – Validar Requisitos	
SP 3.5-2 – Validar Requisitos com métodos compreensíveis	SP 3.5-2 – Validar Requisitos com métodos compreensíveis
<i>GG 1 – Conseguir Objetivos específicos</i>	
GP 1.1 – Execute melhores práticas	
<i>GG 2 – Instituir um processo gerenciado</i>	<i>GG 2 – Instituir um processo gerenciado</i>
GP 2.1 – Estabeleça uma política organizacional	GP 2.1 – Estabeleça uma política organizacional
GP 2.2 – Planeje o processo	GP 2.2 – Planeje o processo
GP 2.3 – Provenha recursos	GP 2.3 – Provenha recursos
GP 2.4 – Atribua responsabilidades	GP 2.4 – Atribua responsabilidades

GP 2.5 – Treine pessoas	GP 2.5 – Treine pessoas
GP 2.6 – Gerencie as configurações	GP 2.6 – Gerencie as configurações
GP 2.7 – Identifique e envolva os stakeholders relevantes	GP 2.7 – Identifique e envolva os stakeholders relevantes
GP 2.8 – Monitore e controle os processos	GP 2.8 – Monitore e controle os processos
GP 2.9 – Avalie objetivamente a aderência	GP 2.9 – Avalie objetivamente a aderência
GP 2.10 – Revise os status com maior nível de gerenciamento	GP 2.10 – Revise os status com maior nível de gerenciamento
<i>GG 3 – Instituir um processo definido</i>	<i>GG 3 – Instituir um processo definido</i>
GP 3.1 – Estabeleça um processo definido	GP 3.1 – Estabeleça um processo definido
GP 3.2 – Colete informações de melhoria	GP 3.2 – Colete informações de melhoria
<i>GG 4 – Instituir um processo gerenciado quantitativamente</i>	
GP 4.1 – Estabeleça objetivos quantitativos para o processo	
GP 4.2 – Estabilize o desempenho dos sub-processos	
<i>GG 5 – Instituir um processo otimizado</i>	
GP 5.1 – Assegure a melhoria contínua do processo	
GP 5.2 – Corrija as causas raiz dos problemas	

Tabela 1 – Comparativo entre as abordagens contínua e por estágios, em Requisitos

- SG – Specified Goals (Objetivos Específicos).
- GG – Generic Goals (Objetivos Genéricos).
- SP – Specified Practices (Práticas Específicas).
- GP – Generic Practices (Práticas Genéricas).

## **6 – SPICE**

### **6.1 – A ORIGEM DO SPICE**

A resolução 144 aprovada em junho de 1.991 pela ISO/IEC JTC 1/SC7 aprovava um período de estudo e investigação de requisitos para uma padronização das melhorias de processo de desenvolvimento de software. Com esses estudos a ISO obteve as seguintes conclusões:

- A certeza da necessidade de um compromisso internacional para uma padronização dos processos de melhoria do desenvolvimento de software;
- A necessidade da verificação das experiências dos usuários para um relatório técnico de pré-publicação para uma base para tal padrão internacional.
- A necessidade da criação de uma consciência no mercado para um padrão de processo de desenvolvimento de software.

O artigo foi aprovado em janeiro de 1.993 e a partir de junho do mesmo ano foi estabelecido o programa de trabalho aprovado pela ISO/IEC para uma padronização internacional dos processos de desenvolvimento de software, fundamentando então suas pesquisas nas conclusões mencionadas anteriormente. Em junho de 1.995 o grupo terminou então os primeiros esboços do seu trabalho.

As diretrizes orientadoras de ISO/IEC indicaram um relatório técnico que consistia das seguintes partes:

- 1 – Conceitos e guia introdutório;
- 2 – Um modelo de gerenciamento de processos;

3 – Processos de avaliação;

4 – Guia de melhoria de condução;

5 – Construção, seleção e uso de instrumentos e ferramentas de avaliação;

6 – Qualificação e treinamento de assessores;

7 – Guia para melhor uso dos processos;

8 – Guia para determinação de potencialidade de fornecedor;

9 – Vocabulário.

## 6.2 – O QUE É O SPICE

O Spice (ISO/IEC 15504) é um modelo que, como o C.M.M.I., possui foco na melhoria dos processos de desenvolvimento de software e a determinação da capacidade de processos de uma organização. Uma melhoria neste processo exige um exame minucioso para decisão e determinação de qual etapa é mais importante e deve ser implantada em primeiro lugar em uma organização. Esta decisão tem relação direta com o contexto, a regra de negócio e os riscos de sua implantação.

Spice prega algumas boas práticas fundamentais para uma boa engenharia de software. A arquitetura do framework organiza essas práticas em duas abordagens distintas:

- Práticas base com foco em atividades essenciais para processos específicos agrupados por tipo de atividade;
- Práticas genéricas para qualquer processo que representa fundamentalmente as atividades necessárias para a gestão do processo.

O Spice possui processos bem definidos para cada etapa do desenvolvimento e/ou manutenção de um produto de software, dentro de cada processo são delineadas uma série de atividades específicas para a conclusão da implantação de cada processo. Os processos definidos pelo Spice são:

- Cliente-Fornecedor, o qual consiste dos processos que possuem relação direta com o cliente;
- A Engenharia que cuida especificamente da especificação, execução, manutenção e documentação de um sistema de software;
- O Projeto que foca o controle dos recursos do projeto;
- Suporte que possui ênfase numa melhor performance de sustentação para o projeto;

- Organização que controlam os processos que estabelecem os objetivos de negócio da organização desenvolvendo processos, produtos e recursos.

Ao contrário do C.M.M.I., o Spice possui seis níveis de capacidade (de 0 a 5) de uma organização definidos conforme segue:

- Nível 0 – Não executado: É a organização que não possui nenhum tipo de processo sendo definido também como uma organização caótica.

- Nível 1 – Informalmente executado: Trata-se da organização que possui algumas práticas básicas porém não documentadas e sem planejamentos rigorosos, geralmente o êxito em seus projetos depende de esforços individuais dos colaboradores.

- Nível 2 – Planejado e rastreado: Os produtos de trabalho começam a apresentar uma especificação padrão e requisitos bem delineados.

- Nível 3 – Bem definido: Neste nível a organização já possui planejamento e organização usando processos padronizados.

- Nível 4 – Controlado quantitativamente: Neste nível a organização já possui ferramentas para medição da melhoria dos seus processos tendo o desempenho objetivamente gerenciado.

- Nível 5 – Melhoria contínua: A organização possui um 'feedback' do seu desempenho e implementa novas idéias com inovações tecnológicas. A idéia é que a organização conquiste um nível de controle e gerenciamento que possa detectar falhas em seus processos e corrigi-las.

O Spice possui mecanismos de pontuação que possuem uma escala ordenada de quatro valores, escolhidos de acordo com um percentual de atendimento aos requisitos do atributo de processo. De 0 a 15% é declarado como não atendido, de 16% a 50% parcialmente atendido, de 51% a 85% largamente atendido e de 86% a 100% é definido como totalmente atendido. Uma organização pode ser considerada do nível 2 quando

todos os atributos dos níveis inferiores são totalmente atendidos e todos os atributos do nível são, pelo menos, largamente atendidos.

O Spice possui os processos separados por categoria e subdivididos em atividades conforme especificado abaixo:

## **CUS Cliente-Fornecedor**

### **CUS.1 Identificação do produto de software e/ou serviço**

- CUS.1.1 Identificar as necessidades
- CUS.1.2 Definir os requisitos
- CUS.1.3 Preparar estratégias de aquisição
- CUS.1.4 Preparar pedido para proposta
- CUS.1.5 Selecionar o fornecedor do software

### **CUS.2 Estabelecimento de contrato**

- CUS.2.1 Revisar antes da finalização do contrato
- CUS.2.2 Negociar contrato
- CUS.2.3 Determinar relações com agentes independentes
- CUS.2.4 Determinar relações aos sub-contratantes

### **CUS.3 Identificação de necessidades do cliente**

- CUS.3.1 Obter requisitos e solicitações de cliente
- CUS.3.2 Entender as expectativas do cliente
- CUS.3.3 Manter os clientes informados

### **CUS.4 Execução de exames e revisões**

- CUS.4.1 Estabelecer exames e revisões
- CUS.4.2 Preparar para exames e revisões pelo cliente
- CUS.4.3 Conduzir o gerenciamento de revisões
- CUS.4.4 Conduzir revisões técnicas

CUS.4.5 Revisão da aceitação de suporte

CUS.4.6 Executar a avaliação do processo

### **CUS.5 Pacote, entrega e instalação do software**

CUS.5.1 Identificar requisitos de instalação

CUS.5.2 Preparar local para instalação

CUS.5.3 Empacotar o software

CUS.5.4 Entregar o software

CUS.5.5 Verificar o repositório correto

CUS.5.6 Instalar o software

CUS.5.7 Fornecer procedimentos de manipulação e armazenamento

### **CUS.6 Suporte operacional do software**

CUS.6.1 Identificar riscos operacionais

CUS.6.2 Executar testes operacionais

CUS.6.3 Operar o software

CUS.6.4 Resolver problemas operacionais

CUS.6.5 Verificar pedidos de usuário

CUS.6.6 Documentar trabalho temporário

CUS.6.7 Monitorar capacidade do sistema e o serviço

### **CUS.7 Fornecimento de serviço para cliente**

CUS.7.1 Treinar o cliente

CUS.7.2 Estabelecer um suporte ao produto

CUS.7.3 Monitorar o desempenho

CUS.7.4 Instalar atualizações de produto

### **CUS.8 Avaliação da satisfação do cliente**

CUS.8.1 Determinar o nível de satisfação do cliente

CUS.8.2 Comparar com competidores

CUS.8.3 Comunicar a satisfação do cliente

## **ENG Engenharia**

### **ENG.1 Desenvolvimento de requisitos e projeto**

- ENG.1.1 Especificar os requisitos do sistema
- ENG.1.2 Descrever a arquitetura do sistema
- ENG.1.3 Alocar requisitos
- ENG.1.4 Determinar estratégia de liberação

### **ENG.2 Desenvolvimento de requisitos do sistema**

- ENG.2.1 Determinar os requisitos do software
- ENG.2.2 Analisar os requisitos do software
- ENG.2.3 Determinar impactos de ambiente
- ENG.2.4 Avaliar requisitos do cliente
- ENG.2.5 Atualizar requisitos para próxima iteração

### **ENG.3 Desenvolvimento do projeto de software**

- ENG.3.1 Desenvolver o projeto arquitetural do software
- ENG.3.2 Projetar interfaces com nível superior
- ENG.3.3 Desenvolver projeto detalhado
- ENG.3.4 Estabelecer traceabilidade

### **ENG.4 Implementação do projeto de software**

- ENG.4.1 Desenvolver unidades de software
- ENG.4.2 Desenvolver procedimentos de verificação de unidade
- ENG.4.3 Verificar unidades de software

### **ENG.5 Integração e teste de software**

- ENG.5.1 Determinar estratégia para testes de regressão
- ENG.5.2 Agregar configurações de unidades de software
- ENG.5.3 Desenvolver testes para agregações
- ENG.5.4 Testar as agregações do software
- ENG.5.5 Desenvolver testes de software

ENG.5.6 Testar a integridade do software

### **ENG.6 Integração e teste do sistema**

ENG.6.1 Agregar configurações para elementos do sistema

ENG.6.2 Desenvolver testes para agregações

ENG.6.3 Testar agregações de sistema

ENG.6.4 Desenvolver testes para sistema

ENG.6.5 Testar integridade do sistema

### **ENG.7 Manutenção do sistema e do software**

ENG.7.1 Determinar requisitos de manutenção

ENG.7.2 Analisar problemas de usuário e melhorias

ENG.7.3 Determinar modificações para próximas atualizações

ENG.7.4 Implementar e testar modificações

ENG.7.5 Atualizar sistema do usuário

## **PRO Projeto**

### **PRO.1 Planejamento de ciclo de vida do projeto**

PRO.1.1 Avaliar opções para desenvolvimento do produto

PRO.1.2 Selecionar modelo de ciclo de vida do software

PRO.1.3 Descrever atividade e tarefas

PRO.1.4 Estabelecer seqüências de tarefas

PRO.1.5 Documentar atividades

### **PRO.2 Estabelecimento de plano de projeto**

PRO.2.1 Desenvolver estrutura de avaria do trabalho

PRO.2.2 Identificar padrões de projeto

PRO.2.3 Identificar facilidade especializadas

PRO.2.4 Determinar estratégia de reuso

PRO.2.5 Desenvolver estimativas de projeto

PRO.2.6 Identificar riscos iniciais de projeto

- PRO.2.7 Identificar métricas de projeto
- PRO.2.8 Estabelecer programação do projeto
- PRO.2.9 Estabelecer comprometerimentos com projeto
- PRO.2.10 Documentar os planos do projeto

### **PRO.3 Identificação dos times do projeto**

- PRO.3.1 Definir times de projeto
- PRO.3.2 Dividir times de projeto
- PRO.3.3 Manter iterações com equipe de projeto
- PRO.3.4 Gerenciar versões de projeto

### **PRO.4 Gerenciamento de requisitos**

- PRO.4.1 Obter concordância com os requisitos
- PRO.4.2 Estabelecer uma linha base de requisitos de cliente
- PRO.4.3 Gerenciar mudanças de requisitos de cliente
- PRO.4.4 Usar requisitos de cliente
- PRO.4.5 Manter traceabilidade

### **PRO.5 Gerenciamento de qualidade**

- PRO.5.1 Estabelecer objetivos de qualidade
- PRO.5.2 Definir métricas de qualidade
- PRO.5.3 Identificar atividades de qualidade
- PRO.5.4 Avaliar atividade de qualidade
- PRO.5.5 Avaliar a qualidade
- PRO.5.6 Aplicar ações de correção

### **PRO.6 Gerenciamento de riscos**

- PRO.6.1 Estabelecer escopo de gerenciamento de riscos
- PRO.6.2 Identificar riscos
- PRO.6.3 Analisar e priorizar riscos
- PRO.6.4 Desenvolver estratégias de diminuição de riscos
- PRO.6.5 Definir métricas de riscos

- PRO.6.6 Implementar métricas de diminuição de riscos
- PRO.6.7 Avaliar resultados das estratégias de diminuição de riscos
- PRO.6.8 Implementar ações de correção

#### **PRO.7 Gerenciamento e controle de recursos**

- PRO.7.1 Adquirir recursos
- PRO.7.2 Progredir com trabalho
- PRO.7.3 Conduzir gerenciamento de revisões
- PRO.7.4 Conduzir revisões de técnicas
- PRO.7.5 Gerenciar o comprometimento

#### **PRO.8 Gerenciamento de sub-contratados**

- PRO.8.1 Estabelecer procedimentos de trabalho
- PRO.8.2 Qualificar o potencial dos sub-contratados
- PRO.8.3 Selecionar sub-contratados
- PRO.8.4 Estabelecer e gerenciar comprometimentos
- PRO.8.5 Manter comunicação
- PRO.8.6 Avaliar a conformidade
- PRO.8.7 Avaliar a qualificação do sub-contratado

### **SUP Suporte**

#### **SUP.1 Desenvolvimento de documentação**

- SUP.1.1 Determinar requisitos de documentação
- SUP.1.2 Desenvolver o documento
- SUP.1.3 Checar o documento
- SUP.1.4 Distribuir o documento
- SUP.1.5 Manter o documento

#### **SUP.2 Execução do gerenciamento de configuração**

- SUP.2.1 Estabelecer sistema de gerenciamento de configuração
- SUP.2.2 Identificar itens de configuração

- SUP.2.3 Manter descrições dos itens de configuração
- SUP.2.4 Gerenciar solicitações de mudanças
- SUP.2.5 Controlar mudanças
- SUP.2.6 Liberar produtos de configuração
- SUP.2.7 Manter histórico de itens de configuração
- SUP.2.8 Relatar o status da configuração

### **SUP.3 Execução da garantia de qualidade**

- SUP.3.1 Selecionar padrões de projeto
- SUP.3.2 Revisar atividades de engenharia de software
- SUP.3.3 Examinar produtos de trabalho
- SUP.3.4 Relatar resultado
- SUP.3.5 Verificar desvios

### **SUP.4 Execução definições de problemas**

- SUP.4.1 Preparar relatório de problemas
- SUP.4.2 Auditar relatório de problemas
- SUP.4.3 Priorizar problemas
- SUP.4.4 Determinar resolução
- SUP.4.5 Corrigir o defeito
- SUP.4.6 Distribui a correção

### **SUP.5 Execução de revisões em par**

- SUP.5.1 Selecionar produtos de trabalho
- SUP.5.2 Identificar padrões de revisão
- SUP.5.3 Estabelecer critérios de conclusão
- SUP.5.4 Estabelecer revisão de critérios
- SUP.5.5 Distribuir materiais de revisão
- SUP.5.6 Revisão de conduta em par
- SUP.5.7 Documentar itens de ação
- SUP.5.8 Auditar itens de ação

## **ORG Organização**

### **ORG.1 Planejamento do negócio**

- ORG.1.1 Estabelecer visão da estratégia
- ORG.1.2 Estender a visão
- ORG.1.3 Estabelecer cultura de qualidade
- ORG.1.4 Formar equipes integradas
- ORG.1.5 Prover incentivos
- ORG.1.6 Definir planos de carreira

### **ORG.2 Definição do processo**

- ORG.2.1 Definir objetivos
- ORG.2.2 Identificar atividades correntes, regras e responsabilidades
- ORG.2.3 Identificar entradas e saídas
- ORG.2.4 Definir critérios de entrada e saída
- ORG.2.5 Definir controle de pontos
- ORG.2.6 Identificar relações externas
- ORG.2.7 Identificar relações internas
- ORG.2.8 Definir gravações de qualidade
- ORG.2.9 Definir métricas de processo
- ORG.2.10 Documentar padrões de processo
- ORG.2.11 Estabelecer políticas
- ORG.2.12 Estabelecer desempenho e expectativas
- ORG.2.13 Estender o processo

### **ORG.3 Melhoramento do processo**

- ORG.3.1 Identificar oportunidades de melhoria
- ORG.3.2 Definir escopo de melhoria de atividades
- ORG.3.3 Entender os processos
- ORG.3.4 Identificar melhorias

- ORG.3.5 Priorizar melhorias
- ORG.3.6 Definir métricas de impacto
- ORG.3.7 Mudar o processo
- ORG.3.8 Confirmar a melhoria
- ORG.3.9 Melhorar a melhoria

#### **ORG.4 Execução de treinamentos**

- ORG.4.1 Identificar necessidades de treinamento
- ORG.4.2 Desenvolver ou adquirir treinamento
- ORG.4.3 Treinar pessoal
- ORG.4.4 Manter gravações de treinamento

#### **ORG.5 Habilitação de reuso**

- ORG.5.1 Determinar estratégias de reuso da organização
- ORG.5.2 Identificar componentes reusáveis
- ORG.5.3 Desenvolver componentes reusáveis
- ORG.5.4 Estabelecer biblioteca de reuso
- ORG.5.5 Certificar componentes reutilizáveis
- ORG.5.6 Integrar o reuso no ciclo de vida
- ORG.5.7 Propagar as mudanças com cuidado

#### **ORG.6 Fornecimento de ambiente para programação**

- ORG.6.1 Identificar requisitos de ambiente para engenharia de software
- ORG.6.2 Prover ambiente para engenharia de software
- ORG.6.3 Prover suporte para desenvolvedores
- ORG.6.4 Manter ambiente para engenharia de software

#### **ORG.7 Fornecimento de facilidades de trabalho**

- ORG.7.1 Fornecer espaço de trabalho produtivo
- ORG.7.2 Assegurar a integridade dos dados
- ORG.7.3 Prover backups dos dados
- ORG.7.4 Prover facilidades de trabalho

ORG.7.5 Prover facilidades de acesso remoto

## 6.3 – CONCEITUAÇÃO DA ENGENHARIA DE REQUISITOS NO SPICE

A engenharia de requisitos aparece no Spice em diversos processos e atividades:

- Definição de requisitos: A definição de requisitos aparece no processo cliente-fornecedor mais especificamente na categoria identificação do produto de software e/ou serviço e tem como objetivo preparar os requisitos do sistema para satisfazer todas as necessidades para um produto novo e/ou um serviço. O Spice cita que tal definição pode ou não ser completamente definida pelo fornecedor, contudo, aborda a atividade de desenvolvimento dos requisitos como essencial para a continuação deste processo.
- Obtenção de requisitos e solicitações do cliente: Esta atividade pertence à categoria de identificação de necessidades do cliente e é importante na validação das solicitações oriundas do cliente e dos usuários do sistema abordando a revisão das propostas do cliente, o ambiente físico do sistema e outros documentos inerentes aos requisitos.
- Especificação dos requisitos do sistema: Já no processo de Engenharia na categoria de desenvolvimento de requisitos e projeto, a especificação se preocupa com definições como funcionalidades, potencialidades, desempenho, segurança, confiabilidade, segurança, interfaces, requisitos de manutenção entre outros.
- Requisitos de instalação: Dentro do processo de pacotes, entrega e instalação de software é mencionada uma atividade com relação aos requisitos de instalação, os principais objetivos desta atividade são a identificação dos requisitos de empacotamento, entrega e instalação do sistema com foco em detalhes como tipo de mídia a ser entregue, documentação, licenças, políticas de cópia de backup e versões críticas de segurança.
- Determinação de requisitos de documentação – Esta atividade faz parte do processo de Suporte na categoria desenvolvimento de documentação e tem como objetivo a identificação de um esboço para uma documentação original a ser construída incluindo

título, audiências, finalidade, objetivo a ser conseguido, esboço de índice, meios de distribuição.

- Alocar requisitos: A alocação de requisitos está na mesma categoria da especificação de requisitos do sistema e se preocupa com a arquitetura do sistema, tendo como artefato final uma documentação de configuração do produto que descreva a posição de cada elemento na arquitetura.

- Determinação dos requisitos de software: A atividade de determinação de requisitos é parte do processo de engenharia na categoria desenvolvimento dos requisitos e tem a principal finalidade de determinar e documentar as especificações de requisitos comentadas anteriormente.

- Análise dos requisitos de software – Nesta etapa a análise dos requisitos do software se propõe a um refinamento dos requisitos se preocupando com variantes como a integrabilidade, a facilidade de compreensão e verificação, a testabilidade, a validade e a consistência do software.

- Identificação de requisitos de ambiente para engenharia de software – Esta atividade pertence ao processo da organização dentro da categoria fornecimento de ambiente para programação e determina requisitos de ambiente como processos, papéis, atividades que deve suportar, questões de segurança, compartilhamento de dados, 'backups' , restauração de informações e etc.

- Avaliação dos requisitos do cliente – Neste momento é necessária à revisão dos requisitos e a apresentação dos requisitos finais ao cliente. A prototipação é uma atividade altamente recomendada nesta etapa por diminuir os riscos do projeto e ser um método apropriado de avaliar os requisitos junto ao cliente.

- Obter concordância com os requisitos – Esta atividade está presente no processo de projeto na categoria gerenciamento de requisitos e se preocupa com a concordância da equipe de análise e desenvolvimento com relação aos requisitos do sistema.

- Atualização dos requisitos para próxima iteração – Esta atividade é a última do processo de engenharia na categoria desenvolvimento de projeto e propõe que após as etapas de projeto, codificação e testes, sejam obtidos um ‘feedback’ do cliente para uso na próxima iteração.
- Gerência de mudança de requisitos – A atividade de gerência de mudança de requisitos pertence ao processo projeto e a categoria gerenciamento de requisitos e aborda a questão da gerência do impacto das mudanças solicitadas pelo cliente assegurando que a equipe diretamente afetada pelas mudanças avalie o impacto e os riscos tomando ações apropriadas de controle.
- Determinação dos requisitos para manutenção – Esta atividade pertence ao processo de engenharia na categoria manutenção do sistema e do software e se preocupa com a identificação dos elementos do sistema que deverão ser mantidos e/ou melhorados.
- Estabelecimento de linha-base de requisitos – Esta atividade da categoria de gerenciamento de requisitos propõe a documentação dos requisitos do cliente para que este sirva de base para todo o projeto.
- Uso dos requisitos do cliente – Esta atividade verifica se todos os requisitos do cliente estão sendo obedecidos conforme estabelecido. Esta atividade verifica principalmente o plano de projeto, as especificações de requisitos, os produtos e as atividades de trabalho.

## **7 – COMPARAÇÃO ENTRE C.M.M.I. E SPICE**

O CMMI e o Spice possuem várias diferenças, mesmo ambas tendo incluídas aspirações e propostas, de um mesmo modelo, o antigo CMM. Começando pela definição, enquanto o Spice se define por um framework para avaliação integrada de processos de software visando à melhoria contínua de processos, o CMMI, conforme o SEI, é um framework de modelos que serve como guia para a melhoria dos processos e habilidades de organizações visando gerenciar o desenvolvimento, a aquisição e a manutenção de produtos e serviços tecnológicos.

Nestas definições percebe-se que o CMMI propõe-se a disponibilizar mais de um tipo de modelo dependendo do tipo de enfoque que a organização pretende dar ao problema dos processos. O Spice, por sua vez, adquire uma visão única de framework o que pode dificultar a adoção do framework.

O propósito pelo qual o Spice se propõe é demonstrar um tipo de avaliação e melhoria contínua de processos utilizando uma auto-compreensão do estado de processos de software, uma autodeterminação de adequabilidade de processos para determinados requisitos e processos de uma organização (fornecedora) no atendimento a um contrato particular.

Conforme o SEI, o CMMI aborda como propósito fundamental o amadurecimento da habilidade de executar, controlar, melhorar e demonstrar melhoria numa área de processo estabelecendo dessa forma, melhorias evolutivas no processo. O CMMI pretende ainda a redução de custo da implementação de melhoria de processo multidisciplinar baseada em modelos por meio de eliminação de inconsistências, redução de duplicidades, melhoria da clareza e entendimento, utilização de terminologia comum e estilo consistente, estabelecimento de regras de construção uniformes, manutenção de componentes comuns, consistência com as normas ISO (principalmente a ISO/IEC 15504), sensibilidade às implicações dos esforços legados.

Quanto este tópico, fica clara uma diferença bastante simples entre os dois frameworks que é o 'entendimento'. O Spice faz questão de especificar que quer que a organização que a utilize saiba por que está adotando este processo, saiba por que está fazendo as melhorias e ainda, quer que a organização entenda o porquê o Spice define que tais práticas sejam executadas pela organização, isto é uma grande vantagem para organizações que queiram andar com suas próprias pernas, pois o conhecimento gerado pela utilização do framework é algo que o próprio framework aconselha.

Enquanto isso, o CMMI usa a palavra 'demonstrar' para explicar sua existência, o que leva a entendermos que o CMMI quer muito mais do que o conhecimento da própria organização, que o mercado saiba que a organização está adotando-o. Talvez isto seja um dos motivos para que a aceitação do CMMI tenha um alcance maior do que o do Spice, a organização, mais do que melhorar, quer mostrar para seus clientes, fornecedores e concorrentes que a organização está respeitando as regras de qualidade dos processos de fabricação do produto-software, os motivos desse objetivo pelas organizações são óbvios. Uma outra característica marcante e bastante importante do CMMI é a objetividade, o que fica muito claro na afirmação 'o CMMI pretende a redução de custo da implementação de melhoria de processo'.

O Spice costuma pregar que o objetivo principal de uma avaliação de processos é a determinação da capacidade no atendimento a requisitos e determinação de melhorias a serem implantadas. Já o CMMI acredita que a seqüência de níveis de maturidade estabelece padrões evolucionários de melhoria de processo, que estabilizam uma parte importante dos processos de uma organização.

Quanto à questão do objetivo da avaliação de processos é notável que o Spice está mais focado na questão da determinação de quais processos precisam de melhorias e quais melhorias devem ser feitas em tais processos enquanto o CMMI preocupa-se na estabilização dos processos da organização, como uma cartilha de boas práticas de

melhores práticas de processo como foco no produto-software. Ambas as abordagens possuem pontos positivos e negativos. O Spice preocupando-se tanto nas melhorias pode por consequência fazer a organização ficar mais comprometida com a implementação e uso do framework já que estará tudo mais claro e objetivo para os colaboradores do por que seguir o caminho especificado pelo framework, porém, tal abordagem pode fazer com que as melhorias demorem a serem percebidas pelos clientes da organização. No CMMI a abordagem de estabilização obtendo-se níveis de maturidade é excelente para o mercado, mas por outro lado, pode fazer com que uma organização fique engessada no meio de processos e atividades que não entende muito bem porque está fazendo, tudo em nome da qualidade exigida pelo CMMI.

O Spice geralmente é aplicado em organizações envolvidas com planejamento, gestão, monitoramento, controle e melhoria de processos de aquisição, fornecimento, desenvolvimento, operação, evolução e suporte de software.

O CMMI é altamente recomendável para organizações em que as características dos processos sejam imaturas a ponto de não haver maneiras sistemáticas de produção de software, não haver ambiente estável de desenvolvimento, de o sucesso do desenvolvimento depender mais da competência e por vezes, do heroísmo dos desenvolvedores do que da sistemática do processo da empresa, alta dependência de pessoas específicas resultando em queda de produtividade verificação de problemas nos cronogramas ou orçamentos estabelecidos preocupação com práticas voltadas à garantia de qualidade dos produtos, porém, abandono dessas práticas em momento de pressão.

Mais uma vez percebe-se que o CMMI possui um foco mais objetivo nos pontos em que uma organização pode não estar praticando as suas atividades corretamente, o que pode acabar em uma organização confusa pela escolha de qual framework de melhoria de processos utilizar pela comparação de cenários existentes com os descritos pelo CMMI. Enquanto a definição de organizações que o Spice sugere leva a impressão de que apenas grandes organizações com faturamento alto e com grande disponibilidade de

recursos podem utilizar, talvez este seja um dos pontos mais negativos nessa definição por parte dos autores do Spice.

O Spice define um conjunto de requisitos para a construção de processos de pontuação, aonde tais resultados devem ser consistentes, repetíveis e representativos. A entrada de dados deve ser definida e um assessor deve assumir a responsabilidade pela qualidade dos resultados usando componentes pontuáveis, escalas, ponderações e registros de saída de dados.

O CMMI se pauta com base em agregação de evidências coletadas através de observações (ferramentas, apresentações, documentos, entrevistas) e com base em suas duas abordagens pontua e classifica uma organização em um dos cinco níveis de maturidade possíveis, assim, para uma organização estar no nível 2, é necessário que todas as áreas-chave deste nível estejam institucionalizadas. Para estar no nível 3, é preciso cumprir todas as áreas no nível 2 e todas do nível 3. E assim por diante. Uma organização no nível 2 pode, por exemplo, possuir práticas de níveis mais altos, mas ser apenas nível 2, por não possuir o conjunto completo das áreas do nível mais alto. Já na abordagem chamada de “modelo contínuo”, cada área-chave de processo possui características relativas a mais de um nível. Assim, uma área-chave que, no modelo em estágios, pertence exclusivamente ao nível 2, no modelo contínuo pode ter características que a coloquem em outros níveis.

A objetividade e maior simplicidade do CMMI aparecem também nos processos de pontuação das organizações em que atua. Tal simplicidade pode ser extremamente benéfica a ponto de não causar dúvida ou suspeita nos investidos de uma organização que está se certificando, porém em sua abordagem seqüencial o CMMI pode pecar no mesmo ponto que o Spice, a falta de definição e a falta de clareza nas áreas-chave que uma organização está madura.

O Spice aborda uma série de instrumentos de avaliação, em papel ou computador, que incorporam vários indicadores de avaliação padronizados que vão desde gerenciamento

de processos, passando por caracterização de produtos de trabalho diferentes, mapeamento de processos para produtos de trabalho, mapeamento de práticas base para produtos de trabalho até a tabulação através de indicadores de processos, porém, não possui explicitamente um método de avaliação.

O CMMI adota métodos de mensuração de qualidade como ferramenta para seus modelos. Um dos métodos mais conhecidos é o SCAMPI. O SCAMPI verifica, entre outros pontos, a satisfação dos critérios de qualidade a melhoria interna do processo, a determinação da capacidade externa, as práticas de seleção de fornecedores e monitoramentos gerais. O SCAMPI permite avaliar os pontos fortes e fracos de processos de engenharia de uma organização conforme o modelo CMMI, priorizar planos de melhoria de processos, produzir um escore de avaliação padronizada para níveis de capacidade e maturidade.

A falta de pelo menos um método de avaliação explícito no Spice e a força do método SCAMPI do CMMI contribuem para uma avaliação positiva quanto ao quesito avaliação por parte do CMMI. É de extrema importância que as organizações juntamente com consultorias especializadas tenham métodos claros e objetivos de avaliação, até mesmo para a confiabilidade do trabalho executado e do resultado divulgado.

Os vários objetivos do CMMI sejam genéricos ou específicos possuem equivalentes no modelo Spice, fazendo com que ambos atendam e foquem o mesmo tipo de organização.

Contudo, o Spice possui um foco mais genérico e abrangente que o C.M.M.I., percebe-se tal diferença quando se verifica que, enquanto o C.M.M.I. trata de melhorias de processos tanto para sistemas incluindo ou não de software, o Spice também se preocupa com detalhes tais como preocupação na contratação de um fornecedor de software, estabelecimentos de contratos comerciais entre as partes, preocupação com os serviços de manutenção, preocupações com relação à instalação, entrega do software, avaliação da satisfação do cliente entre outros.

Deste modo, o Spice apresenta-se como um modelo mais flexível do que o C.M.M.I. mesmo em sua abordagem seqüencial. Tal flexibilidade acomoda melhor o conceito de evolução do nível de capacidade do processo.

Outra vantagem do Spice com relação ao C.M.M.I. é a grande quantidade de formatos de apresentação dos resultados, devido à maneira de como os mecanismos de pontuação foram implementados.

Por outro lado, a flexibilidade do Spice aumenta também a sua complexidade, já que ele não possui roteiros claros para a melhoria do processo.

O Spice mostra-se desta forma, a melhor opção para organizações que desejam maior grau de controle em todos os processos relevantes dentro de um desenvolvimento de um produto de software.

O C.M.M.I. possui a grande vantagem de ser uma melhoria de um modelo já testado e com um grande número de empresas usuárias que é o C.M.M., além disso, sua maior simplicidade devido à estruturação em uma escala crescente de níveis e objetivos, na sua abordagem por estágios, facilita a sua aplicação em programas de melhoria.

Desta forma, resumidamente temos uma tabela comparativa entre o CMMI e o Spice:

	<b>CMMI</b>	<b>Spice</b>
<b>Vantagem</b>	Um modelo menos abrangente, por isso mais simples de ser aplicado e utilizado.	Um modelo mais abrangente e por isso contempla pontos que o CMMI não menciona.  É o modelo sugerido pela

		ISO.
<b>Desvantagem</b>	Por ser um modelo menos abrangente, pode haver etapas em que a organização não tenha o apoio necessário do modelo e não saiba o que fazer.	Por ser um modelo mais abrangente, tem a sua complexidade aumentada fazendo com que os profissionais busquem maiores informações.
<b>Semelhança</b>	Ao contrário do CMM, o CMMI também se utilizou da abordagem seqüencial que já era utilizada pelo Spice.	Usou como modelos de referência o CMM assim como o Trilium, o STD e o Bootstrap de forma harmoniosa e compatível, tendo então várias características semelhantes ao CMMI, principalmente na abordagem por estágios.
<b>Diferença</b>	O CMMI pode ser considerado como uma parte do Spice, pois os dois modelos vieram basicamente do mesmo modelo (CMM) possuindo assim um foco mais amplo que o CMM, porém mais restrito que o Spice,	O Spice contempla controles que o CMMI não possui, como por exemplo a preocupação na contratação de um fornecedor de software, estabelecimentos de contratos comerciais entre as partes, preocupação

	fazendo com que não contemple o nível de controle que o Spice sugere.	com os serviços de manutenção, preocupações com relação à instalação, entrega do software, avaliação da satisfação do cliente entre outros.
--	-----------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------

Tabela 2 – Vantagens, desvantagens, semelhanças e diferenças entre CMMI e Spice.

## 7.1 – COMPARAÇÃO COM FOCO EM ENGENHARIA DE REQUISITOS

Quanto à engenharia de requisitos, o Spice mais uma vez mostra-se um modelo mais abrangente no que diz respeito ao escopo de suas atividades. Um claro exemplo disso são os requisitos de pacotes, entrega e instalação do software a ser desenvolvido ou a preocupação com o refinamento de requisitos ou até mesmo requisitos para a formulação de documentações. Por outro lado o CMMI também propõe atividades que não se encaixam claramente na engenharia de requisitos como a rastreabilidade de requisitos e a viabilidade dos mesmos.

Na abordagem por estágios do C.M.M.I. verificamos alguns problemas com relação as atividades que estão inter-relacionadas no mundo real, estarem separadas neste framework como no caso do gerenciamento de requisitos (presente no nível dois) e a viabilidade de requisitos que faz parte da atividade de desenvolvimento de requisitos (presente no nível três). Tal contexto pode ser problemático caso o canal de comunicação com o cliente ainda não seja muito confiável podendo causar impactos significativos ao projeto.

Esta desvantagem reflete o fato de algumas atividades que estão intimamente ligadas terem de ser melhoradas obrigatoriamente em separado por pertencerem a áreas-chave de processo diferentes. Particularmente na engenharia de requisitos esta desvantagem é agravada por tratar-se de uma área crítica do processo. Percebe-se claramente essa separação verificando-se as atividades de gerenciamento e desenvolvimento de requisitos conforme exposto acima. Esta abordagem pode ainda levar os responsáveis a perderem de vista os processos tentando continuar a melhoria do processo por outra atividade que não pertence a área-chave de processo.

Contudo, essa clara separação no CMMI em sua abordagem por estágios permite que a equipe responsável por uma única atividade, foque melhor o que precisa ser melhorado independentemente de outros processos, fornecendo assim melhores formas para

medições e análises e permitindo ainda que a organização foque nas áreas vitais no desenvolvimento de software que tipicamente podem obstruir o desempenho do processo em um estágio particular do ciclo de vida.

Este problema em potencial da separação dos estágios não é verificado no Spice e nem no CMMI em sua abordagem contínua, uma vez que este gerenciamento é descrito como uma atividade que pode normalmente ser uma das primeiras a serem desenvolvidas pela organização. Deste modo, a organização pode customizar o modelo atendendo suas necessidades de negócio, melhorando a velocidade e satisfação de implementação do modelo e tendo um maior controle do caminho que a equipe de melhoria de software irá percorrer.

Apesar de esta liberdade poder ser traduzida como uma melhor satisfação da organização também pode abrir um leque de possibilidades, talvez, grande demais para uma equipe com pouca experiência. Um erro em uma decisão de prioridade em atividades que deverão ter seus processos melhorados pode ser fatal para o sucesso do processo como um todo, fazendo-se necessário uma consultoria focada na área de melhoria de processos ou de profissionais especializados atuando junto à organização.

Com relação as atividades presentes no Spice e no C.M.M.I., algumas estão presentes mesmo que não façam parte do que se espera de um processo de engenharia de requisitos.

A tabela abaixo mostra uma comparação entre o Spice e o C.M.M.I. com relação a abordagem da O.F.P. na engenharia de requisitos.

Atividades da Engenharia de Requisitos	Spice	CMMI
<b>Área Um: Análise de Negócio</b>	-	-

Análise do Perfil dos Stakeholders	Não contempla	Não contempla
Análise do Cliente	Contemplado na Obtenção de requisitos e solicitações do cliente	Contemplado na Obtenção de entendimento dos requisitos
Análise do Concorrente	Não contempla	Não contempla
Análise de Mercado	Contemplado na Especificação dos requisitos do sistema	Contemplado no Estabelecimento de requisitos de produto e de produto-componente
Análise da Tecnologia	Contemplado na Especificação dos requisitos do sistema	Contemplado no Estabelecimento de requisitos de produto e de produto-componente
Análise do Usuário	Contemplado na Obtenção de requisitos e solicitações de clientes	Contemplado na Obtenção de entendimento dos requisitos
<b>Área Dois: Visão</b>	-	-
Visão de Negócio	Contemplado parcialmente na Obtenção de concordância com os	Contemplado na Coleta de necessidades

	requisitos	
Visão de Sistema	Contemplado na Especificação de requisitos de software e Determinação dos requisitos de software	Contemplado no Estabelecimento de requisitos de produto e de produto- componente e nos Requisitos de interface
Visão de Aplicação	Contemplado na Especificação de requisitos de software e Determinação dos requisitos de software	Contemplado no Estabelecimento de requisitos de produto e de produto- componente e nos Requisitos de interface
Visão de Componente	Não contempla	Contemplado no Estabelecimento de requisitos de produto e de produto- componente e na Alocação de requisitos de produto- componente
<b>Área Três: Desenvolvimento de requisitos</b>	-	-
Identificação dos Requisitos	Contemplado na	Contemplado na

	Definição de requisitos e Identificação de requisitos de ambiente para engenharia de software	Obtenção de entendimento dos requisitos e na Coleta de necessidades
Reuso dos Requisitos	Não contempla	Contemplado na Análise dos requisitos com foco em equilíbrio
Análise dos Requisitos	Contemplado na atividade de Definição de requisitos, Obtenção de requisitos e solicitações de cliente, Alocação requisitos e na Análise dos requisitos de ambiente para engenharia de software	Contemplado na Identificação de inconsistências entre requisitos e produtos de trabalho, no Desenvolvimento dos requisitos do cliente e na Análise de requisitos
Prototipagem dos Requisitos	Contemplado parcialmente na Avaliação dos requisitos do cliente	Não contempla
Especificação dos Requisitos	Contemplado na Avaliação dos requisitos do cliente	Contemplado no Estabelecimento de uma definição de funcionalidade

		requerida
Validação dos Requisitos	Contemplado na Análise dos requisitos de software, Avaliação dos requisitos do cliente, Obter concordância com os requisitos e no Uso dos requisitos do cliente	Contemplado no Desenvolvimento dos requisitos do cliente e na Validação dos requisitos com métodos compreensíveis
<b>Área Quatro: Gerenciamento de requisitos</b>	-	-
Gerenciamento de requisitos	Contemplado na Gerência de mudança de requisitos e no Estabelecimento de linha-base de requisitos	Contemplado no Gerenciamento de mudanças de requisitos e contemplado parcialmente na Alocação de requisitos de produto-componente
<b>Outros*</b>	-	-
Requisitos de Documentação	Contemplado na Determinação de requisitos de documentação e Determinação dos requisitos de software	Não contempla

Refinamento de Requisitos	Contemplado na Análise dos requisitos de software	Não contempla
Requisitos de Ambiente	Contemplado na Identificação de requisitos de ambiente para engenharia de software	Contemplado no Estabelecimento de conceitos operacionais e cenários
Atualização e Melhoria Contínua de Requisitos	Contemplado parcialmente na Atualização dos requisitos para próxima iteração e na Determinação dos requisitos para manutenção	Contemplado parcialmente na Obtenção do comprometimento com os requisitos
Traceabilidade de Requisitos	Não contempla	Contemplado na Manutenção da traceabilidade bidirecional dos requisitos
Viabilidade de Requisitos	Contemplado na Análise dos requisitos	Contemplado na Análise dos requisitos com foco em equilíbrio

Tabela 3 – Diferenças entre o CMMI e o Spice com foco na Engenharia de Requisitos.

Atividades que não estão mencionadas dentre as atividades padrão da engenharia de requisitos conforme a OFP.

Quanto à análise de negócio, CMMI e Spice estão em níveis similares, tanto na análise do perfil dos stakeholders quando na análise de concorrente, os dois frameworks não apresentam sugestões para contemplar essas atividades.

Na análise do cliente o Spice contempla na sua atividade de obtenção de requisitos e nas solicitações do cliente, pois ambas as atividades sugerem uma análise profunda do cliente, enquanto no CMMI essa atividade é verificada na obtenção do entendimento dos requisitos.

A análise de mercado é atribuída no CMMI na atividade de estabelecimento de produto e de produto-componente enquanto no Spice essa atividade é referenciada nas especificações dos requisitos do sistema, onde há a necessidade de verificar o que o mercado anda utilizando para determinar o caminho a seguir no sistema.

A análise da tecnologia contempla-se nas especificações dos requisitos do sistema e no estabelecimento de requisitos de produto e de produto-componente no Spice e CMMI respectivamente.

Com referência a análise do usuário o Spice aborda essa atividade em dois objetivos: obtenção de requisitos e solicitações do cliente. Já o CMMI menciona tal atividade na obtenção de entendimento dos requisitos.

Quanto à área de visão o CMMI suporta melhor as atividades sugeridas pela engenharia de requisitos, enquanto o Spice contempla apenas parcialmente a visão de negócio e não comenta a visão de componentes, o CMMI mostra-se mais robusto e completo com relação às atividades dessa área.

Na visão de negócio o Spice contempla apenas parcialmente essa atividade pois a obtenção de concordância com os requisitos sugere que os responsáveis obtenham tal concordância verificando também a visão de negócio da organização do cliente, porém a idéia dessa visão é um pouco mais abrangente obtendo concordância até mesmo em critérios de sucesso, em novas iniciativas e nas aplicações de negócio da organização do cliente, o CMMI, no entanto, aborda mais objetivamente essa visão na coleta de necessidades do cliente, aonde tais preceitos são conceituados.

Já a visão de sistema e de aplicação são abordadas no Spice pela atividade de especificação de requisitos de software e na determinação de tais requisitos enquanto no CMMI é sugerido o estabelecimento de requisitos de produto e de produto-componente e mencionado ainda nos requisitos de interface, assim como a visão de aplicação é exemplificada no CMMI no objetivo de estabelecer requisitos de produto e de produto-componente e nos requisitos.

A visão de componente não é comentada no Spice enquanto no CMMI é verificado no estabelecimento de requisitos de produto e de produto-componente e na alocação de requisitos de produto-componente.

Quanto à área de desenvolvimento de requisitos, mais uma vez CMMI e Spice atendem similarmente suas atividades. A identificação dos requisitos é apontada no Spice na definição de requisitos e na identificação de requisitos de ambiente para engenharia de software, já no CMMI essa identificação é abordada na obtenção de entendimento dos requisitos e na coleta de necessidades.

Quanto à questão do reuso de requisitos, o Spice não contempla tal abordagem, enquanto o CMMI possui essa preocupação que, segundo ele, é relevante à atividade de análise dos requisitos com foco em equilíbrio.

A análise dos requisitos é amplamente trabalhada pelos dois frameworks. No Spice, essa atividade relaciona-se com a atividade de definição, alocação, análise e obtenção de requisitos e solicitações de cliente, enquanto no CMMI essa atividade é

abordada na identificação de inconsistência entre requisitos e produtos de trabalho, no desenvolvimento de requisitos do cliente e na análise de requisitos.

Quanto à prototipagem dos requisitos, nenhuma abordagem desse tipo é adotada pelo CMMI, enquanto o Spice possui esse cuidado parcialmente na avaliação dos requisitos do cliente, mas infelizmente não menciona a publicação de tais requisitos prototipados pelos participantes dessa fase.

A especificação dos requisitos é mencionada no estabelecimento de definições de funcionalidades requeridas no CMMI e na avaliação de requisitos do cliente no Spice.

A validação dos requisitos é focada na análise dos requisitos de software, avaliação requisitos do cliente e na obtenção da concordância com os requisitos e, ainda, no uso dos mesmos pelo cliente no Spice. As atividades 'desenvolvimento dos requisitos do cliente' e 'validação dos requisitos com métodos compreensíveis' aborda esse tema no CMMI.

A área de gerenciamento de requisitos é atendida plenamente pelos dois frameworks que possuem preocupações relevantes e similares quanto a essas atividades.

O Spice aborda esse gerenciamento na atividade de gerência de mudança de requisitos e no estabelecimento de linha-base de requisitos, enquanto o CMMI suporta essa atividade no gerenciamento de mudanças de requisitos e também o menciona na alocação de requisitos de produto-componente.

Verificou-se também que em ambos os frameworks algumas atividades tiveram preocupações que não foram encontradas em atividades-padrão da engenharia de requisitos. Quanto a estas atividades o Spice obteve larga vantagem sobre o CMMI por possuir preocupações que o CMMI não levanta.

Requisitos de documentação foram contemplados no Spice na determinação de requisitos de documentação e na determinação dos requisitos de software, enquanto no CMMI não há nenhum tipo de abordagem neste sentido.

O refinamento de requisitos é outra preocupação do Spice que não há similar no CMMI, essa preocupação pôde ser percebida na atividade de análise de requisitos de software.

Requisitos de ambiente foram identificados no estabelecimento de conceitos operacionais e cenários no CMMI e na identificação de requisitos de ambiente para engenharia de software no Spice.

A atualização e melhoria contínua dos requisitos foram mencionadas, não tendo propriamente este foco, na atualização dos requisitos para próxima iteração e na determinação dos requisitos para manutenção no Spice e abordado, também de forma parcial, na obtenção de comprometimento com os requisitos do CMMI.

A rastreabilidade de requisitos foi o único ponto entre outras atividades não-relacionadas na abordagem padrão da engenharia de requisitos em que o CMMI preocupa-se e que o Spice não aborda, tal atividade é verificada na manutenção da rastreabilidade bidirecional dos requisitos do CMMI.

A viabilidade de requisitos foi contemplada na análise dos requisitos no Spice e na análise dos requisitos com foco em equilíbrio no CMMI.

Tais comparações mostram claramente as vantagens e desvantagens entre CMMI e Spice apontando indicadores para a conclusão de que o Spice possui uma abordagem um pouco mais abrangente que o CMMI. O fato de o Spice ter demonstrado melhor desempenho em atividades que não estão relacionadas no na abordagem padrão de engenharia de requisitos prova essa teoria.

Um outro fato interessante é a vantagem quanto à área de visões do CMMI em relação ao Spice, isto ajuda a demonstrar a objetividade do CMMI, já que a área de visão tem como um dos objetivos absorve exatamente qual o tipo de cliente que estamos trabalhando, o foco do negócio, a visão da aplicação no ambiente sistêmico, permitindo dessa forma, que os responsáveis pela implementação desse framework possam ter

maior consciência e entendimento do processo em si, pela sua abordagem mais clara e objetiva.

## **8 – ESTUDOS DE CASO**

### **8.1 – IMPLEMENTAÇÃO DA ISO/IEC 15504 (SPICE)**

A Senior Sistemas, fundada em 1988 com sede na cidade de Blumenau, é uma empresa nacional que atualmente desenvolve, evolui e comercializa quatro produtos: VETORH, para administração de recursos humanos, SAPIENS, para gestão empresarial, REGENTE, para administração de agências de viagem e RONDA para controle de acesso. A Senior pode ser considerada, para avaliação de processos, como composta por seis unidades organizacionais: uma para cada um dos produtos, uma para a área comum dos produtos e outra para a empresa como um todo.

Em 1999, a Sênior realizou uma avaliação de processos baseada na ISO/IEC TR 15504 aonde foram selecionados cinco processos (Customer Support Process, Quality Assurance Process, Project Management Process, Organizational Alignment Process e Process Establishment Process). Na análise dos motivos do não-atendimento de alguns objetivos, percebeu-se a dificuldade da empresa em estabelecer processos e garantir a qualidade dos produtos. No final da implantação, que ocorreu entre fevereiro de 2000 e março de 2001, todos os objetivos foram atendidos com a criação de um departamento de qualidade.

Em maio de 2002 foi realizada uma avaliação de processos com a norma 15504., com o objetivo de comparar o estado atual com o estado avaliado em 1999. A avaliação dos processos foi realizada por dois avaliadores em três fases seqüenciais:

- 1) Preparação da avaliação;
- 2) Obtenção e análise de dados e determinação dos resultados;
- 3) Produção do relatório de avaliação.

Durante os meses de abril e maio de 2002 foi planejada a avaliação e aplicados os questionários sobre processos a pessoas selecionadas da empresa. Foram selecionados 7 processos da norma 15504 para serem avaliados até o nível 3 de capacidade, dentre as quais 5 da avaliação anterior.

<p>CUS.4.2 Customer Support Process (Suporte ao Cliente): O propósito deste processo é estabelecer e manter um nível aceitável de serviços ao cliente/usuário que apóie o uso efetivo do produto de software.</p>
<p>SUP.3.2 Quality Assurance Process (Garantia da Qualidade): O propósito deste processo é garantir que os processos e produtos de trabalho satisfaçam seus requisitos específicos e sejam consistentes com seus planos estabelecidos.</p>
<p>MAN.2 Project Management Process (Gerenciamento de Projeto): O propósito deste processo é identificar, coordenar e acompanhar atividades, tarefas e recursos necessários para um projeto produzir um produto ou serviço que satisfaça os requisitos.</p>
<p>ORG.1 – Organization Aligment Process (Alinhamento Organizacional): O propósito deste processo é garantir que os indivíduos da organização compartilhem de uma mesma visão, cultura e entendimento dos objetivos de negócio para que possam executar suas funções de modo mais efetivo.</p>
<p>ORG.2.1 – Process Establishment (Estabelecimento de Processo): O propósito deste processo é estabelecer um conjunto de processos organizacionais para todos os processos do ciclo de vida de software que se aplicam às atividades de negócio.</p>
<p>ENG.1.2 – Software Requirements Analysis (Análise de Requisitos): O propósito deste processo é estabelecer os requisitos para os componentes de</p>

software do sistema.

ORG.5 – Measurement Process (Medição de Processo): O propósito deste processo é coletar e analisar dados relativos aos produtos desenvolvidos e processos implementados na unidade organizacional, para apoiar a gerência efetiva dos processos e demonstrar de forma objetiva a qualidade dos produtos.

Tabela 3 – Processos escolhidos pela organização Senior para implantação da ISO/IEC 15504.

Quatro fontes de informações foram utilizadas: resposta a questionários, entrevistas, análise de documentos e discussão sobre resultado preliminar. Estas observações foram utilizadas como base para a atribuição das notas de cada processo. Foram identificadas ao todo cerca de 70 observações positivas e 50 oportunidades de melhoria, relacionadas às práticas descritas no Spice.

O resultado da avaliação em 2002 indicou uma melhoria em termos de capacidade de processo. Em geral houve uma melhoria do nível 0 para o nível 2, com alguns casos de melhoria para o nível 1. A prática de utilização de indicadores para acompanhar as atividades e subsidiar a tomada de decisões também existe na empresa. Existem atualmente doze indicadores relacionados ao desenvolvimento e quatro relacionados ao suporte ao cliente, aonde praticamente tudo é analisado e registrado, incluindo reclamações dos clientes e requisitos para evolução dos produtos.

No trabalho de planejamento de melhorias foram identificadas evidências do sucesso do início do trabalho, mas já existia a consciência de que o sucesso seria atingido com a melhoria dos processos desenvolvidos pela empresa.

## 8.2 – IMPLEMENTAÇÃO DO C.M.M.I.

A Microsiga, fundada em 1983, é uma empresa nacional com filiais espalhadas por toda a América Latina. Em 2005, a Microsiga decidiu implantar o CMMI na sua fábrica de software.

A fábrica de software é uma área técnica dedicada ao desenvolvimento de customizações e projetos especiais a serem agregados ao Protheus, ERP Microsiga. A fábrica de software existe desde 1994 e conta com cerca de sessenta profissionais com muita experiência no desenvolvimento de soluções agregadas a ERP e com domínio das ferramentas de criação no ambiente Protheus.

A estrutura da fábrica de software concentra-se nas funções: Vice-presidência de Tecnologia e Sistemas, Líder (Analista de Projetos Sr.), Líder (Analista de Projetos Pl.), Líder (Analista de Projetos Mt.), Coordenador de Consultoria em Soluções de Negócio, Gerente de Fábrica de Software, Coordenador de Negócios, Consultor de Negócios, Analistas de Projetos (Sr., Pl., Jr., Tr.) e Estagiário.

Todos os projetos da fábrica de software gerenciam os requisitos dos produtos e os acordos com fornecedores, estabelecem e mantêm planos de trabalho que possibilitam o acompanhamento do projeto além de estruturas e critérios de medições que garantam a integridade e visibilidade do processo e produto.

Os objetivos da implementação do CMMI na Microsiga foram:

- Aumentar a rentabilidade do network;
- Redução de despesas;
- Diminuição de re-trabalhos;
- Redução de erros;

- Melhor qualidade de produtos;
- Assertividade de projetos;
- Maiores e melhores ofertas de produtos;
- Redução de tempo de colocação de produtos no mercado;
- Motivação do processo de melhoria;
- Oportunidade de descrever um caminho estruturado e contínuo no desenvolvimento.

Os grandes benefícios esperados pela adoção do CMMI foram a oportunidade de possuir uma metodologia única aonde possa haver um gerenciamento único de todos os projetos, clara definição de papéis e redução de re-trabalhos.

Os principais desafios encontrados na implementação do CMMI foram a mudanças de cultura, o envolvimento das equipes e pessoas, a capacitação das equipes, o retorno a médio e longo prazo, a adoção simultânea de novas práticas nos processos, a adequação á maior rigidez de sistemas e ferramentas, a alocação de pessoas-chave nos processos e a comunicação.

Os principais sub-processos utilizados para a obtenção do nível 2 foram o levantamento de necessidades do cliente com uma carta de entrega e especificação do produto, a elaboração da proposta do desenvolvimento do projeto, a alocação de recursos, a elaboração do projeto lógico, o desenvolvimento do projeto com as devidas validações, a disponibilização do ambiente de homologação e treinamento de usuários-chave, a implantação no ambiente de produção e o processo de aceite do cliente.

O total de horas gastas para a implantação do CMMI foi de 4.553 horas, sendo que 800 horas foram em definições, 350 horas para a institucionalização dos novos processos com

a equipe de qualidade e 1.957 horas em definições e 1.346 em institucionalização com a equipe da fábrica de software.

As dificuldades surgidas durante o processo de implementação tiveram relação com a quebra de paradigma no que tange a importância da melhoria de processo, a institucionalização do novo processo e a aceitação de PPQA em todos os projetos.

Os principais indicadores do sucesso da implementação do nível 2 do CMMI na Microsiga se refletem nos benefícios alcançados pela organização como a definição clara de papéis na organização, a melhor padronização de artefatos e processos, o aumento de satisfação dos clientes, a maior confiabilidade no processo e a grande motivação profissional citada pelos colaboradores da organização.

## 9 – CONCLUSÕES

Definidas as principais diferenças e semelhanças entre os modelos Spice e C.M.M.I. com um foco aprofundado na engenharia de requisitos, verifica-se que cada modelo é melhor definido e possui um escopo que abrange, inclusive, parte do escopo trabalhado pelo outro modelo. Neste caso, o Spice trabalha com um escopo de maior grandeza, definindo, dessa forma, atribuições que não se desenvolvem em outros modelos tais como o C.M.M.I.

O C.M.M.I., por sua vez, como é um modelo com um escopo de menor grandeza, se comparado com o Spice, possui uma abordagem mais clara, concisa e detalhada. Estes atributos lhe dão vantagens com relação ao custo, treinamento e aperfeiçoamento.

Ao fim, a escolha de um modelo de melhoria de processo de software deve levar em consideração questões como os custos, os prazos, a viabilidade e a abrangência. Organizações podem disponibilizar maiores recursos para a implementação deste processo de melhoria e exigirem que outras atividades, mesmo que não diretamente ligadas ao desenvolvimento, mas, que sejam relevantes ao desenvolvimento de software seja incluída no escopo da organização da equipe de melhorias, neste caso a escolha do modelo Spice seria uma opção mais viável do que a do C.M.M.I.

E, por fim, empresas de desenvolvimento de software que necessitem de melhorias nos seus processos de desenvolvimento de software, porém com custos e prazos mais baixos, maior facilidade de entendimento e foco maior no desenvolvimento, possui forte tendência a adotar o modelo C.M.M.I.

É relevante afirmar ainda que uma organização deve procurar entender as semelhanças, diferenças, vantagens e desvantagens entre os modelos de melhoria de processo de desenvolvimento de software, não apenas verificando o CMMI e o Spice como também outras alternativas como o Trillium, o Software Technology Diagnostic, o Bootstrap, o MPS-

BR, entre outros, ou até mesmo utilizando-se de um conjunto de características apresentadas por cada modelo focando as atividades que melhor representam a necessidade da organização levando em consideração a política interna, a visão, a missão, o cliente, entre outras características específicas de cada empresa.

## 10 – CONSIDERAÇÕES FINAIS

O autor considera que o conhecimento é melhorado e aperfeiçoado sempre que aplicado e por isso, espera que os este trabalho tenha contribuído de alguma forma ao leitor.

E como perspectiva de trabalho futuro sugere a análise mais detalhada dos mecanismos de pontuação do Spice, quais as vantagens e desvantagens desses mecanismos, a abordagem da instalação, entrega e avaliação da satisfação do cliente contida no Spice e uma maneira de controlar tais atividades com menor complexidade como mesmo no Spice, ou ainda, a explanação de como o modelo C.M.M.I. pode ser adequado com referência ao Spice, de modo a implementar um modelo não tão abrangente quanto o Spice e nem tão específico quanto o C.M.M.I., tentando melhorar a questão do custo-benefício avaliando o que cada modelo possui de melhor.

Outra perspectiva a ser explorada é a questão da gerência de mudanças de requisitos quanto a novas propostas, abordagens e sugestões para minimizar o efeito quase sempre incisivo nos projetos de desenvolvimento de software. A engenharia de requisitos possui muitos pontos a serem melhorados e por isso, o autor acredita que é uma área interessante e que possui muitas perspectivas, sendo de fundamental importância para o futuro do desenvolvimento de software.

## 11 – BIBLIOGRAFIA

CHRISISSIS, Mary Beth; KONRAD, Mike; SHRUM, Sandy. CMMI – Guidelines for Process Integration and Product Improvement. U.S. Corporate and Government Sales: Addison Wesley, 2003

PRESSMAN, Roger. Engenharia de Software: Makron, 1995.

SOMMERVILLE, Ian. Engenharia de Software: Addison Wesley, 2003.

FIORINI, Soeli T.; STAA, Arndt Von; BAPTISTA, Renan Martins – Engenharia de Software com CMM. Rio de Janeiro: Brasport, 1998

Softex; MPS.BR – Melhoria de Processo do Software Brasileiro: Campinas, 2005

ISSO/IEC - SPICE, Software Process Assessment - Part 1: Concepts and Introductory Guide.

OPFRO - ( Open Process Framework Repository Organization ), 1990 disponível em

<http://www.opfro.org>.

MCCONEEL Steven C., Software Project Survival Guide, 1997-1998 disponível em <http://www.stevemccconnell.com/sgreq.htm>.

CÔRTEZ, Mario L. – IINF310 – Modelos de Qualidade de Software, 2005 Capítulo 7: ISSO;IEC 15504 – Spice, 2005.

SEI, 2002. Capability Maturity Model\_ Integration (CMMI), Version 1.1.,

TSUKUMO, Alfredo C.; RÉGO, Claudete M.; SALVIANO, Clênio F.; AZEVEDO, Luciano K. Meneghetti; COSTA, Márcia C.C.; CARVALHO, Mário Bento de; COLOMBO, Regina M.T. - Qualidade de Software: Visões de Produto e Processo de Software, 1997

SILVA, Ricardo Pereira e, Perfil das empresas de desenvolvimento de software classificáveis como CMMI, nível 1 – subsídios para um 'exame de consciência', 2005.

VOLPE, Renato Luiz Della; JOMORI, Sergio Massao, ZABEU, Ana Cecília Peixoto, CMM – CMMI, Principais conceitos, diferenças e correlações, 2003.

FERNANES, Jorge H.C., Modelos de qualidade em TI e Software: CMMI, COBIT, ITIL e SPICE, 2004.

PAULK, Mark C.; KONRAD, Michael D.; GARCIA, Suzanne M., CMM Versus Spice Architectures, 2000

WEBER, Kival C.; ROCHA, Ana Regina; ALVES, Ângela; AYALA, Arnaldo M.; GONÇALVES, Austregésilo; PARET, Bento; SALVIANO, Clênio; MACHADO, Cristina F.; SCALET, Danilo; PETIT, Djalma; ARAÚJO, Erastóstenes; BARROSO, Márcio Girão; OLIVEIRA, Kathia, OLIVEIRA, Luiz Carlos A.; AMARAL, Márcio P.; CAMPELO, Renata Endriss C.; MACIEL, Tereza – Modelo de referência para melhoria de processo de software: uma abordagem brasileira, 2005.

SIMPROS- (VII Simpósio Internacional de Melhoria de Processo de Software), 2005 disponível em <http://www.simpros.com.br>

SALVIANO, Clênio F., INTRODUÇÃO A MELHORIA DE PROCESSO DE SOFTWARE COM ISSO/IEC 15504 E CMMI, 2004.