

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Engenharia de Produção

Identificação dos Fatores Críticos de Sucesso e Boas Práticas no Processo de Desenvolvimento de Software

Stephanny Andressa Rodrigues de Oliveira

TCC-EP-96-2012

Maringá - Paraná
Brasil

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Engenharia de Produção

Identificação dos Fatores Críticos de Sucesso e Boas Práticas no Processo de Desenvolvimento de Software

Stephanny Andressa Rodrigues de Oliveira

TCC-EP-96-2012

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Produção, do Centro de Tecnologia, da Universidade Estadual de Maringá.
Orientador(a): Prof^(a). Edwin Cardoza Galdamez

**Maringá - Paraná
2012**

DEDICATÓRIA

À minha família e aos meus
amigos.

RESUMO

Muitas empresas de software têm buscado meios para melhorar seus processos de desenvolvimento. Modelos de maturidade de processo, como o CMMI (*Capability Maturity Model Integration*), trazem consigo, além de um arcabouço de práticas que estimulam a melhoria dos processos, a possibilidade de obtenção de uma certificação reconhecida e requisitada no mercado de TI. Este trabalho tem por objetivo identificar os fatores críticos observados na implantação do modelo CMMI nível 2 em uma empresa local, como também boas práticas. A revisão bibliográfica é composta por temas relacionados, como processo de desenvolvimento de software, modelos de desenvolvimento de software, qualidade de processo e de produto de software, assim como modelos para melhoria do processo de desenvolvimento de software. Por meio de entrevistas e questionários aplicados, foi possível identificar o que foi crítico no processo de implantação, assim como as boas práticas alcançadas e lições aprendidas que servirão de referência em projetos futuros da empresa.

Palavras-chave: Modelos de maturidade de processos; CMMI; melhoria de processos em software; fatores críticos no processo de desenvolvimento de software.

SUMÁRIO

LISTA DE FIGURAS	vii
LISTA DE QUADROS.....	viii
LISTA DE ABREVIATURAS E SIGLAS	ix
1. INTRODUÇÃO	10
1.1 JUSTIFICATIVA	12
1.2 DEFINIÇÃO E DELIMITAÇÃO DO PROBLEMA	12
1.3 OBJETIVOS	12
1.3.1 <i>Objetivo geral</i>	12
1.3.2 <i>Objetivos específicos</i>	13
1.4 ESTRUTURA DO TRABALHO	13
2. REVISÃO DA LITERATURA	14
2.1 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	14
2.1.1 <i>Metodologias Tradicionais de Desenvolvimento de Software</i>	15
2.1.2 <i>Metodologias Ágeis de Desenvolvimento de Software</i>	17
2.1.3 <i>Análise Comparativa entre Desenvolvimento Tradicional e Ágil</i>	21
2.2 QUALIDADE DE PRODUTO E PROCESSO NO DESENVOLVIMENTO DE SOFTWARE	22
2.2.1 <i>Avaliação e Monitoração da Qualidade do Processo de Software e Produto</i>	23
2.3 MODELOS PARA MELHORIA NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	25
2.3.1 <i>Modelo de Maturidade de Capacitação Integrada (CMMI)</i>	25
2.3.2 <i>Modelo de Melhoria de Processo do Software Brasileiro (MPS.BR)</i>	34
3. DESENVOLVIMENTO	36
3.1 METODOLOGIA	36
3.2 CONTEXTO DA EMPRESA	36
3.3 VISÃO DE NEGÓCIO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	37
3.3.1 <i>Desenvolvimento</i>	38
3.4 CERTIFICAÇÃO CMMI NÍVEL 2	40
3.4.1 <i>Práticas do CMMI nível 2</i>	42
3.5 AVALIAÇÃO DO PROCESSO DE IMPLANTAÇÃO DO CMMI	44
3.5.1 <i>Fatores críticos e dificuldades</i>	44
3.5.2 <i>Nível de desempenho dos diferenciais competitivos</i>	45
3.5.3 <i>Práticas executadas propostas para o Guia CMMI-dev para o nível 2</i>	46
3.6 ANÁLISE DE RESULTADOS E LIÇÕES APRENDIDAS	49
4. CONCLUSÃO	51
4.1 CONSIDERAÇÕES FINAIS	51
4.2 LIMITAÇÕES DA PESQUISA	51
4.3 TRABALHOS FUTUROS	52

REFERÊNCIAS	55
APÊNDICE 1	59
APÊNDICE 2	60
APÊNDICE 3	61
ANEXO I	65

LISTA DE FIGURAS

FIGURA 1 – Modelo em “cascata”	16
FIGURA 2 – Ciclo de vida do Scrum	20
FIGURA 3 – Principais fatores de qualidade dos produtos de software	21
FIGURA 4 – Representações do CMMI	26
FIGURA 5 – Visão de negócio do processo de desenvolvimento	38
FIGURA 6 – Processo de desenvolvimento de software baseado no modelo cascata.....	39
FIGURA 7 – Práticas executadas por área de processo.....	47
FIGURA 8 – Boas práticas por área de processo	47
FIGURA 9 – Nível de dificuldade de implantação das práticas propostas	48
FIGURA 10 – Nível de dificuldade de implantação por área de processo	49

LISTA DE QUADROS

QUADRO 1 – Comparação entre as abordagens de desenvolvimento de software	21
QUADRO 2 – Áreas de processo do CMMI	28
QUADRO 3 – Práticas por área de processo do CMMI propostas pelo Guia CMMI-dev	31
QUADRO 4 – Práticas e artefatos gerados na implantação do modelo CMMI	42
QUADRO 5 – Respostas ao questionário 2	45

LISTA DE ABREVIATURAS E SIGLAS

CMMI *Capability Maturity Model Integration*

CMM *Capability Maturity Model*

MPS.BR *Melhoria de Processo do Software Brasileiro*

NATO *North Atlantic Treaty Organisation*

RUP *Rational Unified Process*

SEI *Software Engineer Institute*

SQA *Software Quality Assurance*

XP *Extreme Programming*

1. INTRODUÇÃO

No fim da década de 70, o mundo passava pela chamada “crise do software”. Segundo Sommerville (2004), essa crise foi originada a partir do surgimento do poderoso hardware de terceira geração, dado que sua capacidade de processamento tornava viável programas de computador até então impensáveis. Essa mudança tecnológica afetou drasticamente a produção de software (KOSCIANSKI; SOARES, 2007), visto que a capacidade do hardware aumentou e permitiu que programas complexos fossem criados, porém os programadores não estavam preparados para desenvolver programas desse nível.

Koscianski e Soares (2007) relatam que, em um relatório originário de uma conferência realizada pelo Comitê de Ciência da NATO (*North Atlantic Treaty Organisation* – Organização do Tratado do Atlântico Norte) em 1968, juntamente com outros documentos referentes ao assunto da década de 1970, foi realizado um levantamento dos problemas comuns ao desenvolvimento de software:

- Cronogramas deixados de lado;
- Projetos abandonados devido à quantidade de problemas;
- Módulos que não interagem corretamente quando combinados;
- Programas que não atendem ao esperado;
- Programas descartados por serem de difícil uso;
- Programas que simplesmente deixam de funcionar

Koscianski e Soares (2007) discorrem que esses problemas persistem nos projetos de desenvolvimento de software. De acordo com Sommerville (2004), a engenharia de software, que começou a ser desenvolvida devido à crise do software, ajudou em grande escala o desenvolvimento de projetos desta área, porém problemas como atender expectativas do cliente, prazos e custos ainda hoje são corriqueiros, o que leva alguns autores (PRESSMAN, 2005) a considerar que a engenharia de software está em um estado de aflição crônica.

Sommerville (2004) destaca que tem havido um grande interesse por parte da comunidade de engenharia de software pela melhoria de processos, almejando melhorar a qualidade do processo e produto, assim como reduzir custos e tempo de desenvolvimento. Na visão de

Hock et al. (2000, *apud* KUBOTA, 2006), o desenvolvimento de software, quando de grande porte, é complexo, podendo envolver milhões de linhas de código. Ainda assim, muitas organizações não adotam processos de desenvolvimento estruturados, o que pode ocasionar em perdas para os *stakeholders*, produtividade abaixo do esperado e desmotivação dos desenvolvedores. Nesse cenário, foram desenvolvidas diversas certificações de qualidade de software no mercado.

De acordo com Kubota (2006), uma das certificações de software de maior renome é o *Capability Maturity Model (SW-CMM)*. Foi proposto pelo *Software Engineer Institute (SEI)* da *Carnegie-Mellon Institute*, instituto fundado pelo Departamento de Defesa dos Estados Unidos, que na década de 80 realizou estudos sobre como avaliar a capacitação de fornecedores de projetos de desenvolvimento de software, estudos esses que originaram o modelo de maturidade de capacitação CMM (SOMMERVILLE, 2004). Em 2000, o modelo CMM evoluiu para o *Capability Maturity Model Integration (CMMI)*.

Para Marçal (2009, p.29):

O CMMI representa uma abordagem de melhoria de processos que provê elementos essenciais para um processo efetivo de desenvolvimento de software. Reúne melhores práticas que abrangem o desenvolvimento e manutenção, cobrindo o ciclo de vida de produto desde a sua concepção até a sua entrega e manutenção.

De acordo com Pressman (2005), o ciclo de vida clássico da engenharia de software propõe uma abordagem sistemática, sequencial ao desenvolvimento de software, que se inicia no nível do sistema e avança ao longo da análise, projeto, codificação, teste e manutenção. Pressman (2005) também cita que, numa definição formal, um software pode ser descrito como estruturas de dados que possibilitam que os programas manipulem adequadamente a informação, entretanto, por ser um elemento do sistema lógico, e não físico, não deve ser gerido como se fosse um projeto de manufatura clássico, sendo que os custos de software estão concentrados no trabalho de engenharia.

Nesse contexto, o presente trabalho propõe analisar as melhorias alcançadas na qualidade do processo de desenvolvimento e no produto de software, a partir da implantação do modelo de maturidade CMMI (*Capability Maturity Model Integration*) nível 2 em uma empresa desenvolvedora de software em Maringá, Paraná.

1.1 Justificativa

A proposta do trabalho é elaborada tendo em vista que existem poucos estudos efetivos realizados em empresas de software na região no que diz respeito à melhoria dos seus processos, tanto referente aos processos de desenvolvimento de software como aos processos de negócio. Igualmente, a visão do Engenheiro de Produção, em especial os que cursaram a ênfase de “software”, pode ser um grande diferencial em uma área em que os colaboradores caracteristicamente possuem uma visão técnica do processo ou uma visão gerencial, raramente os dois. Com a realização deste trabalho, será possível identificar quais foram os fatores críticos de sucesso com a implantação do modelo CMMI e as boas práticas alcançadas, além de destacar as lições aprendidas, que poderão auxiliar na preparação da empresa em estudo para o próximo de nível de certificação do modelo CMMI.

1.2 Definição e delimitação do problema

O presente trabalho discorrerá sobre a Empresa Benner Saúde, atuante no desenvolvimento de sistemas de informação para operadoras de saúde. Este estudo tem por objetivo reconhecer quais as vantagens e desvantagens de se ter os processos da organização geridos conforme o modelo CMMI, além de definir quais foram os fatores críticos de sucesso no processo de desenvolvimento de software da empresa em estudo, assim como as boas práticas originadas.

1.3 Objetivos

1.3.1 Objetivo geral

Identificar e avaliar os fatores críticos de sucesso e boas práticas com a implantação do modelo CMMI (*Capability Maturity Model Integration*) no processo de desenvolvimento de software.

1.3.2 Objetivos específicos

- ✓ Revisão bibliográfica sobre os temas: Processo de Desenvolvimento de Software, Software, Modelos de desenvolvimento de software (tradicionais e ágeis), Qualidade de Processo e Produto no Desenvolvimento de Software e Modelos para Melhoria no Processo de Desenvolvimento de Software, (CMMI e MPS.Br).

- ✓ Relatar o processo de implantação do modelo CMMI nível 2 em uma fábrica de software

- ✓ Analisar as vantagens, desvantagens, fatores críticos, boas práticas e lições aprendidas com a implantação do modelo CMMI nível 2 na empresa estudada.

1.4 Estrutura do Trabalho

O Capítulo 1 introduz o tema abordado e apresenta a justificativa da realização do estudo, assim como os objetivos e contextualização geral do trabalho.

O Capítulo 2 apresenta a Revisão Bibliográfica, abordando os temas: Processo de Desenvolvimento de Software, Modelos de desenvolvimento de software, Qualidade de Processo e Produto no Desenvolvimento de Software e Modelos para Melhoria no Processo de Desenvolvimento de Software.

No Capítulo 3 é apresentado o desenvolvimento do trabalho de campo, a metodologia adotada, a descrição da empresa em estudo, o método de implantação do modelo, as práticas executadas e a análise dos resultados.

O Capítulo 4 aborda a conclusão do trabalho, discorrendo sobre as contribuições, limitações e propostas de melhoria.

2. REVISÃO DE LITERATURA

Neste tópico são abordados os conceitos de: Processo de Desenvolvimento de Software, Software, Processo de software, Modelos de desenvolvimento de software, Modelo em “cascata”, RUP, XP, *Scrum*, Análise comparativa entre desenvolvimento tradicional e desenvolvimento ágil, Qualidade de Processo e Produto no Desenvolvimento de Software, Qualidade, Qualidade no produto de software, Modelos para Melhoria no Processo de Desenvolvimento de Software, Modelo de Maturidade de Capacitação Integrada (CMMI), Modelo de Melhoria no Processo de Software Brasileiro (MPS.Br), Avaliação e monitoração da qualidade do processo de desenvolvimento de software e produto. Estes elementos darão a base teórica para o desenvolvimento do seguinte trabalho.

2.1 Processo de Desenvolvimento de Software

Um sistema de software consiste em uma série de programas separados, arquivos de configuração que norteiam a configuração desses programas, documentação do sistema descrevendo as estruturas do mesmo e a documentação do usuário, que explicita como utilizar o software. Os engenheiros de software tem por objetivo desenvolver produtos de software, em outras palavras, software para ser vendido para um cliente (SOMMERVILLE, 2004).

De acordo com Sommerville (2004), um processo de software é um conjunto de atividades e entregas associadas, que resultam em um produto de software. Pressman (2005) discorre que o processo de desenvolvimento de software contém três fases genéricas, independente do modelo de desenvolvimento escolhido. Essas três fases “definição”, “desenvolvimento” e “manutenção”, são encontradas em qualquer desenvolvimento de software, independentemente do tipo de aplicação, dimensão do projeto ou complexidade. Segundo Pressman (2005), a fase de definição é onde o desenvolvedor de software realiza um esforço para identificar quais informações devem ser processadas, qual função e desempenho são esperados, quais interfaces devem ser estabelecidas, quais as restrições existentes em projeto e quais os critérios de validação são requeridos para se definir um sistema bem-sucedido. Já na fase de desenvolvimento, o desenvolvedor de software definirá como será a estrutura de dados e a arquitetura de software, como os detalhes procedimentais devem ser implementados, como o projeto será traduzido em uma linguagem de programação e como os testes devem ser

realizados. Por fim, a fase de manutenção concentra-se nas alterações que estão associadas à correção de erros e adaptações requeridas, à medida que o ambiente do software evolui e implementações são produzidas, por exigências do cliente.

2.1.1 Metodologias tradicionais de desenvolvimento de software

As metodologias de desenvolvimento de software, ou modelo de processo de desenvolvimento, são uma representação abstrata de um processo de software, sobre um ponto de vista particular, de modo que proporciona apenas informações parciais sobre o processo (SOMMERVILLE, 2004). A seguir, serão apresentados dois métodos clássicos para desenvolvimento de software: o modelo em “cascata” e o RUP (*Rational Unified Process*).

O modelo em “cascata”

De acordo com Sommerville (2004), um modelo de processo de software é uma abstração do processo real que está sendo descrito, de maneira simplificada. O primeiro modelo publicado do processo de desenvolvimento de software teve origem de processos das Engenharias (ROYCE, 1970 *apud* SOMMERVILLE, 2004). Na visão de Pressman (2005), este modelo clássico, também chamado de “cascata”, requer uma abordagem sistemática, sequencial ao desenvolvimento do software, que tem início no nível do sistema e avança ao longo da análise, projeto, implementação, testes, e manutenção. As principais etapas do modelo retratam as atividades de desenvolvimento primordiais (SOMMERVILLE, 2004), conforme ilustra a Figura 1.

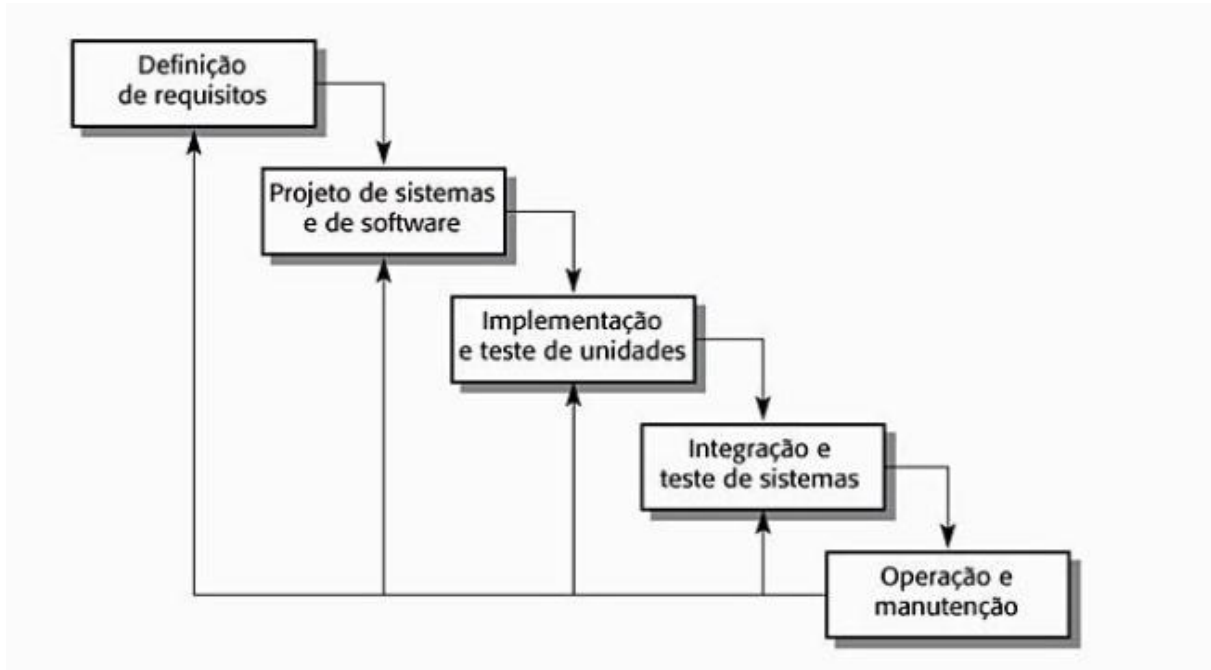


Figura 1 – modelo em “cascata”

Fonte: SOMMERVILLE (2004)

Sommerville (2004) discorre sobre as etapas do modelo:

- **Definição de requisitos:** As funcionalidades, restrições e objetivos do sistema são estabelecidos consultando os usuários do sistema, de modo a elaborar uma especificação do sistema.
- **Projeto de sistemas e de software:** o processo de projeto de sistema reúne os requisitos em sistemas de hardware ou software. Estabelece uma arquitetura de sistema genérica. O projeto de software abrange a identificação e descrição das principais abstrações do sistema de software, assim como suas relações.
- **Implementação e teste de unidades:** nesta etapa, o projeto de software é entendido como unidades de programa, sendo que o teste de unidade tem por princípio verificar que cada unidade atenda à sua especificação.
- **Integração e teste de sistemas:** as unidades de programa são integradas e testadas como um sistema completo, de modo a assegurar que os requisitos de software foram atendidos. Após esta etapa, o sistema é entregue ao cliente.
- **Operação e manutenção:** o sistema é entregue ao cliente e posto em produção. A manutenção engloba corrigir erros que não foram detectados em etapas anteriores do

ciclo de vida, melhorando a implementação das unidades de sistema e adicionando novas funcionalidades à medida que novos requisitos forem descobertos.

RUP

O RUP (*Rational Unified Process*) é um processo de desenvolvimento de software iterativo, incremental e ajustável, podendo ser adaptado para variados tipos de produtos e projetos de software (LEAL, 2010).

Vasco *et al.* (2005) afirma que a metodologia RUP divide cada ciclo de desenvolvimento do projeto em quatro fases, cada qual com um marco definido (*milestone*). Os *milestones* são indicadores de evolução do projeto, e são utilizados para tomada de decisão quanto a continuar, abandonar ou mudar o rumo do projeto. As fases do RUP são (VASCO *et al.*, 2005):

1. Início: definição do escopo de desenvolvimento, sendo elaborada uma visão do produto final a partir de um caso de uso geral.
2. Elaboração: elaboração do plano de atividades e recursos requeridos, onde serão definidas as funcionalidades e arquitetura a serem desenvolvidas.
3. Construção: codificação do software, sendo que esta etapa pode ser dividida em várias iterações, visando facilitar o gerenciamento.
4. Transição: o produto é disponibilizado para o usuário final, sendo que nesta fase ocorre o treinamento e a avaliação do produto.

De acordo com Marinho *et al.* (2003), o RUP incentiva o controle de qualidade e o gerenciamento de riscos contínuo e com propósito. A avaliação de qualidade está presente em todas as atividades do processo. O gerenciamento de riscos acontece no processo de maneira à identificar os riscos que são impeditivos ao sucesso do projeto e propor soluções no início do processo de desenvolvimento. O desenvolvimento com RUP é centrado na arquitetura, o que pressupõe a minimização de retrabalho, assim como aumenta a reutilização de componentes e a competência de manutenção do sistema.

2.1.2 Metodologias ágeis de desenvolvimento de software

Os métodos ágeis surgiram em resposta à crise crônica do software (FOWLER, 2005, *apud* MARÇAL, 2009), reagindo aos modelos tradicionais de desenvolvimento e como alternativa à necessidade de se criar algo diferente dos processos caracterizados pelo foco dado à elaboração de documentações extensas (BECK *et al.*, 2001). Os princípios das metodologias ágeis de desenvolvimento de software, citadas no *Manifesto for Agile Software Development* (BECK *et al.*, 2001), pregam a valorização das pessoas a processos, e a satisfação do cliente com a entrega de um software de valor. Escolheu-se discorrer sobre as metodologias XP e SCRUM visto serem as metodologias ágeis mais utilizadas atualmente.

XP

De acordo com Dias (2005), o *Extreme Programming* (XP) é um Método Ágil voltado para equipes pequenas e médias desenvolverem software em ambientes que sofrem com mudança de requisitos. O XP possui iterações curtas, objetivando flexibilidade no projeto e diminuição de custos com possíveis mudanças, e usa o código gerado como indicador de evolução do projeto (LEAL, 2010).

O XP prioriza o desenvolvimento rápido do projeto e tem por objetivo assegurar a satisfação do cliente, auxiliando também no cumprimento das estimativas. A metodologia XP considera quatro valores principais: comunicação, simplicidade, *feedback* e coragem (SOARES, 2004).

- **Comunicação:** a finalidade da comunicação é prover a melhor relação possível entre cliente e desenvolvedor, priorizando conversas pessoais à outros meios de comunicação.
- **Simplicidade:** refere-se ao código, que não deve conter funções desnecessárias, sendo melhor implementar algo simples hoje e modificar futuramente, se necessário, do que codificar funções que poderão não ser utilizadas.
- **Feedback:** permite o programador obter informações constantes do cliente e do código, sendo que possíveis erros e não conformidades são identificados e corrigidos com mais rapidez e frequência.
- **Coragem:** a coragem é necessária para adotar e praticar os três valores anteriores, assim como a metodologia em si.

O ciclo da metodologia segue as seguintes etapas (LEAL, 2010):

- Exploração: são escritas estórias, cada qual contendo uma funcionalidade a ser desenvolvida no *release*;
- Planejamento: define-se o que é prioritário destas estórias com o cliente, sendo que nesta fase é realizada as estimativas de custo e prazo para o desenvolvimentos destas estórias;
- Iterações para *release*: são realizadas iterações até o primeiro *release* ser finalizado, de acordo com as prioridades preestabelecidas com o cliente;
- Validação para Produção: são executados testes para validação do software;
- Manutenção: após a liberação do primeiro *release*, haverá o desenvolvimento de outros *releases* com adição de novas funcionalidades;
- Morte: quando não há mais funcionalidades a serem desenvolvidas, pois o cliente está satisfeito com o software.

SCRUM

O *Scrum* foi criado em 1996 como um método que leva em conta que o desenvolvimento de software não é previsível e formaliza a abstração, sendo aplicável em ambientes instáveis (SCHWABER, 2009). O Scrum se diferencia dos demais métodos ágeis devido à maior ênfase dada ao gerenciamento do projeto (UDO; KOPPERNSTEINER, 2003, *apud* MARÇAL, 2009), possuindo atividades de monitoramento e *feedback*, reuniões diárias e curtas com toda a equipe, com o intuito de identificar e corrigir quaisquer não conformidades e impedimentos possíveis no processo de desenvolvimento do software.

O *Scrum* possui três papéis principais (SCHWABER, 2009): O *Product Owner*, que representa os interesses de todos os envolvidos no projeto; o *Team*, que são todos os que desenvolvem as funcionalidades do software; e o *ScrumMaster*, que assegura que todos sigam as regras e práticas do *Scrum*, além de ser o responsável por verificar e remover todos os impedimentos possíveis do projeto.

Um projeto com *Scrum* inicia-se com uma visão do produto que será desenvolvido, sendo que essa visão contém uma relação de características do produto, além de premissas e restrições estabelecidas pelo cliente (SCHWABER, 2009). Com essas informações, é criado o *Product*

Backlog, listando todos os requisitos conhecidos, e, após a elaboração do *Product Backlog*, ele é priorizado e dividido em *releases*.

Schwaber (2009) discorre que cada *Sprint* (iteração com duração entre duas a quatro semanas) tem início com uma reunião de planejamento (*Sprint Planning Meeting*), na qual o *Product Owner* e o *Team* entram em acordo sobre o que deverá ser implementado (*Select Product Backlog*), priorizando os requisitos mais importantes e definindo o *Sprint Backlog* (tarefas necessárias para implementar as funcionalidades escolhidas no *Product Backlog*), além de definir o que será realizado nas próximas *Sprints*. No decorrer das *Sprints*, são realizadas reuniões rápidas (*Daily Scrum Meeting*) para verificar o andamento das tarefas, o que está feito, o que está para fazer e o que está sendo impeditivo. Ao fim da *Sprint* é realizada a reunião de revisão (*Sprint Review Meeting*) para que o *Team* demonstre o que foi desenvolvido durante a *Sprint* ao *Product Owner*; sendo que nesse momento as funcionalidades poderão ser verificadas e validadas, e adaptações no projeto poderão ser realizadas. Após, é realizada uma reunião de retrospectiva, ministrada pelo *ScrumMaster* para o *Time*, com o intuito de melhorar os processos e/ou entregas e/ou envolvidos para a próxima *Sprint*. O ciclo de vida do *Scrum* é demonstrado na Figura 2.

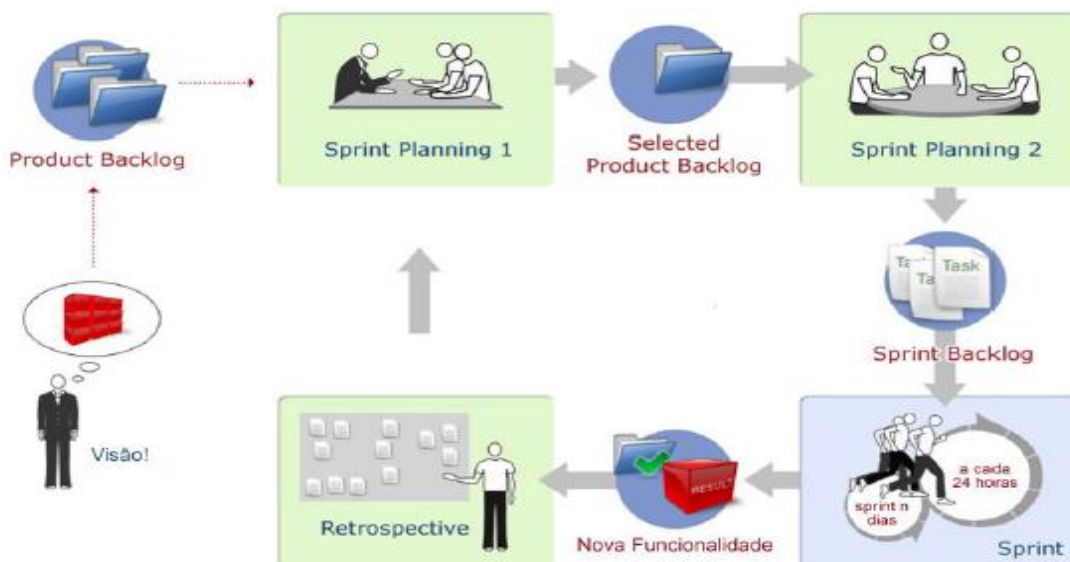


Figura 2 – Ciclo de vida do Scrum

Fonte: MARÇAL (2009)

2.1.3 Análise Comparativa entre Desenvolvimento Tradicional e Ágil

De acordo com SOARES (2004), as metodologias ágeis têm sido apontadas como uma alternativa às abordagens tradicionais para o desenvolvimento de software. Tanto as metodologias tradicionais quanto às ágeis possuem características distintas, tendo vantagens e desvantagens em relação umas às outras. O Quadro 1 apresenta uma comparação entre a abordagem clássica e a abordagem ágil para o desenvolvimento de software (DIAS, 2005, *apud* MARÇAL, 2009).

Quadro 1 – Comparação entre as abordagens de desenvolvimento de software

	Desenvolvimento Clássico	Desenvolvimento Ágil
Premissa fundamental	O software é totalmente especificável e previsível e pode ser construído por meio de um planejamento detalhado e extensivo	O software é adaptativo podendo ser construído por times pequenos, com princípios de melhoria contínua do projeto e o desenvolvimento baseado no rápido <i>feedback</i> e na aceitação de mudanças
Estilo de Gerenciamento	Comando e Controle	Liderança e colaboração
Distribuição de Papéis	Individual favorecendo a especialização	Equipes auto-organizadas, encorajando a multidisciplinaridade de papéis
Papel do cliente	Importante	Crítico

As metodologias tradicionais devem ser utilizadas em situações em que os requisitos do software são estáveis. Em ambientes instáveis, as metodologias ágeis se encaixam melhor, por possuírem a característica de serem adaptativas às mudanças. Além disso, as metodologias ágeis permitem entregas constantes ao cliente, ao contrário das tradicionais, em que o cliente tem acesso ao software no fim do desenvolvimento. Por fim, as metodologias ágeis se mostram uma alternativa viável em ambientes com requisitos instáveis e promove maior

envolvimento com o cliente, porém ainda peca em alguns aspectos, como a falta de análise de riscos. (SOARES, 2004).

2.2 Qualidade de produto e processo no desenvolvimento de software

O conceito “qualidade” não é estranho às pessoas, porém deve se levar em conta de que esse conceito não tem o mesmo significado para todos. Segundo Paladini (2004), apesar de ser uma palavra de uso comum, ao se propor uma definição técnica de qualidade, dois fatores devem ser levados em consideração:

- A definição proposta não deve se opor a noção intuitiva que se tem sobre o assunto
- Por ser de uso cotidiano das pessoas, não é possível delimitar seu significado com acurácia

Assim sendo, existem várias definições para qualidade, com diversos enfoques. Segundo Juran e Gryna (1991, *apud* PALADINI, 2004), “qualidade é adequação ao uso”. Para Deming (2000, *apud* COSTA *et al.*, 2001), o significado de qualidade é atender e, se possível, exceder às expectativas do cliente. De acordo com Campos (1999), um produto ou serviço de qualidade é aquele que atende de forma acessível, confiável e no prazo correto as necessidades do consumidor.

O gerenciamento de qualidade leva em conta a seguinte premissa: a qualidade do processo de desenvolvimento interfere diretamente na qualidade do produto fornecido. Segundo Sommerville (2004), quando o produto é tangível, a relação produto/processo é clara: melhorar um processo visando diminuir falhas conduzirá a um melhor produto. Porém essa relação é menos óbvia quando o produto é intangível e dependente de pessoas no processo. Sommerville (2004) ressalta que a qualidade de software é dependente do processo de projeto, em que considerações das pessoas envolvidas são significativas, sendo que, em alguns casos, o processo utilizado é o diferencial mais relevante para a qualidade do produto.

Existem quatro fatores que podem afetar a qualidade de software. Esses fatores são determinantes, dependendo da característica da equipe de desenvolvimento e tipo de projeto. Para sistemas muito grandes, com uma equipe grande, o principal fator que influencia a qualidade do produto é o processo. Já para pequenos projetos, com poucos membros, a

qualidade do pessoal é mais importante do que o processo de desenvolvimento. Além disso, a tecnologia de desenvolvimento é mais importante para pequenas equipes do que para as grandes, visto que uma boa tecnologia influencia diretamente em sua produtividade, algo que não acontece nas grandes equipes (sua produtividade não é influenciada pelas ferramentas de desenvolvimento). Porém, o fator “Custo, tempo e cronograma” é fundamental para qualquer tipo de projeto, visto que, sem os recursos necessários, não é possível ter uma implementação eficaz, afetando conseqüentemente o processo e a qualidade do produto (SOMMERVILLE, 2004). Os quatro fatores são demonstrados na Figura 3.

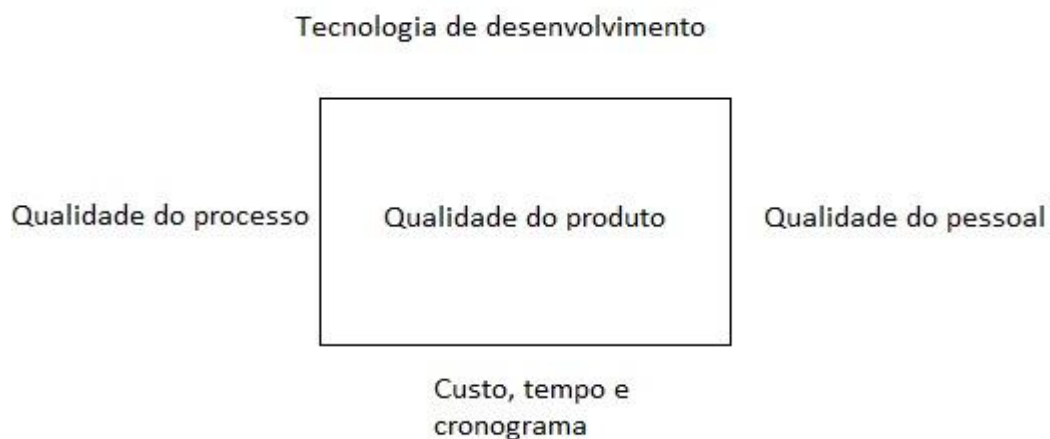


Figura 3 – Principais fatores da qualidade de produtos de software

Fonte: SOMMERVILLE (2004)

2.2.1 Avaliação e monitoração da qualidade do processo de desenvolvimento de software e produto

Sommerville (2004) discorre que a qualidade de software é um conceito complexo, que não se consegue definir de maneira simplificada. A definição clássica citada por Crosby (1979, *apud* SOMMERVILLE, 2004), que diz que se tem um produto de qualidade quando ele cumpre com sua especificação, não é totalmente aplicável ao software, conforme Sommerville (2004) relata nos seguintes problemas:

1. A especificação de requisitos deve refletir a necessidade do cliente, porém nem todos os requisitos de fato estarão inclusos na especificação.
2. É complexo definir determinadas características da qualidade, de maneira objetiva.

3. Existe a dificuldade de se escrever a especificação completa do software; desse modo, é possível que o produto atenda aos requisitos da especificação mas ainda assim não satisfazer por completo o cliente.

Na visão de Pressman (2005, p. XXX), a qualidade de software pode ser definida como “a conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente desenvolvido”.

Reconhecendo o problema da especificação imperfeita e incompleta, deve-se estabelecer procedimentos para melhorar a qualidade dentro desse cenário. A princípio, o gerenciamento da qualidade envolve definir padrões e procedimentos que devem ser usados no decorrer de todo o desenvolvimento do software, e verificar se estão sendo cumpridos por todos os envolvidos (SOMMERVILLE, 2004).

As atividades de garantia de qualidade estabelecem uma estrutura para alcançar a qualidade do software, sendo que esse processo compreende definir padrões que devem ser aplicados no processo de desenvolvimento de software e no produto (SOMMERVILLE, 2004). A garantia da qualidade do software, também conhecida pela sigla SQA (*Software Quality Assurance*) envolve diversas tarefas associadas a sete grandes atividades (PRESSMAN, 2005):

1. Aplicação de métodos técnicos: métodos e ferramentas técnicas auxiliam os envolvidos a entregarem produtos de trabalho de alta qualidade.
2. Realização de revisões técnicas formais: a revisão técnica formal é uma reunião realizada com os envolvidos no projeto com o intuito de descobrir problemas de qualidade, sendo consideradas tão efetivas quanto os testes para descobrir problemas da qualidade
3. Atividades de teste de software: múltiplos casos de teste são projetados para garantir a detecção de erros e não conformidades, sendo esta atividade uma espécie de “rede de segurança” para a garantia da qualidade.
4. Aplicação de padrões: quando existem padrões e procedimentos formais estabelecidos na organização, uma atividade SQA deve ser definida para garantir que eles sejam seguidos, como uma auditoria.
5. Controle de mudanças: o processo de controle de mudanças colabora efetivamente para a qualidade de software ao formalizar pedidos de mudança, averiguar sua natureza e

controlar seu impacto, sendo esse controle realizado tanto durante o processo de desenvolvimento quanto na fase de manutenção do software.

6. Medição: um importante objetivo da SQA é rastrear a qualidade de software e averiguar o impacto das mudanças de métodos e procedimentos sobre a qualidade, utilizando, para esse fim, métricas de software, que deverão ser coletadas e analisadas.

7. Manutenção de registros e reportagem: oferecem procedimentos para a coleta e disseminação de informações de SQA em relação aos resultados das revisões, auditorias, controle de mudanças, testes e outras atividades SQA.

2.3 Modelos para Melhoria no Processo de Desenvolvimento de Software

2.3.1 Modelo de Maturidade de Capacitação Integrada (CMMI)

Desenvolvido pelo SEI (*Software Engineer Institute* – Instituto de Engenharia de Software), instituto ligado à universidade *Carnegie-Mellon*, o CMMI (do inglês *Capability Maturity Model Integration*) representa uma abordagem para melhoria de processos, reunindo melhores práticas que comporta todo o ciclo de vida, desde sua concepção até a entrega final, sendo um guia para melhorar o gerenciamento do desenvolvimento e manutenção de produtos e serviços (MARÇAL, 2009). Dentre os modelos desenvolvidos pelo o SEI, o CMMI-dev (*Capability Maturity Model Integration for Development*) é específico para melhoria de processos em software (MONTONI, 2010).

Segundo Zanatta e Vilain (2006), o CMMI-dev possui 4 disciplinas (áreas de conhecimento): Engenharia de Sistemas, Engenharia de Software, Produto Integrado e Desenvolvimento de Processos, sendo que estas disciplinas são formadas por áreas de processo. Segundo Fiorini (1998, *apud* ZANATTA; VILAIN, 2006, pag. 2) “área de processo é um conjunto de práticas relacionadas em uma área que quando executadas coletivamente satisfazem um conjunto de objetivos importantes para a melhoria significativa daquela área.” Ao executar-se as áreas de processo, é alcançada a melhoria do processo na disciplina. Além disso, as áreas de processo possuem objetivos específicos (SP – *specific practice*) e objetivos genéricos (SG – *specific goal*). Os objetivos específicos relacionam-se especificamente a uma área de processo, enquanto os objetivos genéricos abrangem várias áreas de processo e também a organização, e dizem respeito ao que deve ser cumprido para que a institucionalização das práticas na organização possa ocorrer (MONTONI, 2010).

O CMMI-dev descreve 22 áreas de processo e possui duas representações para melhoria de processos (SEI, 2008). As representações são demonstradas na Figura 4.

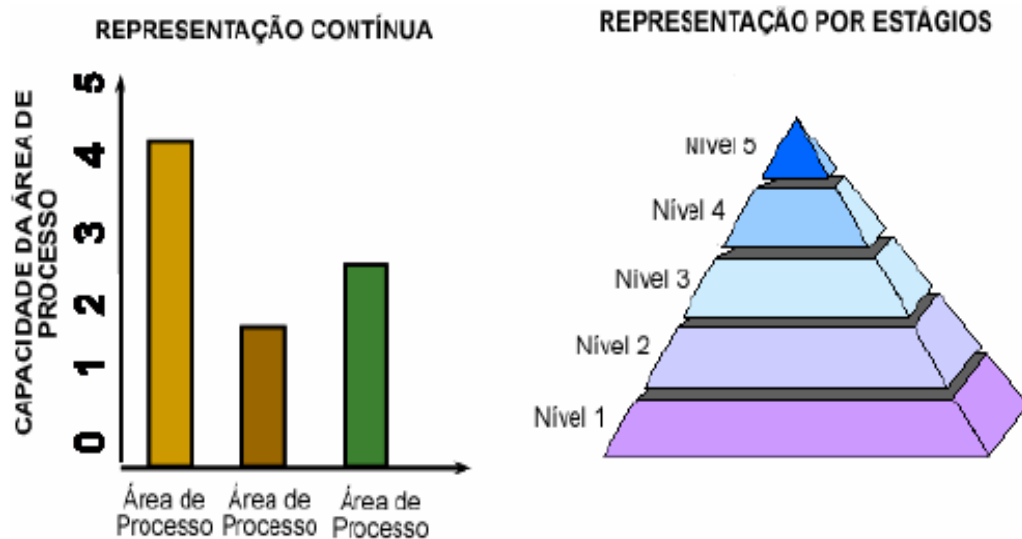


Figura 4 – Representações do CMMI

Fonte: MARÇAL (2009)

A representação “Estágio” possui 5 níveis de maturidade: 1) Inicial, 2) Gerenciado, 3) Definido, 4) Gerenciado Quantitativamente, 5) Em Otimização, sendo que cada estágio possui várias áreas de processo que se encontram no mesmo nível.

A representação “Contínua” não depende de níveis de maturidade, sendo que uma determinada área do processo poderá ter sua capacidade avaliada independentemente das outras áreas de processo. A representação “Contínua” tem 6 níveis para dimensão de capacitação: 1) Incompleto, 2) Executado, 3) Gerenciado, 4) Definido, 5) Gerenciado Quantitativamente, 6) De otimização (ZANATTA; VILAIN, 2006).

Os cinco níveis de maturidade do modelo na representação por estágios possuem as seguintes características (MARÇAL, 2009):

1. Inicial: Os processos não são definidos e o ambiente da organização é instável, sendo que seu sucesso depende da competência de colaboradores e não do uso de processos

comprovados. Apesar desse ambiente informal e caótico, essas organizações conseguem entregar produtos e serviços; todavia, elas frequentemente extrapolam o orçamento e o cronograma de seus projetos.

2. Gerenciado: Os requisitos, processos, produtos de trabalho e serviços são geridos. O estado dos produtos de trabalho e a entrega dos serviços são evidentes para o gerenciamento em marcos definidos. Compromissos são estabelecidos entre os interessados relevantes e são revistos de acordo com a necessidade. Os produtos de trabalho são revistos com os interessados e controlados. Os produtos de trabalho e serviços cumprem seus requisitos, padrões e objetivos são estabelecidos.

3. Definido: Todos os processos padrão da organização são definidos e melhorados continuamente. Os projetos estabelecem seus processos definidos adequando o conjunto de processos padrão da organização. A alta gestão define os objetivos dos processos e garante que esses estão sendo conduzidos de forma adequada. Os processos são geridos de forma mais proativa, utilizando um entendimento da integração das atividades de processos e medidas detalhadas do processo, seus produtos de trabalho e seus serviços.

4. Gerenciado Quantitativamente: São escolhidos os subprocessos que colaboram consideravelmente para o desempenho geral do processo. A qualidade e o desempenho do processo são compreendidos em termos estatísticos e são geridos durante o ciclo de vida dos processos. Novos indicadores, medidas de qualidade e desempenho de processos serão disponibilizados, podendo fornecer suporte a futuras decisões baseadas em fatos ocorridos.

5. Otimizado: Os processos são melhorados sistematicamente, com base em uma percepção quantitativa das causas usuais de variações intrínsecas aos processos.

O Quadro 2 mostra as vinte e duas áreas de processo do modelo CMMI com suas categorias e classificações em relação ao nível de maturidade conforme a representação por estágios do modelo.

Quadro 2 – Áreas de processo do CMMI (MARÇAL, 2009)

Nível	Área de processo	Sigla	Categoria
2	Gerenciamento de Requisitos	REQM	Engenharia
	Planejamento do Projeto	PP	Gerenciamento de projetos
	Controle e Monitoramento do Projeto	PMC	Gerenciamento de projetos
	Gerenciamento e Acordo com Fornecedores	SAM	Gerenciamento de projetos
	Medição e Análise	MA	Suporte
	Garantia da Qualidade do Processo e do Produto	PPQA	Suporte
	Gerenciamento de Configuração	CM	Suporte
3	Desenvolvimento de Requisitos	RD	Engenharia
	Solução Técnica	TS	Engenharia
	Integração de Produtos	PI	Engenharia
	Verificação	VER	Engenharia
	Validação	VAL	Engenharia
	Foco no Processo Organizacional	OPF	Gerenciamento de Processos
	Definição do Processo Organizacional + IPPD	OPD	Gerenciamento de Processos
	Treinamento Organizacional	OT	Gerenciamento de Processos
	Gerenciamento Integrado de Projetos Desenvolvimento Integrado do Produto e do Processo	IPM +IPPD	Gerenciamento de Projetos
	Gerenciamento de Riscos	RSK	Gerenciamento de Projetos
Análise de Decisões e Resoluções	DAR	Suporte	
4	Desempenho do Processo Organizacional	OPP	Gerenciamento de Processos
	Gerenciamento Quantitativo do Projeto	QPM	Gerenciamento de Projetos

Nível	Área de processo	Sigla	Categoria
5	Inovação e Desenvolvimento Organizacional	OID	Gerenciamento de Processos
	Análise de Causas e Resoluções	CAR	Suporte

As áreas de processos se dividem em 4 categorias (CMMI, 2008):

- **Gestão de processos:** concentra as áreas relacionadas aos projetos, associados à definição, ao planejamento, à implementação, ao monitoramento, ao controle, a medição e à melhoria do processo. Essas áreas fornecem à organização a capacidade de documentar e disponibilizar as melhores práticas, lições aprendidas e ativos de processo.
- **Gestão de projetos:** abrange as atividades de gestão ligadas à planejamento, monitoramento e controle do projeto, atividades relacionadas ao estabelecimento e manutenção do plano de projeto e compromissos, verificação do progresso em relação ao plano estabelecido, adoção de ações corretivas e gestão de acordo com fornecedores.
- **Engenharia:** tratam de atividades de desenvolvimento e manutenção de variadas disciplinas da Engenharia, integrando os processos relacionados à diversas áreas da Engenharia em um único processo de desenvolvimento do produto, propondo uma estratégia de melhoria de processo orientada à produto.
- **Suporte:** possui as atividades que apoiam o desenvolvimento e a manutenção do produto, tendo uma atenção nos os processos que são utilizados no contexto de execução de outros processos.

O nível 2 da representação por estágios do modelo possui as seguintes áreas de processo (CMMI, 2008):

- **Gerenciamento de Requisitos (REQM):** descreve atividades para adquirir e controlar alterações de requisitos e certificar que outros dados relevantes mantenham-se atualizados, fornecendo rastreabilidade de requisitos, desde o cliente até o produto. Além disso, garante que as mudanças ocorridas nos requisitos sejam percebidas em planos, atividades e produtos do trabalho de projeto.

- Planejamento do Projeto (PP): trata da elaboração do plano de projeto, o envolvimento dos *stakeholders* e a obtenção do comprometimento com o plano e sua manutenção. O plano de projeto abrange as diversas atividades de gestão e desenvolvimento de projeto executadas no contexto do projeto. O projeto verifica outros planos que o influenciam, gerados pelos *stakeholders*, estabelecendo compromissos a respeito de suas colaborações com o projeto.
- Controle e Monitoramento do Projeto (PMC): abrange atividades de monitoramento, controle e implantação de ações corretivas, sendo que o nível de monitoramento é determinado pelo plano de projeto. Ações corretivas são implementadas quando o *status* do projeto diverge do que é esperado.
- Gerenciamento e Acordo com Fornecedores (SAM): cuida das necessidades de obtenção de partes do trabalho que são produzidas pelos fornecedores. Faz-se a verificação de quais fontes de produtos são necessários para satisfazer os requisitos do projeto, assim, o fornecedor é selecionado e é definido um contrato para gerenciá-lo, sendo que o desempenho do fornecedor é acompanhado por meio do monitoramento dos processos e produtos de trabalho selecionados.
- Medição e Análise (MA): fornece práticas específicas que norteiam os projetos e as organizações no alinhamento das necessidades e objetivos de medição, obtendo resultados que poderão ser utilizados para implantação de ações corretivas, e auxiliando na tomada de decisão.
- Garantia da Qualidade do Processo e do Produto (PPQA): fornece práticas específicas para avaliar com objetividade processos, produtos de trabalho e serviços, correlacionando com as descrições dos processos e procedimentos aplicáveis. Esta área de processo garante a entrega de produtos e serviços de qualidade, concedendo a todos os envolvidos visibilidade e *feedback* sobre os processos e produtos de trabalho no decorrer do ciclo de vida do projeto
- Gerenciamento de Configuração (CM): estabelece e mantém a integridade de todos os produtos de trabalho, usando identificação, controle, balanço de atividades e auditorias de configuração. Os produtos de trabalhos que são geridos pela configuração abrangem os produtos de trabalho interno, produtos adquiridos e quaisquer ferramentas utilizadas para desenvolver esses produtos de trabalhos.

O Quadro 3 lista todas as práticas por áreas de processo, objetivos específicos (SP) e objetivos genéricos (SG) (CMMI, 2002).

Quadro 3 – Práticas por áreas de processo do CMMI propostas pelo Guia CMMI-dev (MARÇAL, 2009)

Práticas por área de processo	
Gestão de Configuração (CM)	
SG 1 Estabelecer <i>Baselines</i>	SP 1.1 Identificar Itens de Configuração
	SP 1.2 Estabelecer um Sistema de Gestão de Configuração
	SP 1.3 Criar ou Liberar <i>Baselines</i>
SG 2 Acompanhar e Controlar Mudanças	SP 2.1 Acompanhar Solicitações de Mudança
	SP 2.2 Controlar Itens de Configuração
SG 3 Estabelecer Integridade	SP 3.1 Estabelecer Registros de Gestão de Configuração
	SP 3.2 Executar Auditorias de Configuração
Medição e Análise (MA)	
SG 1 Alinhar Atividades de Medição e Análise	SP 1.1 Estabelecer Objetivos de Medição
	SP 1.2 Especificar Medidas
	SP 1.3 Especificar Procedimentos de Coleta e Armazenamento de Dados
	SP 1.4 Especificar Procedimento de Análise
SG 2 Fornecer Resultados de Medição	SP 2.1 Coletar Dados Resultantes de Medição
	SP 2.2 Analisar Dados Resultantes de Medição
	SP 2.3 Armazenar Dados e Resultados
	SP 2.4 Comunicar Resultados
Monitoramento e Controle de Projeto (PMC)	
SG 1 Monitorar o Projeto em Relação ao Plano	SP 1.1 Monitorar os Parâmetros de Planejamento do Projeto
	SP 1.2 Monitorar Compromissos
	SP 1.3 Monitorar Riscos do Projeto
	SP 1.4 Monitorar a Gestão de Dados
	SP 1.5 Monitorar o Envolvimento das Partes Interessadas

Quadro 3 – Práticas por áreas de processo do CMMI propostas pelo Guia CMMI-dev (MARÇAL, 2009) – continuação

Práticas por área de processo	
Monitoramento e Controle de Projeto (PMC)	
SG 1 Monitorar o Projeto em Relação ao Plano	SP 1.6 Conduzir Revisões de Progresso
	SP 1.7 Conduzir Revisões de Marco
SG 2 Gerenciar Ações Corretivas até sua Conclusão	SP 2.1 Analisar Questões Críticas
	SP 2.2 Implementar Ações Corretivas
	SP 2.3 Gerenciar Ações Corretivas
Planejamento de Projeto (PP)	
SG 1 Estabelecer Estimativas	SP 1.1 Estimar o Escopo do Projeto
	SP 1.2 Estabelecer Estimativas para Atributos de Produtos de Trabalho e de Tarefas.
	SP 1.3 Definir Ciclo de Vida do Projeto
	SP 1.4 Determinar Estimativas de Esforço e Custo
SG 2 Elaborar um Plano de Projeto	SP 2.1 Estabelecer Orçamento e Cronograma
	SP 2.2 Identificar Riscos do Projeto
	SP 2.3 Planejar Gestão de Dados
	SP 2.4 Planejar Recursos do Projeto
	SP 2.5 Planejar Habilidades e Conhecimento Necessários
	SP 2.6 Planejar o Envolvimento das Partes Interessadas
	SP 2.7 Estabelecer o Plano do Projeto
SG 3 Obter Comprometimento com o Plano	SP 3.1 Revisar Planos que Afetam o Projeto
	SP 3.2 Conciliar Carga de Trabalho e Recursos
	SP 3.3 Obter Comprometimento com o Plano

Quadro 3 – Práticas por áreas de processo do CMMI propostas pelo Guia CMMI-dev (MARÇAL, 2009) – continuação

Práticas por área de processo	
Garantia da Qualidade de Processo e Produto (PPQA)	
SG 1 Avaliar Objetivamente Processos e Produtos de Trabalho	SP 1.1 Avaliar Objetivamente os Processos
	SP 1.2 Avaliar Objetivamente Produtos de Trabalho e Serviços
SG 2 Fornecer Visibilidade	SP 2.1 Comunicar e Assegurar a Solução de Não conformidades
	SP 2.2 Estabelecer Registros
Gestão de Requisitos (REQM)	
SG 1 Gerenciar Requisitos	SP 1.1 Obter Entendimento dos Requisitos
	SP 1.2 Obter Comprometimento com os Requisitos
	SP 1.3 Gerenciar Mudanças nos Requisitos
	SP 1.4 Manter Rastreabilidade Bidirecional dos Requisitos
	SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos
Gestão de Contrato com Fornecedores (SAM)	
SG 1 Estabelecer Contratos com Fornecedores	SP 1.1 Determinar Tipo de Aquisição
	SP 1.2 Selecionar Fornecedores
	SP 1.3 Estabelecer Contratos com Fornecedores
SG 2 Cumprir Contratos com Fornecedor	SP 2.1 Executar Contrato com Fornecedor
	SP 2.2 Monitorar Processos Selecionados do Fornecedor
	SP 2.3 Avaliar Produtos de Trabalho Selecionados do Fornecedor
	SP 2.4 Aceitar Produto Adquirido
	SP 2.5 Transferir Produtos

Dificuldades e fatores críticos na adoção do modelo

Segundo Santos (2008), o investimento em certificações possibilita às empresas conquistar clientes no exterior, assim como o aumento de suas receitas, além do fato de que as certificações evidenciam a busca pela qualidade.

MEZZENA e ZWICKER (2007) citam alguns fatores fundamentais para obtenção de resultados eficazes na implantação do modelo CMMI, tais como o comprometimento dos colaboradores participantes e o envolvimento da alta gerência, que facilita a superação das eventuais resistências à mudanças entre as pessoas da organização. Entretanto, existem também problemas relacionados à adoção. MEZZENA e ZWICKER (2007) discorrem que o volume de investimentos requeridos para implantar o modelo mostra-se como uma dificuldade, visto que o custo com consultoria externa e avaliação do modelo pode ser elevado. Outro ponto associado à problemas com a adoção do modelo CMMI está relacionado ao fato de que nem sempre a implantação do modelo atende aos objetivos estratégicos das organizações de software, visto que o modelo visa buscar a excelência operacional por meio da inovação dos processos da organização, e não necessariamente dos produtos de software (MONTONI, 2010).

2.3.2 Modelo de Melhoria de Processo do Software Brasileiro (MPS.Br)

Foi criado em dezembro de 2003, pela associação para a promoção da excelência de software brasileiro, o modelo de Melhoria de Processo do Software Brasileiro (SANTANA, 2006). Este modelo visa a melhoria de qualidade dos processos de desenvolvimento de software em organizações que possuem poucos recursos para investimento em qualidade, sendo que este modelo não define novos conceitos, mas adequa estratégias de implementação já existentes para a realidade brasileira (WEBBER et al., 2004, *apud* SANTANA, 2006).

O Modelo MPS inclui um Modelo de Referência para Melhoria do Processo (MR – MPS) e um Método de Avaliação para Melhoria do Processo de Software (MA – MPS) com os seguintes aspectos (WEBER *et al*, 2006):

- Conformidade com as normas ISO/IEC 12207 e ISO/IEC 15504;
- Compatibilidade com o modelo CMMI;
- Baseado nas melhores práticas da Engenharia de Software;

- Criado para a realidade das empresas brasileiras;

O modelo define níveis de maturidade que são uma combinação entre os processos e sua capacidade (MPS.Br, 2011). As atividades e tarefas não são descritas no modelo, sendo essa função cargo dos usuários a realizarem.

Os níveis de maturidade estipulam patamares evolutivos dos processos. O MR – MPS define sete níveis de maturidade (MPS.Br, 2011): A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado).

De acordo com Weber *et al.* (2008), o primeiro nível de maturidade é o G (Parcialmente Gerenciado), sendo composto pelos processos mais essenciais de gerência. O segundo nível é o F (Gerenciado), no qual são institucionalizados processos de apoio para o desenvolvimento de software. O nível E (Parcialmente Definido) possui o grupo de processos que apoiam a implantação e melhoria dos processos padrão para conduzir projetos de software. O quarto nível é o D (Largamente Definido), cujo o foco é a melhoria dos processos técnicos da engenharia de software. O próximo nível, o C (Definido), é constituído de processos de engenharia de software que complementam o gerenciamento de projetos. Finalizando, os níveis B (Gerenciado Quantitativamente) e A (Em Otimização) são de elevada maturidade, com o foco na melhoria contínua dos processos.

3. DESENVOLVIMENTO

3.1 Metodologia

Segundo Moresi (2003, p.8), “pesquisa é um conjunto de ações, propostas para encontrar a solução para um problema, que têm por base procedimentos racionais e sistemáticos.”

A pesquisa realizada neste trabalho possui caráter exploratório. Envolve a pesquisa bibliográfica, composta de livros, artigos científicos e revistas referentes ao tema investigado.

A pesquisa de campo caracteriza-se por um estudo de caso. De acordo com Moresi (2003), estudo de caso são investigações de acontecimentos à medida que ocorrem, sem interferência do pesquisador, objetivando a compreensão do evento em investigação, analisando o contexto e os processos envolvidos. O estudo de caso também é visto como o delineamento mais preciso para a investigação de um fenômeno contemporâneo inserido em seu contexto real, onde as fronteiras entre o fenômeno e o contexto não são percebidos com clareza (YIN, 2001, *apud* GIL, 2002).

Este estudo de caso foi conduzido em uma empresa (fábrica) de software. O levantamento de dados foi realizado por meio de observação, entrevistas e questionários com colaboradores. A análise dos dados permitiu a identificação dos fatores críticos e boas práticas, a partir do repositório de práticas formado e delineamento da opinião dos entrevistados, assim como lições aprendidas com o projeto de implantação.

3.2 Contexto da empresa

A empresa Benner Sistemas é a terceira maior fornecedora nacional de softwares de gestão, conforme notícia divulgada em sua homepage institucional. É composta de fábricas de software, cuja filial em Maringá (doravante Benner Maringá) oferece soluções de Tecnologia de Informação para gestão da saúde, tendo como clientes operadoras de planos de saúde em todo o Brasil, tais como: Porto Seguro Saúde, CASSI, Unimed Curitiba, BACEN, TRF, entre outros.

Sua história em Maringá começou em 1999, quando os sócios da empresa identificaram uma oportunidade de mercado na região e montaram uma fábrica para desenvolver o software “Benner Saúde”, solução voltada para gestão de operadoras de saúde, que contribuiu para o desenvolvimento da empresa e a levou à liderança nesse segmento.

Os principais serviços oferecidos pela empresa, a partir da aquisição do software de gestão da Saúde, são: desenvolvimento de novas funcionalidades de acordo com demanda; suporte técnico e manutenção do sistema. A partir de uma quantidade de novos requisitos solicitados pelos clientes, é estipulada uma nova versão do sistema, com prazos de liberação acordados com os *stakeholders*. O suporte técnico funciona em horário comercial, podendo existir colaboradores de plantão aos fins de semana para dar suporte a clientes em situações críticas.

O foco principal da Benner Sistemas é no cliente, provendo relacionamento e atendimento personalizados, visando permitir, com seus sistemas, uma gestão enxuta e simplificada. Sua missão: *Proporcionar prazer em gerir e desenvolver negócios.*

Em 2011, a empresa se une por meio de uma *joint venture* ao grupo TBA e lançam a empresa Globalweb, sendo essa uma das 10 maiores empresas de tecnologia de informação no Brasil e com a previsão de até 2014 atingir o faturamento de R\$ 500 milhões.

3.3 Visão de negócio do processo de desenvolvimento de software

Todo o processo de desenvolvimento de software da Benner Saúde é gerenciado por projetos, com início e fim definidos, sendo que existem vários projetos sendo executados paralelamente. Apesar dos projetos serem diferentes, seguem as mesmas etapas. A visão de negócio do processo de desenvolvimento é ilustrada na Figura 5.

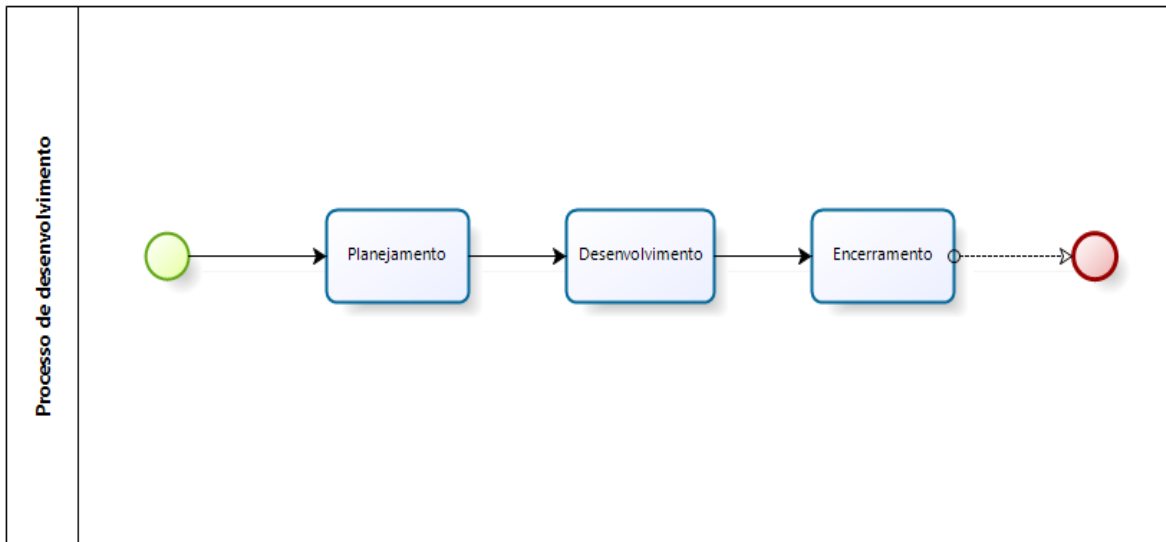


Figura 5 – Visão de negócio do processo de desenvolvimento

✓ **Planejamento**

É a etapa onde é estipulado o cronograma do projeto, assim como os custos e esforços necessários para sua realização, partindo da demanda (requisitos) dos clientes. Ao fim do planejamento, serão selecionados quais requisitos são os mais importantes (na situação em que nem todos possam ser implementados) e é elaborado o plano do projeto.

✓ **Desenvolvimento**

A partir dos requisitos são desenvolvidas e validadas as novas funcionalidades para uso no sistema. A parte de desenvolvimento será detalhada no tópico 3.2.1.

✓ **Encerramento**

Ao término do desenvolvimento são disponibilizadas aos clientes as novas funcionalidades desenvolvidas. Há uma reunião de avaliação e o encerramento do projeto.

3.3.1 Desenvolvimento

O processo de desenvolvimento de software da Benner Saúde é baseado no modelo cascata e aderente ao CMMI nível 2. As principais entregas da empresa constituem em novas funcionalidades implementadas e suporte técnico (manutenção do sistema).

Novas implementações

Todas as necessidades de alteração solicitadas por um cliente, seja uma nova implementação ou correção, são vinculadas a uma SMS (Solicitação de Manutenção de Sistema), sendo as SMSs geridas por uma ferramenta que permite o controle dessas solicitações, e gerencia todo o ciclo de vida de desenvolvimento.

O processo de desenvolvimento é ilustrado na Figura 6. A partir da análise de viabilidade e orçamento dos esforços necessários para desenvolver as novas funcionalidades requisitadas, é realizada a análise de negócio, discorrendo sobre as regras de negócio a serem introduzidas/alteradas. A partir da análise de negócio, é realizada a análise técnica, que é visão de negócio em "baixo nível", ou seja, o que deverá ser inserido/modificado no sistema, em uma linguagem técnica. Com a análise técnica pronta, é possível iniciar a parte de codificação, que transformará a documentação da análise técnica em código de máquina. Ao término da codificação, são realizados os testes para verificação e validação dos requisitos implementados. Após os testes, são documentados todos os novos processos/alterações do sistema, e, ao disponibilizar as novas funcionalidades para o cliente, sempre pode existir a necessidade de manutenção, caso sejam encontrados erros na homologação pelo cliente.

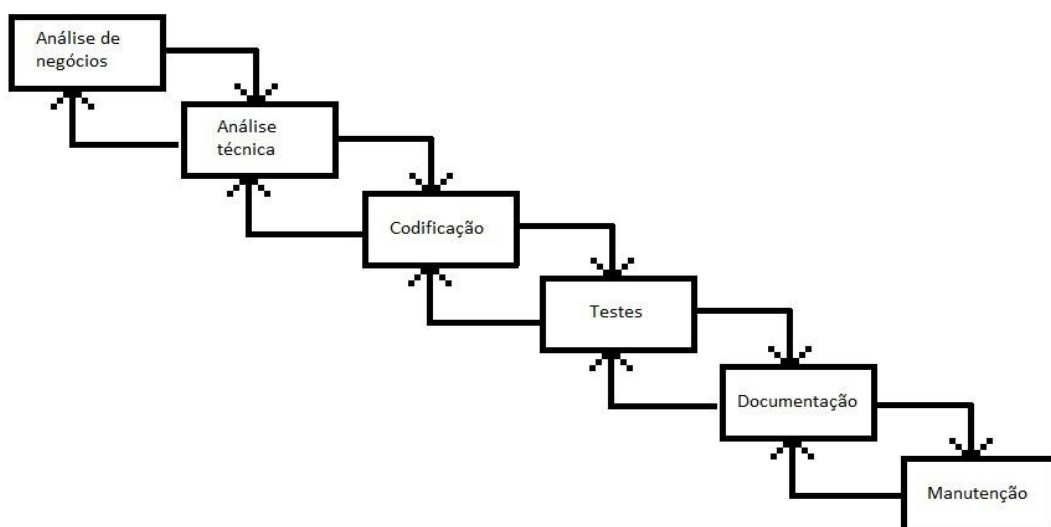


Figura 6 – Processo de desenvolvimento de software baseado no modelo "cascata"

Na situação de não-conformidades em qualquer uma das etapas, é possível voltar à etapa anterior para correção, sendo este tipo de movimentação previsto pela ferramenta gerencial da empresa, que controla o fluxo das SMSs desta maneira.

Suporte técnico (manutenção)

A Benner Saúde disponibiliza suporte técnico para seus sistemas. O processo é ilustrado pela Figura 6. O cliente, quando possui dúvida a respeito do sistema ou encontra um possível erro no mesmo, abre um novo chamado (protocolo). Esse chamado será verificado pelos recursos competentes (técnicos de suporte), que irão averiguar o conteúdo do mesmo. Caso seja uma dúvida a respeito do sistema, retornará o protocolo para o cliente com as respostas cabíveis. Caso ateste que foi encontrado um erro no sistema, e de fato o é, abrirá uma SMS de correção pontual para tal não conformidade. Todo este processo é gerido pela ferramenta gerencial de ciclo de vida da empresa.

3.4 Certificação CMMI nível 2

O método de implantação do modelo CMMI nível 2 na Benner Maringá foi identificado por meio de entrevistas realizadas com os colaboradores que participaram ativamente do processo de implantação. Por meio das respostas, foi possível identificar quais foram as ações tomadas pela empresa para alcançar a certificação. O roteiro da entrevista pode ser verificado no Apêndice 1.

A busca da Benner Saúde, uma das maiores filiais do grupo Benner Sistemas, pela otimização dos seus processos se fez presente com o crescimento da empresa e a expectativa de novos clientes. A aderência ao modelo CMMI (*Capability Maturity Model Integration*) traria, além de um arcabouço de práticas para uma eficiente gestão de processos, uma certificação de renome, reconhecida internacionalmente. A situação anterior à implantação do modelo CMMI na empresa era caracterizada por processos que não eram formalmente definidos e uma gestão de projetos com muito a melhorar. Com a intenção de mudar este cenário, foi criado um grupo com pessoas-chave da organização (SEPG – *Software Engineering Process Group*) para trabalhar com melhoria de processos, e, posteriormente, seus esforços foram direcionados para iniciar a implantação do modelo CMMI na empresa em Maringá.

O primeiro passo constituiu a contratação de uma consultoria especializada para apoio e direcionamento. Além disso, foi contratado um colaborador para atuar especificamente no projeto de implantação.

Com o auxílio da consultoria, foram identificadas as atividades necessárias (tópico 3.3.1) para que fossem atendidas as áreas de processo do nível 2 do modelo, e a partir disso foi estabelecido um plano de melhorias e o WBS (*work breakdown structure* – estrutura analítica de projeto, Anexo I), com divisão de responsabilidades e priorização das melhorias. Dentre as áreas de processos sugeridas pelo Guia CMMI-dev para o nível 2 (tópico 2.3.1, pág. 24), foi decidido que a área de processo “Gerenciamento e Acordo com Fornecedores” (SAM) não seria implantada, devido ao fato de não existir uma equipe local que lidasse com fornecedores, logo foi decidido com a consultoria que esta área de processo seria definida em um momento futuro.

Para realização das atividades propostas, foram formados grupos de trabalho com colaboradores-chave. O andamento das atividades era gerido por uma ferramenta específica da consultoria, que listava todas as práticas necessárias para aderência ao modelo, além de mostrar se determinada prática estava ou não concluída, assim como sua evidência (produto de trabalho).

Com todas as atividades finalizadas, foi efetuada uma avaliação de validação dos produtos de trabalho gerados, avaliação essa realizada pela consultoria, e iniciou-se a institucionalização das novas práticas. Foram realizados treinamentos referentes aos novos processos com colaboradores-chaves das equipes de desenvolvimento, os novos guias (Quadro 3, “Artefatos”, tópico 3.3.1) contendo as práticas foram disponibilizados em um repositório para acesso a todos os colaboradores, e gradativamente todos os projetos de desenvolvimento foram adequados ao modelo.

O monitoramento da institucionalização foi realizado continuamente. A auditoria interna era realizada por um colaborador específico, responsável por averiguar se as práticas estavam sendo cumpridas conforme o esperado, e, do encontro de não conformidades, tomar as ações cabíveis. Além disso, foram escolhidos os colaboradores que atuariam como Auditores na avaliação oficial, que passaram pelos treinamentos necessários (CMMI, e SCAMPI – treinamento para o método de avaliação).

Atingido o *deadline* estabelecido pelos sócios, foi realizado o *Pre Appraisal*, que constitui em entrevistas realizadas com alguns envolvidos nos projetos auditados, com o intuito de verificar se a empresa estava apta a passar pela Auditoria Oficial. Confirmado isso, a empresa

passou pela Auditoria Oficial do CMMI, por suas duas fases: *readness* e entrevistas. Todas as áreas de processo foram avaliadas, excetuando a SAM, que, como dito anteriormente, não foi implantada.

A empresa foi bem-sucedida nessa empreitada e obteve a certificação após pouco menos de um ano do início das atividades relacionadas ao CMMI, assim como iniciou a construção de um arcabouço de boas práticas que agregam valor aos seus processos e produtos.

3.4.1 Práticas do CMMI nível 2

Consultando o WBS definido (Anexo I), e o repositório de evidências, foi possível verificar as seguintes práticas e artefatos, de acordo com cada área de processo, listados no Quadro 3.

Quadro 4 – Práticas e artefatos gerados na implantação do modelo CMMI

Área	Prática	Artefato
Planejamento do Projeto	Estabelecimento de um plano de melhorias	Plano de melhorias
	Definição do conceito de projeto no contexto da empresa	Guia de plano, monitoramento e controle do projeto
	Definição do que é “requisito” no contexto de desenvolvimento do projeto	
Gerenciamento de Requisitos	Definição do que é “requisito” no contexto de desenvolvimento do projeto	Guia de requisitos
	Explicitado no plano de projeto onde estão localizados e quais são os requisitos no projeto de desenvolvimento	
Área	Prática	Artefato

Gerenciamento de Requisitos	Estabelecido o papel do “fornecedor de requisitos”	Guia de requisitos
Controle e Monitoramento do Projeto	Definição de como seria realizado o acompanhamento do projeto	Guia de monitoramento e controle do projeto
Medição e Análise	Definição do objetivo de medição e indicadores básicos para acompanhamento do projeto de desenvolvimento de software	Indicadores de projeto
		Indicadores de processo
	Criação de indicadores organizacionais e de processo	Indicadores organizacionais
	Criação do Guia de Medição com os objetivos de medição e indicadores	Guia de medição
Garantia da Qualidade do Processo e do Produto	Estabelecimento do <i>checklist</i> para auditoria dos projetos	<i>Checklist</i> de auditoria
	Estabelecimento de um sistema de registro de não-conformidades encontradas durante a auditoria, de modo que possam ser vinculadas aos projetos de desenvolvimentos	Ajuste da ferramenta gerencial da empresa para o registro de não conformidades

Quadro 4 – Práticas e artefatos gerados na implantação do modelo CMMI - continuação

Área	Prática	Artefato
Garantia da Qualidade do Processo e do Produto	Definição do cronograma das auditorias que irão acontecer na empresa	Cronograma de auditorias
Gerenciamento de Configuração	Definição do que é “release” e o que é “versão” em cada equipe de desenvolvimento	Guia de Gerenciamento de Configuração
	Estabelecimento das <i>baselines</i> dos projetos de desenvolvimento	

3.5 Avaliação do processo de implantação do CMMI

3.5.1 Fatores críticos e dificuldades

Foi desenvolvido um questionário com questões abertas para verificar as seguintes dimensões: o método de implantação do modelo (tópico 3.3), fatores críticos, dificuldades e boas práticas. Esse questionário foi respondido por dois líderes de processo (um da área da Qualidade do produto, e outro da área do Suporte), ambos participantes do processo de implantação da certificação. O questionário pode ser consultado no Apêndice 1.

Três itens foram considerados como fatores críticos para a implantação do modelo CMMI na empresa: o apoio constante da alta gerência, o envolvimento dos colaboradores, que marcavam reuniões semanais para discutir sobre atividades que envolviam a implantação e organizavam grupos de trabalho para definir e executar práticas, e o apoio da consultoria contratada.

As dificuldades apontadas dizem respeito à cultura organizacional, como resistência às mudanças e aceitação das novas práticas; a adequação da ferramenta de gestão da empresa; e o prazo curto entre a institucionalização das novas práticas e a avaliação final. No decorrer da pesquisa de campo, diversas boas práticas foram observadas, sendo que a maioria das citadas se refere à melhoria de gerenciamento dos projetos, tais como: melhor organização e controle

de atividades; indicadores mais confiáveis, devido às novas práticas organizacionais; maior controle de riscos do projeto; uma gerência de configuração por projeto. Foi citada uma boa prática observada no processo de desenvolvimento: a revisão do código fonte, realizada por um colaborador diferente do que desenvolveu o código, permitindo uma maior detecção de não conformidades.

3.5.2 Nível de desempenho dos diferenciais competitivos

Confeccionou-se um questionário para verificar o nível de desempenho de diferenciais competitivos percebidos pela empresa com a obtenção da certificação CMMI nível 2. O questionário pode ser consultado no Apêndice 2. As respostas podem ser conferidas no Quadro 5.

Quadro 5 – Respostas ao questionário 2

Diferencial	Desempenho		
	Negativo	Neutro	Positivo
Melhoria dos processos			X
Qualidade do processo			X
Qualidade do produto			X
Produtividade		X	
Aumento das receitas	X		
Novos clientes		X	

Entre os diferenciais competitivos levantados, analisando as respostas dos entrevistados pode-se averiguar que metade teve nível de desempenho considerado positivo: os itens “Melhoria dos processos”, “Qualidade do processo” e “Qualidade do produto”. Além disso, pode-se constatar que apenas um diferencial competitivo, “Aumento das receitas”, obteve um nível de desempenho negativo. Como já descrito, custos são considerados como uma das dificuldades para as empresas de software na adoção do modelo CMMI, logo, este item comprova um dos fatores críticos citados no tópico 3.5.1: o apoio da alta gerência, que deve ter uma visão de

médio a longo prazo no que diz respeito a possível elevação das receitas com a implantação do modelo, consequência da otimização dos processos, aumento de produtividade e conquista de novos clientes. Inclusive dois destes fatores anteriormente citados estão relacionados, cada um, diretamente com os dois diferenciais competitivos que tiveram desempenho médio: “Produtividade” e “Novos clientes”.

3.5.3 Práticas executadas (propostas pelo Guia CMMI-dev para o nível 2)

Elaborou-se um questionário para verificar, das atividades propostas pelo Guia CMMI-dev para a certificação do nível 2, quais foram de fato executadas, qual o nível de dificuldade e quais foram adotadas como “boas práticas”. Este questionário foi preenchido por um gerente de projeto com o auxílio de um gerente de configuração, ambos colaboradores que participaram ativamente da certificação do CMMI nível 2 na empresa. Visou-se verificar quais práticas foram implantadas, qual o nível de dificuldade de implantação (1 – muito fácil, 2 – fácil, 3 – médio, 4 – difícil, 5 – muito difícil) e quais são consideradas “boas práticas” (práticas que perceptivelmente agregam valor ao processo/produto e/ou trouxeram benefícios relevantes em sua utilização). As respostas podem ser consultadas no Apêndice 3. Considerou-se fundamental analisar os resultados também por áreas de processo.

Práticas propostas x Práticas executadas

Analisando os resultados do Apêndice 3, pode-se constatar que, das 56 práticas propostas pelo Guia CMMI-dev (tópico 2.3.1, Quadro 2) para o nível 2:

- i) 46 foram executadas
- ii) Das 46 práticas executadas, 41 são consideradas “boas práticas”

Todas as áreas de processo teve grande parte de suas práticas executadas, exceto a área de processo SAM (Gerenciamento e Acordo com Fornecedores), que não teve práticas executadas, como descrito anteriormente (tópico 3.3). A Figura 7 demonstra o gráfico da quantidade de práticas executadas por área de processo.

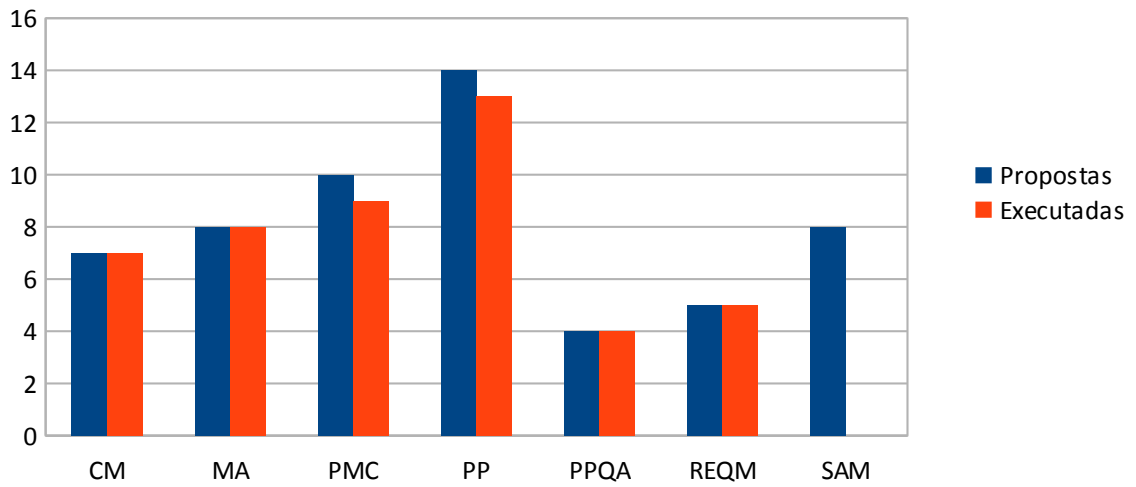


Figura 7 – Práticas executadas por área de processo

A Figura 8 demonstra o comparativo entre as práticas executadas e as consideradas “boas práticas”, por área de processo.

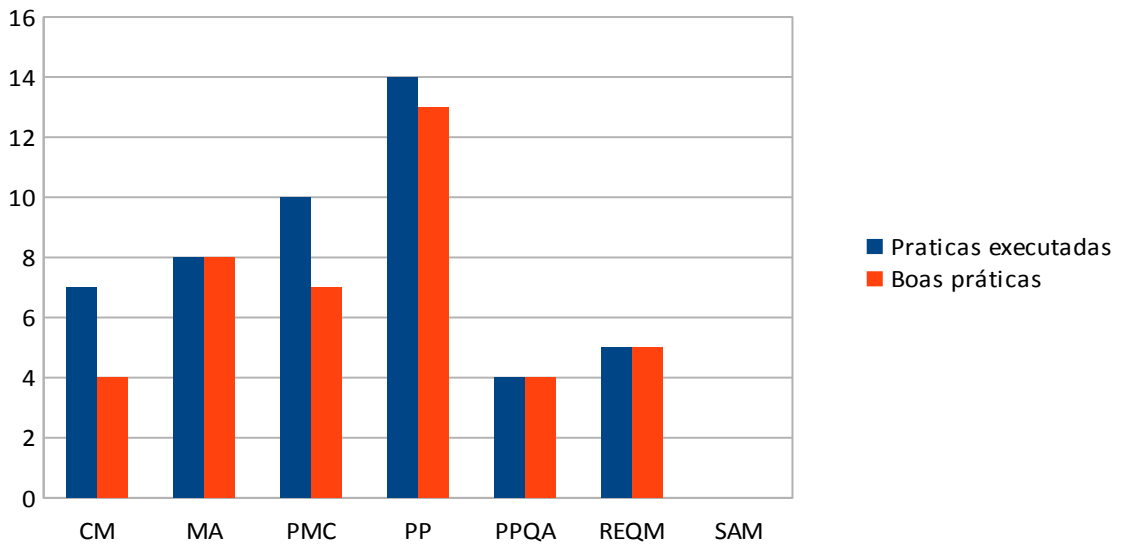


Figura 8 – Boas práticas por área de processo

Nível de dificuldade de implantação

O resultado referente ao nível de dificuldade na implantação das práticas propostas pelo Guia CMMI pode ser observado na Figura 9. Verifica-se que 35% das práticas foram consideradas de “muito fácil” e “fácil” implantação, sendo que 35% práticas foram consideradas como de dificuldade média, e 30% foram consideradas entre “difícil” e “muito difícil”.

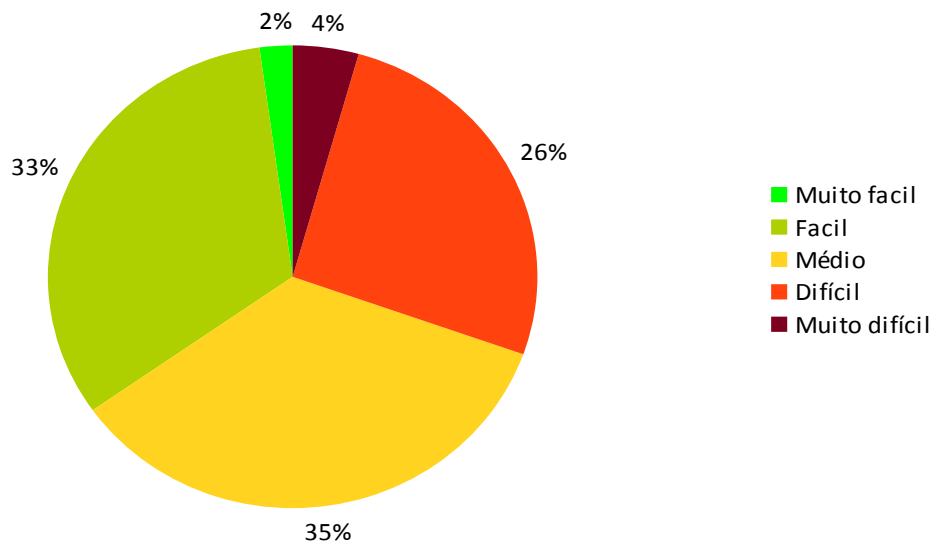


Figura 9 – Nível de dificuldade na implantação das práticas propostas

O nível de dificuldade de implantação das práticas por área de processo pode ser observado na Figura 9. Verifica-se que a área de processo MA (Medição e Análise) teve a maioria de suas práticas consideradas de difícil implantação (Figura 10), contudo, houve um reflexo positivo, pois todas são consideradas boas práticas (Figura 8), e umas das boas práticas citadas no questionário 1 refere-se a essa área (“Indicadores mais confiáveis, devido às novas práticas organizacionais”, tópico 3.5.1). Já a área de processo PP (Planejamento do Projeto) foi a que proporcionalmente teve mais práticas consideradas de fácil implantação (8 práticas, de 14 executadas). A empresa possuir diretrizes para gestão de projetos anteriores à implantação e uma ferramental gerencial aderente ao CMMI nível 2 nesse quesito pode ter sido fatores determinantes para o nível de dificuldade observado nesta área de processo.

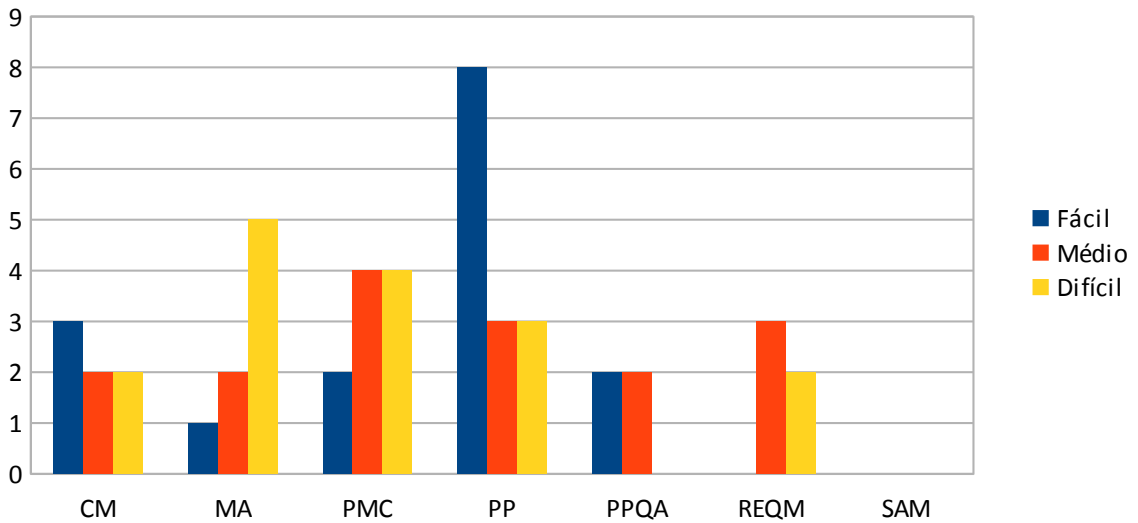


Figura 10 – Nível de dificuldade na implantação por área de processo

3.5.4 Análise de Resultados e Lições Aprendidas

Verificou-se que a alta direção da empresa possui uma visão de longo prazo, que entende que a adoção de uma abordagem para melhoria de processos não deve ser realizada em prol de resultados financeiros imediatos. Isso pode ser constatado pelo fator crítico “apoio da alta gerência”, citado em 3.5.1, que é verificado como plausível ao se averiguar que não foi notado um aumento de receitas devido à adoção ao modelo, nem o incremento da produtividade e conquistas de novos clientes.

O envolvimento dos colaboradores, igualmente mencionado como fator crítico, também pode ser comprovado pela eficiência da execução das práticas (46 práticas, das 56 propostas pelo guia CMMI-dev), mesmo que a maior parte delas tenha sido considerada de “médio” a “difícil” implantação (65% do total de práticas). O sucesso da implantação das práticas também se deve a outro fator crítico relatado, o apoio da consultoria contratada, que auxiliou constantemente no processo de adequação das atividades e processos ao modelo CMMI. Igualmente, pode-se afirmar que, pelo nível de desempenho percebido nos itens de “Melhoria dos processos”, “Qualidade do processo” e “Qualidade do produto”, a adoção do modelo CMMI foi positiva para a empresa.

Das práticas executadas por área de processo, pode se constatar que a mais crítica foi a área de “Medição e Análise”, pelo nível de dificuldade na implantação observado. Além disso, verificou-se que a área de processo “Planejamento do Projeto” foi a que teve mais práticas consideradas de fácil execução. Não foram encontrados estudos de caso que analisam especificamente a implantação destas áreas de processo em empresas de software, porém sabe-se que a medição de software ainda é uma prática incomum, enquanto a prática de planejamento de projeto já é tida como de fundamental aplicação em empresas de software (SOMMERVILLE, 2004,).

Pode-se destacar como lições aprendidas: a consciência da alta direção de que a obtenção de uma certificação de melhoria de processos trará resultados em médio ou à longo prazo; a importância da cooperação entre a consultoria especializada e os colaboradores participantes da implantação; a importância da execução das práticas propostas, mesmo que sejam de difícil execução, pois trazem resultados satisfatórios.

4. CONCLUSÃO

4.1 Considerações Finais

O objetivo do trabalho foi identificar os fatores críticos e boas práticas percebidas na obtenção da certificação CMMI nível 2 em uma empresa desenvolvedora de software. A identificação desses itens permitiu a observação de lições aprendidas, que podem ser utilizadas em projetos futuros semelhantes.

O levantamento bibliográfico permitiu um melhor entendimento sobre como o modelo de maturidade de processo CMMI se relaciona com o processo de desenvolvimento de software, e serviu de comparativo para os resultados obtidos na pesquisa de campo.

A pesquisa de campo foi fundamental para entender a motivação da empresa em obter a certificação CMMI, além de permitir a avaliação do método de implantação, a identificação dos fatores críticos e a relação destes com o nível de desempenho dos diferenciais competitivos e as práticas executadas propostas pelo Guia CMMI.

O questionário das práticas executadas propostas pelo Guia CMMI inclusive permitiu avaliar a efetividade deste modelo como um todo e por área de processo, por meio da avaliação dos resultados obtidos, relacionando as variáveis: práticas propostas, práticas executadas, nível de dificuldade de implantação e “boas práticas”.

Por fim, este trabalho proporcionou maior entendimento nos temas abordados, possibilitando o uso do conhecimento adquirido em projetos futuros relacionados ao CMMI e a melhoria de processos na empresa investigada.

4.2 Limitações da Pesquisa

Uma limitação encontrada foi que as alterações trazidas pelo CMMI nível 2 não foram completamente percebidas por todos os colaboradores da empresa, mais sim pelos participantes ativos do processo da implantação e pessoas de cargos gerenciais, logo não se

pode registrar a percepção da maioria. Isso se deve ao fato de que o nível 2 do CMMI atua mais na parte gerencial do processo de desenvolvimento de software.

Outro fator limitante observado remete ao fato que a empresa investigada já havia obtido a certificação CMMI nível 2 quando do início da pesquisa de campo. Pelo fato deste trabalho não ter acompanhado a implantação em tempo real, informações relevantes podem não ter sido mencionadas.

Outra limitação encontrada diz respeito ao fato de que existem poucos estudos efetivos sobre a implantação do modelo CMMI por área de processo, o que impossibilitou comparações sobre a efetividade de execução de práticas do Guia CMMI por esta dimensão. Este tipo de estudo possibilitaria, além de um referencial teórico para trabalhos futuros sobre este tema, uma fonte de consulta para pequenas e médias empresas que desejam obter a certificação CMMI trabalhando inicialmente com uma ou poucas áreas de processo por vez, e depois procurando atingir a maturidade nas outras, como no caso exposto por Borba *et al.* (2008).

4.3 Trabalhos futuros

Uma proposta para trabalho futuro na empresa estudada seria de criar um repositório único de lições aprendidas, que fosse divulgado e disponibilizado para acesso a todos os colaboradores. Conscientizando as pessoas a utilizarem as melhores práticas identificadas nos processos, possivelmente o retrabalho diminuirá, possibilitando o aumento de produtividade.

Uma segunda proposta é relacionada a expor à empresa investigada a possibilidade do uso de uma metodologia ágil de desenvolvimento de software. Esta se prova como uma tendência do mercado de TI, por possuir características que são consideradas fatores competitivos (atendimento de prazos) e promover melhoria dos processos (LEITÃO, 2010). Além disso, estudos comprovam que algumas dessas metodologias, como o SCRUM, são aderentes ao modelo CMMI (MARÇAL, 2009).

Outra proposta diz respeito a identificar pequenas empresas que tenham interesse em melhorar seus processos, e sugerir a implantação de uma ou algumas áreas de processo propostas pelo modelo CMMI, conduzindo um estudo mais detalhado a respeito desta implantação. Como citado anteriormente, uma das principais dificuldades das pequenas

empresas em relação à certificação CMMI diz respeito aos altos custos da implantação, logo, implantar poucas áreas de processo permitiriam que essas empresas desfrutassem de melhores práticas com custos menores, e, de certa maneira, as preparariam para a obtenção da certificação futuramente.

REFERÊNCIAS

BECK, K. et al. **Manifesto for Agile Software Development**. 2001. Disponível em: <<http://agilemanifesto.org>>. Acesso em 29 jul. 2012

BORBA, M. et al. **Engenharia de Requisitos e Estratégia Organizacional aliadas na implantação de CMMI em Pequenas Empresas**. 2008. Disponível em: <http://www.marquioni.com.br/manter_artigos/arquivos/aad128dd12c331ca4b19251c91e0c249.pdf>. Acesso em 29 set. 2012.

CAMPOS, Vicente Falconi. **TQC – Controle da Qualidade Total** (no estilo japonês). 8. ed. Belo Horizonte: Editora de Desenvolvimento Gerencial, 1999.

COSTA, A.F.B et al.. **Controle estatístico da qualidade**. São Paulo, Atlas, 2004.

DIAS, M. V. B. **Um novo enfoque para gerenciamento de projetos de software**. 2005. 202 f. Dissertação (Mestrado). Universidade de São Paulo. 2005.

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

KOSCIANSKI, André; SOARES, Michel dos Santos. **Qualidade de Software**. São Paulo, Novatec, 2007.

KUBOTA, Luis Claudio. Desafios para a Indústria de Software. **Publicação do Instituto de Pesquisa Econômica Aplicada**. Brasília, 2006. Disponível em: <http://www.redmercosur.org/iepcim/RED_MERCOSUR/biblioteca/ESTUDOS_BRASIL/BR_A_88.pdf>. Acesso em 01 mar. 2012

LEAL, G. C. L **Uma abordagem integrada de desenvolvimento e teste de software para equipes distribuídas**. 2010. 169 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Maringá. 2010.

LEITÃO, Michele de Vasconcelos. **Aplicação de Scrum em Ambiente de Desenvolvimento de Software Educativo**. 2010.72 f. Monografia (Bacharelado em Engenharia da

Computação) – Escola Politécnica de Pernambuco – Universidade de Pernambuco. 2010.
Disponível em:

< http://tccmix.googlecode.com/svn/trunk/TCC_MicheleVasconcelos_Correcao.pdf >. Acesso em 13 out. 2012.

MARÇAL, Ana Sofia Cysneiros. **SCRUMMI – Um processo de gestão ágil e aderente ao CMMI**. 2009, 205 f. Dissertação (mestrado) – Universidade de Fortaleza, 2009. Disponível em: <<http://www.cin.ufpe.br/~if720/downloads/SCRUMMI%20-%20AnaSofia.pdf>>. Acesso em: 31 mar. 2012

MARINHO, F. et al. **Uma Proposta de um Repositório de Padrões de Software**

Integrado ao RUP. 2003. Disponível em:

<http://www.cin.ufpe.br/~sugarloafplop/articles/spa/spa_RepositorioPadroesSoftwareRUP.pdf>. Acesso em 01 jun.2012

MEZZENA, Bruno; ZWICKER Ronaldo. **Benefícios e dificuldades do modelo CMM de melhoria do processo de software**. 2007. Disponível em:

< <http://www.revistasusp.sibi.usp.br/pdf/rege/v14n3/v14n3a7.pdf>>. Acesso em 28 set. 2012.

MONTONI, Mariano Angel. **Uma investigação sobre os fatores críticos de sucesso em iniciativas de melhoria de processos de software**. 2010. 387 f. Tese (Doutorado). Universidade Federal do Rio de Janeiro. 2010.

MORESI, Eduardo. **Metodologia de Pesquisa**. 2003. Disponível em: <<http://www.inf.ufes.br/~falbo/files/MetodologiaPesquisa-Moresi2003.pdf>>. Acesso em 14 out. 2012.

PALADINI, Edson Pacheco. **Gestão da qualidade: teoria e prática**. São Paulo, Atlas, 2004.

PRESSMAN, Roger S. **Engenharia de Software**. 3 ed. São Paulo. Pearson Education do Brasil. 2005.

SANTOS, Gilmar Souza; CAMPOS, Fernando Celso de. **Vantagem Competitiva em Certificações de Produção de Software e Gestão de Serviços de TI: Lições das Empresas de TI Indianas**. 2008. Disponível em:

< http://www.abepro.org.br/biblioteca/enegep2008_TN_STO_070_499_11229.pdf >. Acesso em 19 ago. 2012.

SANTANA, C. et al. **Mapeamento do modelo de Melhoria do Processo de Software Brasileiro (MPS.Br) para empresas que utilizam Extreme Programming (XP) como metodologia de desenvolvimento**. 2006. Disponível em:

< <http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2006/009.pdf> >. Acesso em 27 mai. 2012.

SCHWABER, Ken. **Agile Project Management with Scrum**. Microsoft, 2009.

(SEI) Software Engineering Institute. **CMMI – Dev, v.1.2**. Guia de desenvolvimento. 2008.

SOARES, M. S. **Metodologias Ágeis Extreme Programming e Scrum Para o Desenvolvimento de Software**. 2004.

Disponível em: < <http://revistas.facecla.com.br/index.php/reinfo/article/download/146/38> >. Acesso em 01 jun. 2012

SOARES, F.S. F. et al. **Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de Software**. 2007. Disponível em:

<http://www.sistemaengenharia.com.br/admIN/upload/obras_arquivos/ArquivoPDF_9697.pdf>. Acesso em 01 jun. 2012.

SOARES, Michel do Santos. **Comparação Entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**. 2004. Disponível em:

<http://xpsproject.googlecode.com/svn!/svn/bc/6/trunk/outros/Met._Ageis.pdf>. Acesso em 23 set. 2012

SOFTEX. **MPS.Br – Melhoria de Processo do Software Brasileiro**. Guia geral. 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 6 ed. São Paulo, Addison Wesley, 2004.

VASCO, C. G. et al. **Comparação entre as metodologias RUP e XP**. 2005. Disponível em: <http://edilms.eti.br/uploads/file/bd/RUPvsXP_draft.pdf > . Acesso em 01 jun. 2012.

WEBER, K. et al. **Melhoria de Processo do Software Brasileiro (MPS.Br): um programa mobilizador**. 2006. Disponível em: < http://sites.google.com/site/jcpantunes/es_paper4.pdf >. Acesso em 27 mai. 2012.

WEBER, K. et al. **MPS.BR - Melhoria de Processo do Software Brasileiro: resultados alcançados e lições aprendidas**. 2008. Disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments/_mpsbr/Artigo_CLEI_2008_vFina111.pdf > . Acesso em 01 jun. 2012

ZANATTA, Alexandre L.; VILAIN, Patricia. **Uma análise do método ágil Scrum conforme abordagem nas áreas de processo Gerenciamento e Desenvolvimento de Requisitos do CMMI**. 2006. Disponível em: <http://wer.inf.pucrio.br/WERpapers/pdf_counter.lua?wer=WER05&file_name=alexandre_zanatta.pdf>. Acesso em 01 mar. 2012.

APÊNDICE 1 – Roteiro para Entrevista

- Na sua opinião, quais foram os principais fatores que fizeram com que a empresa aderisse ao modelo CMMI?
- Como você caracterizaria o processo de desenvolvimento antes da certificação CMMI nível 2?
- Qual foi a estratégia adotada para iniciar a implantação do modelo? (definição de times de trabalho? Plano de melhoria?)
- Como foram escolhidos os colaboradores que atuaram ativamente neste projeto? Houve contratações para este fim?
- Quais foram as atividades propostas pela consultoria para adequar os processos da empresa ao modelo CMMI?
- Quais foram os problemas encontrados na implantação do modelo?
- Quais foram os fatores críticos de sucesso para a certificação?
- Quais foram as boas práticas adotadas a partir da implantação do modelo?

APÊNDICE 2 – Questionário sobre Nível de Desempenho dos Diferenciais Competitivos

Questionário

Sobre os diferenciais competitivos trazidos pela certificação CMMI, como você classificaria os seguintes, em relação a sua influencia, após a certificação, para a empresa?

Certificação reconhecida internacionalmente

1. Não influenciou totalmente
2. Não influenciou parcialmente
3. Indiferente
4. Influenciou parcialmente
5. Influenciou totalmente

Melhoria dos processos

1. Não influenciou totalmente
2. Não influenciou parcialmente
3. Indiferente
4. Influenciou parcialmente
5. Influenciou totalmente

Qualidade do processo

1. Não influenciou totalmente
2. Não influenciou parcialmente
3. Indiferente
4. Influenciou parcialmente
5. Influenciou totalmente

Qualidade do produto

1. Não influenciou totalmente
2. Não influenciou parcialmente
3. Indiferente
4. Influenciou parcialmente
5. Influenciou totalmente

Produtividade

1. Não influenciou totalmente
2. Não influenciou parcialmente
3. Indiferente
4. Influenciou parcialmente
5. Influenciou totalmente

Retorno sobre investimento

1. Não influenciou totalmente
2. Não influenciou parcialmente
3. Indiferente
4. Influenciou parcialmente
5. Influenciou totalmente

APENDICE 3 – Práticas do Guia CMMI-dev executadas

O exposto abaixo visa verificar quais práticas foram implantadas, qual o nível de dificuldade da implantação (1 – muito fácil, 2 – fácil, 3 – médio, 4 – difícil, 5 – muito difícil), e quais viraram “boas práticas” (práticas que perceptivelmente agregam valor ao processo/produto e/ou trouxeram benefícios relevantes com sua utilização)

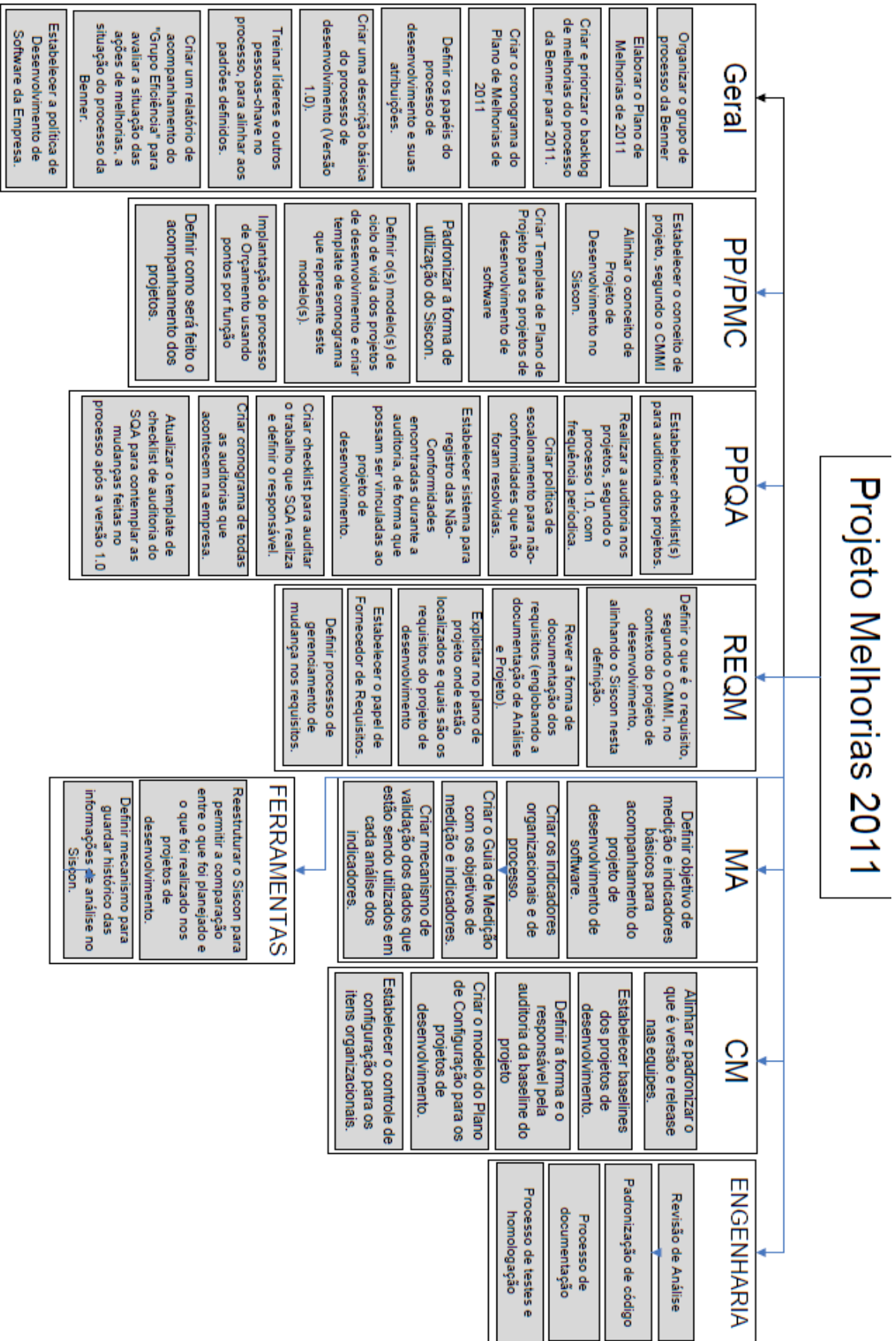
Práticas por área de processo		S	N	Dificul.	BP?	
Gestão de Configuração (CM)						
SG Estabelecer <i>Baselines</i>	1	SP 1.1 Identificar Itens de Configuração	X		4	N
		SP 1.2 Estabelecer um Sistema de Gestão de Configuração	X		3	N
		SP 1.3 Criar ou Liberar <i>Baselines</i>	X		2	S
SG Acompanhar e Controlar Mudanças	2	SP 2.1 Acompanhar Solicitações de Mudança	X		4	S
		SP 2.2 Controlar Itens de Configuração	X		3	N
SG Estabelecer Integridade	3	SP 3.1 Estabelecer Registros de Gestão de Configuração	X		2	S
		SP 3.2 Executar Auditorias de Configuração	X		2	S
Medição e Análise (MA)						
SG 1 Alinhar Atividades de Medição e Análise		SP 1.1 Estabelecer Objetivos de Medição	X		4	S
		SP 1.2 Especificar Medidas	X		4	S
		SP 1.3 Especificar Procedimentos de Coleta e Armazenamento de Dados	X		4	S
		SP 1.4 Especificar Procedimento de Análise	X		3	S
SG 2 Fornecer Resultados de Medição		SP 2.1 Coletar Dados Resultantes de Medição	X		5	S
		SP 2.2 Analisar Dados Resultantes de Medição	X		4	S

	SP 2.3 Armazenar Dados e Resultados	X		1	S
	SP 2.4 Comunicar Resultados	X		3	S
Monitoramento e Controle de Projeto (PMC)					
SG 1 Monitorar o Projeto em Relação ao Plano	SP 1.1 Monitorar os Parâmetros de Planejamento do Projeto	X		3	S
	SP 1.2 Monitorar Compromissos	X		4	S
	SP 1.3 Monitorar Riscos do Projeto	X		5	N
	SP 1.4 Monitorar a Gestão de Dados	X		4	N
	SP 1.5 Monitorar o Envolvimento das Partes Interessadas		X	4	N
	SP 1.6 Conduzir Revisões de Progresso	X		2	S
	SP 1.7 Conduzir Revisões de Marco	X		2	S
SG 2 Gerenciar Ações Corretivas até sua Conclusão	SP 2.1 Analisar Questões Críticas	X		3	S
	SP 2.2 Implementar Ações Corretivas	X		3	S
	SP 2.3 Gerenciar Ações Corretivas	X		3	S
Planejamento de Projeto (PP)					
SG 1 Estabelecer Estimativas	SP 1.1 Estimar o Escopo do Projeto	X		4	S
	SP 1.2 Estabelecer Estimativas para Atributos de Produtos de Trabalho e de Tarefas.	X		2	S
	SP 1.3 Definir Ciclo de Vida do Projeto	X		2	S
	SP 1.4 Determinar Estimativas de Esforço e Custo	X		2	S

SG 2 Elaborar um Plano de Projeto	SP 2.1 Estabelecer Orçamento e Cronograma	X		3	S
	SP 2.2 Identificar Riscos do Projeto	X		3	S
	SP 2.3 Planejar Gestão de Dados	X		2	S
	SP 2.4 Planejar Recursos do Projeto	X		2	S
	SP 2.5 Planejar Habilidades e Conhecimento Necessários	X		4	S
	SP 2.6 Planejar o Envolvimento das Partes Interessadas		X	5	N
	SP 2.7 Estabelecer o Plano do Projeto	X		3	S
SG 3 Obter Comprometimento com o Plano	SP 3.1 Revisar Planos que Afetam o Projeto	X		2	S
	SP 3.2 Conciliar Carga de Trabalho e Recursos	X		2	S
	SP 3.3 Obter Comprometimento com o Plano	X		2	S
Garantia da Qualidade de Processo e Produto (PPQA)					
SG 1 Avaliar Objetivamente Processos e Produtos de Trabalho	SP 1.1 Avaliar Objetivamente os Processos	X		3	S
	SP 1.2 Avaliar Objetivamente Produtos de Trabalho e Serviços	X		3	S
SG 2 Fornecer Visibilidade	SP 2.1 Comunicar e Assegurar a Solução de Não conformidades	X		2	S
	SP 2.2 Estabelecer Registros	X		2	S
Gestão de Requisitos (REQM)					
SG 1 Gerenciar Requisitos	SP 1.1 Obter Entendimento dos Requisitos	X		3	S
	SP 1.2 Obter Comprometimento com os Requisitos	X		4	S
	SP 1.3 Gerenciar Mudanças nos Requisitos	X		3	S
	SP 1.4 Manter Rastreabilidade Bidirecional dos Requisitos	X		4	S
	SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos	X		3	S

Gestão de Contrato com Fornecedores (SAM)					
SG 1	Estabelecer Contratos com Fornecedores	SP 1.1 Determinar Tipo de Aquisição	X	5	N
		SP 1.2 Selecionar Fornecedores	X	5	N
		SP 1.3 Estabelecer Contratos com Fornecedores	X	5	N
SG 2	Cumprir Contratos com Fornecedor	SP 2.1 Executar Contrato com Fornecedor	X	5	N
		SP 2.2 Monitorar Processos Selecionados do Fornecedor	X	5	N
		SP 2.3 Avaliar Produtos de Trabalho Selecionados do Fornecedor	X	5	N
		SP 2.4 Aceitar Produto Adquirido	X	5	N
		SP 2.5 Transferir Produtos	X	5	N

ANEXOS
Anexo I – WBS



Universidade Estadual de Maringá
Departamento de Engenharia de Produção
Av. Colombo 5790, Maringá-PR CEP 87020-900
Tel: (044) 3011-4196/3011-5833 Fax: (044) 3011-4196