UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

ALFREDO COLENCI NETO

Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS.Br

São Carlos
2008

ALFREDO COLENCI NETO

Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS.Br

Tese de doutorado apresentada à Escola de Engenharia de São Carlos, como parte dos requisitos para obtenção do título de doutor em Engenharia de Produção.

ORIENTADOR: Prof. Dr. Edson Walmir Cazarini

São Carlos
2008

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

C693p

Colenci Neto, Alfredo
Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS.Br / Alfredo Colenci Neto ; orientador Edson Walmir Cazarini. -- São Carlos, 2008.

Tese (Doutorado) - Programa de Pós-Graduação e Área de Concentração em Engenharia de Produção -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2008.

1. Desenvolvimento de produto. 2. Desenvolvimento de software. 3. Modelo de referência. I. Título.

FOLHA DE JULGAMENTO

Candidato: Tecnólogo **ALFREDO COLENCI NETO**

Tese defendida e julgada em 12/12/2008 perante a Comissão Julgadora:

Aprovado

Prof. Dr. **EDSON WALMIR CAZARINI (Orientador)**
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Associado **LUIZ CESAR RIBEIRO CARPINETTI**
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Dr. **DANIEL CAPALDO AMARAL**
(Escola de Engenharia de São Carlos/USP)

Aprovado.

Profª. Associada RENATA PONTIN DE MATTOS FORTES
(Instituto de Ciências Matemáticas e de Computação/USP)

APROVADO

Dr. **FREDY JOÃO VALENTE**
(COSS Consulting – São Carlos)

Prof. Associado **REGINALDO TEIXEIRA COELHO**
Coordenador do Programa de Pós-Graduação em
Engenharia de Produção

Prof. Associado **GERALDO ROBERTO MARTINS DA COSTA**
Presidente da Comissão da Pós-Graduação da EESC

Dedicado a minha esposa Ana Vivian, e minhas filhas Laura e Maria Luiza.

Agradecimentos

À Deus pela minha vida.

Agradecimento especial ao meu orientador, Prof. Dr. Edson Walmir Cazarini, pelos vários anos de convivência e ensinamentos transmitidos nesta jornada.

Aos meus pais Alfredo e Ana Maria por me darem a vida e me ensinarem a vive-la com dignidade, sem nunca ter medido esforços na minha educação. Vocês são os melhores !

Aos meus irmãos Pedro Luciano, Ana Teresa e meu cunhado Wilder pelos momentos de descontração e incentivo.

Aos professores e funcionários do Departamento de Engenharia de Produção da Escola de Engenharia de São Carlos em especial ao Prof. Dr. Daniel Capaldo do Amaral e José Luis Chiaretto.

A todos colaboradores da COSS Consulting em especial ao Dr. Fredy João Valente, Sr. Antonio Rigo e Sr. Luis Carlo Colella Ferro.

Aos amigos Janaina e Anderson pela colaboração nos momentos finais.

Resumo

COLENCI NETO. A. (2008). **Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS.Br.** Tese (doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.

Esta tese propõe um modelo de referência aplicável às pequenas empresas produtoras de softwares, para dar suporte ao seu processo de desenvolvimento de produtos (PDP). Utilizando uma revisão bibliográfica que permitiu a contextualização teórica, ao estado da arte, e tendo por base o modelo de qualidade MPS.Br, são apresentados e discutidos os conceitos de qualidade e de produtividade com abordagem voltada para as pequenas empresas. A seguir, procede-se a uma avaliação contextualizada, com base no método de avaliação MARES, em algumas empresas do setor, para se obter uma constatação da efetiva situação das mesmas quanto as suas práticas no desenvolvimento de sistemas informatizados. Da análise dos resultados e baseado no referencial teórico, objetiva-se, como contribuição ao melhor domínio do tema, disponibilizar-se um modelo de referência capaz de harmonizar o atingimento de qualidade assegurada e produtividade elevada com atuação rápida e eficaz, de modo a garantir competitividade às pequenas empresas produtoras de software no Brasil.

Palavras Chave: *Desenvolvimento de Produto, Desenvolvimento de Software, Modelo de Referência.*

Abstract

COLENCI NETO. A. (2008). **Proposal of a referencial model for software development with focus on MPS.Br.** Tese (doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.

This thesis proposes a product development process (PDP) reference model for software companies. The PDP reference model was developed based on the MPS.Br existing model and an extensive bibliography revision which allowed for a state of the art context analysis.

This work also presents the main quality and productivity concepts targeted at small software development companies. Following that path, a context evaluation based on the evaluation method MARES was applied to a group of small software companies in order to produce a present stage concerning their software development practices, per company. By analyzing and comparing the evaluation results and the theoretical material surveyed, a new PDP reference model was proposed aiming to ease the introduction of quality and productivity practices for small software development companies. It is expected that the adoption of the proposed reference model can help small companies leverage their competitive advantage in Brasil.

Keywords: *Product Development, Software Development, Referencial Model.*

Lista de Ilustrações

Figura 1. Seqüência das atividades da pesquisa. 24
Figura 2. Processos relacionados com desenvolvimento de produtos. ROZENFELD et al. (2006). 27
Figura 3. Modelo cascata. SOMERVILLE (2003). 33
Figura 4. Modelo espiral resumido de Boehm. Adaptado de Pressman (2005). 35
Figura 5. Modelo Incremental. PRESSMAN (2005). 36
Figura 6. Rational Unifield Process. KRUCHTEN (2003). 38
Figura 7. O processo Extreme Programming. PRESSMAN (2005). 44
Figura 8. Fluxo de desenvolvimento Scrum. Adaptado de COCHANGO (2006). 47
Figura 9. Níveis de maturidade do CMMi. CMMI-DEV (2006). 49
Figura 10. Diagrama da relação entre processos. Adaptado de MOPROSOFT (2005). 56
Figura 11. Estrutura do MR-MPS. MPS (2006). 57
Figura 12. Classes de Processos do MPS.Br. MPS (2006). 61
Figura 13: Níveis de Maturidade: CMMI x MR-MPS. Adaptado de PINTO (2006). 63
Figura 14. O modelo MPS.Br. WEBER (2005). 77
Figura 15. Etapas do processo de avaliação. 82
Figura 16. Modelo de referência. ROZENFELD et al, (2006). 94
Figura 17. Visão geral do modelo de referência para desenvolvimento de software. 108
Figura 18. Fluxo de atividades da gerência de requisitos 117
Figura 19. Fluxo de atividades para gerência de projetos. 125
Figura 20. Fluxo de trabalho na implantação do processo. 139
Figura 21. Tela do Share-Point. 144
Figura 22. Arquitetura da solução WELCOSS. 146
Figura 23. Fluxo de kanbans na empresa. 148
Figura 24. Arquitetura do sistema DVR. 149

Lista de Tabelas

Tabela 1 - Áreas de Processo do CMMi. 51

Tabela 2 - Organizações com certificação CMM/CMMi no Brasil 53

Tabela 3 - Classificação das empresas segundo o porte 66

Tabela 4 - Resultado da aderência das atividades do modelo de referência ao MPS.Br – Gerência de Projeto. 104

Tabela 5 - Resultado da aderência das atividades do modelo de referência ao modelo MPS.Br – Gerência de Requisitos 106

Lista de Quadros

Quadro 1. Representação da classificação metodológica da pesquisa..... 22
Quadro 2. Características de cada nível do CMMi..... 50
Quadro 3. Níveis de maturidade do MPS.Br..... 58
Quadro 4. Quadro resumo da empresa A. 84
Quadro 5. Quadro resumo da empresa B. 85
Quadro 6. Quadro resumo da empresa C. 86
Quadro 7. Quadro resumo da empresa D. 87
Quadro 8. Quadro resumo da empresa E. 88
Quadro 9: Resumo dos projetos analisados..... 150

Lista de Abreviaturas e Siglas

ABES	Associação Brasileira de Empresas de Software
BID	Banco Interamericano de Desenvolvimento Econômico e Social
BNDES	Banco Nacional de Desenvolvimento
CMM	Capability Maturity Model
CMMi	Capability Maturity Model Integrated
CRM	Customer Relationship Management
EAP	Estrutura Analítica de Processos
ERP	Enterprise Resource Planning
FINEP	Financiadora de Estudos e Projetos
GCQ	Grupo de Controle da Qualidade
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IPT	Instituto de Pesquisas Tecnológicas do Estado de SP
MA-MPS	Modelo de Avaliação - Melhoria do Processo de Software Brasileiro
MCT	Ministério da Ciência e Tecnologia
MSF	Microsoft Solution Framework
MR-MPS	Modelo de Referência - Melhoria do Processo de Software Brasileiro
MPS.Br	Melhoria do Processo de Software Brasileiro
O.O.	Orientação a Objetos
PDS	Processo de Desenvolvimento de Software
PDP	Processo de Desenvolvimento de Produto
PMBOK	Project Manager Body of Knowledgement
PMI	Project Manager Institute
RFID	Radio Frequency Identification
RUP	Rational Unified Process
SEI	Software Engineering Institute
SPICE	Software Process Improvement and Capability dEtermination
SVN	Subversion
TI	Tecnologia da Informação
UNAM	Universidad Nacional Autónoma del México
UML	Unified Modeling Language
XP	eXtreme Programming

Sumário

RESUMO 1

ABSTRACT 1

LISTA DE ILUSTRAÇÕES 1

LISTA DE TABELAS 1

LISTA DE QUADROS 1

LISTA DE ABREVIATURAS E SIGLAS 1

1. INTRODUÇÃO 13

1.1 Objetivos 17

1.2 Hipótese 18

1.3 Justificativa 19

1.4 A Escolha do Método de Pesquisa 20

1.4.1 Aspectos Conceituais da Pesquisa 20

1.4.2 Atividades da Pesquisa 23

1.5 Estrutura do Trabalho 25

2. REVISÃO TEÓRICA 26

2.1 Desenvolvimento de Produto 26

2.2 Processo de desenvolvimento de produto 28

2.3 Processo de desenvolvimento de software 29

2.4 Modelos de Processo de software 32
2.4.1 Modelo Cascata 33
2.4.2 Modelo Espiral 34
2.4.3 Modelo Incremental 35
2.4.4 Rational Unified Process (RUP) 37

2.5 Metodologias ágeis 39
2.5.1 eXtreme Programming 41
2.5.2 SCRUM 45

2.6 Qualidade no Processo de Software 47

2.6.1 CMMi 48
2.6.2 ISO/IEC 12207 54
2.6.3 MoProSoft 55
2.6.4 MPS.Br 56
2.6.5 Comparativo CMMi e MPS.Br 61

3. CARACTERÍSTICAS DA INDÚSTRIA DE SOFTWARE 64

3.1 Setor de Software no Brasil 64
3.1.1 Análise das Condições de Atuação 67

3.2 Avaliação do processo de desenvolvimento de software 71
3.2.1 Procedimentos de avaliação 73
3.2.2 Considerações Gerais 74
3.2.3 Métodos de Avaliação 74

3.3 Procedimento de avaliação de processo 80
3.3.1 Processo de avaliação 81

3.4 Resultados da pesquisa 83
3.4.1 Sobre as empresas 83
3.4.2 Análise 88

4. PROPOSTA DE UM MODELO DE REFERÊNCIA 91

4.1 Modelos de referência 91

4.2 O modelo de desenvolvimento de produto 93
4.2.1 Macro-Fases 94

4.3 Detalhamento do nível G do MPS.Br 96
4.3.1 Gerência de Projetos 96
4.3.2 Gerência de Requisitos 101

4.4 Pontos comuns entre MPS.br e o modelo de referência 103

4.5 Proposta do Modelo 106
4.5.1 Processo Padrão de Gerência de Requisitos 114
4.5.2 Processo Padrão de Gerência de Projetos 123

5. APLICAÇÃO DO MODELO: UMA PESQUISA-AÇÃO 136

5.1 A empresa 136

5.2 Estruturação da metodologia de trabalho na empresa 138
5.2.1 Implantação do Processo de Gerência de Requisitos 139
5.2.2 Implantação do Processo de Gerência de Projetos 141

5.5 Disponibilização do modelo de processo na empresa 143

5.6 O modelo em aplicação: a mensuração dos resultados **145**

5.7 Mensuração das melhorias **150**
5.7.1 Resultados 151

6. CONCLUSÃO **161**

REFERÊNCIAS BIBLIOGRÁFICAS **165**

ANEXO I **170**

ANEXO II **178**

1. INTRODUÇÃO

A Tecnologia da Informação (TI) representa um conjunto de conhecimentos que permite o tratamento integrado e sistêmico das informações, proporcionando o rápido crescimento do conhecimento em todas as atividades humanas. Drucker (1968) afirma que através da TI, o conhecimento se comporta como no princípio dos vasos comunicantes, da Física, promovendo desenvolvimentos em direção a seqüentes crescimentos em todas as áreas e avançado gradualmente. O software, entendido como um sistema integrador e sistematizador das informações, representa o principal meio para a gestão do conhecimento e para sua operacionalização. Deste modo, a atividade de produção de software ganha importância fundamental para o sucesso da gestão das tecnologias de informação e tem merecido a atenção de inúmeros pesquisadores em todo o mundo.

Examinando-se o formalismo processual no desenvolvimento de software e sua constatação com a prática vigente nas pequenas empresas, sediadas no mercado brasileiro, percebe-se claramente um distanciamento das práticas adotadas frente aos procedimentos recomendados. De fato, podem ser constatadas certas dificuldades advindas de falhas desse processo, que, via de regra, peca pela ação não disciplinada capaz de bem sistematizar seu desenvolvimento e aplicação, o que lhe acarreta os efeitos da não qualidade do sistema, nocivos aos usuários e a própria organização.

As pesquisas dos últimos anos em projeto de produto e da própria engenharia de software mostraram que a qualidade de um produto final é conseqüência de um processo de desenvolvimento bem disciplinado, capaz de levar em conta não apenas requisitos previamente estruturados mas também aqueles a serem percebidos como exigências

correlacionadas às reais condições de demanda, ou seja, os semi-estruturados e os não estruturados.

Historicamente, a produção de software chega a se mostrar um procedimento caótico. Isso de deve principalmente aos poucos anos de experiência dessa área que possui não mais que seis décadas. Desta forma, muitos métodos, técnicas e ferramentas estão ainda sendo criados e implantados, no sentido de se melhorar a qualidade do produto.

Na década de 90 surgiram modelos de qualidade cujo propósito era ser um guia para que empresas pudessem amadurecer o próprio processo de forma gradual. Modelos como *Capability Maturity Model Integrated* (CMMi) e a própria norma ISO são referências nesse sentido.

O CMMi é um modelo de maturidade para o desenvolvimento e manutenção de software e dos serviços que abrangem o ciclo de vida do produto, desde sua concepção, entrega e manutenção. O modelo enfatiza às disciplinas de engenharia de sistemas e também à engenharia de software e à integração necessária para se construir e manter os produtos de forma abrangente. O CMMi oferece um conjunto de boas práticas agrupadas de acordo com áreas de atividades correlatas e níveis de maturidade. Os níveis correspondem a etapas progressivas de eficácia gerencial e se apresentam como um caminho evolucionário para qualquer organização que pretenda melhorar seus processos de desenvolvimento e manutenção de software. CHRISSIS; KONRAD; SHRUM, (2003).

Desde que as empresas começaram a orientar a adaptação dos processos de desenvolvimento a esses modelos e normas, muitos relatos de insucesso foram apresentados em todo o mundo, principalmente no segmento de pequenas empresas.

O desafio das empresas atuais é atingir um grau de qualidade em seu processo sem que se perca produtividade. Esse fato é marcado principalmente porque essas empresas dependem de níveis satisfatórios de produtividade para concorrer e prosperar neste mercado competitivo. Ocorre que, em algumas empresas, a busca pela produtividade acaba se tornando um empecilho na garantia da qualidade, em conseqüência do mal uso dos recursos na medida e no tempo adequado.

Com foco voltado para o mercado nacional, no qual mais de 70% das empresas são pequenas segundo a ABES (2007) e a SOFTEX (2005), foi criado em 2003 o modelo brasileiro de qualidade de software chamado de MPS.Br – Melhoria do Processo de Software Brasileiro que conta com o apoio do Ministério da Ciência e Tecnologia (MCT) e de centros de pesquisa como a SOFTEX e a RIOSOFT. Este modelo foi baseado nos modelos CMM, ISO 12207 e ISO 15504 para que empresas nacionais pudessem amadurecer seu processo com impacto reduzido, ou seja produtivamente.

O programa apresenta um modelo de referência para melhoria (MR-MPS) e também para avaliação de processo (MA-MPS). Atualmente existem 104 empresas que já se certificaram em algum dos 7 níveis de maturidade, que são uma combinação entre processos e capacidade de processos. MPS (2007).

Mesmo com esse modelo bem definido, ainda se encontram no mercado empresas que não possuem formas de alcançar as metas propostas para garantir qualidade e isto se dá devido à falta de uma estrutura já definida dos processos, ferramentas, documentos, papéis que auxiliariam no curso correto a ser seguido, ou seja, carecem ainda de um tratamento sistêmico.

Sabe-se que um processo de desenvolvimento compreende atividades associadas seguidas de forma ordenada que tenha início e fim bem definidos, que ao término tenha-se um produto que cumpra com seus requisitos pré-estabelecidos, ABNT (2000).

No desenvolvimento de software, pode-se definir um procedimento de forma similar ao de qualquer outro produto. Um processo de software maduro e consolidado é de extrema importância pois garante que o que foi desenvolvido permite obter, no mínimo, realmente o que foi planejado. Muito esforço já foi despendido, ao longo dos anos, na busca de metodologias de desenvolvimento que pudessem nortear equipes na construção de sistemas. Entre essas metodologias se destacam: o modelo clássico, espiral, incremental, prototipagem. Todos esses, entretanto, apresentam limitações e problemas que os afastam de situações reais. PFLEEGER (2004).

De forma mais atualizada e garantindo uma eficácia maior, surgiram metodologias chamadas ágeis, que deixam de lado os aspectos burocráticos do desenvolvimento para valorizar produtividade. Entre elas se destacam: Extreme Programming e SCRUM. O efeito sentido do sacrifício da parte burocrática do processo para o prestígio da agilidade de resposta é a perda da garantia da qualidade, tantas vezes percebida e reclamada pelos clientes.

Esta investigação propõe o desenvolvimento de metodologia própria e consolidada, capaz de eliminar o aparente antagonismo de objetivos que se estabelece ao se minimizar o protocolo e a burocracia e se prestigiar a redução de tempo de processamento do projeto. Assegurar qualidade e reduzir tempo são objetivos a serem perseguidos harmonicamente, se o processo de projeto se dá com o adequado tratamento sistêmico.

Nesse sentido, este trabalho apresenta uma pesquisa de campo com cinco empresas desenvolvedoras de software para levantamento da real situação em que as mesmas se encontram com relação ao processo de desenvolvimento, em confronto com a teoria já disponível.

A partir desses dados e confrontando com dados nacionais que mostram a evolução da indústria de software brasileira, pode-se ter uma real situação de como estas organizações

desenvolvem seus produtos e mediante dificuldades e problemas apresentados propõe-se um modelo de desenvolvimento otimizado para esse segmento, voltado principalmente para cumprir com os propósitos do modelo de qualidade MPS.Br.

Tem-se como base para a criação de um modelo adequado para desenvolvimento de software, o modelo de referência de desenvolvimento de produto criado em ação conjunta de grupos de três universidades brasileiras: NUMA (EESC-USP), GEPEQ (DP-USFCAR) e GEPP (UFSC). Segundo Rozenfeld et al (2006), essa ação, teve como resultado um modelo de referência que contém os conceitos e as melhores práticas em desenvolvimento de produtos (DP) com a missão de se tornar "referência" para a derivação de outros modelos, adequados a um setor ou tipo específico de produto. A justificativa para uso desse modelo é que o mesmo já foi utilizado em outras pesquisas com resultados satisfatórios.

Para se validar o modelo proposto, faz-se uma aplicação dos processos de gerência de projetos e de gerência de requisitos, em dois projetos distintos, em uma pequena empresa localizada na cidade de São Carlos – SP. Após a aplicação que levou 15 meses, propôs-se uma avaliação qualitativa junto aos membros participantes do projeto para mensuração dos resultados.

1.1 Objetivos

O objetivo principal deste trabalho é propor um modelo de referência para desenvolvimento de software, aplicável a pequenas empresas, de modo a permitir o alcance de elevados padrões de qualidade e produtividade na produção de software, em conformidade com o modelo MPS.Br .

Para atingir o objetivo principal, são propostos os seguintes objetivos específicos:

- Realizar uma investigação sobre os processos de desenvolvimento de software atuais, presentes na literatura e utilizados nas empresas, tanto grandes quanto pequenas;
- Realizar uma pesquisa bibliográfica sobre métodos de avaliação de processos de desenvolvimento de software;
- Realizar uma pesquisa de campo contextualizando a atual situação de cinco empresas que desenvolvem software e os procedimentos adotados atualmente quanto ao processo de desenvolvimento;
- Acompanhar e avaliar em uma empresa desenvolvedora de software a aplicação do modelo proposto em um projeto real;
- Analisar possíveis discrepâncias de procedimentos e propor uma metodologia própria capaz de garantir a competitividade.

1.2 Hipótese

Para atingir os objetivos propostos, a seguinte hipótese de pesquisa será examinada através de revisão bibliográfica e análise qualitativa:

Hipótese 1: A garantia da qualidade e o alcance de elevados padrões de produtividade serão atendidos efetivamente, com uso adequado do modelo proposto.

1.3 Justificativa

Conforme dados do ABES (2007), 94% das empresas desenvolvedoras de software no Brasil são pequenas, em um total aproximado de 1.894. Sabe-se que tais empresas enfrentam barreiras para amadurecer seu processo de desenvolvimento em decorrência das pressões do mercado e mesmo, da pouca integração de suas equipes, o que, se superados, lhes garantirá alcançar melhores resultados em qualidade, produtividade e tempo de atendimento, na qualidade de seus produtos. Não dominando perfeitamente o gerenciamento do processo, essas empresas passam a viver situações complexas, com perda de clientes e de rentabilidade chegando rapidamente ao insucesso.

Estendendo a abordagem ao mercado globalizado, pode-se perceber estar ocorrendo um intenso processo de terceirização no desenvolvimento de software, conhecido como *off shore*, onde empresas americanas e européias vivendo um estágio mais intensivo em tecnologia, buscam parceiros para desenvolverem sistemas capazes e com custos mais competitivos.

Neste cenário macroambiental, países como Índia e China têm tido vantagens principalmente por possuírem certificações de qualidade no processo de desenvolvimento. Cabe as empresas desenvolvedoras de software no Brasil, neste momento, criar meios para competir nesse mercado bastante promissor.

Uma das maneiras de se criar essa possibilidade no cenário brasileiro é justamente fortalecendo e melhorando a forma pela qual as empresas criam seus produtos. Existe, atualmente, um número considerável de processos para desenvolver sistemas computadorizados, com sérias restrições ao uso em pequenas empresas, conforme será apresentado neste estudo. Entre esses processos se destacam modelos tradicionais como espiral, incremental, cascata e mais modernamente as metodologias ágeis Extreme

Programming e Scrum como também o RUP (*Rational Unified Process*) que sofrem críticas quanto garantia da qualidade final dos produtos.

Quanto a qualidade do processo de desenvolvimento, tem-se modelos como a norma ISO 12207 e o CMMi, mundialmente conhecidos, mas que apresentam restrições a adoção por pequenas empresas. Surge como alternativa, para esse segmento grande de mercado, o modelo brasileiro chamado de MPS.Br. Todos esses modelos indicam o que deve ser cumprido em termos de metas para se garantir qualidade, porém fica na incumbência das organizações proverem meios para atingir os objetivos necessários. Na literatura existem poucos relatos de uma estrutura que possa ser utilizada como referência nas pequenas empresas.

Justifica-se então a elaboração de um modelo de referência, com o objetivo de ser justamente, um guia a pequenas empresas na melhoria do seu processo de desenvolvimento de software em conformidade com o modelo MPS.Br.

1.4 A Escolha do Método de Pesquisa

Nesta seção apresentam-se considerações sobre a metodologia da pesquisa empregadas no trabalho, sua classificação e as atividades seguidas, bem como a exposição de motivos que justifiquem a sua escolha.

1.4.1 Aspectos Conceituais da Pesquisa

A escolha da metodologia da pesquisa permitiu estabelecer uma revisão bibliográfica do contexto teórico a partir do qual procedeu-se a escolha de procedimento melhor aplicável ao caso em investigação. De início, estabeleceu-se uma classificação metodológica que, para

melhor compreensão, destacando-se quatro aspectos: natureza, objetivos, forma da abordagem do problema e procedimentos técnicos adotados.

- **Quanto à natureza:** Para Gil (2001) e Silva (2001), pode-se classificar uma pesquisa quanto a sua natureza entre básica e aplicada. Neste caso, em se tratando de uma pesquisa onde a hipótese é aplicada para se comprovar sua veracidade, este trabalho é caracterizado como uma pesquisa aplicada, pois gera conhecimento prático dirigido à solução de um problema específico, envolvendo verdades e interesses locais.
- **Forma de abordagem do problema:** Quanto à abordagem do problema que é caracterizado por considerar a idéia de que pequenas empresas não possuem um modelo de referência, adequado à sua realidade, para o desenvolvimento de software, esta pesquisa considera que existe uma relação entre o mundo real e o sujeito que não é traduzido em números e que segundo Silva (2001), reflete ao fato de que os dados e resultados devem ser, na sua maioria, indutivamente analisados, não traduzindo em números opiniões e informações para classificá-los e analisá-los, caracterizando a pesquisa como qualitativa. Sendo assim, não se faz uso de métodos e técnicas estatísticas.
- **Quanto aos objetivos:** Para Gil (2001) e Demo (1994) uma pesquisa pode ser classificada de três formas quanto aos seus objetivos: pesquisas exploratórias, pesquisas descritivas e pesquisas explicativas. Em se tratando de um estudo para familiaridade com o problema com vista a torná-lo explícito e construir hipóteses, este trabalho é melhor caracterizado como pesquisa exploratória, pois envolve levantamento bibliográfico; entrevistas e visitas com empresas que possuem o problema pesquisado.

- **Procedimentos técnicos adotados:** Para Gil (2001), divide-se a classificação quanto aos procedimentos técnicos adotados nos seguintes itens: pesquisa bibliográfica, pesquisa documental, pesquisa experimental, pesquisa ex-pos-facto, levantamento, estudo de caso, pesquisa-ação e pesquisa participante. Neste aspecto, esta pesquisa tem características comuns a mais de uma classificação. Por compreender um levantamento teórico do assunto ao estado da arte, pode ser classificada como pesquisa bibliográfica. Também é caracterizado como uma pesquisa-ação, pois o escopo da pesquisa foi aplicado e demonstrado em uma organização de forma colaborativa entre o pesquisador e seus colaboradores. A pesquisa-ação ou pesquisa participante se mostra bastante enriquecedora pois ultrapassa a simples obtenção de dados para, através das possibilidades de comunicação biunívoca entre entrevistador e entrevistado, colher opiniões e mesmo, esclarecer o entrevistado sobre suas condições de atuação, segundo Brandão (1985) e Thiollent (1988).

O quadro 1, apresenta uma classificação da pesquisa, de forma geral para uma melhor visualização, destacando, em negrito, aquelas que foram adotadas neste trabalho.

Quadro 1. Representação da classificação metodológica da pesquisa.

CLASSIFICAÇÃO	CARACTERÍSTICA			
Natureza	Básica	**Aplicada**		
Forma de abordagem do problema	**Qualitativa**	Quantitativa		
Objetivos	**Exploratória**	Descritiva	Explicativa	
Procedimentos técnicos adotados	**Levantamento Bibliográfico**	Estudo de Caso	Pesquisa Experimental	Pesquisa Participante
	Levantamento	Pesquisa Expos-acto	Pesquisa Documental	**Pesquisa-Ação**

1.4.2 Atividades da Pesquisa

Escolhido o procedimento metodológico a ser adotado, estabeleceu-se a seqüência de atividades a serem desenvolvidas na sua realização, conforme se apresenta na figura 1, a seguir.

Durante a revisão da literatura buscou-se estabelecer o contexto teórico, ao estado da arte, de modo a oferecer subsídios para a identificação dos temas que cuja relevância pudessem contribuir para o melhor entendimento do assunto.

Com a justificativa traçada foram propostos os problemas da pesquisa que serviram para definir a hipótese e objetivos apresentados na seção 1.1 deste trabalho.

Para a hipótese estabelecida, procedeu-se a uma pesquisa de campo, entre empresas representativas. Buscou-se entre as empresas desenvolvedoras de software aquelas que pudessem oferecer contribuição efetiva. A seguir foram realizados os procedimentos para o processo de avaliação composto das atividades de identificação das empresas pesquisadas, aplicação do método, análise dos resultados, conforme será apresentado no capítulo 4.

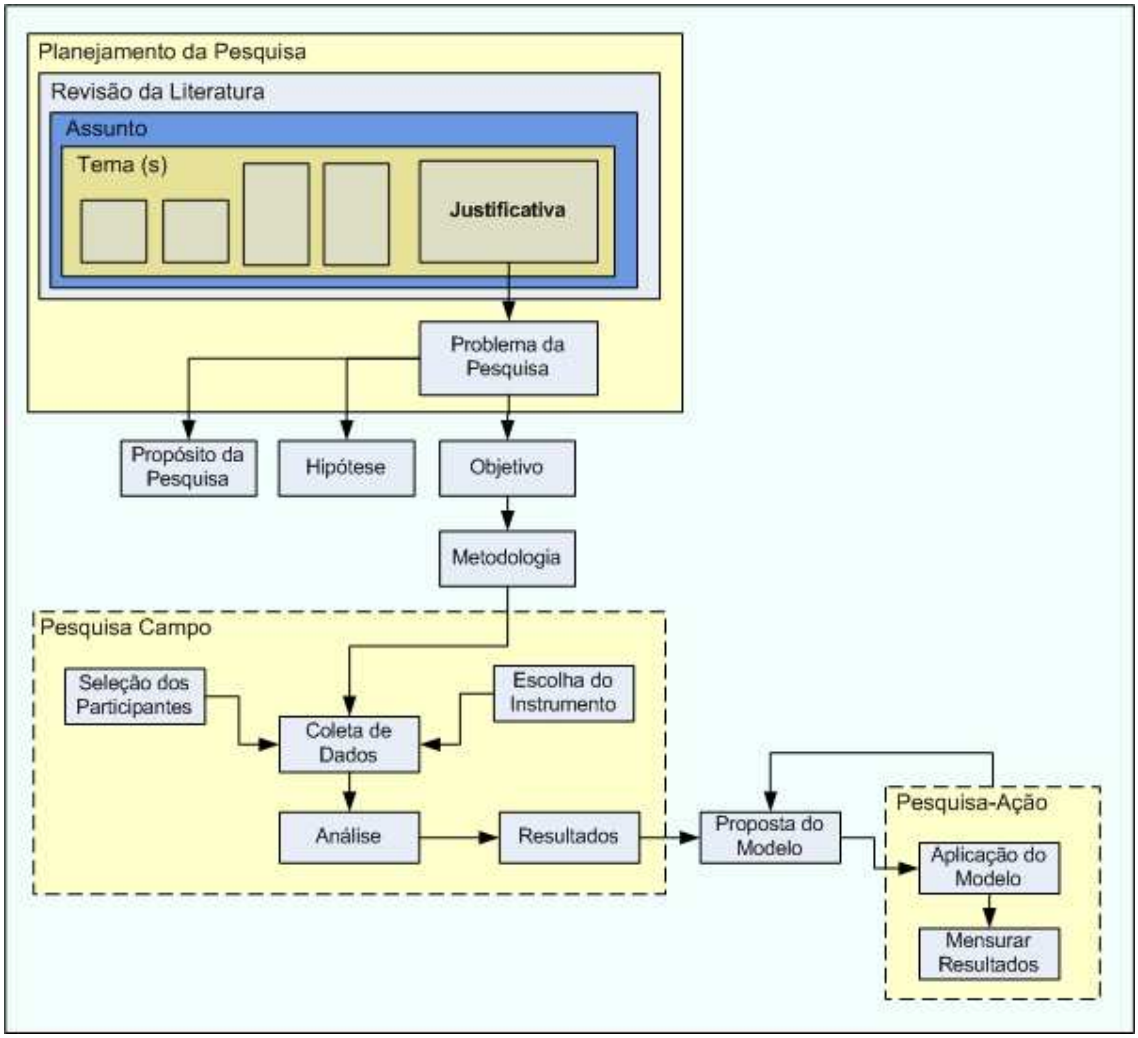

Figura 1. Seqüência das atividades da pesquisa.

Complementarmente, com a análise desses dados, pode-se confrontar os resultados com estudos do Ministério da Ciência e Tecnologia (MCT), para assim propor o modelo desejado e posteriormente aplicá-lo em projetos reais de uma empresa como forma de comprovação da hipótese e mensuração dos resultados.

1.5 Estrutura do Trabalho

O presente trabalho está dividido em seis capítulos, cujo conteúdo é descrito a seguir:

O Capítulo 1, que é este, faz a introdução do assunto, apresenta os objetivos, as justificativas, metodologia e hipóteses para a realização do trabalho, ou seja, apresenta como o trabalho está estruturado.

O Capítulo 2, através de revisão bibliográfica, discorre os conceitos de desenvolvimento de produto, desenvolvimento de software e qualidade.

O Capítulo 3 realiza uma avaliação baseada no método MARES com empresas de pequeno porte, para traçar a real situação destas quanto ao seu procedimento de desenvolvimento. Faz-se também um levantamento amplo acerca dos vários métodos de avaliação existentes.

O Capítulo 4, com base nos dados levantados na pesquisa de campo, apresenta a proposta do trabalho baseada no modelo de referência proposto por Rozenfeld et al (2006) para desenvolvimento de produto. Neste capítulo contextualizam-se também conceitos e definições acerca do que vem a ser um modelo de referência.

O Capítulo 5 demonstra os resultados da aplicação do modelo em uma empresa.

O Capítulo 6, finalmente, aponta as considerações finais do trabalho e discussão a respeito de possibilidade de novas investigações, a partir da pesquisa realizada.

2. REVISÃO TEÓRICA

Este capítulo aborda tópicos que se relacionam à temática de pesquisa e que sustentam os processo de investigação, são eles: desenvolvimento de produto, processo de desenvolvimento de produto, processo de desenvolvimento de software, metodologias para desenvolvimento de software e modelos de qualidade aplicados ao processo de desenvolvimento de software.

2.1 Desenvolvimento de Produto

Desenvolver produtos consiste em um conjunto de atividades por meio da qual busca-se, a partir da necessidade do mercado e das possibilidades tecnológicas chegar as especificações de projeto de um produto e sua posterior construção.

Ao longo dos anos, tem-se a percepção clara de que empresas que desenvolvem seus produtos de forma a propiciar uma inovação, tem tido destaque frente ao mercado competitivo. Essa inovação se faz possível devido a um processo de desenvolvimento bem estruturado.

O desenvolvimento de produtos gerenciado de forma eficaz e eficiente tem se mostrado um diferencial competitivo para as organizações. Através de um processo bem definido e gerenciado, as organizações de qualquer segmento são capazes de se manterem no mercado de forma competitiva.

Mais do que desenvolver o produto, busca-se procedimentos capazes de promover inovações produtivas com eficácia. Sendo assim, o processo do projeto tem que ser administrado em virtude do envolvimento de recursos financeiros, físicos e humanos.

Neste sentido, Kaminski (2005) cita que o desenvolvimento de produtos é uma tarefa complexa, envolvendo diversos interesses e habilidades, tanto pelo lado do consumidor quanto do desenvolvedor, equipe de vendas, distribuição e suporte, tornando assim o desenvolvimento de novos produtos uma solução de compromissos.

Entende-se que a atividade de desenvolver produtos deve se integrar com outras áreas funcionais para garantir um produto que realmente atenda ao que foi planejado. As áreas são: processo de vendas, suporte, suprimentos, produção, distribuição, pesquisa e desenvolvimento, planejamento estratégico, atendimento ao cliente, monitoria do mercado. A Figura 2 ilustra essa integração entre os processos.

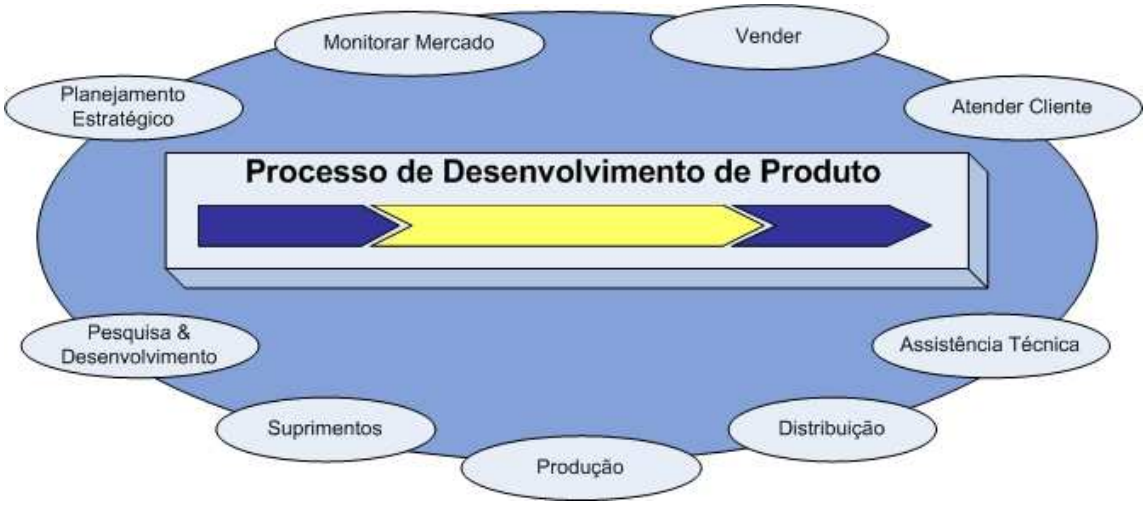

Figura 2. Processos relacionados com desenvolvimento de produtos. ROZENFELD et al. (2006)

Pela importância que o desenvolvimento de produtos representa para as organizações, este capítulo faz uma explanação sobre o PDP (processo de desenvolvimento de produto) e o PDS (processo de desenvolvimento de software).

2.2 Processo de desenvolvimento de produto

O processo de desenvolvimento de um produto, segundo Pugh (1990), é uma atividade sistemática, da identificação de necessidades de mercado/usuários até a venda do produto que atenda com êxito aquela necessidade, ou seja, uma atividade que envolve produto, processos, pessoas e organização, de modo a caracterizar os requisitos do projeto.

O PDP é considerado como um processo pelo qual uma organização transforma dados sobre oportunidades de mercado e possibilidades técnicas em informações de valor para o produto final, segundo Clark e Fujimoto (1991).

Adicionalmente, Cooper (1993) idealizador do chamado *stage-gate*, define o PDP como um modelo formal, mapa, *template* ou processo pensado para orientar um projeto de um novo produto do estágio de idéias até depois do seu lançamento. Avançando em suas contribuições Rozenfeld et al. (2006) defendem que o processo de desenvolvimento de produtos é operacionalizado através de projetos separados. Sendo assim, propõem um modelo de referência genérico que adotado neste trabalho e que será melhor discutido no capítulo 5.

Uma forma de se caracterizar o processo de desenvolvimento de produto é por meio das seguintes dimensões, as quais estão presentes no modelo de referência desenvolvido pelo Grupo de Engenharia Integrada: ROZENFELD et al. (2006)

- **atividades/fases:** Há muitas formas de se classificar as fases e atividades do processo de desenvolvimento de produto. Na Abordagem do Grupo EI e no modelo de referência são identificadas sete fases: Conceber Produto, Conceituar Produto, Projetar Produto e Processo, Homologar Produto, Homologar Processo e Ensinar Empresa. O modelo de referência apresenta as atividades dispostas em cada uma destas etapas;

- **recursos:** compõe-se de todos conceitos/filosofias, métodos/técnicas e ferramentas/sistemas que podem ser aplicados no processo de desenvolvimento de produto;

- **organização:** refere-se a não só a estrutura organizacional responsável e executora das atividades de desenvolvimento de produto como também os elementos como cultura, qualificação profissional, formas de comunicação entre os indivíduos, etc... , ligados aos aspectos de organização do trabalho;

- **informação:** dimensão que representa o fluxo de informação existente neste processo: os dados, sua estrutura e o formato como estes circulam (relatórios, fichas, telas de computador, etc...).

2.3 Processo de desenvolvimento de software

O Processo de desenvolvimento de software (PDS) se baseia no conceito generalizado de processo, que pode ser definido como uma seqüência de estados de um sistema que se transforma. SPINOLA (1999).

Entende-se então que o conceito de PDS segue a mesma definição utilizada para outros processos, sendo um conjunto de atividades inter-relacionadas que quando executadas produzem um bem ou serviço, que neste caso é o software.

Apesar de possuir conceitos similares para definição de processos, Humphrey (1989), define que a gestão de software se diferencia da gestão de um processo típico de manufatura, em virtude dos seguintes aspectos:

- software geralmente é mais complexo do que outros produtos manufaturados;
- em virtude de a engenharia de software ser relativamente nova, não há muitos gerentes e profissionais que tenham condições de avaliar um processo efetivo;

- em outros campos da engenharia, a entrega para manufatura segue uma disciplina natural que não é encontrada no processo de software;
- a disciplina de software não é uma ciência natural. Não se baseia em fundações naturais estáveis da ciência, como a Física;
- software é a parte mais visível de um sistema para os usuários; portanto,objeto de constantes mudanças e reclamações;
- custos insignificantes de se reproduzir software faz com que a descoberta de problemas ocorra muito tarde no ciclo de vida.

Sommerville (2003) cita que o processo de software é definido como um conjunto de atividades e resultados associados que produzem um produto de software. Pressman (2005) apresenta a definição de um processo de software como um *framework* para as tarefas que são necessárias para a construção de um software de alta qualidade.

O *Software Engineering Institute* (SEI) da *Carnegie Melon University,* propõe o seguinte:

> "*Um processo é uma seqüência de passos realizados para um dado propósito. Colocado de maneira mais simples, processo é aquilo que você faz. Processo é aquilo que as pessoas fazem, usando procedimentos, métodos, ferramentas, e equipamentos, para transformar matéria prima (entradas) em produto (saída) que tenha valor para o cliente*". PAULK (1995).

Em meados dos anos 70, Schwartz já apontava as fases principais do processo de produção de um sistema de software:

1. <u>Especificação de Requisitos</u>: tradução da necessidade ou requisito operacional para uma descrição da funcionalidade a ser executada.

2. <u>Projeto de Sistema</u>: tradução destes requisitos em uma descrição de todos os componentes necessários para codificar o sistema.

3. Programação: (Codificação): produção do código que controla o sistema e realiza a computação e lógica envolvida.

4. Verificação e Integração: (*Check-out*): verificação da satisfação dos requisitos iniciais pelo produto produzido.

A definição mais atual oferecida por Sommerville (2003) é similar; e divide as atividades como especificação, desenvolvimento, validação e evolução.

Um processo de desenvolvimento de software tem, segundo Booch, et al. (2005), quatro objetivos fundamentais:

1) Providenciar orientação sobre a seqüência de realização das atividades envolvidas.
2) Especificar os modelos descritivos do sistema que devem ser desenvolvidos.
3) Dirigir as tarefas dos participantes e da equipe como um todo.
4) Providenciar critérios para a monitoração e avaliação dos modelos e atividades do projeto.

O não cumprimento eficiente dessas tarefas por parte de alguns projetistas se deve a diversos fatores, inclusive os de caráter histórico e é justificado pela dificuldade que esses projetistas enfrentam em definir um processo realístico para um produto que é predominantemente lógico e não exatamente, físico.

Dessa forma, empresas que definirem seus processos, de forma eficiente, garantirão sua sobrevivência e destaque em um mercado globalmente concorrido, tendo estimativas, custos, prazos e o cumprimento dos requisitos fielmente garantidos, ou seja eficiência no processo e eficácia no produto. Inclusive com reais condições de comprovarem a maturidade de seu processo, conquistando em função disto, uma certificação de qualidade correspondente.

Entre os mais conhecidos se destacam o CMMi (*Capablity Maturity Model Integrated*), a norma ISO 12207 e o MPS.Br (Melhoria do processo de software brasileiro).

Desde a década de 60 tem-se procurado estabelecer procedimentos sistematizados de desenvolvimento que pudessem garantir um produto de qualidade. Desde que os primeiros sistemas computacionais começaram a ser criados surgiram metodologias que tinham como intuito servirem de guia para a criação de software. Entre eles se destacam, o Modelo Clássico (Cascata), Espiral, Incremental, Prototipagem e mais recentemente as metodologias chamadas de Ágeis, como o *Extreme Programming* (XP) e o SCRUM.

Estas metodologias sofrem críticas quanto a sua eficácia de aplicação, dando margem para que adaptações sejam implementadas. O que se tem ainda, de fato é um panorama desfavorável quanto a oferta de algum modelo eficaz que garanta produtividade e qualidade, principalmente voltado para as pequenas empresas. Esforços neste sentido continuam a serem despendidos.

A partir da definição das fases básicas de um processo de desenvolvimento, muitos pesquisadores idealizaram novos modelos na tentativa de padronizar e facilitar a criação de um software. Alguns desses modelos serão descritos a seguir.

2.4 Modelos de Processo de software

Muitas vezes identificados como paradigmas, os modelos de desenvolvimento de software foram surgindo desde meados dos anos 60 para reverter o que pode ser chamado de forma caótica de criação de sistemas. À medida que novas tecnologias, bem como necessidades foram aparecendo, esses modelos tentavam suprir a carência por um processo disciplinado de desenvolvimento.

Faz-se a seguir uma descrição dos modelos mais difundidos.

2.4.1 Modelo Cascata

Também conhecido como modelo clássico, foi criado por Royce (1970). Este modelo tem como principal característica o fato das atividades estarem distribuídas de forma linear e seqüencial, como mostra a figura 3.

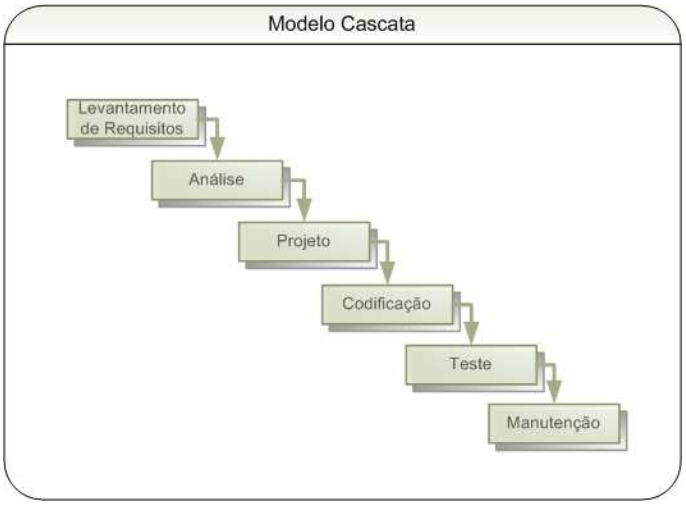

Figura 3. Modelo cascata. SOMERVILLE (2003).

Apesar da proposta inicial ter como característica a retro-alimentação, o que tornava o modelo menos linear, a prática deturpou essa idéia e o modelo passou a ser seqüencial, o que significa que cada etapa deve ser completada na sua totalidade para desencadear a próxima fase. Isso demanda um tratamento ordenado e sistemático ao desenvolvimento do software e exige maior tempo de desenvolvimento. PFLEEGER (2004).

Apesar de ser considerado um modelo bastante didático e fácil de ser compreendido, muitas são as críticas a este paradigma. Uma das críticas se deve ao fato de que o modelo sendo linear e seqüencial, tem um custo muito elevado na manutenção já que o retorno para uma das fases anteriores é difícil.

Para Teles (2004), o modelo assume que todo o sistema é construído de uma única vez. Isso implica que para poder testar todo o sistema é preciso que todas as fases anteriores,

desde o levantamento dos requisitos até a implementação, estejam concluídas e integradas em um todo. Na prática, os erros ocorrem em todas as etapas do desenvolvimento e deixar os testes para depois da implementação é algo altamente ineficiente. Sabe-se que para sistemas complexos onde não se tem todos os requisitos definidos, as dificuldades na utilização desses modelos o torna inoperante.

2.4.2 Modelo Espiral

O pesquisador Boehm (1988) sugeriu um modelo evolucionário, isto é, a medida que o desenvolvimento ocorre, o produto evolui. O processo é delimitado em uma seqüência de fases que resultam versões incrementais do software. Na forma de espiral, cada volta na espiral representa novos incrementos.

São definidas quatro importantes atividades conforme se apresenta na figura 4, a seguir, que são:

(1) Planejamento: determinação dos objetivos, alternativas e restrições.

(2) Análise de riscos: análise de alternativas e identificação/resolução de riscos.

(3) Engenharia: desenvolvimento do produto no “nível seguinte”.

(4) Atualização feita pelo cliente: avaliação dos resultados da engenharia.

Um ciclo se inicia com a fase de planejamento (primeira atividade) onde ocorre o comprometimento dos envolvidos e o estabelecimento de uma estratégia para alcançar os objetivos. Na segunda atividade, faz-se uma análise de riscos. Usando a prototipação como ferramenta. Uma vez que o risco for tido como inaceitável, pode-se não continuar com o

projeto. Na terceira atividade ocorre o desenvolvimento do produto. Na quarta atividade o produto é avaliado e se prepara para iniciar um novo ciclo.

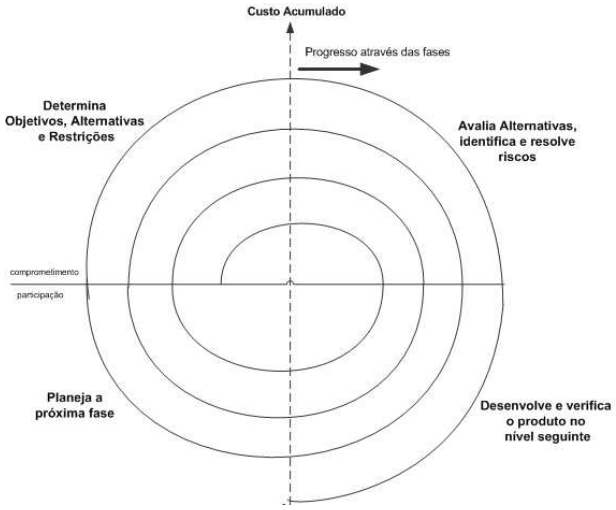

Figura 4. Modelo espiral resumido de Boehm. Adaptado de PRESSMAN (2005).

Em geral, modelos evolucionários têm o propósito de lidar melhor com um conjunto de requisitos que sofrem mudanças constantes ou que são incertos.

2.4.3 Modelo Incremental

Na visão de Pfleeger (2004), assim como o modelo espiral, o modelo incremental também é evolucionário. Parte-se do princípio de que em um produto final não precisa necessariamente ser entregue como um pacote, ao todo. Várias partes, chamadas de incremento, podem ser desenvolvidas separadamente conforme a prioridade.

Tem-se assim um ganho na confiabilidade do produto, já que o incremento entregue antes, será usado e testado de forma antecipada. A figura 5 mostra de forma simplificada este modelo.

A desvantagem é que para muitos usuários receber um produto em partes pode não ser confortável e a integração entre os incrementos pode ser um fator de alto risco se não for bem controlada.

Um processo de desenvolvimento segundo esse modelo divide o desenvolvimento de software em iterações. Em cada iteração, são realizadas as atividades de análise, projeto, implementação e testes para uma parte do sistema. Esta característica contrasta com o modelo em cascata, no qual as fases de análise, projeto, implementação e testes são realizadas uma única vez para o sistema como um todo. PRESSMAN (2005).

Uma vez alocados os requisitos a uma iteração de desenvolvimento, estes requisitos são analisados, projetados, implementados, testados e implantados. Na iteração subseqüente, um outro subconjunto de requisitos é considerado para ser desenvolvido, o que produz uma nova versão (ou incremento) do sistema que contém extensões e refinamentos sobre a versão anterior. Desta forma, o desenvolvimento evolui em versões, através da construção incremental e iterativa de novas funcionalidades até que o sistema completo esteja construído.

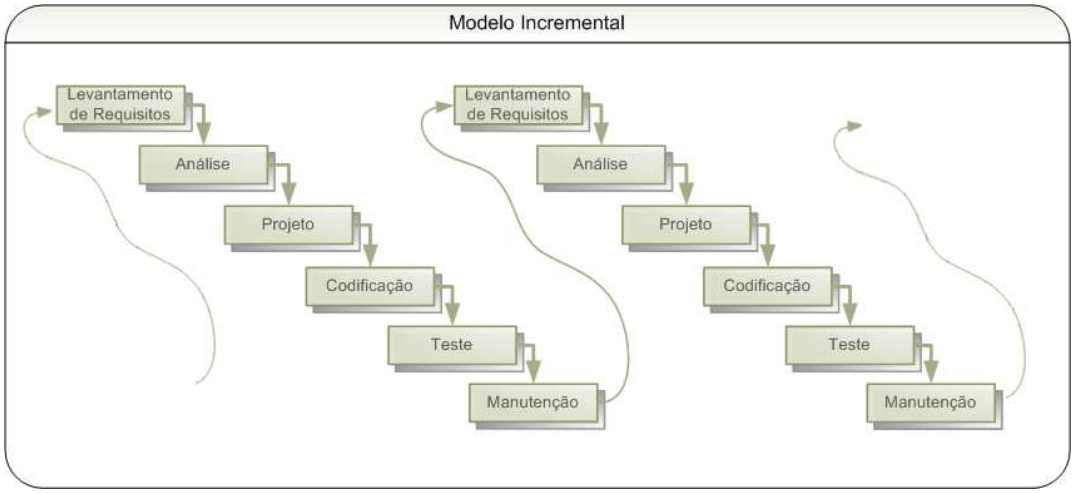

Figura 5. Modelo Incremental. PRESSMAN (2005).

Mesmo com vários modelos criados para disciplinar o processo de desenvolvimento, durante todos esses anos, a indústria de software não conseguiu ter índices de produtividade aceitáveis. Foi então que na última década, alguns pesquisadores criaram o conceito de desenvolvimento ágil idealizando metodologias que poderiam quebrar os paradigmas até então estudados.

2.4.4 Rational Unified Process (RUP)

Este *framework* proprietário foi criado em 1998 pela Rational Software Corporation baseado-se nas melhores práticas de vários outros processos que antecederam sua criação. Tem como finalidade ser um guia para equipes de desenvolvimento na criação de produtos de software. Segundo Booch et al. (2005), o processo unificado consiste em uma abordagem do ciclo de vida de um processo, especialmente adequada à UML (*Unified Modeling Language*).

Kruchten (2003) define o RUP como um processo bem estruturado para desenvolver software com alta qualidade de modo repetível e previsível.

Uma característica do RUP é o fato de apresentar a modelagem de negócios durante o início do ciclo de desenvolvimento do produto, que torna o entendimento das necessidades mais claras. A figura 6 apresenta o modelo de forma gráfica para uma melhor compreensão.

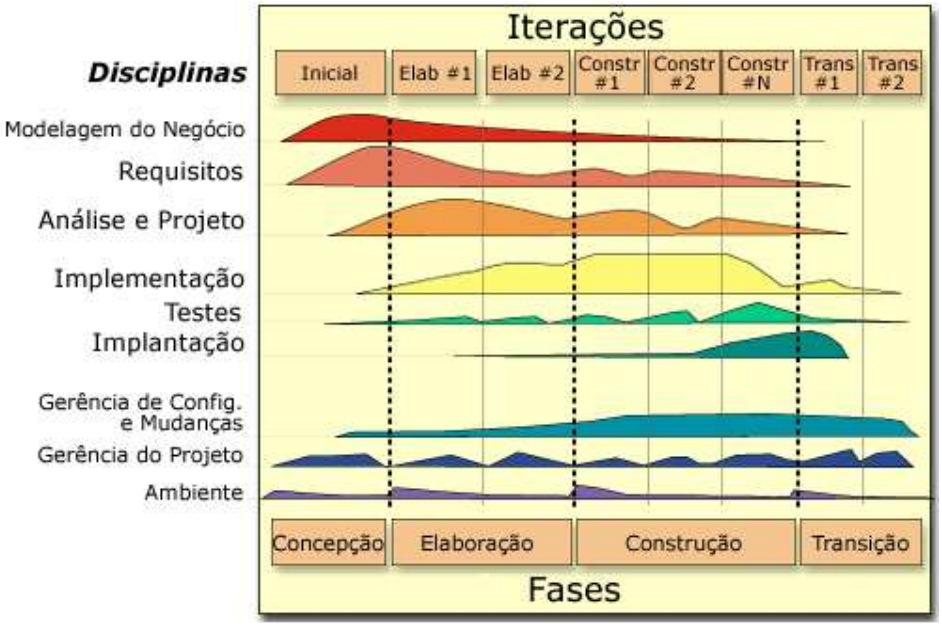

Figura 6. Rational Unifield Process. KRUCHTEN (2003).

Nesta figura estão representados, nas linhas, as chamadas disciplinas, que indica o que será feito durante o desenvolvimento. Já nas colunas, estão as fases do desenvolvimento. Na intersecção entre as linhas e colunas estão representadas as quantidades de esforço a ser realizado por disciplina durante cada uma das fases.

O RUP é composto por 4 fases, que são: concepção, elaboração, construção e transição, cada uma com objetivos específicos, ou seja:

- Fase de concepção: estabelecer o escopo e a viabilidade econômica do projeto.
- Fase de elaboração: elimina os principais riscos e estabelece uma arquitetura estável a partir da qual o sistema poderá evoluir.
- Fase de construção: um produto completo é desenvolvido de maneira iterativa até que esteja pronto para ser passado aos usuários, o que ocorre na fase de transição, onde uma versão beta do sistema é disponibilizada.

- Fase de transição: inicia-se um novo ciclo de desenvolvimento para a evolução do produto, o que envolveria todas as fases novamente. Em todas as fases identifica-se no seu final um marco (*milestone*) de verificação de quais objetivos da fase foram alcançados.

As disciplinas definem o que deve ser feito pelos responsáveis em termos de atividades e artefatos. Uma facilidade que o RUP apresenta é fornecer modelos (*templates*) para cada artefato e roteiros (*guidelines*) para cumprimento das atividades. Por estas práticas o RUP é considerado como sendo um processo de desenvolvimento iterativo e incremental, com enfoque orientado a objetos (OO).

Apesar dessa abordagem ter surgido como uma solução para empresas desenvolvedoras, algumas críticas e limitações devem ser consideradas. A principal delas está relacionada ao fato do modelo ter sido criado por uma empresa comercial que vincula a cada fase e disciplina produtos e ferramentas comerciais proprietários. Outra crítica está relacionada ao alto volume de documentos exigidos em cada atividade o que dificulta a implementação por pequenas empresas que não possuem recursos para criar e gerenciar a documentação.

Diante destas críticas, são poucos os relatos de pequenas empresas que conseguiram implantar o *framework* com sucesso. Este fato remete às organizações a criar derivações e adaptá-lo às suas realidades, sendo que a derivação mais conhecida se chama openUP.

2.5 Metodologias ágeis

As chamadas metodologias ágeis para desenvolvimento de software, são uma resposta às chamadas metodologias pesadas ou tradicionais. Fowler (2006) coloca as metodologias

modernas de desenvolvimento como uma reação aos modelos tradicionais e extremamente conceituais, chamada por ele de monumentais que existem há muito tempo e que nunca ficaram conhecidas por serem particularmente de sucesso. A crítica mais freqüente a elas é que são burocráticas, ou seja, existe documentação excessiva.

Como uma reação a estas metodologias, um novo grupo apareceu na década de 90, com maior flexibilidade de procedimentos que tentam estabelecer um compromisso útil entre nenhum processo e um processo excessivo, provendo apenas o processo necessário e suficiente para fornecer uma vantagem razoável.

A primeira impressão de que as metodologias ágeis de desenvolvimento se opõem as metodologias mais tradicionais, simplesmente porque geram menos documentos, não toca a essência dos métodos ágeis. Duas das principais idéias por trás dos métodos ágeis, as quais os diferenciam das demais metodologias, segundo Fowler (2006), são:

1. Métodos ágeis são mais adaptativos do que previdentes. Métodos tradicionais procuram planejar em detalhes longos períodos, desta forma, é da sua natureza a resistência à mudança, uma vez que o planejamento esteja estabelecido. Métodos ágeis, por sua vez, acolhem a mudança a qualquer momento, a ponto de adaptar a própria metodologia para serem bem sucedidos.

2. Métodos ágeis baseiam-se mais nas pessoas do que nos processos. Enquanto os métodos tradicionais procuram definir processos que vão funcionar com qualquer um que os execute, os métodos ágeis asseguram que nenhum processo pode superar as habilidades de uma equipe. Desta forma, o papel do processo é dar suporte à equipe de desenvolvimento em seu trabalho.

A discussão que se promove neste aspecto, está relacionada ao desenvolvimento de software como um processo definido ou um processo empírico no sentido de não se

conhecerem todas as características do produto nas fases iniciais de desenvolvimento. As metodologias ágeis partem do princípio de que o processo de software é empírico.

Processos empíricos baseiam-se na visibilidade dos aspectos e resultados do processo e em prover maneiras de inspecionar e corrigir o processo e o produto. Dessa forma, processos empíricos aceitam as falhas como conseqüência natural da produção e tentam torná-las visíveis e passíveis de correção o mais cedo possível. Assim, argumenta Schwaber et al (2002), a abordagem empírica é ideal para o ambiente de desenvolvimento de software, onde funcionalidades consideradas "terminadas" muitas vezes não estão suficientemente testadas e aceitas pelo cliente, onde os passos a serem seguidos na criação de um produto só podem ser descritos em alto-nível (como "elicitar requisitos") e onde correções, seguindo um processo definido, têm um custo inaceitável ("modificar os documentos de requisitos, arquitetura e testes além do código-fonte").

Existem diversos processos e metodologias que são consideradas ágeis, sendo *Extreme Programming*, Scrum, Crystal e *Microsoft Solution Framework* (MSF), apenas alguns exemplos.

Das metodologias ágeis existentes no mercado, a Scrum tem se destacado nos últimos anos por concentrar-se nas práticas e atividades de gerenciamento de projetos, reunindo ações de monitoramento e *feedback* que visam identificar e corrigir deficiências e/ou impedimentos no processo de desenvolvimento. A seguir são detalhadas as duas mais conhecidas.

2.5.1 eXtreme Programming

eX*treme Programming* (XP) teve como idealizador o americano Kent Beck que em 1999 propôs uma forma de desenvolvimento que pudesse valorizar a programação ignorando

as atividades, que na visão do autor são demasiadas, como documentação, análise e projeto muito extensos. Sendo assim, esta abordagem é voltada explicitamente para práticas de codificação.

A metodologia está voltada para projetos cujos requisitos são alterados constantemente com equipes pequenas (aproximadamente 12 desenvolvedores) e com desenvolvimento iterativo.

De acordo com Beck (2004) uma empresa interessada em utilizar o XP deve-se basear nos quatro valores defendidos pela metodologia, que são:

- *Feedback:* o cliente estando disponível gera constante retorno a equipe de desenvolvimento;
- Comunicação: uma boa comunicação é essencial para agilizar o retorno do cliente a equipe;
- Simplicidade: deve-se implementar o necessário para atender a necessidade do cliente;
- Coragem: para se conseguir colocar em prática todos os valores e práticas do XP.

Para Jeffries (2001), XP é uma disciplina de desenvolvimento de software baseada em valores que funciona unindo todo o time através de práticas simples, tais como oferecer *feedback* suficiente para todos saberem onde estão posicionados em relação a um projeto e como devem ajustar as práticas para esta situação única.

Além dos valores, têm-se algumas práticas que garantem o bom andamento dos projetos, são elas: cliente presente, planejamento do jogo, *stand up meeting*, programação em par, desenvolvimento guiado pelos testes, refatoração, código coletivo, código padronizado, *design* simples, metáfora, ritmo sustentável, integração contínua e *releases* curtos.

Essas práticas estabelecias por Jeffries (2001), são detalhadas a seguir.

- **Time:** todos os envolvidos fazem parte do projeto e possuem contribuições;
- **Planejamento do jogo:** atividade para definir estimativas, prioridades e funcionalidades juntamente com o cliente;
- **Desenvolvimento orientado a testes:** os testes são planejados antecipadamente ao desenvolvimento;
- **Versões intermediárias: e**ntrega de pequenas versões do software ao cliente que poderá acompanhar o desenvolvimento, diminuindo riscos e dando opiniões sobre o produto;
- **Programação em par:** sempre dois programadores executam a função em conjunto;
- **Projeto simples:** evita-se complexidade desnecessária que poderia comprometer desempenho e manutenibilidade;
- **Refatoração:** consiste na melhoria da legibilidade do código a fim de facilitar seu entendimento e manutenção;
- **Padronização do código:** todos seguem um padrão de código definido;
- **Metáfora:** a comunicação deve ser facilitada utilizando metáforas;
- **Todos são donos de tudo:** todo o código pertence à equipe e qualquer membro pode melhorar o que for necessário;
- **40 horas semanais:** todo cronograma e estimativa são feitos para se trabalhar 40 horas semanais. Se um membro necessitar trabalhar além desse tempo significa que algo está errado.

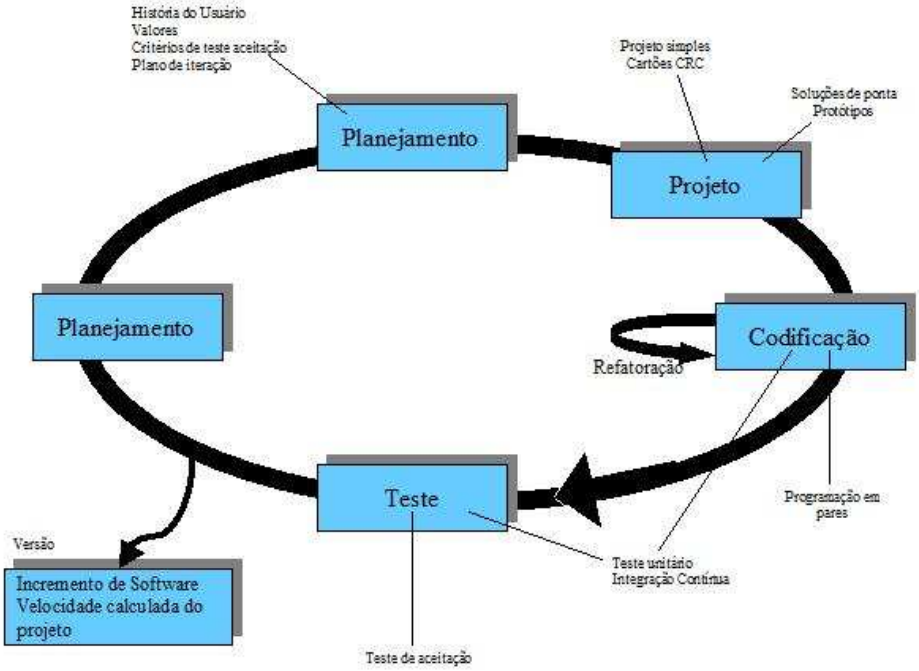

Figura 7. O processo Extreme Programming. PRESSMAN (2005).

As críticas a esta abordagem apontam que o XP não permite reproduzir resultados em outros projetos, já que a documentação e o controle não são tratados de forma adequada e que ela seria impraticável em projetos complexos. Além disso, existem barreiras como a comprovação para o cliente de que um processo nessas condições será eficaz. A disponibilidade do cliente sendo necessária pode se tornar também um empecilho, uma vez que nem todos projetos contarão com essa possibilidade e grandes empresas impõem burocracias para a tomada de decisões e mudar isso é extremamente complexo.

Apesar dessas barreiras e críticas, a metodologia mostra uma tendência por soluções no sentido de se agilizar o desenvolvimento. Dentre as boas práticas, algumas como a refatoração, a programação orientada a testes, a democratização do código fonte e padronização do código, são extremamente válidas e devem ser levadas em consideração.

2.5.2 SCRUM

Assim como *Extreme Programming*, a SCRUM é outra metodologia ágil que apresenta uma comunidade grande de adeptos em todo o mundo. Idealizada por Ken Schwaber na década de 90, Schwaber (2002), possui como objetivo fornecer um processo conveniente para a gerência de projeto e desenvolvimento orientado a objeto. Scrum se destaca das demais por dar mais enfoque a área de gerenciamento.

Trata-se de fato, de um processo de desenvolvimento iterativo e incremental que pode ser aplicado a qualquer produto, não somente em software, ou no gerenciamento de qualquer atividade complexa.

Segundo Schwaber et al. (2002), a Scrum apresenta uma abordagem empírica que aplica algumas idéias da teoria de controle de processos industriais para o desenvolvimento de softwares, re-introduzindo as idéias de flexibilidade, adaptabilidade e produtividade. O foco da metodologia é encontrar uma forma de trabalho dos membros da equipe para produzir o software de forma flexível e em um ambiente em constante mudança.

O fato da Scrum ser baseado em conceitos de processos industriais mostra que outras engenharias podem contribuir e muito com a engenharia de software.

O método baseia-se ainda, conforme Schwaber et al (2002), em princípios como: equipes pequenas (máximas sete pessoas); requisitos que são pouco estáveis ou desconhecidos; e iterações curtas. Divide o desenvolvimento em intervalos de tempos de, no máximo 30 dias, também chamadas de *Sprints*.

Este método não requer ou fornece qualquer técnica ou método específico para a fase de desenvolvimento de software, apenas estabelece conjuntos de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto. As práticas gerenciais do Scrum são:

- **Tarefas do Produto:** define tudo o que é necessário no produto final. Contém uma lista priorizada e constantemente atualizada dos requisitos do sistema que está sendo construído ou otimizado;
- **Estimativa de esforço:** como a Scrum representa um processo iterativo, a estimativa de esforço para executar as tarefas deve ser realizada freqüentemente;
- ***Sprint*:** procedimento de adaptação às mudanças de variáveis de ambiente, como requisitos, tempo, recursos ou tecnologia. S*prints* são intervalos fixos de tempo, em que todo o trabalho é realizado. Na Scrum, um *sprint* tem duração de trinta dias. Durante um *sprint* a equipe Scrum se organiza para produzir um incremento do produto. Reuniões Scrum diárias (*Daily Scrum Meetings*), de aproximadamente quinze minutos realizadas para verificar o progresso do projeto;
- **Reunião de Revisão de *Sprint*:** Apresentação dos resultados no último dia do *sprint*.

As pequenas equipes são formadas de: projetistas, programadores, tecnólogos, engenheiros e gerentes de qualidade. Estas equipes trabalham em cima de funcionalidades definidas no início de cada *sprint* e são responsáveis pelo desenvolvimento destas funções.

A figura 8, a seguir, representa o fluxo de desenvolvimento da metodologia Scrum.

Figura 8. Fluxo de desenvolvimento Scrum. Adaptado de COCHANGO (2006).

Como já foi dito, cabe reforçar um aspecto positivo desta metodologia que é o procedimento de trabalho das equipes que estabelece reuniões diárias e quinzenais. Diariamente se realiza uma reunião de quinze minutos, onde o time apresenta à gerência o que será feito no dia seguinte, e nestas reuniões os gerentes podem levantar os fatores de impedimento e avaliar e o progresso geral do desenvolvimento.

Um dos fatores de destaque é a possibilidade de fornecer um mecanismo de informação de status, atualizado continuamente, e a redefinição pela responsabilidade de tarefas a serem divididas dentro da equipe, de forma explícita.

Nas reuniões quinzenais, a equipe se dedica a proceder a revisões das etapas e do todo, e sendo o caso, efetuar retroações (*feedback*) para corrigir e otimizar o processo de projeto.

2.6 Qualidade no Processo de Software

A norma ISO/NBR 8402 (1994) define qualidade como a totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades implícitas e

explícitas. Necessidades explícitas são aquelas expressas na definição dos requisitos propostos pelo produtor e necessidades implícitas, aquelas que embora não documentadas nas exigências do produtor, são necessárias ao usuário.

Segundo Machado (2001), para muitos projetistas de sistemas, a qualidade do processo de software é tão importante quanto a qualidade do produto. A partir da década de 90 houve uma crescente preocupação com a modelagem e melhoria do processo de software. Abordagens importantes como as normas ISO 15504 e a ISO 12207, o modelo CMMi (*Capability Maturity Model Integrated*) sugerem que melhorando o processo de software se obtém a melhoria da qualidade dos produtos.

Para melhor caracterização dos esforços, são apresentados, a seguir, os diversos modelos criados ao longo do tempo, que estabelecem uma correlação construtiva, entre a melhoria da qualidade do processo e a conseqüente melhoria da qualidade do produto.

2.6.1 CMMi

O *Capability Matururity Model Integrated* é o mais reconhecido modelo de qualidade para processo de desenvolvimento de software no mundo. Idealizado pelo Instituto de Engenharia de Software (SEI) da Universidade Carnegie Mellon - USA, teve sua primeira versão lançada em 1991, com limitação somente voltada às práticas de engenharia de software. Com a evolução do modelo e sua aceitação mundial, vários versões foram surgindo. Como exemplo, tem-se: aquisição de software, gestão de times e desenvolvimento integrado de produto e processo e engenharia de sistemas. O CMMi surgiu para unificar essas versões.

Humphrey (1989), principal idealizador do CMM, descreve que o modelo possui estágios de maturidade pelos quais as organizações passam enquanto evoluem o seu ciclo de desenvolvimento, através de avaliação contínua, identificação de problemas e ações corretivas

dentro de uma estratégia de melhoria dos processos. Este caminho de melhoria é definido por cinco níveis de maturidade, conforme apresentado na Figura 9.

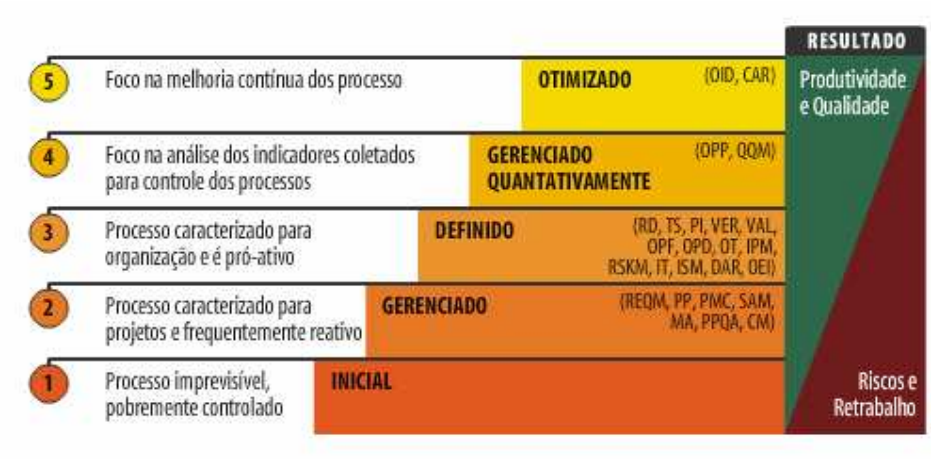

Figura 9. Níveis de maturidade do CMMi. CMMI-DEV (2006).

O CMMi possui duas representações: "contínua" ou "por estágios". Estas representações permitem a organização utilizar diferentes caminhos para a melhoria de acordo com a necessidade:

- **Contínua:** possibilita a organização utilizar a ordem de melhoria que melhor atende os objetivos de negócio da empresa. É composto por níveis de capacidade.
- **Por estágios:** disponibiliza uma seqüência pré-determinada para melhoria baseada em estágios que não deve ser desconsiderada, pois cada estágio serve de base para o próximo, sendo composto por níveis de maturidade.

O quadro 2 apresenta de forma resumida, as características de cada nível estabelecido pelo CMMi:

Quadro 2. Características de cada nível do CMMi.

1 – Inicial	No nível 1 os processos são informais e caóticos. Existe uma completa falta de planejamento e controle dos processos. Os funcionários estão focados basicamente em atividades corretivas que surgem a todo instante.
2 – Gerenciado	Os processos básicos de gerenciamento de projetos para planejar e acompanhar custos, prazos e funcionalidades são estabelecidos. Compromissos são firmados e gerenciados. A disciplina de processo permite repetir sucessos de projetos anteriores em aplicações similares. Tipicamente, possui gerenciamento de projetos estabelecido; alguns procedimentos técnicos escritos; acompanhamento de qualidade; gerência de configuração inicial; atividades básicas de medição e análise. O sucesso depende basicamente do gerenciamento do projeto.
3 - Definido	Atividades de gerenciamento básico e de Engenharia de Software são documentadas, padronizadas e integradas em processos-padrão. Todos os projetos de desenvolvimento ou manutenção de softwares utilizam uma versão de um desses processos adaptada às características especificas de cada projeto. Possui processos gerenciais e técnicos bem definidos, possibilidade de avaliação do processo; ferramentas e metodologias padronizadas; medições iniciais de desempenho; inspeções e auditorias rotineiras; testes padronizados; gerência de configuração; evolução controlada dos processos técnicos e gerenciais.
4 – Gerenciado Quantitativamente	Métricas detalhadas do processo de software e da qualidade do produto são coletadas. Tanto o processo como o produto de software são quantitativamente compreendidos, avaliados e controlados. Relatórios estatísticos são gerados. Tipicamente, encontra-se estabelecido e em uso rotineiro um programa de medições. A qualidade é planejada por um grupo dedicado, sendo rotineiramente avaliada e aprimorada
5 - Otimizado	A melhoria contínua do processo é estabelecida por meio de sua avaliação quantitativa e da implantação planejada e controlada de tecnologias e idéias inovadoras. Projetos-piloto são realizados para a absorção e internalização de novas tecnologias. Tipicamente, um alto nível de qualidade e de satisfação dos clientes é alcançado rotineiramente, com grande foco na melhoria contínua.

Fonte: FIORINI (1998).

Estruturado em 5 níveis de maturidade, o CMMi abrange 22 áreas de processo que segundo o CMMI-DEV (2006), são um conjunto de práticas interrelacionadas, que, quando implementadas coletivamente, satisfazem os objetivos considerados importantes para obter melhoramentos numa determinada área. Essas áreas de processos, por sua vez, são divididas em 4 categorias:

- Gerenciamento de projetos;
- Gerenciamento de processos;
- Engenharia;
- Suporte.

A Tabela 1, a seguir, representa as 22 áreas de processo do CMMi, sua categoria e nível de maturidade;

Tabela 1 - Áreas de Processo do CMMi.

Área de Processo	*Categoria*	*Nível de Maturidade*
Análise e Resolução de Causas	Suporte	5
Gerenciamento de Configuração	Suporte	2
Análise de Decisão e Resolução	Suporte	3
Gerência de Integração do Projeto	Gerenciamento de projetos	3
Medição e Análise	Suporte	2
Inovação e Implantação Organizacional	Gerenciamento de processos	5
Definição do Processo Organizational	Gerenciamento de processos	3
Foco no Processo Organizacional	Gerenciamento de processos	3
Desempenho do Processo Organizacional	Gerenciamento de processos	4
Treinamento Organizacional	Gerenciamento de processos	3
Integração do Produto	Engenharia	3
Monitoração e Controle de Projeto	Gerenciamento de projetos	2
Planejamento de Projeto	Gerenciamento de projetos	2
Garantia da Qualidade do Processo e do Produto	Suporte	2
Gerência Qualitativa do Projeto	Gerenciamento de projetos	4
Desenvolvimento de Requisitos	Engenharia	3
Gerência de Requisitos	Engenharia	2
Gerência de Risco	Gerenciamento de projetos	3
Gerência de Acordo com Fornecedor	Gerenciamento de projetos	2
Solução Técnica	Engenharia	3
Validação	Engenharia	3
Verificação	Engenharia	3

Fonte: CMMI-DEV (2006).

Apesar de toda notoriedade que o CMMi apresenta perante o mercado mundial, existe uma discussão acerca de sua real aplicabilidade pelas pequenas empresas e pela adoção de metodologias ágeis. Paulk (1998) apresenta uma visão sobre a utilização do modelo em pequenas empresas, defendendo que o modelo em si, somente diz o que deve ser cumprido em termos de processos, deixando a cargo das pequenas empresas a forma de trabalho, tendo então que se buscar meios para isso.

Com relação a adoção de metodologias ágeis, Paulk (2001), em um estudo sobre o uso do XP e CMMi ressalta que:

- XP foca no trabalho técnico e o CMMi nas questões de gerenciamento;
- XP não aborda questões relacionadas à institucionalização das práticas em uma organização, o que é considerado crucial para o CMM;
- XP aborda, até um determinado limite, questões de gerenciamento e acompanhamento de projetos.

Pode-se entender então que, de acordo com estes relatos, o uso de algumas práticas, seja de XP ou Scrum, podem ser utilizadas para auxiliar empresas, independentemente do porte a cumprir com as metas do CMMi.

Modelos de qualidade e metodologias ágeis não são antagônicas e talvez possam coexistir na mesma organização. O CMMi pode servir como orientação para dizer quais processos e práticas de gerencia de projetos devem estar presentes e Scrum para dizer como esses processos devem ser executados.

Todavia, são poucos os relatos reais de implantação do CMMi em pequenas empresas ou de adoção de metodologias ágeis. Segundo MCT (2006), apenas 70 empresas no Brasil têm certificação CMM ou CMMi sendo que apenas seis estão no nível 5, das quais, todas são

médias ou grandes empresas. A tabela 2 apresenta uma cronologia da uso do CMM e do CMMi , no Brasil e sua adoção por organizações produtoras de software.

Tabela 2 - Organizações com certificação CMM/CMMi no Brasil

Desde	Nível Atual								No ano	Até o ano
	2		3		4		5			
	CMM	CMMi	CMM	CMMi	CMM	CMMi	CMM	CMMi		
1997			1						1	1
1998	1								1	2
1999										2
2000										2
2001			4						4	6
2002	3								3	9
2003	16			1	1				18	27
2004	6		2					1	9	36
2005	14	11	1	2				2	30	66
2006				1				3	4	70
Total	**40**	**11**	**8**	**4**	**1**			**6**	**70**	

Fonte: MCT (2006).

Diante dessa constatação e das dificuldades encontradas pelas pequenas empresas, algumas ações colocam-se como alternativa ao CMMi. Entre eles pode-se destacar o MOPROSOFT no México e o MPS.Br no Brasil, ressaltando-se que ambos são fortemente baseados nas diretrizes da norma ISO/IEC 12207 apresentada a seguir.

2.6.2 ISO/IEC 12207

A Norma ISO/IEC 12207 foi criada pela *International Organization for Standardization* (ISO) e o *International Electrotechnical Commission (*IEC).

Esta norma estabelece os processos, atividades e tarefas a serem aplicados durante a aquisição, fornecimento, desenvolvimento, operação e manutenção de software. Definindo de forma ampla os processos que guiam a adaptação da sua utilização nos projetos de software em uma empresa.

Como objetivo, passa a contar com o auxílio dos envolvidos na produção de software, de modo a melhor definir suas funções, por meio de processos bem estabelecidos, e assim, proporcionar às empresas que a utilizam um entendimento melhor das atividades a serem seguidas nas operações correlacionadas.

A versão brasileira da norma, publicada em 1998, tem o mesmo nome que da norma internacional, acrescida das iniciais NBR.

Para Nogueira (2003), a adoção da norma ISO/IEC 12207 pelo desenvolvedor de software, estabelece os procedimentos de como estruturar e gerenciar o ciclo de desenvolvimento, proporcionando a possibilidade de acompanhamento de todo o processo e, permitindo que o software venha representar mais fielmente, a realidade da empresa modelada, para geração de um sistema customizado, atendendo assim de modo adequado os requisitos e necessidades demandados por essa empresa.

2.6.3 MoProSoft

Este é um modelo para a melhoria e manutenção de processo de desenvolvimento para a indústria de software mexicana. Desenvolvido pela Associação Mexicana para a qualidade em engenharia de software, através da Faculdade de Ciências da Universidade Nacional Autônoma do México (UNAM) e da Secretaria de Economia com a finalidade de se estabelecer uma norma mexicana que fosse apropriada às características de tamanho da grande maioria das empresas daquele país que são pequenas uma vez que elas representam 92% das empresas. MOPROSOFT (2006).

O modelo proposto, assim como os outros apresentados a seguir, neste trabalho, está focado em processos e considera três níveis básicos estruturais de uma organização: alta direção, gerencial e operacional.

A alta direção engloba os processos de Gestão de Negócios. O nível gerencial engloba Gestão de Processos, Gestão de Projetos e Gestão de Recursos. Este último está constituído pelos sub-processos: recursos humanos e ambiente de trabalho, bens, serviços e infraestrutura e conhecimento da organização. O nível operacional é integrado pelos processos de administração de projetos específicos e de desenvolvimento e manutenção de software.

Em cada processo estão definidos os responsáveis pela execução das práticas. As regras são direcionadas às pessoas da organização, de acordo com suas habilidades e capacitação para desenvolve-la.

A figura 10, representa as relações de processos e sub-processos, estabelecidos pelo MOPROSOFT (2005) e seus níveis de responsabilidade intrínseca.

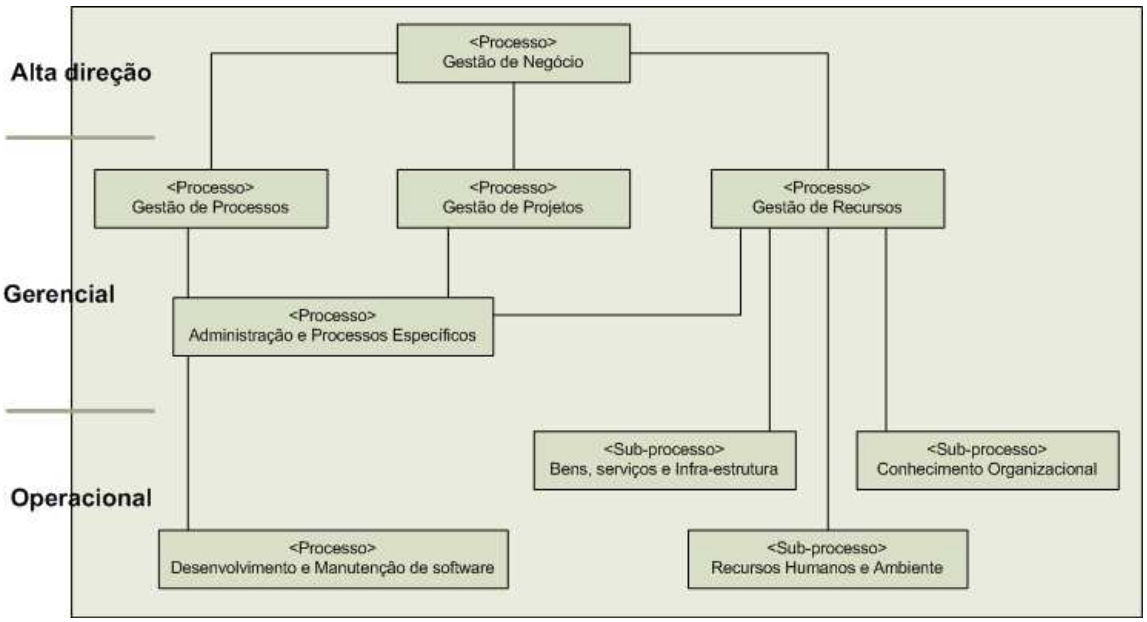

Figura 10. Diagrama da relação entre processos. Adaptado de MOPROSOFT (2005).

2.6.4 MPS.Br

O MPS.Br é um programa para Melhoria de Processo do Software Brasileiro, coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), contando com apoio do Ministério da Ciência e Tecnologia (MCT), da Financiadora de Estudos e Projetos (FINEP) e do Banco Interamericano de Desenvolvimento (BID).

Conforme descrito em MPS (2007), esse programa tem como propósito definir e aprimorar um modelo de melhoria e avaliação de processo de software, em atendimento preferencialmente às pequenas e médias empresas, de forma a satisfazer as suas necessidades de negócio e ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software.

O MPS.Br também estabelece um processo e um método de avaliação, que oferece sustentação e permite verificar que o mesmo esteja sendo empregado de forma coerente com as definições. Pode-se dizer que esta nova possibilidade representa uma sensível melhoria e contribuição ao domínio do tema.

Tomou-se como base para a construção deste modelo de melhoria e avaliação de processo de software, as normas NBR ISO/IEC 12207, a norma internacional ISO/IEC 12207 e a ISO/IEC 15504, o que permite considerar, portanto, que o modelo está em conformidade com estas normas. Este modelo também garante o conteúdo do CMMI-SE/SW, através da inclusão de processos e de resultados, segundo WEBER (2005).

O modelo de referência MR-MPS define Níveis de Maturidade, que são uma combinação entre processos e capacidade de processos, conforme a estrutura apresentada na figura 11.

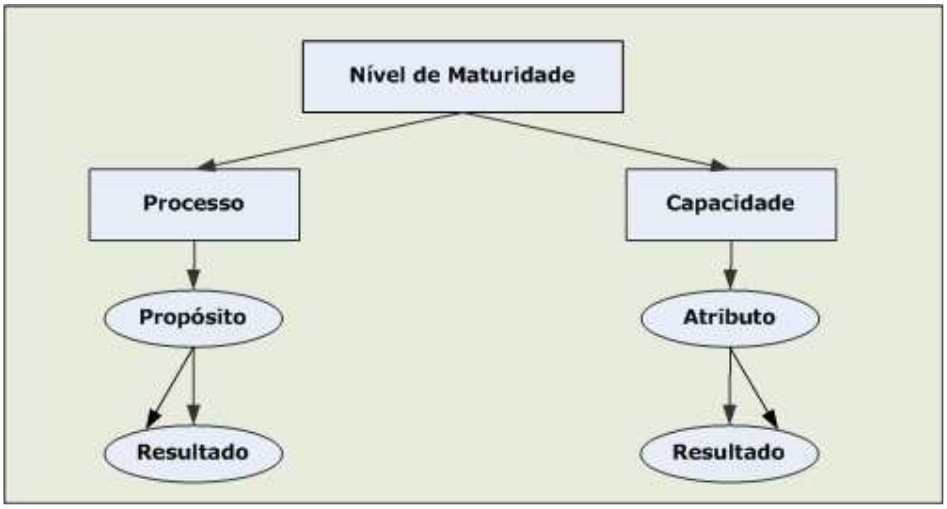

Figura 11. Estrutura do MR-MPS. MPS (2006).

A definição dos processos neste caso segue a forma apresentada na ISO/IEC 12207, declarando o propósito e os resultados de sua execução. Isso permite avaliar e atribuir graus de efetividade na execução dos processos. As atividades e tarefas necessárias para atender ao propósito e aos resultados são definidas pelos usuários do MR-MPS.

2.6.4.1 Níveis de maturidade

De fato, os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria de implementação de processos na organização. O nível

de maturidade em que se encontra uma determinada organização permite prever seu desempenho futuro em uma ou mais disciplinas. O MR-MPS define sete níveis de maturidade: A (Em Otimização); B (Gerenciado Quantitativamente); C (Definido); D (Largamente Definido); E (Parcialmente Definido); F (Gerenciado) e G (Parcialmente Gerenciado). A escala de maturidade se inicia no nível G e progride até o nível A. Para cada um destes sete níveis de maturidade foi atribuído um perfil de processos e de capacidade de processos que indicam por exemplo, onde a organização tem que colocar esforço para melhoria de forma a atender os objetivos do negócio, ou destaca seus pontos fortes.

O Quadro 3 apresenta os níveis de maturidade do modelo, seus processos e as capacidades necessárias para cada nível.

Quadro 3. Níveis de maturidade do MPS.Br

Nível	Processo	Capacidade
A	Inovação e Implementação na Organização Análise e Resolução de Causas	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
B	Desempenho do Processo Organizacional Gerência Quantitativa do Projeto	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
C	Análise de Decisão e Resolução Gerência de Risco	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
D	Desenvolvimento de Requisitos Solução Técnica Integração do Produto Instalação do Produto Liberação do Produto Verificação Validação	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
E	Treinamento Avaliação e Melhoria do Processo Organizacional Definição do Processo Organizacional Adaptação do Processo para Gerência de Projeto	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
F	Medição Gerência de Configuração Aquisição Garantia da Qualidade	AP1.1, AP2.1 e AP2.2
G	Gerência de Requisitos Gerência de Projetos	AP1.1 e AP2.1

Fonte: MPS (2007).

O progresso e o alcance do nível de maturidade são obtidos, quando são atendidos todos os resultados e propósitos do processo, e também os atributos do processo relacionados àquele nível e aos anteriores.

A divisão em estágios, tendo como referência os níveis de maturidade do CMMi tem uma escala diferenciada, com o propósito de almejar uma implementação e avaliação mais gradual e adequada às pequenas e médias empresas. A possibilidade de se realizarem avaliações considerando maior detalhamento e um maior número de níveis, permite ao gerenciador uma maior visibilidade dos resultados de melhoria dos processos o que acarretará maior confiabilidade e menores custos ou tempo de execução.

2.6.4.2 Capacidade do Processo

A capacidade do processo é um conjunto de atributos descrito em termos de resultados os quais proporcionam o cumprimento dos atributos de processo. A capacidade estabelece o grau de refinamento e institucionalização com que o processo é executado na organização. Cabe destacar que à medida que se evolui nos níveis, um maior ganho de capacidade para desempenhar o processo é exigido pela organização.

Para melhor entendimento, a capacidade do processo se caracteriza segundo cinco atributos de processos (AP) que são:

AP 1.1 – O processo é executado;

AP 2.1 – O processo é gerenciado;

AP 2.2 – Os produtos de trabalho do processo são gerenciados;

AP 3.1 – O processo é definido, e;

AP 3.2 – O processo é implementado.

2.6.4.3 Processos

Por processos entende-se o conjunto de operações realizadas segundo métodos próprios, desenvolvidos para o entendimento de necessidades estabelecidas ou implícitas. Responde ao **como fazer**.

Os processos, de acordo com MPS (2006), são descritos em termos de:

- Propósito: descreve o principal objetivo a ser atingido durante a execução do processo e os prováveis resultados obtidos com a efetiva implementação do mesmo;
- Resultados do processo: podem ser evidenciados por um artefato produzido, uma mudança significativa de estado e um atendimento às especificações;
- Informações adicionais: são referências que podem ajudar na definição do processo pela organização.

Os processos são agrupados, por uma questão de sistematização, de acordo com o seu objetivo principal no ciclo de vida de software. Esse agrupamento caracteriza em três diferentes classes de processos, que são, segundo MPS (2007):

- Fundamental: atendem o início e a execução do desenvolvimento, operação ou manutenção dos produtos de software e serviços correlatos durante o ciclo de vida de software;
- De Apoio: auxiliam um outro processo e contribuem para o sucesso e qualidade do projeto de software;
- Organizacional: uma organização pode empregar estes processos em nível corporativo para estabelecer, implementar e melhorar um processo do ciclo de vida.

A figura 12 apresenta as classes de processos fundamentais, organizacionais e de apoio do MPS.BR, destacando os pontos fundamentais a serem considerados no desenvolvimento de software.

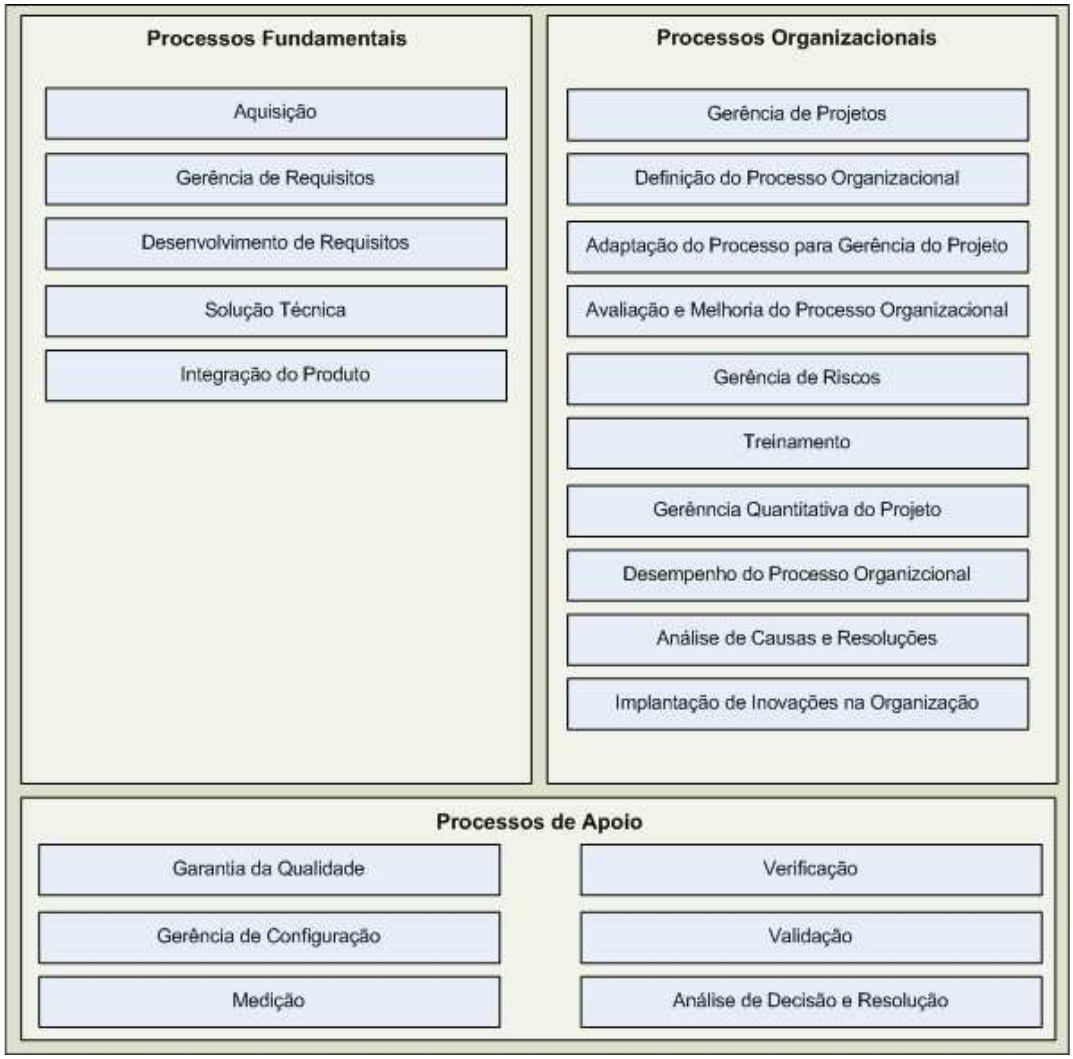

Figura 12. Classes de Processos do MPS.Br. MPS (2006).

2.6.5 Comparativo CMMi e MPS.Br

Feitas as apresentações individuais dos diferentes modelos de processo destacando suas características e especificidades, cabe explicitar comparativamente suas vantagens e limitações.

O MPS.Br possui uma série de diferenciais (ou vantagens) se relacionado com outros modelos de processo. Segundo Couto (2007) alguns desses diferenciais são:

- Os sete níveis de maturidade do MPS.Br permitem uma implementação mais gradual, adequada à micro, pequena e média empresa, além de aumentar a visibilidade do processo de melhoria;
- O MPS.Br possui compatibilidade com CMMI e SPICE (*Software Process Inprovement and Capability dEtermination*), que são modelos reconhecidos internacionalmente;
- O modelo MPS.Br foi criado pensando na realidade da empresa brasileira, com foco na micro, pequena e média empresa de *software*;
- Permite sua aplicação a custos acessíveis (em R$) adequados a realidade brasileira.
- A qualificação da empresa que o aplica é mantida por avaliações periódicas (de dois em dois anos).

A comparação do MPS.Br com o CMMi é também interessante, pois o CMMi é um dos modelos de maturidade mais importantes, reconhecido internacionalmente. Fica evidente que o MPS.Br não tem como objetivo a criação de algo novo no que se refere às normas e modelos de maturidade, mas sim uma adaptação inovadora dos modelos existentes para a realidade das empresas brasileiras. Com isso, pode-se dizer que a adequação da empresa às normas de maturidade descritas no MR-MPS pode ser vista, em alguns casos, como uma preparação para uma avaliação posterior do CMMi.

Cabe lembrar que o CMMi possui cinco níveis de maturidade designados pelos números de 1 a 5, diferente do MR-MPS que possui sete níveis designados pelas letras de A a G. Os níveis de maturidade do CMMi possuem uma relação com os níveis de maturidade do MR-MPS, conforme apresentado na figura 13, a seguir:

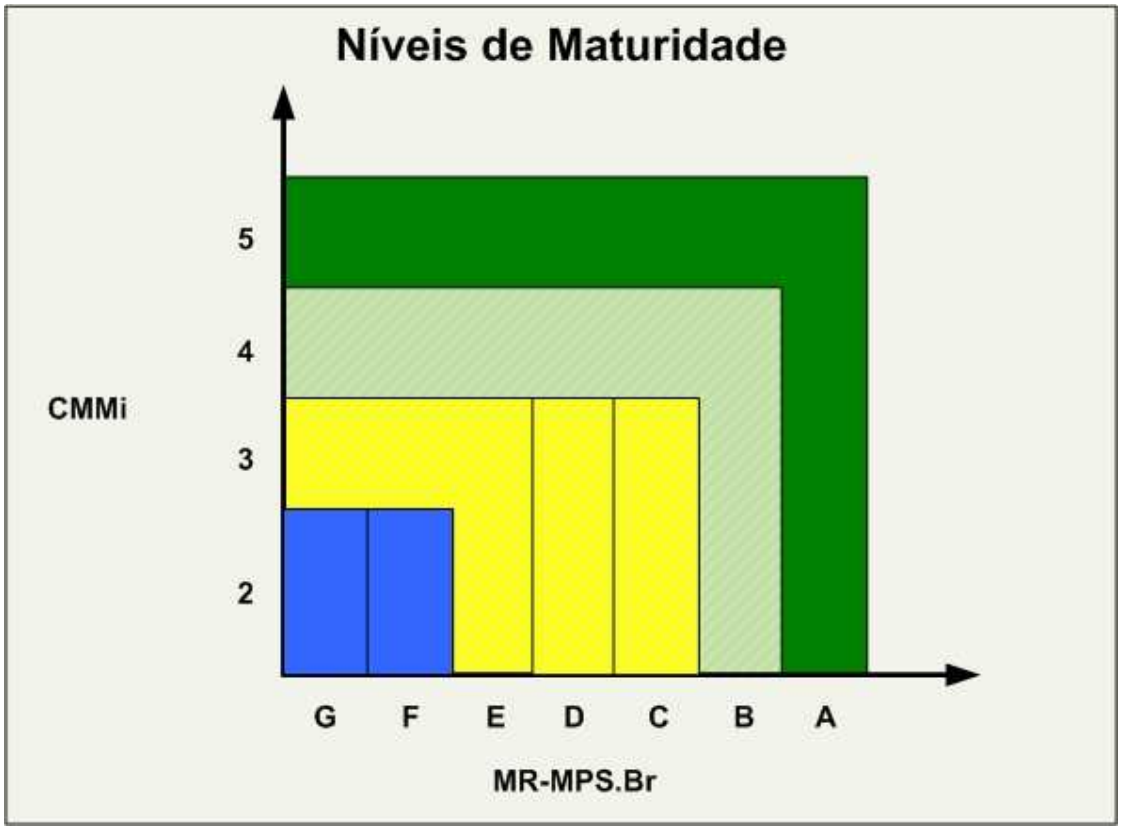

Figura 13: Níveis de Maturidade: CMMI x MR-MPS. Adaptado de PINTO (2006).

Pode-se dizer que no MR-MPS a divisão dos níveis de maturidade permite uma implementação mais gradual e adequada às micros, pequenas e médias empresas brasileiras. A possibilidade de se realizarem avaliações, considerando um número maior de níveis permite uma visibilidade dos resultados de melhoria de processo, na empresa e no país, com prazos mais curtos, COUTO (2007).

Como se observa, não há critérios conflitantes nos diferentes modelos, apenas adaptações estabelecidas por força das especificidades do mercado brasileiro.

3. CARACTERÍSTICAS DA INDÚSTRIA DE SOFTWARE

Este capítulo tem a finalidade de avaliar pequenas empresas desenvolvedoras de software, com o objetivo de possibilitar melhor entendimento das efetivas condições pelas quais essas empresas, de forma geral, realizam os processos de produção de software. Para tanto, procedeu-se à uma revisão bibliográfica sobre as características da indústria de software no Brasil e sua evolução e tendências, de modo a permitir a compreensão do macro e do microambiente que circunstanciam a atuação do setor.

Para a realização desta avaliação, procedeu-se a um levantamento amplo dos mais destacados critérios de avaliação, de modo a se identificarem suas efetivas condições materiais e intelectuais de produção, seu ambiente de atuação e suas práticas, diante de suas potencialidades e limitações. Para se ter um balizador nacional, foi realizada uma comparação com os resultados de pesquisas do MCT, estabelecido como um referencial macro, de modo a permitir comparação de pontos comuns ou discrepantes, sobre a efetividade de atuação da amostra considerada. Fez-se também um estudo da evolução do setor de software no país, com o intuito de melhor compreender as características das empresas no Brasil.

3.1 Setor de Software no Brasil

O Brasil possui 7.818 empresas de software, dedicadas ao desenvolvimento, produção e distribuição de software e prestação de serviços. Daquelas que atuam no desenvolvimento e produção de software, 94% das empresas são classificadas como micro e pequenas, em um total de 1.894. ABES (2007).

Os gráficos 1 e 2, apresentam um panorama da divisão das empresas por porte e por tipo de atividade respectivamente:

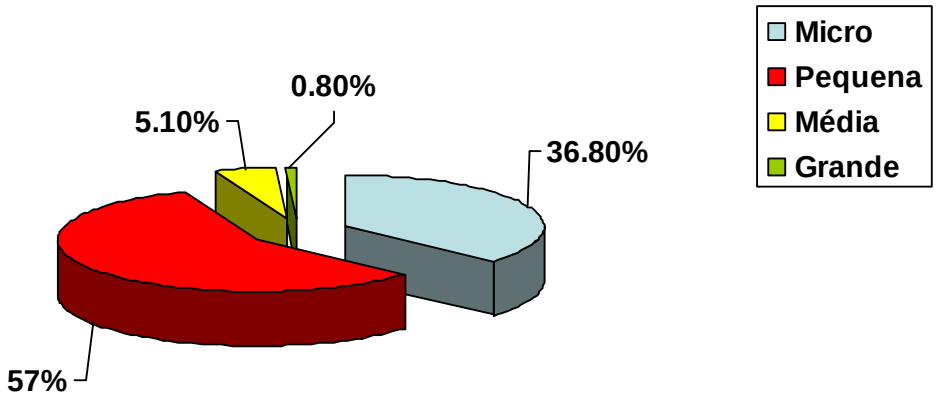

Gráfico 1: Divisão por porte da empresa. ABES (2007).

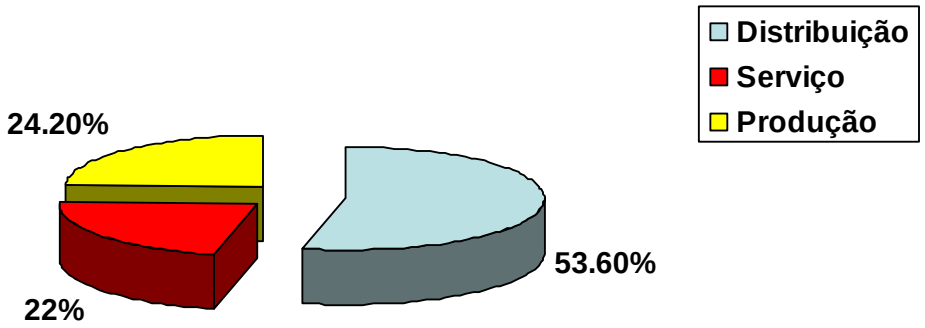

Gráfico 2: Divisão das empresas por tipo de atividade. ABES (2007).

A caracterização do tamanho da empresa é estabelecida pelo Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e pelo Instituto Brasileiro de Geografia e Estatística (IBGE), segundo critérios diferentes. São consideradas micro e pequenas empresas aquelas que, de acordo com o SEBRAE (1998) apresentam:

- faturamento menor que R$ 1,2 milhões por ano.
- possui menos que 50 empregados.

O BNDES (Banco Nacional de Desenvolvimento Econômico e Social) apresenta definição diferente para classificar as micro e pequenas empresas, baseando-se na receita operacional bruta anual. Sendo:

- até R$ 1,2 milhão - micro empresa.
- superior a R$ 1,2 milhão e inferior ou igual a R$ 10,5 milhões - pequenas empresa.

O SEBRAE utiliza como critério de classificação, o faturamento a partir da comercialização bruta anual das empresas, o que, neste caso, inclui valores provenientes da comercialização de software e de outros produtos e serviços de informática ou não, de acordo com a faixa de valores a que pertencem, consideração que pode mascarar o quantitativo específico da produção de softwares, com outras atividades.

A tabela abaixo apresenta uma comparação entre os valores definidos pelo Sebrae e pelo BNDES quanto a classificação por faturamento.

Tabela 3 - Classificação das empresas segundo o porte

Agência	Porte			
	Micro Empresa	Pequena Empresa	Média Empresa	Grande empresa
SEBRAE	Até 19 empregados. Faturamento anual de R$ 244 mil	Até 99 empregados. Faturamento anual de até R$ 1,2 milhão		
BNDES	Receita operacional bruta anual ou anualizada de até R$ 1,2 milhão	Receita operacional bruta anual ou anualizada superior a R$ 1,2 milhão e inferior ou igual a R$ 10,5 milhões	Receita operacional bruta anual ou anualizada superior a R$ 10,5 milhões e inferior ou igual a R$ 60 milhões	Receita operacional bruta anual ou anualizada superior a R$ 60 milhões.
MCT	de 1 a 9 pessoas.	de 10 a 49 pessoas.	de 50 a 99 pessoas.	100 ou mais pessoas.

A classificação por faturamento transparece algumas inadequações quando aplicadas às empresas de software, visto que existem casos de empresas com um número reduzido de colaboradores mas que apresentam faturamento alto. Este fato, pode significar o alto grau de inovatividade contida nos softwares, o que lhe agrega valor.

Sendo assim, é utilizada neste trabalho, a classificação definida pela MCT (2005) que considera a força de trabalho, quer seja efetiva (sócios, dirigentes e empregados efetivos) quanto total (efetivos mais terceiros prestadores de serviço, bolsistas e estagiários) e adotando-se como critério:

- microempresas (de 1 a 9 pessoas),
- pequenas (de 10 a 49 pessoas),
- médias (de 50 a 99 pessoas) e
- grandes (100 ou mais pessoas).

3.1.1 Análise das Condições de Atuação

Entender as condições de atuação da empresa de software é um dos mais importantes pontos para se compreender as contingências de execução do processo de produção de software, pois representa a maturidade gerencial e o profissionalismo de atuação.

Um aspecto interessante desta análise é compreender que as pequenas empresas representam em seu conjunto, uma significativa porcentagem do faturamento do setor e empregam um número elevado da mão de obra, MCT (2005), mas que possuem como característica marcante o fato de não apresentarem um processo de desenvolvimento de software maduro. Essa dificuldade em amadurecer o processo de desenvolvimento, tem conseqüências no faturamento, na sua própria competitividade, na qualidade dos produtos

gerados, entre outros fatores. Pode-se dizer que há um interrelacionamento entre os fatores: competência, incorporação tecnológica, nível de soluções oferecidas e faturamento.

3.1.1.1 Características gerais de atuação

Estudos qualitativos de caráter exploratório destacam que, as características marcantes das pequenas empresas desenvolvedoras de software no Brasil são derivados de um processo de software imaturo e inconcluso. Entre essas características se destacam, segundo FERNADES (2004):

- Informalidade: não existem processos previamente definidos e adotadas pela empresa. A produção de software é fortemente dependente de prática individual, baseada na formação e capacidade dos desenvolvedores;
- Não há ambiente estável de desenvolvimento: nada novo projeto demanda o emprego de toda a atenção e competência no seu tratamento, desde as etapas de definição de requisitos até a entrega do produto. Assim, sucessos anteriores não se repetem, de forma sistematizada;
- O sucesso se deve à competência: o processo é fortemente marcado pelo esforço dos desenvolvedores, em atos que beiram ao heroísmo, pelos esforços despendidos;
- Dependência de pessoas específicas: o que resulta em queda acentuada de produtividade ou mesmo incapacidade de realizar alguma tarefa quando certas pessoas deixam a empresa, pois grande parte do conhecimento pode ser “não estruturado”, vai com quem o detém.
- Incapacidade de controle do processo: o que ocasiona na constante verificação de problemas nos cronogramas ou orçamentos estabelecidos, resultando em atrasos no desenvolvimento ou na sobrecarga das pessoas para evitar prováveis atrasos;

- Falta de Comprometimento com Qualidade: existe a preocupação com práticas voltadas à garantia de qualidade dos produtos, porém, freqüentemente ocorre o abandono dessas práticas em momentos de pressão e *stress*.

Em questões relacionadas ao tratamento dos requisitos de software, tem-se:

- Definição de requisitos de software de maneira informal, apenas por meio de contatos pessoais (reuniões), as vezes sem registro, gerando futuras discórdias entre desenvolvedores e clientes;
- Desconhecimento ou não aplicação de técnicas de Engenharia de Requisitos;
- Procedimentos informais de validação dos requisitos sem um comprometimento formal do cliente;
- Alterações dos requisitos de uma maneira informal, às vezes até através de contatos telefônicos, sem avaliação de efeito sobre custos ou cronograma;
- Alterações efetuadas por pessoas não qualificadas ou autorizadas a fazê-lo;
- Falta de atenção aos requisitos ao longo do desenvolvimento, levando à frustrações do cliente com requisitos não considerados ou mal interpretados;

Tais condicionantes oneram qualitativa e quantitativamente as empresas por afetarem sua imagem de competência, ás vezes de maneira irrecuperável.

Nos processos de Gerenciamento do Desenvolvimento, ficam evidentes os seguintes pontos, conseqüência direta da falta de controle sobre as operações:

- Falta de abordagens metodológicas bem como de dados históricos para estimativas de custo e esforço;
- Cronograma e orçamento com elevado nível de incerteza;
- Falta de argumentos objetivos (números) para respaldar uma negociação de cronograma e orçamento de clientes;

- Execução e planejamento, sem a correspondente medição periódica ao longo do desenvolvimento, resultando em perda do controle do estado do desenvolvimento;
- Falta de capacidade de acompanhar o progresso de um desenvolvimento de forma objetiva (quantitativa);
- Falta de controle de responsabilidades mútuas, levando a empresa a arcar com atrasos que não são seus, como por exemplo, quando um cliente atrasa uma avaliação prevista no cronograma;
- Falta de preocupação com gerenciamento de riscos, que são percebidos apenas quando se materializam em forma de problemas reais;
- Falta de capacidade de antecipação de possíveis atrasos, levando a constatação do problema às vésperas do encerramento de prazos, quando mudanças de planejamento são inviáveis;
- Não aplicação de ações corretivas no momento adequado, quando o planejado não se efetiva.
- Impossibilidade de se estabelecer *landmakers*, ou indicadores de desempenho para verificação de progresso ao longo dos prazos estabelecidos.

No aspecto preservação de dados, constata-se forte dispersão causada por:

- Perda de versões anteriores de um software;
- Armazenamento dos produtos (versão anteriores, inclusive) dependente de disciplina individual;
- Alterações bem intencionadas e não discutidas levando ao caos;
- Indefinição de quem pode alterar documentos e produtos e de como as alterações são definidas e avaliadas;
- Falta de controle de mudanças;

- Dificuldade para desfazer uma mudança após contatar resultado insatisfatório.

Assim sendo, as experiências e os resultados anteriores, não sendo sistematicamente documentados, não permitem criar um histórico e um referencial de aprendizagem, contido numa base de dados produtiva.

A observação desta circunstância de atuação das empresas e a experiência desenvolvida ao longo do tempo, ensejaram um aprofundamento no tema, merecendo esta pesquisa.

Com o intuito de entender com maior precisão as características e problemas enfrentados pelas empresas, foi realizada uma pesquisa de campo em cinco empresas, todas localizadas no interior do estado de São Paulo, com o objetivo de avaliar a real condição sob as quais elas desenvolvem seus produtos.

A seguir, se apresentam as considerações sobre a pesquisa de campo, assim como o processo e método de avaliação escolhido, buscando-se correlacionar o contexto teórico com a prática reinante e suas circunstâncias.

3.2 Avaliação do processo de desenvolvimento de software

De início cabe destacar que qualquer processo de desenvolvimento de software apresenta um significativo grau de dificuldade, principalmente por se tratar da concepção e materialização de uma solução de alto conteúdo abstrato e não físico, onde questões subjetivas e não estruturadas têm forte participação. Mesmo assim, uma aproximação com as práticas de organização sistêmica e de planejamento estratégico podem representar uma significativa contribuição à melhoria dos procedimentos de gestão neste tipo de atuação. Como exemplo,

pode-se recorrer ao planejamento estratégico de Porter (2004), para análise dos pontos fortes, pontos fracos, ameaças e oportunidades (PFOA); ou ao Balanced Scorecard – BSC de Kaplan e Norton (1997) que permite estabelecer o planejamento estratégico, de início, sob quatro perspectivas: a econômica (geração de renda e redução de custos); a do cliente (na pré-venda, venda e pós-venda); a dos processos internos que dão suporte à atividade principal da empresa, no caso, a produção e desenvolvimento de software e a aprendizagem e inovação.

O entendimento **do que** uma empresa faz e **como** ela faz é um passo importante para traçar uma tentativa de gerenciamento, planejamento ou melhoria das atividades.

Segundo Wangenheim (2005), as análises são feitas por meio de uma avaliação de processo, que é um exame disciplinado das atividades usadas por uma organização, baseada em um modelo de referência com o objetivo de determinar a capacidade destes processos e a maturidade da organização.

Tanto o CMMi, o MPS.Br e a ISO/IEC 12207, possuem métodos próprios e normas para avaliação de processo de software, porém apresentam limitações para uma avaliação de contextualização.

No processo de melhoria como um todo, existem basicamente três tipos de avaliações, que podem ser usadas para avaliar o processo, motivar equipes, direcionar estratégias e determinar um estado em comparação a um modelo ou norma, ou formar uma linha base para acompanhar a melhoria. SALVIANO (2004).

Os principais tipos de avaliação incluem:

- <u>Avaliações de contextualização:</u> ponto inicial no processo de melhoria onde gera-se um entendimento da sua situação atual considerando os principais pontos fortes e fracos.

- Avaliações detalhadas: são realizadas para avaliar em mais detalhes os processos importantes e críticos.
- Avaliações contínuas: podem ser integradas com programas de medição existentes para monitorar os processos selecionados durante e depois da implantação das ações de melhoria.

3.2.1 Procedimentos de avaliação

Os procedimentos de avaliação encontram dificuldades em sua realização ocasionadas entre outras, pela rapidez da inovação, pela especificidade de cada projeto e por decorrência das mudanças estruturais da própria empresa.

Comparando-se por exemplo, com a indústria metal-mecânica, que atua sob encomenda, na execução de produtos unitários, repetem-se as dificuldades semelhantes:

- Prazos dilatados de concepção e desenvolvimento;
- Necessidade de se contar com equipes com conhecimento intensivo e competências de planejamento, concepção, desenvolvimento, testes e consolidação de procedimentos;
- Grande possibilidade de migração da empresa desenvolvimentista do software para a empresa aplicadora (*Spin off* de profissionais);
- Dificuldade de se estabelecer ganhos de escala, advindos da padronização, da repetitividade ou por analogia na aprendizagem do processo, com casos semelhantes, anteriormente tratados;
- Dificuldade de se estabelecerem padrões de produção e portanto, controles de produção efetivos, o que se agrava, pela imaterialidade e subjetividade na criação de soluções.

Um ponto importante a ser destacado neste caso, refere-se ao fato das dificuldades em se realizar tais avaliações. Principalmente por se tratar de um produto tecnológico que sofre mudanças rápidas, além da especificidade de cada projeto e do aspecto gerencial onde mudanças contínuas e rápidas na estrutura da empresa afetam o processo de desenvolvimento.

3.2.2 Considerações Gerais

Feitas estas considerações, nascidas da análise das circunstâncias de atuação das empresas, fortalece-se a idéia de se aprofundar no estudo do assunto. Sendo o objetivo deste trabalho identificar a situação atual de algumas empresas com relação ao processo de desenvolvimento, a avaliação de contextualização foi usada no primeiro momento, por permitir um rápido diagnóstico das condições de atuação da organização.

3.2.3 Métodos de Avaliação

Método de avaliação de processo de desenvolvimento de software é um exame disciplinado dos processos usados por uma organização baseado em um modelo de referência com o objetivo de determinar a capacidade destes processos e/ou maturidade de uma organização, segundo Wangenheim (2005).

Paulk *et al.* (1995) salientam que um método de avaliação fornece a estrutura básica para a investigação do processo e permite um rápido e consistente desenvolvimento das constatações que apontam os pontos fortes e fracos da organização.

A elaboração de métodos de avaliação não é algo recente, tendo seu início em meados da década de 70 quando pesquisadores da IBM, preocupados com atrasos nos cronogramas, falta de controle de versões entre outros problemas criaram mecanismos para avaliar o

processo e futuramente melhorá-los. Desde então, dezenas de métodos foram criados, cada qual com suas especificações e característica.

Apresenta-se a seguir, de maneira sucinta, alguns métodos aplicados mundialmente e que foram adotados como referência nesta pesquisa.

3.2.3.1 QuickLocus

Criado em 2003 por pesquisadores do Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, trata-se de um método de avaliação voltado para pequenas e médias empresas.

Segundo Kohan (2003), destina-se às organizações de software com até cinqüenta pessoas, utilizando um tempo reduzido de aplicação e cujo resultado fornece a base para um plano de melhoria do processo de desenvolvimento. A aplicação do QuickLocus consome, em geral, apenas um dia de trabalho na organização, quando é avaliado um escopo reduzido do modelo de referência.

Este método permite estabelecer a escolha do modelo de avaliação que será utilizado, como o CMMi ou a ISO/IEC 15504.

3.2.3.2 RAPID

Criado na Austrália o método RAPID - *Rapid Assessment for Process Improvement for Software Development* teve como motivação atender as pequenas e médias organizações que não possuiam recursos para custear as avaliações mais detalhadas que duravam cerca de quatro dias.

Assim como os demais, o método é compatível com a ISO/IEC 15504, tornando possível que o resultado de suas avaliações possam ser comparados com o resultado de outras organizações que o utilizam.

Tem como característica a necessidade de avaliadores qualificados para sua aplicação, sendo considerado um bom método por fornecer subsídios para melhoria de processo.

3.2.3.3 MARES

Método criado por pesquisadores do Laboratório de Qualidade e Produtividade de Software da UNIVALI (Universidade do Vale do Itajaí), com objetivo de desenvolver um modelo e método de avaliação de processo de software integrando o CMMi-SW e a norma ISO/IEC 15504 em uma forma adaptada às características das micros e pequenas empresas brasileiras de software visando posterior melhoria de processo.

De acordo com Anacleto (2004), este método se diferencia dos demais, principalmente, pela definição de uma estrutura que auxilia na caracterização da organização e seleção dos processos relevantes, passíveis de serem melhorados no contexto específico.

Este método encontra-se disponível publicamente, incluindo uma descrição detalhada de todas as atividades da avaliação com guias para sua adaptação a contextos específicos. Outra característica positiva é que o método não exige conhecimento específico dos representantes das empresas participantes sobre avaliação de processos e sobre o modelo de referência utilizado, o que facilita sua aplicação.

Segundo Wangenheim (2006) este método se destaca justamente por ser voltado a pequenas empresas que possuem vários fatores limitantes quanto a utilização de outros métodos.

3.2.3.4 MA-MPS

Este método permite a realização de avaliações em empresas seguindo o modelo MPS.Br (melhoria do processo de software brasileiro). A figura 14, a seguir, destaca o método de avaliação do MPS.Br dentro do modelo.

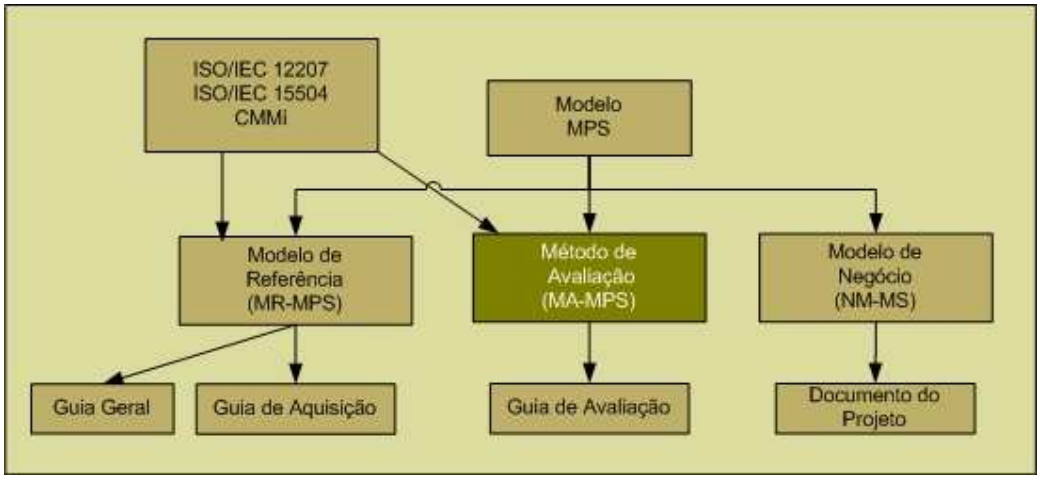

Figura 14. O modelo MPS.Br. WEBER (2005).

Os requisitos do MA-MPS são baseados nos requisitos dos métodos de avaliação definidos na norma ISO/IEC 15504, os requisitos definidos no CMMi e os próprios requisitos definidos pelo MPS.Br.

Conforme apresentado em MA-MPS (2007) é um método avaliativo voltado para pequenas e médias empresas e se destaca por possuir facilitadores que garantem seu uso de forma acessível a essas organizações.

A avaliação utilizando este método exige um líder e membros da equipe com certificação do modelo.

3.2.3.5 FAME

Este é um método de avaliação unificado, desenvolvido na Alemanha e que infelizmente oferece pouca referência disponível. Seu foco não é exclusivamente voltado às pequenas empresas, podendo ser aplicável nas demais. Suporta avaliações tanto com o objetivo de melhoria, quanto determinação da capacidade. O método auxilia determinar os pontos fortes e fracos dos processos de software atuais e auxilia na tomada de decisões para a melhoria de processo. Este método utiliza o modelo de avaliação padrão da ISO/IEC TR 15504.

3.2.3.6 TOPS

O projeto TOPS (*Toward Organised Process em SMEs – small and medium enterprises*) foi formado pela Comunidade Européia com o propósito de difundir as melhores práticas de software e, particularmente, propor caminhos de melhoria para empresas regionais da Itália.

Desde o início do projeto foi identificado que a melhor maneira de incentivar programas de melhoria é contatar diretamente a empresa e executar uma avaliação rápida de processo de software para identificar e planejar ações específicas. O projeto desenvolveu um método de avaliação rápida, adaptando o SPICE (ISO/IEC TR 15504) e oferecendo a avaliação, sem custos, para as empresas que se integrassem a ele. O programa de avaliações teve como objetivos, estimular interesse em avaliações de processo de software e suas melhorias, contribuir na definição de planos de melhoria, coletar dados e estatísticas sobre melhorias de processo de software, enfim estabelecer uma referência para a contínua otimização.

3.2.3.7 ISO/IEC 15504

A ISO, em 1992, destacou a necessidade de se elaborar uma norma que fosse aplicável à melhoria de processos e à determinação da capacidade. Este padrão deveria considerar os métodos e normas já existentes (como por exemplo, o SW-CMM e a ISO 9001), abranger todos os processos de software e ser construído pelos especialistas que já desenvolviam e trabalhavam com os métodos e normas existentes. Como resultado desse primeiro trabalho, a ISO iniciou em janeiro de 1993, o projeto SPICE (*Software Process Improvement and Capability dEtermination*) com o objetivo de produzir inicialmente um relatório técnico que fosse, ao mesmo tempo, mais geral e abrangente que os modelos existentes e mais específico que a norma ISO 9001 originando assim a norma ISO/IEC 15504.

Esta norma se presta à realização de avaliações de processos de software com dois objetivos:

- melhoria de processos;
- determinação da capacidade de processos de uma unidade organizacional.

Se o objetivo for a melhoria de processos, a unidade organizacional pode realizar uma avaliação com o objetivo de gerar um perfil dos processos que será usado para a elaboração de um plano de melhorias. A análise dos resultados identifica os pontos fortes, os pontos fracos e os riscos inerentes aos processos. No segundo caso, a organização tem como meta avaliar um fornecedor, obtendo o seu perfil de capacidade. O perfil de capacidade permite ao comprador estimar o risco associado à contratação daquele fornecedor para auxiliar na tomada de decisão de contratá-lo ou não. Esta análise de risco permite alimentar o processo de decisão de se contratar ou não determinada empresa. Cabe lembrar que o insucesso na aquisição de um

software extrapola os custos de aquisição em si e repercute em perdas às vezes irreparáveis, decorrentes da não disponibilidade no tempo certo, de uma ferramenta estratégica.

3.3 Procedimento de avaliação de processo

A avaliação do processo de desenvolvimento em uma organização pode variar de acordo com o método escolhido. Normalmente, são escolhidos avaliadores capacitados, um modelo de avaliação compatível e um processo de avaliação.

Para a realização da pesquisa de campo deste trabalho foi escolhido o método MARES. O fator principal que justifica a escolha deste método é o fato do método apresentar uma avaliação de contextualização facilmente aplicável e que apresenta resultados satisfatórios, sendo que o mesmo já foi aplicado em centenas de empresas brasileiras. Os outros métodos destacados nesta pesquisa estão voltados para avaliações detalhadas ou avaliações contínuas e na maioria dos casos exige pessoal certificado para aplicar a pesquisa.

Outros fatores que justificam a escolha desse método de avaliação são:

- Baixo custo;
- Rapidez na aplicação;
- Obtenção de resultados confiáveis;
- Casos relatados de aplicações bem sucedidas já realizadas com o método;
- Dados obtidos servem de base para o processo de melhoria;
- Fácil acesso aos questionários e material.

3.3.1 Processo de avaliação

Uma avaliação deve ser conduzida de acordo com um processo documentado, capaz de atender aos propósitos previamente estabelecidos. O processo documentado de avaliação tem que conter as atividades de planejamento, coleta de dados, validação dos dados e a apresentação dos resultados. Segue-se o detalhamento de cada um destes requisitos:

Planejamento: o planejamento do procedimento de avaliação deve ser desenvolvido em função da empresa em estudo e deve estar contido num plano, devidamente documentado, de modo a incluir os seguintes elementos: entradas requeridas, atividades a serem executadas, recursos e cronograma, identidade e as responsabilidades definidas dos participantes da avaliação, critérios de ponderação dos fatores quanto a sua importância relativa, descrição dos resultados esperados e considerações relevantes;

Coleta de Dados: os dados requeridos para a avaliação dos processos dentro do escopo da avaliação e informações adicionais devem ser coletados de forma sistemática, considerando-se no mínimo, um estudo prévio sobre estratégia e técnica para a seleção, coleta, análise e justificativa para atribuição de notas; estabelecimento de correspondência entre os processos das unidades organizacionais e os elementos do modelo de avaliação de processo.

Validação dos Dados: os dados coletados necessitam ser validados para confirmar se as evidências coletadas são objetivas, para garantir que as mesmas são suficientes e representativas para cobrir o escopo e o propósito da avaliação e para garantir que os dados coletados, como um todo, sejam consistentes.

Apresentação dos Resultados: os resultados da avaliação, incluindo no mínimo as saídas especificadas, têm que ser documentados e comunicados ao patrocinador da avaliação ou ao seu representante. Os resultados devem conter no mínimo, os seguintes elementos: data da avaliação, as entradas da avaliação, a identificação das evidências objetivas utilizadas, a

abordagem utilizada para a avaliação, a identificação do processo documentado de avaliação e o conjunto de perfis de processo resultante da avaliação, sendo um perfil de processo para cada processo avaliado. A figura 15, detalha as etapas do processo de avaliação.

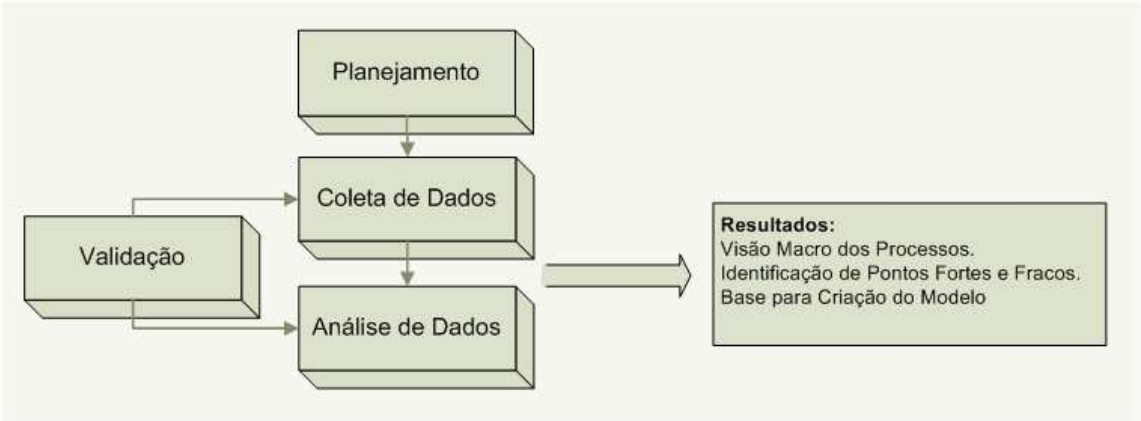

Figura 15. Etapas do processo de avaliação.

Para este trabalho foi primeiramente idealizada, a elaboração de um método de avaliação próprio, porém com o avanço dos estudos, pôde-se constatar a existência de métodos já consagrados que atenderiam aos objetivos previamente estabelecidos. O método MARES foi escolhido, por comparação entre alternativas.

Escolhido o método, iniciou-se a etapa de planejamento, com o contato com empresas conhecidas, seis foram selecionadas. Esse contato aconteceu inicialmente por telefone e por email. Cabe destacar que uma das empresas teve seus resultados posteriormente desconsiderados por apresentar dados não relevantes e, por ter encerrado suas atividades. Nesta etapa buscou-se fazer um levantamento prévio da empresa através do site, para um rápido entendimento das suas atividades.

A etapa de coleta de dados durou 3 meses, entre as entrevistas e as visitas. Cada entrevista teve duração de 1 dia. As dificuldades dessa etapa ficaram relacionadas ao

agendamento das entrevistas com representantes da empresa, e a busca por dados entendidos como confidenciais que foi solucionada com a assinatura de um acordo de confidencialidade.

Com os questionários preenchidos, iniciou-se a análise dos dados. Esta etapa levou 16 dias para ser concluída e teve como objetivo, identificar as práticas de cada empresa com relação aos processos de desenvolvimento, seus pontos fortes e fracos, seguindo o modelo de Porter (2004).

A etapa de validação foi realizada em conjunto com as empresas para garantir os dados coletados e a posterior análise. Aproveitou-se nesta etapa para, como forma de contribuição à empresa, realizar uma apresentação aos colaboradores mostrando os pontos fortes e fracos levantados na pesquisa e discutindo-se melhorias aplicáveis, ocorrendo uma interação amigável entre entrevistador e entrevistados.

Os resultados da análise dessa pesquisa de campo são apresentados a seguir.

3.4 Resultados da pesquisa

3.4.1 Sobre as empresas

- **EMPRESA A**

Trata-se de uma empresa com sete anos no mercado que iniciou suas atividades desenvolvendo portais corporativos, sendo ganhadora inclusive de prêmios nacionais pelos trabalhos realizados. Passou a desenvolver sistemas sob-encomenda e atualmente tem um foco em desenvolvimento de software *web* para o setor de logística. Em 2005 realizou a exportação de seus produtos para países da América do Sul, fato que exigiu mudanças no processo de desenvolvimento, de modo a atender aos padrões de qualidade exigidos pelo cliente.

Quanto ao desenvolvimento de software a empresa apresentava algumas práticas positivas como métricas para estimativas e outras boas práticas de gestão de projetos e codificação, porém, não de maneira institucionalizada. Revelou falhas nas atividades de levantamento de requisitos e testes, já que essas tarefas não possuem padrão de documentação e o gerenciamento de documentação, quando feito, é falho. Cada projeto segue um processo diferenciado perdendo-se boas práticas já realizadas em projetos anteriores e os correspondentes ganhos de aprendizagem.

A empresa conta com o apoio de uma consultoria para criar o planejamento estratégico da organização, evidenciando o interesse pela melhoria de seus processos visando crescimento nas exportações.

Ramo	Portais Corporativos e Logística
Número de colaboradores	7
Localização	Jundiaí - SP
Ano da Fundação	2000
Entrevistado (Cargo)	Proprietário / Gerente de Projeto

Quadro 4. Quadro resumo da empresa A.

- **EMPRESA B**

Empresa familiar com foco em desenvolvimento de software para instituições educacionais, tem entre seu produto-chave um ERP (*Enterprise Resource Planning*) para gestão escolar. O ponto chave da organização é o grau de conhecimento dos sócios quanto a tecnologias de informação e suas correlações.

Não existe em sua atuação, processo definido de desenvolvimento e apresenta poucas práticas de gestão de projetos, o que acarreta problemas principalmente em gerenciar requisitos. Pelo pequeno número de colaboradores existe uma sobreposição de funções e certa negligência em atividades-chave.

A empresa possui dezenas de clientes e o foco maior está na customização e manutenção de produtos, porém não existe um processo padrão definido o que acarreta, inclusive a dificuldade de reutilização de código-fonte.

Apresenta necessidade de formalizar seu processo de desenvolvimento em todos os aspectos.

Ramo	Software Gestão Educacional
Número de colaboradores	3
Localização	Tambaú – SP
Ano da Fundação	2004
Entrevistado (Cargo)	Sócio da Empresa

Quadro 5. Quadro resumo da empresa B.

- **EMPRESA C**

Como foco voltado para projetos agropecuários, a empresa se destaca por desenvolver software com alto grau de inovação e pioneirismo, a partir de financiamento de órgãos de fomento à pesquisa.

A organização possui um organograma bem definido, porém em momentos críticos, as atribuições são deixadas de lado para cumprimento de metas e prazos prementes. Existe portanto um conflito entre a estrutura formal, contida no organograma e a efetividade dos

níveis decisórios, que diante de situações-problema, são centralizados em uma ou duas pessoas. Existe um processo de desenvolvimento definido e documentado, todavia sem muito detalhamento e com falta de padrões de documentação. O padrão de codificação existe, porém não ocorre um processo de verificação de seu cumprimento. O processo definido pela empresa necessita ser adaptado a uma realidade mais voltada ao mercado, com ganhos de produtividade.

A duração dos projetos é de aproximadamente 6 meses, sem muito controle do cronograma e de estimativas, e o gerenciamento correto depende direta e fortemente da experiência dos gerentes.

Ramo	Software Agropecuário
Número de colaboradores	16
Localização	São Carlos - SP
Ano da Fundação	2003
Entrevistado (Cargo)	Gerência de Projeto

Quadro 6. Quadro resumo da empresa C.

- **EMPRESA D**

Empresa pioneira em algumas tecnologias, possui significativos projetos com clientes de grande porte, no setor de automação tendo foco no desenvolvimento de hardware. Como ponto favorável tem-se a gerência de projetos bem definida com uso de softwares e métodos declarados, bem como na padronização de código-fonte, seguida por todos. Por ter uma forte propensão ao uso de software livre em seus produtos, a empresa se utiliza também para

atividades internas, de várias ferramentas de código aberto, o que agrega valor e facilita o cumprimento de tarefas, como trabalho colaborativo, controle de versão, entre outros.

Esta empresas, não apresenta um programa definido de qualidade e várias atividades são dependentes de esforço individual de colaboradores que na maioria dos projetos desempenham papéis não previamente definidos. A documentação dos projetos é falha, o que fica evidente na falta de padrão dos documentos gerados. Não existe atualmente um processo declarado de desenvolvimento de software.

Ramo	Automação e software livre
Número de colaboradores	13
Localização	São Carlos - SP
Data da Fundação	2003
Entrevistado (Cargo)	Diretor de Tecnologia

Quadro 7. Quadro resumo da empresa D.

- **EMPRESA E**

Empresa voltada para sistemas de automação comercial e industrial, possuindo 14 produtos em seu portfólio. Apresenta a maioria dos funcionários lotados nos processo de suporte, que não são padronizados, ocasionando problemas na manutenção.

Nenhum documento técnico é gerado durante o projeto, apenas documentos contratuais. Esse fato remete a empresa a problemas sérios de rastreabilidade de requisitos e de configuração de software toda vez que novas funcionalidades são exigidas.

Outro ponto negativo detectado foi o fato de não existir qualquer tipo de controle de versão ou gerência de configuração.

Como ponto positivo se destaca o fato da empresa atuar em um mercado competitivo dominar o mercado em um raio de 130 km, de sua sede.

Ramo	Automação Comercial e Industrial
Número de colaboradores	12
Localização	Porto Ferreira - SP
Data da Fundação	2000
Entrevistado (Cargo)	Proprietário / Gerente de Projetos

Quadro 8. Quadro resumo da empresa E.

3.4.2 Análise

De acordo com a pesquisa de campo realizada, pode-se constatar que todas as cinco empresas analisadas são caracterizadas como pequena empresa, uma vez que possuem menos de 49 colaboradores. Elas apresentam algum tipo de processo de desenvolvimento, porém não formalizado e documentado, o que acarreta deseconomias e perdas de produtividade em termos de cumprimento das metas e de uma definição de documentos e padrões a serem seguidos. Outro fator é que, não operando como processos institucionalizados, em momentos de crise tende-se a abandonar o planejado, como recorrência compensatória.

Em quase todas as organizações pesquisadas existe algum tipo de preocupação com qualidade como, por exemplo na documentação de especificações, na padronização de código e de testes, entretanto, a falta de um modelo, acarreta nestas empresas a impossibilidade de repetir boas práticas já adquiridas em outros projetos e de se beneficiar com os ganhos da

aprendizagem, ou seja, não se verificarem crescimentos ao longo das curvas de aprendizagem decorrentes dos processos.

Outra característica marcante detectada na pesquisa foi a constatação de que todas as empresas já enfrentaram problemas pela não formalização de documentos, como plano de projeto ou documento de requisitos, inclusive com descontrole de versões de ativos (softwares e documentos). Nas fases inicias estes procedimentos informais levam a validação dos requisitos sem um comprometimento do cliente, dificultando inclusive as alterações dos requisitos, pois não há meios de rastreá-los

A falta de abordagens metodológicas bem como de dados históricos para estimativas de custo e esforço torna cada novo projeto a um elevado nível de incerteza quanto a cronograma e orçamento;

Faz-se a seguir uma descrição de outros problemas constatados:

- Desconhecimento e conseqüente falta de uso de métodos para análise de riscos.
- Inexistência de um ambiente estável de desenvolvimento.
- Dependência marcante de colaboradores chave.
- Execução e planejamento, sem a correspondente medição periódica ao longo do desenvolvimento, resultando em perda do controle do estado do desenvolvimento.
- Falta de capacidade de acompanhar o progresso do desenvolvimento de forma objetiva (quantitativa);
- Falta de planejamento estratégico definido para alinhamento e direcionamento dos projetos a serem iniciados.
- Falta de capacidade de antecipação de possíveis atrasos,
- Alterações bem intencionadas e não discutidas levando ao caos;

Diante deste diagnóstico, constata-se, uma forte necessidade de se estabelecer um modelo de desenvolvimento de software aplicável às pequenas empresas com características semelhantes, de modo a melhorar seu processo e conseqüentemente manter-se competitivamente no mercado.

4. PROPOSTA DE UM MODELO DE REFERÊNCIA

Pelas razões já apresentadas, a disponibilização de um modelo de referência para o desenvolvimento de softwares aplicável a pequenas empresas, no caso brasileiro representará uma significativa contribuição do entendimento e domínio do tema. Neste sentido, apresentam-se a seguir alguns conceitos.

4.1 Modelos de referência

Modelos de referência podem ser entendidos como uma representação dos processos de negócio, descrevendo suas atividades, informações, recursos e organização. Tem como intuito ser uma referência no sentido de auxiliar empresas e profissionais a planejar, controlar e gerenciar o desenvolvimento de produto.

Barbalho e Rozenfeld (2004) apresentam uma definição bastante completa acerca do que caracteriza um modelo:

> "*Um modelo é uma representação externa e explícita de parte da realidade vista pela pessoa que desejar usá-lo para apoiar a execução de tarefas relacionadas com aquela parte da realidade, sejam operacionais ou gerenciais, sendo expresso em termos de algum formalismo (linguagem) definido por construtos de modelagem.*"

Segundo Vernadat (1996) modelos de referência são modelos parciais que podem ser usados como base para o desenvolvimento ou a avaliação de modelos particulares. Modelos são chamados de parciais quando eles não são totalmente instanciados, ou seja, não atendem

ao processo existente em uma realidade em particular. São genéricos, normalmente direcionados a um determinado setor da economia. Eles são, portanto, *templates* que podem ser adaptados a condições específicas de uma empresa.

Um modelo de referência deve permitir uma visão de como o processo será realizado, explicitando as atividades e responsabilidades de cada participante. O objetivo do modelo de referência é prover a empresa com uma solução inicial para seus processos de negócios, para que, através dessa, seja especificado e detalhado o modelo particular da empresa.

Curtis *et al.* (1992) descrevem quatro aspectos básicos necessários e pertinentes aos modelos:

- Aspectos de informação: descreve os dados que serão utilizados e produzidos e as relações entre eles;
- Aspectos lógicos e seqüenciais: descreve o comportamento, referindo-se ao **como** e ao **quando**;
- Aspectos organizacionais: os aspectos organizacionais descrevem **quem** serão os responsáveis;
- Aspectos funcionais: descreve **o que** deve ser feito.

De acordo com essas visões entende-se que um modelo de referência pode servir para desenvolvimento de produto nas mais variadas áreas, como alimentícia, de manufatura, de software, entre tantas outras. Segundo Vernadat (1996), as vantagens em se adotar modelos de referência consistem em:

- redução de tempo e custo no desenvolvimento do modelo particular;
- comparação das atividades da empresa com as atividades propostas no modelo (exemplo: melhores práticas);
- melhor suporte na implantação de sistemas de gestão empresarial integrado (ERP).

4.2 O modelo de desenvolvimento de produto

Esta pesquisa teve emprego do modelo de referência de desenvolvimento de produto criado pelo esforço conjunto de grupos de pesquisa de três universidades brasileiras:

- Grupo de Engenharia Integrada do Núcleo de Manufatura Integrada (NUMA) do Departamento de Engenharia de Produção da Escola de Engenharia de São Carlos, Universidade de São Paulo;
- Grupo de Estudos e Pesquisa em Qualidade (Gepeq), do Departamento de Engenharia de Produção da Universidade Federal de São Carlos (UFSCar).
- Departamento de Engenharia Mecânica da Universidade Federal de Santa Catarina, o Grupo de Engenharia do Produto e Processo (Gepp).

Essa ação, iniciada em 2001, teve como resultado um modelo de referência que contém os conceitos e as melhores práticas em desenvolvimento de produtos (DP) com a missão de se tornar “referência” para a derivação de outros modelos, adequados a um setor ou tipo específico de produto.

Neste sentido, este modelo já serviu de base para outros trabalhos. Romano (2003) definiu um modelo de referência voltado ao processo de desenvolvimento de máquinas agrícolas. As atividades propostas limitam-se ao processo interno, executadas pelas equipes de vários departamentos internos. O modelo inclui três macro-fases: planejamento, projeção e implementação. A primeira consiste das fases de planejamento do projeto, a macro-fase de projeção é constituída das fases de elaboração dos projetos: informacional, conceitual, preliminar e detalhado do produto e, por fim, processo de manufatura. E, finalmente, a macro-fase de implementação é formada pela fase de produção, lançamento do produto no mercado, validação do produto e encerramento do projeto..

Barbalho (2006), adaptou este modelo para o desenvolvimento de produtos mecatrônicos com aplicação em um caso real. Penso (2003), desenvolveu um trabalho em cima do modelo de referência aplicado a indústria de alimentos.

A figura 16 ilustra uma visão macro do modelo de referência de forma gráfica.

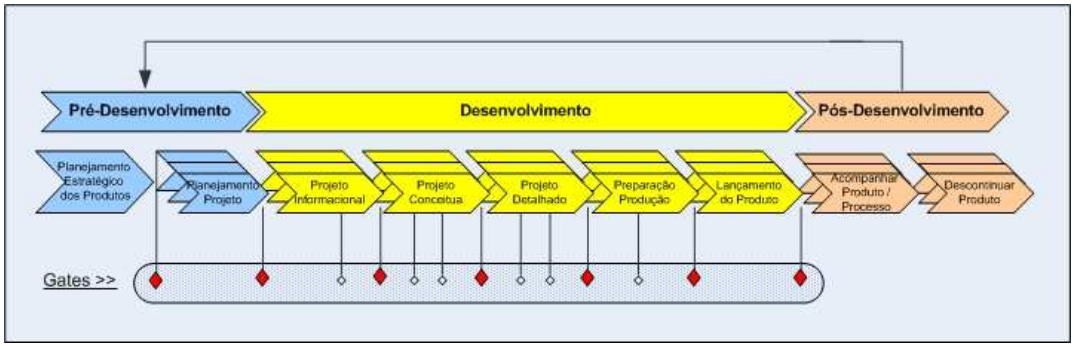

Figura 16. Modelo de referência. ROZENFELD et al, (2006).

Pode-se visualizar a existência de fases e sub-fases bem definidas, na qual papéis e fluxo de atividades são declarados. Além disso a passagem das fases é caracterizada por pontos de controle, chamados de Gates. Faz-se a seguir uma análise resumida das 3 Macro-Fases do modelo.

4.2.1 Macro-Fases

- <u>Pré – Desenvolvimento:</u> esta macro-fase é responsável por alinhar o desenvolvimento de software com o planejamento estratégico da empresa, evitando que projetos fora de contexto sejam levados adiante. Desta forma, o pré-desenvolvimento deve garantir que o direcionamento estratégico, definido pela empresa no seu planejamento estratégico, as idéias de todos os autores internos e externos envolvidos com os produtos e as oportunidades e restrições sejam sistematicamente mapeados e transformados em um conjunto de projetos bem definidos, isto é, o portfólio dos projetos que deverão ser

desenvolvidos. Compreendem esta macro-fase as fases de Planejamento Estratégico e Planejamento do Projeto.

- Desenvolvimento: após a definição do portfólio de produtos e o planejamento dos projetos, inicia-se o desenvolvimento propriamente dito. No modelo, esta fase é integrada às macro-fases de pré e pós-desenvolvimento. Esta tem início a partir das informações geradas pelo pré-desenvolvimento e documentadas no plano de projeto. Divide-se esta fase em quatro outras que se iniciam com o levantamento de requisitos dos interessados e termina com o lançamento do produto no mercado. Utilizam-se técnicas, métodos e ferramentas para o cumprimento das atividades, como por exemplo, técnicas para estimativas, ferramentas de modelagem, modelos para analise de riscos, entre outros. Apesar de aparecerem seqüencialmente, as atividades desta fase não seguem uma linearidade. Têm-se sobreposição de atividades, paralelismo e chegando a ocorrer em algumas vezes atividades cíclicas. As fases que compõem esta macro-fase são: Projeto Informacional, Projeto Conceitual, Projeto Detalhado e Lançamento do Produto. Esta possibilidade oferecida de se avançar e retroceder, de ir e vir, representa uma efetiva oportunidade de se enriquecer o projeto, pela sua otimização.

- Pós-Desenvolvimento: após o desenvolvimento do produto, diferentemente do que acreditam alguns, o projeto ainda não está finalizado. O acompanhamento desse produto, deve ser feito, medindo seu desempenho. Esta macro-fase compreende ainda a retirada do produto do mercado e, finalmente, uma avaliação de todo o ciclo do produto, para que as experiências contrapostas ao que foi planejado anteriormente sirvam de referência a desenvolvimentos futuros, permitindo o *feedback*. Neste

estágio, as fases são: Analisar Satisfação do Cliente, Monitorar Desempenho do Produto, Realizar Auditoria Pós-Projeto e Descontinuar Produto do Mercado. Cabe lembrar que a retirada do produto não deve ser motivo de insatisfação para o cliente, que será atendido por soluções mais avançadas.

4.3 Detalhamento do nível G do MPS.Br

4.3.1 Gerência de Projetos

O processo de gerência de projetos é a primeira camada do processo de engenharia de software que abrange todo o processo de desenvolvimento, do começo ao fim. A gerência de projetos oferece uma compreensão do escopo do trabalho a ser feito, dos riscos, dos recursos exigidos, das tarefas a serem executadas, dos marcos de referência a serem acompanhados, do custo e da programação a ser seguida. PRESSMAN, (2005).

O MPS.Br destaca o processo de gerência de projetos com o objetivo de estabelecer e manter planos que definam as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o seu andamento que permitam a realização de correções quando houver desvios significativos no seu desempenho. MPS (2007).

4.3.1.1 Resultados Esperados

Os resultados esperados pela correta implantação do processo de gerência de projetos, representados pela sigla GPRn, atribuído ao nível G do MPS.Br são apresentados a seguir: MPS (2007).

- GPR1 - O escopo do trabalho para o projeto é definido, apresentando todo o trabalho necessário para entregar um produto e/ou serviço que satisfaça as necessidades, características e funções específicas para o projeto, de forma a concluí-lo com sucesso;
- GPR2 - As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados. Todo o trabalho deve ser decomposto em componentes menores, a fim de facilitar seu gerenciamento. A estrutura de decomposição fornece uma referência para atribuição de tamanho, esforço, cronograma e responsabilidades, e é utilizada como uma estrutura subjacente para planejar, organizar e controlar o trabalho executado no projeto;
- GPR3 - O modelo e as fases do ciclo de vida do projeto são definidos, permitindo planejar o projeto, incluindo marcos importantes para o controle e revisões. No ciclo de vida são definidas as fases e as atividades do projeto, de acordo com o escopo dos requisitos, as estimativas para os recursos e sua natureza. A determinação das fases do ciclo de vida do projeto possibilita períodos planejados de avaliação e de tomada de decisões, nos quais compromissos significativos são realizados com relação aos recursos, abordagem técnica, reavaliação de escopo e custo do projeto;
- GPR4 - O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas, incluindo os dados de custo, esforço e tempo de projetos executados anteriormente, além de dados apropriados de escala para equilibrar as diferenças de tamanho e complexidade;
- GPR5 - O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos. São identificadas as tarefas do projeto e

potenciais gargalos, que são resolvidos quando possível, e o cronograma das atividades é estabelecido, com início, duração e término. Com base no cronograma e na estimativa de custos, refletida pelos recursos requeridos, é estabelecido o orçamento do projeto. Este resultado é importante porque o cronograma e o orçamento são instrumentos fundamentais para o acompanhamento do dia-a-dia do projeto. Desta forma, sempre que necessário, devem ser revistos e atualizados;

- GPR6 - Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados. Para facilitar a identificação dos riscos, é interessante elaborar-se uma lista de riscos mais comuns, que deve ser examinada pelo gerente do projeto e/ou equipe do projeto para identificar quais destes são potenciais riscos para o projeto em questão. Os riscos identificados devem ser registrados, bem como o acompanhamento dos seus estados e ações tomadas. No nível G, os riscos são acompanhados para verificar como afetam o projeto e para se tomar as medidas, aplicáveis mesmo que ainda sem um gerenciamento completo;
- GPR7 - Os recursos humanos para o projeto são planejados considerando-se o perfil e o conhecimento necessário para executá-lo, determinando-se as funções, responsabilidades e relações hierárquicas do projeto. As funções do projeto podem ser designadas para pessoas ou grupos, os quais podem ser internos ou externos à organização;
- GPR8 - As tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados, devendo ser especificadas as tarefas e previstos os recursos e o ambiente necessário, incluindo, por exemplo, equipamentos, ferramentas, serviços, componentes, viagens e requisitos de processo. Este planejamento é importante, pois recursos especiais precisam de suporte

orçamentário, o que pode se tornar crítico em alguns projetos, se não forem previstos;

- GPR9 - Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança. O motivo de se coletar cada dado também deve ser evidenciado, pois coletá-lo, armazená-lo, distribuí-lo de forma controlada e mantê-lo atualizado, implica em custo. A questão da confidencialidade, mesmo quando não declarada pelo cliente, deve ser tratada com extremo cuidado;
- GPR10 - Planos para a execução do projeto são estabelecidos e reunidos no Plano do Projeto. Todas as informações definidas e coletadas para o projeto devem ser organizadas de forma a possibilitar um gerenciamento adequado do projeto. A reunião desses documentos é conhecida como sendo o Plano do Projeto;
- GPR11 - A viabilidade de atingir as metas do projeto, considerando-se as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados. O estudo da viabilidade considera o escopo do projeto e examina aspectos técnicos, financeiros e humanos. Deve-se considerar também os objetivos de negócio da organização. Muitas vezes é preferível não iniciar ou parar um projeto do que prosseguir com um projeto inviável, o que pode levar a perdas maiores, tanto para o fornecedor como para o cliente;
- GPR12 - O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido. Negociações devem ser realizadas quando existem conflitos entre as diversas variáveis do projeto, como requisitos, custos e prazos. A solução dos conflitos e estabelecimento de compromissos é fundamental para que

o projeto possa efetivamente contar com recursos planejados, para atingir as metas definidas;

- GPR13 - O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados. O acompanhamento pode ser realizado utilizando-se ferramentas de planejamento, em que se pode examinar o previsto contra o realizado, usando-se indicadores de progresso e cumprimento de marcos, entre outros. Pode também ser feito por meio de reuniões e comunicação pessoal;
- GPR14 - O envolvimento das partes interessadas no projeto é gerenciado, podendo incluir o cliente e o usuário, a direção da organização e os membros da equipe do projeto. Este resultado é importante porque o distanciamento da gerência do projeto em relação aos interessados pode acarretar desvios em relação às reais necessidades que o projeto deverá atender;
- GPR15 - Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento. Os marcos do projeto precisam ser previamente definidos ao se realizar o planejamento do projeto. As revisões em marcos podem ter um caráter formal, com participação de gerências superiores, representantes do cliente e outras partes interessadas no projeto. Este resultado é importante, porque as revisões em marcos são oportunidades para verificar, de forma ampla, o andamento de todo o projeto, independente do acompanhamento do dia-a-dia;
- GPR16 - Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas. Todos os problemas encontrados com os resultados anteriores devem ser registrados e gerenciados até a sua correção;

- GPR17 - Ações para corrigir desvios em relação ao planejamento e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão. Como resultado dos acompanhamentos citados anteriormente, problemas são identificados, analisados e registrados. Ações corretivas devem ser estabelecidas para resolver problemas que possam impedir o projeto de atingir seus objetivos. O controle dos problemas levantados, as ações tomadas, os responsáveis pelas ações e os resultados devem ser registrados.

4.3.2 Gerência de Requisitos

Os problemas que os projetistas de software têm para solucionar são, muitas vezes, muito complexos. Compreender a natureza desses problemas pode ser muito difícil, especialmente se o sistema é novo. Conseqüentemente, é difícil estabelecer o que o sistema deve fazer. As descrições das funções e das restrições são os requisitos do sistema, e o processo de descobrir, analisar e documentar essas funções e restrições é chamado de engenharia de requisitos. SOMMERVILLE (2003).

O propósito desse processo no Nível G do MPS.Br é gerenciar os requisitos dos produtos e componentes do produto do projeto e identifica. MPS (2007).

Algumas das atribuições do processo de requisitos são: revisão dos requisitos recebidos de um fornecedor de requisitos e prevenir seu mal entendimento; documentar as mudanças nos requisitos e suas justificativas; manter a rastreabilidade bidirecional entre os requisitos e produtos de trabalho em geral; identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto; entre outras. MPS (2007).

4.3.2.1 Resultados Esperados

Segundo MPS (2007), os resultados esperados pela correta implantação do processo de gerência de requisitos, representados pela sigla GREn, atribuído ao nível G do MPS.Br são:

- GRE1 - O entendimento dos requisitos é obtido junto aos fornecedores de requisitos, com a finalidade de garantir que os requisitos estejam claramente definidos. Para que isso ocorra, deve-se rever com o fornecedor de requisitos se as necessidades e expectativas estão sendo atendidas com os requisitos propostos e, com essa confirmação, criar um documento de requisitos. Todas as comunicações com o fornecedor de requisitos devem ser registradas em atas, *e-mails*, ferramentas de comunicação ou outros meios, sendo necessário uma comprovação de que os interessados estão de acordo com o conteúdo registrado;
- GRE2 - Os requisitos de software são aprovados utilizando-se critérios objetivos, envolvendo a equipe técnica da organização e o cliente. Esses critérios devem ser estabelecidos previamente, tais como: possuir identificação única; estar claro e apropriadamente declarado; ser relevante; ser completo; estar consistente com os demais requisitos; ser implementável, testável e rastreável. Deve ter muita cautela ao aprovar novos requisitos, pois uma mudança nos requisitos de um projeto pode impactar significativamente em termos de escopo e estimativas de cronograma e esforço;
- GRE3 - A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida, permitindo rastrear a dependência existente entre eles. Com isso, facilita-se a avaliação do impacto das mudanças de requisitos que possam ocorrer, por exemplo, nas estimativas do escopo, nos produtos de trabalho ou nas tarefas do projeto;

- GRE4 - Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos. Ao identificar alguma inconsistência entre os requisitos e os demais elementos do projeto, estas devem ser registradas e cabendo executar ações corretivas a fim de resolvê-las. Se houver mudanças nos requisitos, deve-se executar essa revisão para identificar se todos os artefatos do projeto estão consistentes;
- GRE5 - Mudanças nos requisitos são gerenciadas ao longo do projeto através de documentação das mudanças e de manutenção do histórico das decisões tomadas acerca dessas mudanças. Estas decisões são tomadas por meio da realização de análises de impacto da mudança no projeto e podem incluir aspectos como: influência em outros requisitos, expectativa dos interessados, cronograma, riscos e custo.

4.4 Pontos comuns entre MPS.br e o modelo de referência

Uma das características marcantes existentes nos modelos de qualidade aplicados ao PDS, como o CMMi e o MPS.Br, se diz ao fato de não existir uma explícita indicação de **como** atingir os objetivos, limitando-se apenas a apresentar **o que** deve ser cumprido, cabendo aos responsáveis pela melhoria do processo criar ou adaptar meios para que os níveis de maturidade possam ser efetivamente atingidos.

No caso do MPS.Br, em todos os níveis de maturidade, encontram-se processos que são identificados no modelo de referência para desenvolvimento de produto. No Nível G, encontram-se os processos de gerência de requisitos e gerência de projetos claramente definidos nas etapas de pré-desenvolvimento e início da etapa de desenvolvimento.

Por outro lado, existem processos como “planejar processo de fabricação e montagem” ou a “fase de preparação da produção do produto” que não são aplicáveis ao PDS devendo ser então eliminados do modelo proposto.

Para uma melhor compreensão dos pontos comuns, faz-se nesta seção, uma análise da aderência sobre o processo de desenvolvimento de produto com o MPS.Br, onde, por meio de uma tabela comparativa, demonstra-se o que cada atividade implementada no decorrer das fases do modelo de referência ficaria representado no resultado esperado do MPS.Br especificamente no processo de gerência de projeto e gerência de requisitos.

As tabelas 4 e 5 estão estruturadas da seguinte maneira: a primeira coluna representa as fases do processo de planejamento de projeto proposto no modelo de referência, a segunda coluna representa cada atividade desenvolvida na fase relacionada, e a terceira coluna representa os resultados esperados no MPS.Br aderentes a cada atividade correlata no modelo de referência. A tabela 4 refere-se à Gerência de Projetos enquanto que a tabela 5, refere-se à Gerência de Requisitos.

Tabela 4 - Resultado da aderência das atividades do modelo de referência ao MPS.Br – Gerência de Projeto.

Fase	Atividade	Resultado Esperado
Planejamento de Projeto	Definir interessados do projeto	GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo; GPR 14. O envolvimento dos interessados no projeto é gerenciado;
	Definir escopo do produto	GPR 2. As tarefas e os produtos de trabalho e as tarefas do projeto são dimensionados utilizando métodos apropriados;
	Definir escopo do projeto	GPR 1. O escopo do trabalho para o projeto é definido;
	Detalhar escopo do projeto	GPR 8. As tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados. GPR 2. As tarefas e os produtos de trabalho e as tarefas do projeto são dimensionados utilizando métodos apropriados;

	Definir atividades e seqüência	GPR 2. As tarefas e os produtos de trabalho e as tarefas do projeto são dimensionados utilizando métodos apropriados; GPR 3. O modelo e as fases do ciclo de vida do projeto são definidas;
	Preparar cronograma	GPR4. O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas. GPR 5. O cronograma e o orçamento do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;
	Detalhar ciclo de vida do produto	GPR 3. O modelo e as fases do ciclo de vida do projeto são definidas;
	Avaliar riscos	GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridades de tratamento são determinados e documentados;
	Preparar orçamento do projeto	GPR 5. O cronograma e o orçamento do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;
	Analisar a viabilidade econômica do projeto	GPR 11. A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário ajustes são realizados;
	Definir indicadores de desempenho	GPR 15. Revisões são realizadas em marcos do projeto e conforme estabelecidos no planejamento. GPR 16. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidas e tratados com as partes interessadas.
	Definir plano de comunicação	GPR 9. Os dados relevantes do projeto são identificados e planejados quanto a forma de coleta, armazenamento e distribuídos. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
	Planejar e preparar aquisições	GPR 8. As tarefas, os recursos e o ambiente de trabalho necessários para completar o trabalho são planejados
	Preparar Plano de Projeto	GPR 10. Planos para execução do projeto são estabelecidos e reunidos no Plano de Projeto.
	Discussão e redação do Plano de Projeto	GPR 12. O plano do projeto é revisado com todos os interessados e seu respectivo compromisso com ele é obtido.
	Avaliar e aprovar Fase	GPR 16. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidas e tratados com as partes interessadas. GPR 17. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até sua conclusão.

Tabela 5 - Resultado da aderência das atividades do modelo de referência ao modelo MPS.Br – Gerência de Requisitos

Fase	Atividade	Resultado Esperado
Projeto Informacional	Definir requisitos do produto	GR1 – O entendimento dos requisitos é obtido junto aos fornecedores de requisitos.
	Definir especificações meta do produto	GR2 – Os requisitos de software são aprovados utilizando critérios objetivos.
	Gerenciar Mudanças de Engenharia	GR5 – Mudanças nos requisitos são gerenciadas ao longo do projeto.
	Propor mudança	GR1 – O entendimento dos requisitos é obtido junto aos fornecedores de requisitos.
	Alterar informações do produto	GR3 – A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida.
	Acompanhar Produto e Processo	GR4 – Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos.

Essa relação, para os dois processos iniciais do MPS.Br, será adotada por ser imprescindível, como base para a proposta de um modelo de referência voltado ao desenvolvimento de software, conforme será justificado a seguir.

4.5 Proposta do Modelo

Com base no modelo de referência para desenvolvimento de produtos e seus pontos de conformidade com a norma MPS.Br se propõe um modelo de referência, chamado de BASE, que serve de guia para as empresas de desenvolvimento de software criarem seus produtos de forma a garantir qualidade dos produtos finais.

Nesta proposta tem-se a definição do modelo representado e estruturado com base em quatro elementos básicos, que representam “**quem**” faz, “**o quê**”, “**como**” e “**quando**”:

- **Papéis e Responsabilidades (quem) -** Um papel define o comportamento e responsabilidades de um profissional ou grupo de profissionais que participam do desenvolvimento do projeto. O comportamento é representado através das atividades que cada papel deve desempenhar ao longo do projeto. As responsabilidades normalmente estão associadas aos artefatos que cada papel deve produzir e manter ao longo das atividades que realiza. Na prática, um mesmo papel pode ser desempenhado por mais de uma pessoa, assim como uma mesma pessoa pode assumir vários papéis ao longo do projeto.
- **Artefatos (o quê) -** Em sentido amplo, o termo artefato representa um produto concreto produzido, modificado ou utilizado pelas atividades de um processo. Os artefatos representados no PDS não incluem todos os artefatos de um projeto de desenvolvimento, mas todos estes devem ser elaborados ao longo do projeto. O PDS disponibiliza modelos (*templates)* para a maioria de seus artefatos, com o objetivo de orientar e facilitar a sua elaboração.
- **Atividades (como) -** Uma atividade no PDS representa um conjunto de passos e tarefas que um profissional, que desempenha o papel responsável por aquela atividade, deve executar para gerar algum resultado. As atividades envolvem a produção e modificação de artefatos do projeto.
- **Fluxo de Atividades (quando) -** O fluxo de atividades do PDS apresenta a seqüência e a dependência entre as atividades do projeto ao longo do tempo. As atividades no fluxo são divididas em fases do ciclo de vida do projeto e nos papéis responsáveis pela execução de cada uma.

A figura 17 apresenta uma visão geral do modelo de referência proposto.

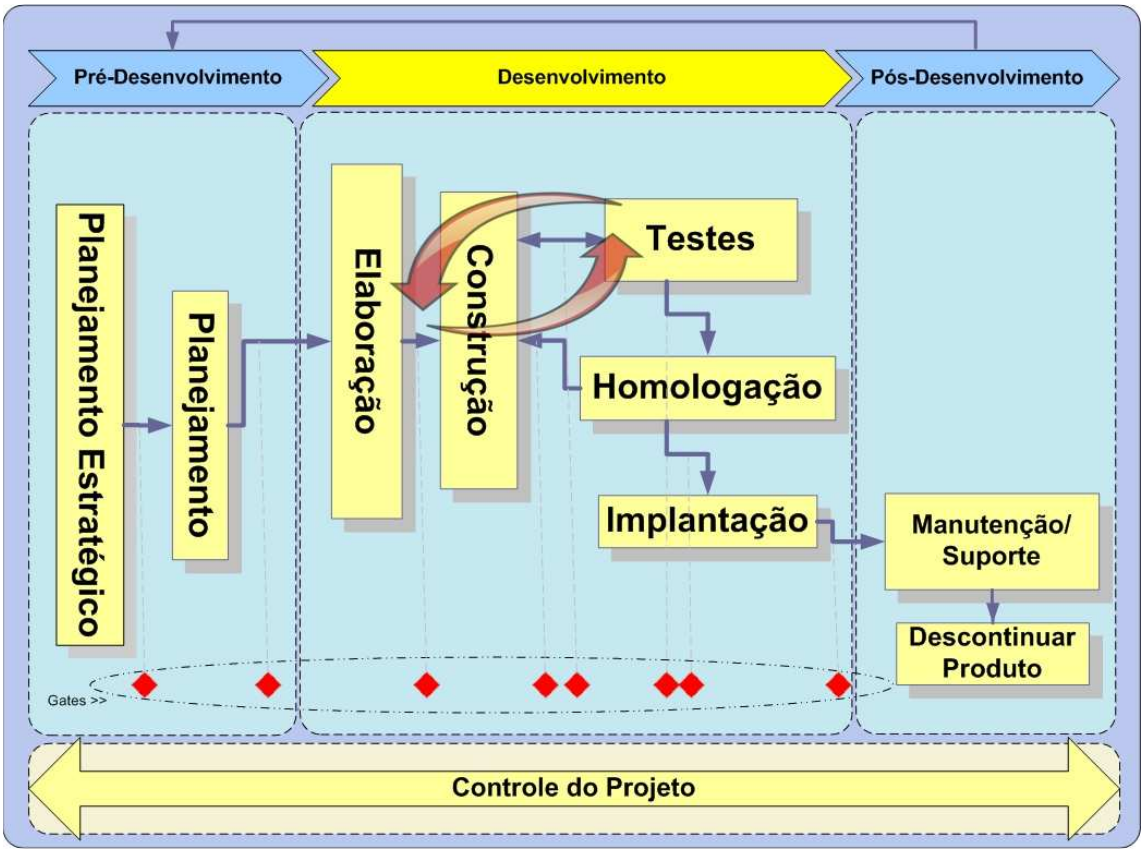

Figura 17. Visão geral do modelo de referência para desenvolvimento de software.

O processo se inicia com uma solicitação por parte do cliente, podendo ser um cliente interno ou externo. Automaticamente este pedido é tratado na fase de planejamento estratégico, onde se analisa o projeto confrontando com as metas da empresa, evitando que projetos fora do contexto estratégico de atuação da empresa sejam levados adiante. Esta fase tem como marco o escopo preliminar. Na fase de planejamento, tem-se como intuito o detalhamento do escopo com o planejamento de recursos necessários e como marco final o Plano de Projeto aprovado. Já na etapa de construção se encontram as fases de Elaboração para elicitação de requisitos junto ao cliente, de construção e testes do produto, além das fases de homologação e implantação. Cabe destacar que apesar de parecer seqüencial e linear, as fases de Elaboração, Construção e Testes apresentam iterações cíclicas. Na fase de Pós-Desenvolvimento estão inclusos os processos de suporte ao produto e a descontinuidade do

mesmo. Paralelamente a todos essas fases, faz-se um controle de qualidade para acompanhar o processo e o produto, tomando ações efetivas quando desvios são detectados.

Cada área dentro do modelo é decomposto em processos que contém sub-processos com atividades, funções e artefatos associados a seu propósito. Em cada processo existem marcos, também chamados de *Gates* que definem pontos de verificação e validação.

Abaixo segue uma descrição mais detalhada de cada área com seus propósitos.

Planejamento Estratégico. Representa a primeira etapa no processo de desenvolvimento do produto e tem o objetivo de se obter ao final um plano que contenha o portfólio de produtos da empresa a partir de seu planejamento estratégico. A importância desta etapa se dá pela filtragem que existirá na seleção de produtos que deverão ser iniciados e aqueles que deverão ser abandonados sempre seguindo a estratégia quanto à dinâmica da tecnologia e à dinâmica do mercado, público alvo, lucratividade, entre outros. Trata-se de uma etapa de gerenciamento macro e de posicionamento da empresa, com relação a sua vocação na produção de projetos para determinadas demandas do mercado.

Planejamento. Esta fase se desenvolve após a filtragem, já na relação direta com o cliente. O objetivo do planejamento é o estabelecimento de um acordo formal, entre a equipe de desenvolvimento e usuários do projeto, quanto ao escopo do produto a ser desenvolvido. A viabilização do projeto pela empresa, sob orientação do planejamento estratégico, marca o início do projeto. A partir do escopo preliminar, a equipe do projeto dá início ao detalhamento do escopo do produto. O escopo detalhado deve ser registrado no Documento de Escopo e posteriormente no Plano de Projeto e aprovado formalmente pelos usuários. O Plano de Projeto resultante desta fase, contém ainda um cronograma do projeto, os recursos a serem

utilizados, o plano de comunicações, a análise de riscos, entre outros pontos importantes, que serão acordados pelo fornecedor e pelo cliente.

Elaboração. Esta fase envolve uma análise detalhada sobre as necessidades e problemas gerais do projeto e a definição de como o sistema será desenvolvido em termos tecnológicos, considerando os requisitos, limitações e restrições identificados durante a fase de Planejamento.

O objetivo da fase de Elaboração é o estabelecimento e validação de uma arquitetura de hardware e software que suporte de forma adequada os requisitos funcionais e não-funcionais do sistema.

Durante esta fase os analistas de sistema da equipe do projeto devem identificar os requisitos detalhados do produto a partir de reuniões e entrevistas de levantamento junto aos usuários. Estes requisitos deverão ser descritos detalhadamente no Documento de Requisitos do projeto e através de um Diagrama de Casos de Uso, representando a visão funcional e as fronteiras do sistema. Esta fase é importante porque focaliza os limites de aplicação do software e elimina um grande número de outras alternativas.

Os casos de uso identificados deverão ser descritos detalhadamente no Modelo de Casos de Uso que, além do diagrama de casos de uso elaborado durante o levantamento de requisitos, inclui também as Especificações de Casos de Uso. Em paralelo, a equipe de analistas de sistema define a arquitetura de hardware e software do sistema, que deverá ser documentada no Documento de Arquitetura. Este documento é formalmente aprovado pelas partes envolvidas.

Construção. Esta fase compreende o desenvolvimento propriamente dito do sistema, em termos de códigos-fonte e componentes de software.

O objetivo desta fase é o desenvolvimento de uma versão operacional do sistema, estável o suficiente para ser disponibilizada para testes e posterior homologação por seus usuários finais, no menor tempo possível, considerando os critérios de qualidade estabelecidos pelo projeto.

Os analistas de sistema da equipe estarão envolvidos no projeto detalhado do sistema - Diagramas de Classes e outros diagramas considerados relevantes para a representação do projeto do sistema, enquanto os projetistas de banco de dados estarão envolvidos na modelagem da versão completa do banco de dados. Com base nestas informações, no Modelo de Casos de Uso, o Documento de Requisitos e no Documento de Arquitetura, os programadores estarão focados na implementação do sistema e de seus componentes, através de atividades de codificação, realização de testes unitários e na integração e compilação de versões intermediárias.

Em paralelo ao desenvolvimento do sistema, os manuais de usuário, guias rápidos, manuais de instalação e administração, *release notes*, entre outros documentos, serão desenvolvidos, e formalmente registrados.

A fase de Construção deverá ser dividida em iterações de acordo com a necessidade identificada pelo gerente do sistema.

A cada iteração será gerada e testada uma nova compilação do sistema, contendo os cenários de casos de uso implementados até o momento.

Teste. Esta fase envolve as atividades necessárias para que o sistema desenvolvido seja adequadamente disponibilizado a seus usuários com a menor presença de erros possíveis. O objetivo principal desse processo é portanto encontrar falhas no sistema durante todo o processo de desenvolvimento, uma vez que os testes são divididos em testes unitários e teste de sistema. Os teste unitários ocorrem cotidianamente e são planejados de forma previa ao

desenvolvimento de um código, sua execução se dá logo após ao término da compilação de uma parte do código. Falhas são documentadas e prontamente corrigidas. Ao término da implementação de todo o sistema são realizados os testes do sistema com todas as funcionalidades integradas.

Homologação. A fase de homologação representa a fase de aceitação do usuário, que quando bem direcionada e com o desenvolvimento do sistema bem instrumentado sempre será encerrada com o documento de avaliação formal sinalizando sucesso. Esta etapa na maioria dos sistemas pode ter uma curta duração, ou até mesmo ser unificada à etapa de teste, porém, processos de homologação mais extensos são suportados.

Estando o sistema homologado junto a seus usuários, ele estará disponível para implantação em ambiente de produção.

Apesar desta fase estar baseada principalmente em atividades externas ao PDS, considera-se importante relacioná-las no ciclo de vida do projeto, uma vez que são consideradas etapas do processo de desenvolvimento como um todo. Entende-se que a passagem do produto para o ambiente de produção (*deployment*) deve ser preocupação constante da equipe de desenvolvimento, e todas as atividades relacionadas devem ser planejadas desde o início do projeto. Todo o PDS tem como objetivo final a disponibilização para os usuários de um produto desenvolvido em conformidade com suas especificações, dentro dos critérios de qualidade previstos para o projeto, e que possa ser prontamente utilizado por seus usuários.

Implantação. Esta etapa representa a entrega e instalação do sistema para uso. Também chamada em outros modelos de *Deployment* ou Produção. Compreende esta etapa uma estratégia para implantação do sistema que é caracterizado pelo desenvolvimento de um diagrama UML chamado de diagrama de implantação (*deployment*). Trata-se de um diagrama

usado para se obter uma visão estática da implantação, assim se pode definir onde e como determinados componentes ou sistemas e subsistemas serão instalados/empacotados. Refere-se também a disponibilização do produto e ao acesso, com a praticidade adequada de uma aplicação eficiente e eficaz.

Manutenção/Suporte. Após a instalação, inicia-se a etapa de acompanhamento do sistema que representa uma importante fase, pois, garante o funcionamento do sistema e eventuais reparos e suporte. Esta fase é caracterizada pelas atividades de suporte e de manutenção. Toda solicitação de suporte é formalmente requisitada, seja por email ou por documento próprio, mesmo as solicitadas por telefone são imediatamente documentadas.

Descontinuidade do Produto. Esta fase é representada pelo fim do suporte ao produto, na qual a empresa e o cliente/usuário formalmente estabelecem a descontinuidade à manutenção do sistema. Dada a velocidade da inovação que afeta a dinâmica da tecnologia e a dinâmica do mercado, conforme Drucker (1968), há que se prever a obsolescência técnica ou esgotamento das demandas específicas ou, a possibilidade de revigorar do produto, com a consideração de sua otimização, o que demandará novo contrato.

Controle do Projeto. Esta fase está relacionada ao acompanhamento do projeto durante todo o ciclo de vida, desde as fases preliminares, com o intuito de monitorar, controlar e registrar o andamento do projeto como um todo, identificando mudanças e possibilitando ajustes. Nesta fase se encontram os processos de gerência de projetos, gerência de requisitos e gerência da qualidade, explicados abaixo com mais detalhes:

Gerência de Requisitos. A gerência de requisitos tem como intuito garantir que mudanças solicitadas e efetuadas durante o desenvolvimento sejam controladas, de forma a

criar um mecanismo que possa atualizar as documentações do produto, negociar as mudanças, analisar o impacto dessas mudanças e reformular o cronograma.

Controle de Projetos. Esta fase é dedicada a todo controle do projeto, previamente planejado. Leva-se em consideração os conceitos do PMI (*Project Manager Institute*) e da metodologia ágil Scrum.

Gerenciamento da Qualidade. Esta fase é realizada durante todo o PDS, sendo de extrema importância para a garantia da qualidade nos padrões estabelecidos pela empresa. Visa o gerenciamento da qualidade no processo e conseqüentemente do produto de software. Esse gerenciamento está alinhado com as diretrizes estabelecidas pelo MPS.Br e deve estar descrita nos planos da empresa.

Esta etapa está sob responsabilidade da equipe de qualidade (GQA) que acompanha cotidianamente o processo, seus artefatos e o produto final, registrando ocorrências e designando os relatórios aos responsáveis para que medidas possam ser tomadas na solução de desvios.

Com foco em aprofundar esse modelo, foram selecionadas os processos inicias de Gerência de Requisitos e Gerência de Projetos. A justificativa para escolha desses dois processos se deve ao fato de serem os pontos iniciais para um programa de melhoria tanto no CMMi como principalmente no MPS.Br.

4.5.1 Processo Padrão de Gerência de Requisitos

A gerência de requisitos compreende uma etapa fundamental do desenvolvimento, pois garante que as mudanças solicitadas após a finalização do projeto ou durante o andamento do projeto serão consideradas e terão um tratamento adequado. Pode-se definir a

gerência de requisitos como um processo sistemático para elicitar, documentar, organizar e rastrear os requisitos variáveis de um sistema.

Tem que se levar em consideração que qualquer mudança deve ser implementada de forma adequada, caso seja aprovada, sem afetar a qualidade final do produto, independentemente de quem solicita e quanto tempo terá para realizar a mudança.

Entre os principais fatores que ocasionam a não gerência efetiva de requisitos por parte de pequenas empresas, pode-se destacar o fato dessas empresas não terem um processo padrão para usar como base. Isso ocasiona problemas como descontrole do cronograma, escopo do projeto maior do que o previamente estipulado e cobrado financeiramente, percepção do cliente de que toda solicitação deverá ser alterada a qualquer tempo, da forma que ele deseja, entre outros.

No intuito de se gerar um processo padrão para a gerência de requisitos, faz-se a seguir um detalhamento das atividades, do fluxo de atividades, dos artefatos necessários e dos papéis a serem desempenhados, em conformidade com a norma MPS.Br.

4.5.1.1 Definição dos fluxos de atividades

O fluxo de atividades do processo de Gerência de Requisitos foi definido com base na literatura atual disponível, ao estado da arte e tendo como objetivo principal atender aos resultados esperados pelo processo de Gerência de Requisitos do MR-MPS, além da influência do modelo de referência para desenvolvimento de produto utilizado neste trabalho.

Através do fluxo de atividades o processo de Gerência de Requisitos é sistematizado de modo que sejam desempenhadas as atividades da forma como apresentado no fluxograma da figura 18.

O processo se inicia com uma requisição, que pode ser externa, solicitada por um cliente ou internamente identificada por um colaborador. De qualquer modo, esta requisição precisa ser formalizada na forma de documento padrão ou email. A partir dessa formalização, faz-se um levantamento mais detalhado para elicitar esse pedido, levantando informações do que o cliente deseja. Constatada a real necessidade de mudança, traça-se uma análise do impacto no projeto como um todo, passando pelo cronograma, necessidade de recursos, etc, para isso, usa-se uma matriz de rastreabilidade dos requisitos. Com essa primeira análise feita, inicia-se uma etapa de negociação na qual as prioridades, custos e o tempo de desenvolvimento da solicitação são acertados, considerando-se que para esta atividade se faz utilizar-se de critérios previamente estabelecidos. Caso não tenha sido aprovada, a solicitação de mudança, o cliente é comunicado dos motivos. Em caso positivo, as mudanças são incorporadas no projeto e a rastreabilidade é estabelecida com a atualização dos documentos de requisitos e plano de projeto.

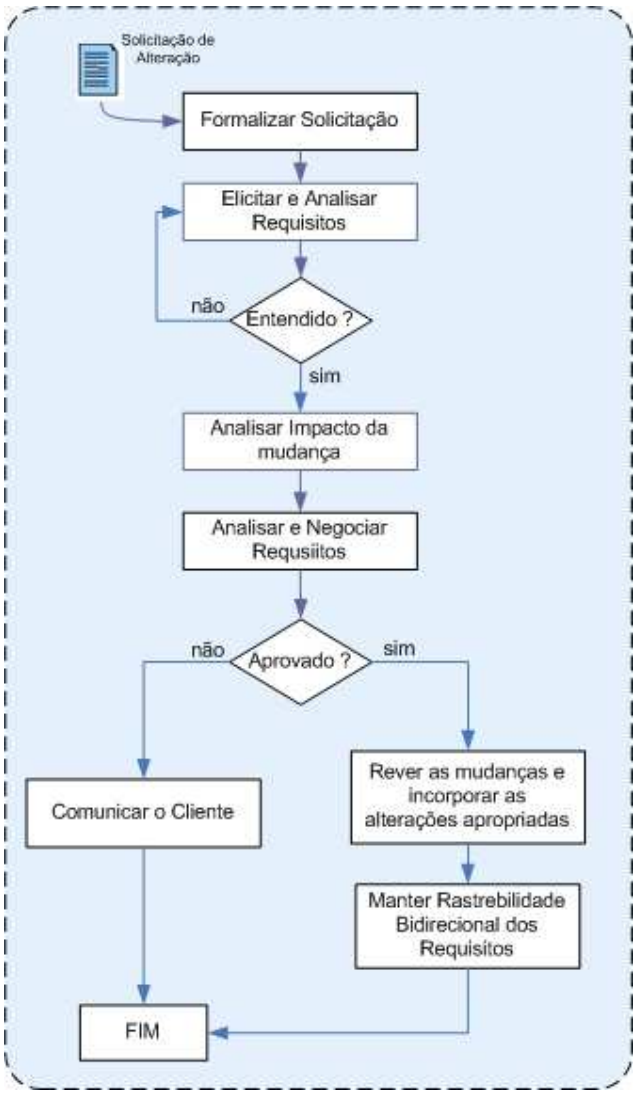

Figura 18. Fluxo de atividades da gerência de requisitos

Deve-se destacar que todas essas ações são definidos em um Plano de Gerência de Requisitos. Os itens abaixo fazem uma breve descrição de cada uma das atividades do fluxo:

4.5.1.2 Atividades

Nome	Formalizar Solicitação
Descrição	Significa receber a solicitação do cliente obtida por um dos meios oficiais declarados na Política Organizacional (PO), registrar solicitação do cliente, agendar obtenção de requisitos e fazer levantamento de informações sobre o cliente (quando externo).
Responsável(is)	Suporte

Participantes	Suporte Cliente
Produtos Requeridos	Email ou formulário de solicitação de mudança
Produtos Gerados	Formulário de Solicitação de mudança preenchido

Nome	Elicitar e Analisar Requisitos
Descrição	Atividade destinada ao levantamento e entendimento das necessidades de forma clara e documentada no padrão esabelecido.
Responsável(is)	Analista de sistemas
Participantes	Analista de sistemas Cliente
Produtos Requeridos	Solicitação de Mudança
Produtos Gerados	Requisitos Especificados

Nome	Analisar impacto da mudança
Descrição	Deverá ser realizada uma análise para entender o impacto que essa mudança irá desencadear no projeto como um todo, em termos de cronograma e da demanda de recursos
Responsável(is)	Gerente do Projeto
Participantes	Analista de Sistemas Gerente de Projeto
Produtos Requeridos	Requisito Especificado. Matriz de Rastreabilidade
Produtos Gerados	Análise do Impacto

Nome	Analisar e Negociar os Requisitos
Descrição	Nesta atividade será analisada a viabilidade de tempo e financeira da mudança e a negociação junto ao cliente.
Responsável(is)	Gerente de Projeto
Participantes	Analista de Sistema Gerente de Projeto Cliente
Produtos Requeridos	Requisitos Específicos
Produtos Gerados	Viabilidade da mudança

Nome	Incorporar alterações apropriadas
Descrição	Com a aprovação da mudança, será executada as mudanças no sistema de forma a atender a necessidade do cliente.
Responsável(is)	Gerente de Projeto
Participantes	Gerente de Projeto Desenvolvedor

	Equipe de Qualidade
Produtos Requeridos	Requisito Especificado
Produtos Gerados	Sistema atualizado

Nome	Manter rastreabilidade bidirecional dos requisitos
Descrição	Com a mudança já executada, deve-se registrar e atualizar os documentos que sofreram alterações.
Responsável(is)	Analista de sistema
Participantes	Equipe de Qualidade Analista de sistema
Produtos Requeridos	Documento de requisitos / Matriz de rastreabilidade / Plano de Projeto
Produtos Gerados	Documentos atualizados.

Nome	Comunicar o Cliente
Descrição	A comunicação com o cliente em caso de inviabilidade deve ser feita de maneira formal explicitando claramente os motivos.
Responsável(is)	Gerente do projeto
Participantes	Gerente do projeto Cliente
Produtos Requeridos	Análise de viabilidade
Produtos Gerados	Email

4.5.1.3 Artefatos

- **Formulário de Solicitação de Mudança:** Documento que registra o pedido de mudança feito pelo cliente interno ou externo, cada formulário gera um número identificando o responsável pela solicitação, descrição e motivo da mudança. Esse formulário pode ser eletrônico ou físico.
- **Documento de Requisitos:** O documento de requisitos registra todos os requisitos do sistema levantados junto aos usuários gestores e finais. Aconselha-se padronizar este documento seguindo o padrão internacional definido pela IEEE 820. Este artefato deve estar sob rígido controle de mudanças uma vez que as alterações em suas informações podem ter impacto por todo o sistema.

- **Documento de Arquitetura:** Descreve as principais decisões de projeto tomadas pela equipe de desenvolvimento e os critérios considerados durante a tomada destas decisões. Suas informações incluem desde a especificação da infra-estrutura de hardware e software do sistema, até descrições detalhadas dos diversos mecanismos de arquitetura de software adotados.
- **Matriz de Requisitos**: A Matriz de Requisitos registra todos os requisitos e as regras de negócios do sistema levantados junto aos usuários gestores e finais. Este artefato deve estar sob rígido controle de mudanças uma vez que as alterações em suas informações podem ter impacto por todo o sistema.
- **Plano de Gerência de Requisitos**: Descreve a documentação de requisitos, os tipos de requisitos e seus respectivos atributos de requisitos, especificando as informações e os mecanismos de controle que devem ser coletados e usados para avaliar, relatar e controlar mudanças nos requisitos do produto.
- **Laudos das revisões realizadas:** Formulário padrão no qual a Equipe de Qualidade periodicamente registra todas os desvios encontrados entre os produtos gerados e a descrição do planejado, enviando aos responsáveis para que medidas sejam tomadas.

4.5.1.4 Papéis e Responsabilidades

- **Analista de Sistemas:** É responsável pelas atividades de análise e projeto do sistema, incluindo o levantamento dos requisitos junto aos *stakeholders*, a especificação dos requisitos em documento apropriado, a modelagem do sistema através de diagramas UML, o projeto da arquitetura do sistema de forma detalhada, além do controle das mudanças nos requisitos e no escopo do projeto.

- **Gerente de Projeto:** O gerente de projeto é responsável por todas as atividades referentes ao gerenciamento e planejamento do projeto, entre os quais se incluem:

- Alocar recursos
- Estabelecer prioridades;
- Coordenar as negociações e acordos com usuários e demais áreas da empresa;
- Definir para cada membro da sua equipe, seus respectivos papéis e as permissões que são atribuídas a cada papel;
- Instituir o uso de uma ferramenta de controle de versão para garantir, de maneira automatizada, que somente alterações autorizadas sejam implementadas no sistema e em todos os itens de configuração. E assim, poder averiguar as mudanças entre o sistema e uma versão futura;
- Autorizar a atualização de uma versão de produção do sistema por uma nova versão; estabelecida pelo Gerente de Configuração. Qualquer mudança em versões de produção deve ser autorizada pelo Gerente do Sistema

O gerente de projeto é o responsável por garantir o planejamento, execução e controle do Processo de Desenvolvimento de Software durante todo o ciclo de vida do projeto.

- **Quality Assurrance:** É responsável pela garantia da qualidade no projeto. Suas ações estão relacionadas ao monitoramento tanto do processo quanto dos produtos gerados com a finalidade de identificar desvios ao planejado devendo relatar ao responsável para que ações corretivas sejam tomadas. Suas atividades são, portanto: investigar os documentos gerados, analisar o código do sistema e registrar as ocorrências.
- **Suporte**: É responsável pelo suporte ao sistema junto ao cliente, podendo ser um suporte remoto ou estar alocado no próprio cliente. Tem como responsabilidade receber as solicitações de manutenção e mudanças registrando de forma correta para gerar tickets para cada chamado. Trata-se do elemento de ligação entre o fornecedor e o cliente, o que lhe exige habilidades específicas de relacionamentos.

4.5.1.5 Atendimento aos Resultados Esperados

Como forma de elucidar o cumprimento dos resultados esperado pelo modelo, segue abaixo, um panorama, considerando-se para cada Resultado de Processo o que se espera, o que deve ser feito e como o modelo proposto cumpre esse resultado.

Resultado Esperado	**GR1 – O entendimento dos requisitos é obtido junto aos fornecedores de requisitos.**
O que fazer ?	*Verificar se os requisitos foram entendidos da mesma forma pelos desenvolvedores e cliente, a partir de documentos prévios ou elaborados com fornecedores de requisitos.*
O que o modelo proporciona	- Define um mecanismo de comunicação contínua com os fornecedores, tais como: reuniões, atas, revisão conjunta. - Identifica os fornecedores de requisitos. - Gera um documento que representa o entendimento dos requisitos.

Resultado Esperado	**GR2 – Os requisitos de software são aprovados utilizando critérios objetivos.**
O que fazer ?	*Com base em critérios pré-definidos deve-se aprovar ou não os requisitos levantados, tanto em termos tecnológicos, financeiros, de tempo, de recursos, entre outros.*
O que o modelo proporciona	*- Define um conjunto de critérios objetivos.* *- Realiza avaliação e aceitação dos requisitos baseados nos critérios.*

Resultado Esperado	**GR3 – A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida.**
O que fazer ?	*Deve-se manter, ao longo do projeto, um mecanismo que possibilite a rastreabilidade bidirecional entre os requisitos e demais produtos do projeto.*
O que o modelo proporciona	*- Criação de uma matriz de rastreabilidade.*

Resultado Esperado	**GR4 – Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos.**
O que fazer ?	*Garante que inconsistências sejam identificadas com base nos requisitos e que sejam realizadas ações para corrigi-las.*
O que o modelo	*- Monitoramento periódico da Equipe de Qualidade (QA) para identificar*

proporciona	*inconsistências e registrar em Laudos as revisões realizadas.*

Resultado Esperado	**GR5 – Mudanças nos requisitos são gerenciadas ao longo do projeto.**
O que fazer ?	*Durante todo o projeto, deve-se registrar os pedidos de mudanças nos requisitos e analisa o impacto da mudança. Essas ações devem estar registradas em um plano de gerência de requisitos.*
O que o modelo proporciona	*- Formalização das mudanças através de email e/ou formulários de solicitação.* *- Cria um plano de gerência de requisitos.*

4.5.2 Processo Padrão de Gerência de Projetos

Existe um consenso, entre os pesquisadores da área de engenharia de software, de que a gestão eficiente de projetos é imperativa na busca pela qualidade de resultados. MAGALHÃES (2005).

Esse conceito reflete bem a realidade mundial que apresenta alguns dados preocupantes. O Chaos Research, principal fonte de dados sobre gerência de projetos mostra que somente 33% dos projetos de software iniciados são de sucesso. STANDISH (2004).

Para o bom entendimento sobre gerência de projetos é interessante destacar o conceito de projeto. Segundo a ABNT (2001), projeto é “Processo único, consistido de um grupo de atividades, controladas com datas para início e término, empreendido para alcance de um objetivo conforme requisitos específicos, incluindo limitações de tempo, custo e recursos.”

Já para o *Project Management Institute*, um projeto pode ser definido, em termos de suas características distintivas “Um empreendimento temporário, planejado, executado e controlado, com objetivo de criar um produto ou serviço único.” PMBOK (2000).

Essas definições apresentadas por instituições que padronizam os conceitos relacionados a projetos seguem uma mesma linha da definição de projeto de software, que para Pressman (2005), um projeto de software para ser bem sucedido, é necessário que alguns parâmetros sejam corretamente analisados, como por exemplo, o escopo do software, as tarefas a serem realizadas, os riscos envolvidos, os recursos necessários, os indicadores a serem estabelecidos e cumpridos, os esforços e custos aplicados e a sistemática a ser seguida. O controle destes parâmetros caracteriza a função típica do gerenciamento de projetos a qual, de forma geral, se inicia antes do trabalho técnico e prossegue à medida que a entrega do software vai se concretizando.

Percebe-se então a importância da gerência de projetos, que de acordo com o PMBOK (2000), é a aplicação de conhecimentos, habilidades, ferramentas e técnicas nas atividades do projeto a fim de atender os requisitos do projeto. Pode ser mais bem compreendido através dos processos que o compõem, que podem ser reunidos em cinco grupos de processos: Iniciação, Planejamento, Execução, Controle e Encerramento. E também, em nove áreas de conhecimento: Gerenciamento da Integração do Projeto, Gerenciamento do Escopo do Projeto, Gerenciamento do Tempo do Projeto, Gerenciamento dos Custos do Projeto, Gerenciamento da Qualidade do Projeto, Gerenciamento dos Recursos Humanos do Projetos, Gerenciamento da Comunicação do Projeto, Gerenciamento dos Riscos do Projeto e Gerenciamento dos Fornecimentos de Bens e Serviços do Projeto.

4.5.2.1 Fluxo das Atividades

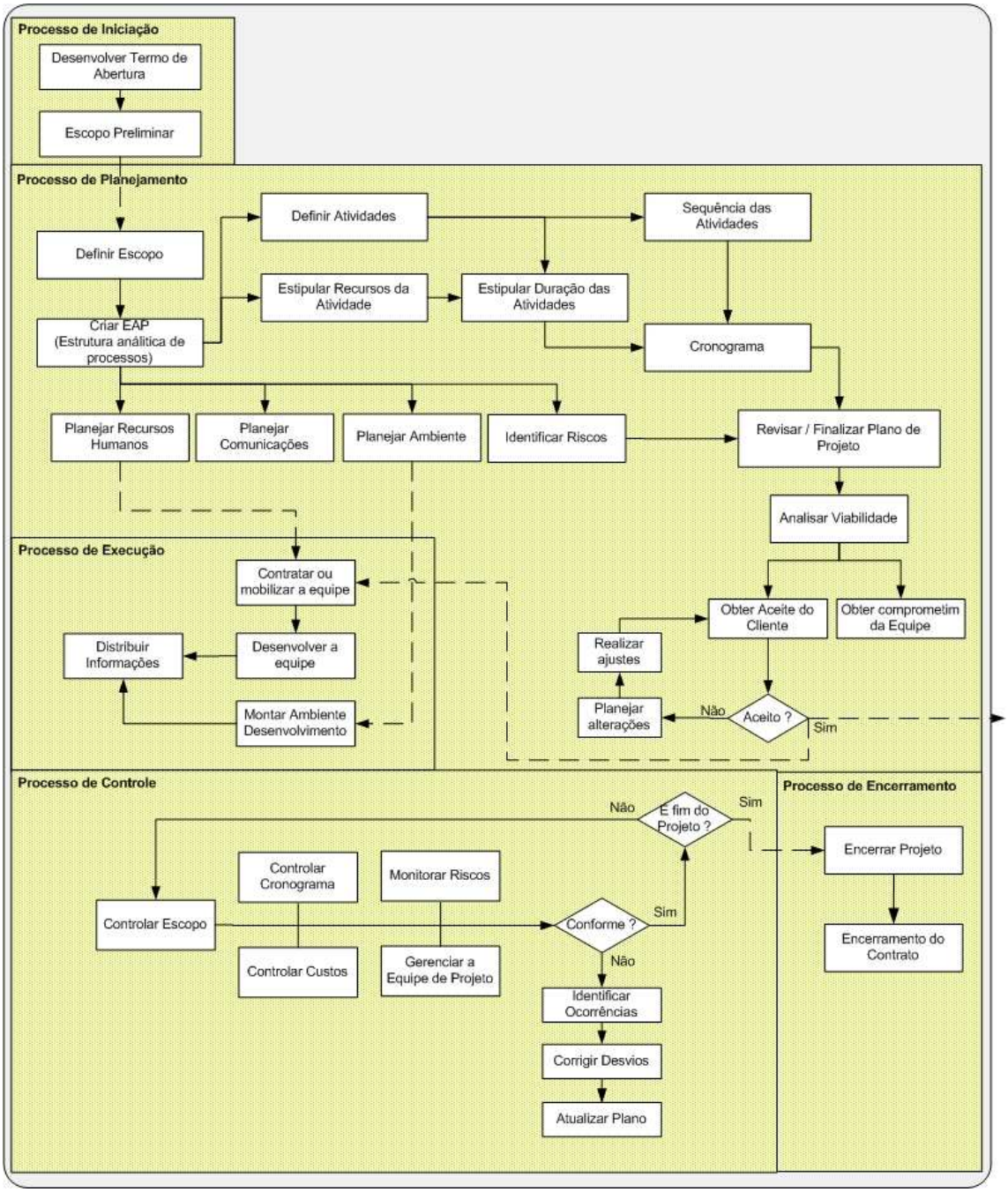

Figura 19. Fluxo de atividades para gerência de projetos.

O processo se inicia com a abertura do projeto que gera um escopo preliminar, já na fase de planejamento esse escopo é detalhado dando subsídio para que se crie uma estrutura analítica de processos (EAP), desencadeando na seqüência outras atividades como

planejamento de esforços, definição de estimativa de tempo e recursos, o que resulta em um cronograma. Esse cronograma juntamente com o planejamento de recursos humanos, planejamento de riscos, planejamento do ambiente de execução do projeto e planejamento das comunicações gera o Plano do Projeto que deverá estar de comum acordo com a equipe de desenvolvimento e com o cliente. A partir da aceitação do cliente são desencadeadas as tarefas de execução na formação e treinamento do time além da montagem do ambiente e divulgação de informações e tarefas. Após o planejamento e execução, é iniciada a fase de controle, na qual o gerente do projeto monitora constantemente riscos, custos, cronograma e a equipe, identificado divergências e tomando decisões para solucionar problemas. O fim do projeto é marcado pela assinatura do termo de aceite e o ato de encerramento do contrato.

Para esse processo padrão, algumas atividades são destacadas, conforme apresentado a seguir:

4.5.2.2 Atividades

Nome	Definir Escopo
Descrição	Esta atividade recebe o escopo preliminar e detalha em um documento oficial contendo descrição dos objetivos, metas e justificativa do projeto como um todo.
Responsável(is)	Gerente do Projeto
Participantes	Gerente do Projeto Recursos Humanos
Produtos Requeridos	Escopo preliminar
Produtos Gerados	Escopo Detalhado

Nome	Criar EAP
Descrição	Cria-se uma estrutura analítica de processos, através de um Brainstorm.
Responsável(is)	Gerente do Projeto
Participantes	Gerente do Projeto Equipe de Desenvolvimento
Produtos Requeridos	Escopo
Produtos Gerados	EAP – Estrutura Analítica de Processos

Nome	Planejar Recursos Humanos

Descrição	Faz-se um levantamento das necessidades de recursos humanos para o projeto. Deve-se estabelecer o perfil, habilidades e conhecimentos necessários, fazendo uma análise se a empresa possui este recurso ou deverá contratar.
Responsável(is)	Gerente de projeto
Participantes	Gerente de Projeto Recurso Humano
Produtos Requeridos	EAP – Estrutura Analítica de Processos
Produtos Gerados	Planilha de alocação de pessoal

Nome	Planejar comunicações
Descrição	Deve-se estabelecer a forma de comunicação entre os membros do projeto e entre o cliente, de forma a estabelecer um caminho para registro de solicitações, pedidos de maneira rápida.
Responsável(is)	Gerente do Projeto
Participantes	Cliente Gerente do Projeto
Produtos Requeridos	Escopo detalhado
Produtos Gerados	Plano de comunicação

Nome	Planejar Ambiente
Descrição	Identificar e descrever os recursos necessários para execução do projeto. Deve-se levar em consideração hardware, software, rede, servidores, entre outros.
Responsável(is)	Gerente do Projeto
Participantes	Gerente do Projeto Equipe de Infra-Estrutura
Produtos Requeridos	Escopo detalhado
Produtos Gerados	Planilha de necessidades de Recursos

Nome	Identificar Riscos
Descrição	Através do método de Moynihan, pretende-se identificar os principais riscos inerentes ao projeto e a probabilidade destes vierem a ocorrer.
Responsável(is)	Gerente de Projeto
Participantes	Gerente de Projeto
Produtos Requeridos	Escopo detalhado EAP
Produtos Gerados	Planilha de Riscos com análise de impacto.

Nome	Definir Atividades, Duração e Seqüência
Descrição	Com a EAP definida, faz-se uma estruturação das atividades de forma seqüencial atribuindo datas e responsabilidades.
Responsável(is)	Gerente do Projeto
Participantes	Gerente do Projeto

Produtos Requeridos	EAP
Produtos Gerados	Atividades seqüenciadas

Nome	Gerar Cronograma
Descrição	Faz-se uma análise das atividades, seus relacionamentos e duração, gerando assim um cronograma do projeto como um todo.
Responsável(is)	Gerente do Projeto
Participantes	Gerente do Projeto
Produtos Requeridos	EAP
Produtos Gerados	Cronograma

Nome	Revisar/Finalizar Plano do Projeto
Descrição	O plano do projeto compreende a unificação de todos os outros planos (RH, Riscos, Ambiente, cronograma, comunicação) de forma a estabelecer formalmente o planejamento do projeto.
Responsável(is)	Gerente do Projeto
Participantes	Gerente do Projeto Cliente Equipe
Produtos Requeridos	Escopo, Plano de RH, Plano de Risco, Plano de Comunicação, Plano de Ambiente e Cronograma.
Produtos Gerados	Plano de Projeto

Nome	Analisar Viabilidade
Descrição	Nesta etapa fundamental para o projeto deverá ser realizado o estudo de viabilidade do projeto, de forma a identificar se a empresa possui tempo para o projeto, se possui recursos, se existe domínio da tecnologia para concluir o projeto.
Responsável(is)	Diretor de Tecnologia
Participantes	Diretor de Tecnologia Gerente de projeto
Produtos Requeridos	Plano de Projeto
Produtos Gerados	Estudo de Viabilidade (Planilha).

4.5.2.3 Papéis

- **Analista de Sistemas:** É responsável pelas atividades de análise e projeto do sistema, incluindo o levantamento dos requisitos junto aos *stakeholders*, a especificação dos requisitos em documento apropriado, a modelagem do sistema através de diagramas

UML, o projeto da arquitetura do sistema de forma detalhada, além do controle das mudanças nos requisitos e no escopo do projeto.

- **Gerente de Projeto:** O gerente de projeto é responsável por todas as atividades referentes ao gerenciamento e planejamento do projeto, incluindo:
    - Alocar recursos
    - Estabelecer prioridades
    - Coordenar as negociações e acordos com usuários e demais áreas da empresa
    - Definir para cada membro da sua equipe, seus respectivos papéis e as permissões que são atribuídas a cada papel
    - Instituir o uso de uma ferramenta de controle de versão para garantir, de maneira automatizada, que somente alterações autorizadas sejam implementadas no sistema e em todos os itens de configuração. E assim, poder averiguar as mudanças entre o sistema e uma versão futura
    - Autorizar a atualização de uma versão de produção do sistema por uma nova versão estabelecida pelo Gerente de Configuração. Qualquer mudança em versões de produção deve ser autorizada pelo Gerente do Sistema

O gerente de projeto é o responsável por garantir o planejamento, execução e controle do Processo de Desenvolvimento de Software durante todo o ciclo de vida do projeto.

- **Quality Assurrance:** É responsável pela garantia da qualidade no projeto. Suas ações estão relacionadas ao monitoramento tanto do processo quanto dos produtos gerados com a finalidade de identificar desvios ao planejado devendo relatar ao responsável para que ações corretivas sejam tomadas. Suas atividades são, portanto: Investigar documentos gerados, analisar código do sistema e registrar ocorrências.

- **Recursos Humanos:** Profissional com a responsabilidade de apresentar ao gerente do projeto os colaboradores atuais da empresa, seus perfis habilidades e disponibilidade.

- **Infra-Estrutura:** Pessoa ou grupo de pessoas que tem como responsabilidade planejar e montar a infra-estrutura para execução do projeto. Compreende esta atividade a instalação de máquinas, softwares, rede, e outros dispositivos necessários. Deve também monitorar cotidianamente o funcionamento desses dispositivos para manter o bom funcionamento.

4.5.2.4 Artefatos

- **Plano de Projeto:** O documento de Plano de Projeto estabelece as principais funcionalidades e fronteiras do sistema e registra informações que delimitam o projeto de desenvolvimento como um todo. Este artefato tem a função de um contrato entre os diferentes interessados no sistema (usuário gestor, usuário final, etc.), registrando aquilo que será desenvolvido ao longo do projeto. Deve ser formalmente aprovado pelas partes envolvidas.
- **Escopo**: Esse documento tem por objetivo descrever uma visão inicial para o projeto, definindo os interessados, objetivos, justificativa e uma descrição do que o sistema irá fazer. Recomenda-se que seja um documento curto de 3 a 5 páginas.
- **Cronogama:** O cronograma é a disposição gráfica do tempo que será gasto na realização de um trabalho ou projeto, de acordo com as atividades a serem cumpridas. Serve para auxiliar no gerenciamento e controle deste trabalho, permitindo de forma rápida a visualização de seu andamento.
- **Plano de Recurso Humano:** Todos as necessidades de recursos humanos devem ser identificadas através de habilidades e perfil desejáveis, além de uma analise se a empresa já possui esses recursos ou se terá que contratar ou treinar.

- **Plano de Risco:** Documento que contempla uma análise de todos os riscos que um projeto de TI (Tecnologia da Informação) pode enfrentar. Nesse documento, deve existir para cada risco, a probabilidade e a importância, além de medidas a serem tomadas em caso de ocorrência.
- **Laudos das revisões realizadas:** Formulário padrão no qual a Equipe de que Qualidade periodicamente registra todas os desvios encontrados entre os produtos gerados e a descrição do planejado, enviando aos responsáveis para que medidas sejam tomadas.

4.5.2.5 Atendimento aos Resultados Esperados

Como forma de elucidar o cumprimento dos resultados esperado, segue um panorama apresentando o resultado que o MR-MPS espera, o que deve ser feito e como o modelo proposto cumpre esse resultado.

Resultado Esperado	GPR1: O escopo do trabalho para o projeto é definido
O que fazer ?	*Deve-se criar um documento de escopo seguindo uma padrão previamente definido.*
O que o modelo proporciona	- Define um escopo do projeto com a visualização do trabalho a ser realizado.

Resultado Esperado	GPR2: As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
O que fazer ?	*Estimar o esforço através de um método definido pela empresa.*
O que o modelo proporciona	*- O modelo proporciona a utilização de bases históricas ou métodos como COCOMO ou Pontos de função baseados em uma EAP (Estrutura Analitica de Projeto).*

Resultado Esperado	GPR3: O modelo e as fases do ciclo de vida do projeto são definidas;
O que fazer ?	*Definir um ciclo de vida apropriado para o projeto em questão com a descrição das atividades.*
O que o modelo proporciona ?	*- Define o ciclo de vida do projeto, indicando suas fases, as relações de seqüência, a interdependência entre as fases e os marcos e pontos de controle do projeto.*

Resultado Esperado	GPR4: O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;
O que fazer ?	*Realiza estimativas de custo e esforço para tarefas e produtos de trabalho com base em dados históricos ou métodos de estimativas .*
O que o modelo proporciona ?	*- Criação de um plano de custos (financeiro ou em horas)* *- Criação de um cronograma planejado com horas de trabalho*

Resultado Esperado	GPR5: O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;
O que fazer ?	*Definir o cronograma com dependência entre tarefas e o orçamento.*
O que o modelo proporciona ?	*- Criação de um cronograma contendo caminho crítico e dependência entre as tarefas.* *- Revisão do cronograma e orçamento, conforme a necessidade ao longo do desenvolvimento.*

Resultado Esperado	GPR6: Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
O que fazer ?	*Utilizar um check-list ou método para identificação dos riscos inerentes a um projeto de tecnologia.*
O que o modelo proporciona ?	*- Utilização de uma lista dos riscos para o projeto, caracterizando seu impacto.* *- Analisa riscos, determina o grau de importância, probabilidade e prioridade de cada risco.*

Resultado Esperado	GPR7: Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;
O que fazer ?	*Criar um plano de recursos humanos*
O que o modelo proporciona	*- Selecionar Recursos humanos a partir das competências requeridas para realizar as atividades do projeto.*

Resultado Esperado	GPR8: As tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados
O que fazer ?	*Criar um plano de infra-estrutura necessária para o projeto, identificando recursos.*
O que o modelo proporciona	*- Planejar a quantidade de recursos e a infra-estrutura necessária para cada tarefa a parir da EAP.*

Resultado Esperado	GPR9: Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
O que fazer ?	*Deve-se definir um plano de gerência de dados listando todos os documentos gerados no projeto, sua distribuição e mídia para armazenamento, proteção (segurança e sigilo) e recuperação desses dados.*
O que o modelo proporciona	*- Criação de um plano de documentação. Tanto para documentos eletrônicos ou não.*

Resultado Esperado	GPR10: Planos para a execução do projeto são estabelecidos e reunidos no Plano do Projeto
O que fazer ?	*Compor o plano de projeto com base nas informações de planejamento do projeto organizadas e relacionadas.*
O que o modelo proporciona	*- Criação de um Plano de Projeto*

Resultado Esperado	GPR11: A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados
O que fazer ?	*Avaliar a viabilidade do projeto, a partir da visão geral dos objetivos e características dos resultados pretendidos, dos recursos financeiros, técnicos, humanos, bem como restrições impostas pelo cliente, ambiente externo e interno e condições de desenvolvimento.*
O que o modelo proporciona	*- Documento de avaliação da viabilidade com critérios estabelecidos*.

Resultado Esperado	GPR12: O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido
O que fazer ?	*Garantir que todos os interessados tomaram conhecimento e se comprometeram com o planejamneto do projeto.*
O que o modelo proporciona	*- Registro do comprometimento com o plano de projeto atavés de assinatura das partes envolvidas.*

Resultado Esperado	GPR13: O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados
O que fazer ?	*Monitorar o projeto, comparando o planejado com o realizado.*
O que o modelo proporciona	*- Relatório de monitoração do projeto.*

Resultado Esperado	GPR14: O envolvimento das partes interessadas no projeto é gerenciado
O que fazer ?	*Planejar o tipo de envolvimento dos interessados no projeto registrando as atividades e evidenciando o envolvimento*
O que o modelo proporciona	*- Reunião com os envolvidos para definição dos papéis e obtenção de assinatura na Ata de reunião.*

Resultado Esperado	GPR15: Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento
O que fazer ?	*Realizar revisões do projeto em marcos previamente estabelecidos*
O que o modelo proporciona	*Geração de laudos das revisões realizadas.*

Resultado Esperado	GPR16: Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas
O que fazer ?	*Identificar e analisar problemas das monitorações*.
O que o modelo proporciona	*- Lista de problemas identificados.*

Resultado Esperado	GPR17: Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão
O que fazer ?	*Definir ações corretivas e acompanhar essas açõea até sua conclusão*
O que o modelo proporciona	*- Plano de ação corretiva com informações de acompanhamento da execução das ações.*

5. APLICAÇÃO DO MODELO: UMA PESQUISA-AÇÃO

Após a idealização de um adequado modelo de referência, procedeu-se a uma aplicação em um projeto real para validar o modelo e mensurar os ganhos com sua implantação em uma empresa de pequeno porte.

A escolha da empresa se justifica pelos seguintes fatores:

- Acessibilidade, isto é, a empresa possui intenção de melhorar seus processos e para tanto, disponibilizou recursos para esse projeto;
- Resultados rápidos;
- O fato de a empresa possuir projetos de diferentes tamanhos em diferentes áreas de atuação, como parte do seu posicionamento junto ao mercado.

5.1 A empresa

A COSS Soluções e Tecnologia Ltda é uma empresa brasileira, fundada em 2005 com sua sede instalada no Pólo de Alta Tecnologia de São Carlos/SP e um escritório em São Paulo/SP. A empresa possui três sócios e conta atualmente com 23 colaboradores sendo 17 deles diretamente ligados a desenvolvimento de software.

Está voltada para atuar na criação, desenvolvimento e implementação de produtos e serviços de alto valor agregado na área de Gestão & Tecnologia da Informação. Atua junto a empresas nacionais e multinacionais do segmento de indústria, serviços e comércio. Trabalha com tecnologias e soluções de classe mundial para entregar o maior valor aos seus clientes e parceiros de negócios, globalmente, como: SAP, Oracle, Microsoft, Java, JBoss, embedded C,

C++, .NET, SAP Netweaver, em ambientes computacionais heterogêneos incluindo dispositivos móveis.

Em 2006, iniciou suas atividades com a prática de implementação de soluções SAP CRM (*Customer Relationship Management*) visando aprofundar o conhecimento dos processos de negócios centrados no cliente. Nessa linha, desenvolveu serviços e metodologias com recursos próprios, implementando de forma objetiva e pragmática esta solução. Construiu relações empresariais, desenvolveu parcerias de negócios com empresas globais e trabalhos colaborativos, treinou profissionais, executivos e empresários, e conquistou financiamento FAPESP para desenvolvimento de plataforma de integração baseado em tecnologia RFID (WELCOSS) com o objetivo de desenvolver soluções próprias e prover ao mercado produtos e serviços de alto valor agregado, nesta área. Esta solução está preparada para atender empresas de diferentes segmentos de mercado como: Automotiva, Química, Bens de Consumo, Varejo, Hospitalar, Governo, Alta-Tecnologia, Aeronáutica, Papel & Celulose, Energia e Manufatura em geral, entre outras.

Possui trabalhos em diversas regiões do Brasil e América Latina. Em 2007, a empresa consolidou a prática de CRM e constituiu o ID SMART Center (lançado em 14/02/2007), trata-se de um centro de competência em tecnologia RFID, preparando mão-de-obra para trabalhar com os mais diferentes dispositivos tecnológicos e soluções de mercado existente nesta área.

A necessidade de poder contar com um modelo de referência estruturado de desenvolvimento de software se dá pela exigência de qualidade por alguns clientes nacionais e internacionais e que fez, de início, com que a empresa adotasse o MPS.Br como estratégia para futuramente implantar o modelo CMMi.

5.2 Estruturação da metodologia de trabalho na empresa

O trabalho de implantação de melhoria de processos na empresa teve início em maio de 2007, com apóio irrestrito da alta administração. Foi formado um grupo chamado de Grupo de Controle da Qualidade (GCQ), constituído por 2 membros, sendo que 1 deles teria dedicação exclusiva a essa atividade durante 15 meses.

A estruturação da metodologia de trabalho na empresa resultou na elaboração de alguns documentos importantes como a Política Organizacional que disserta sobre as questões que se relacionam aos processos de software e que definem a forma de implantação dos dois processos do nível G do MR-MPS: Processo de Gerência de Requisitos e Processo de Gerência do Projeto.

Também foram organizados documentos que padronizam a codificação nas linguagens de programação dos softwares produzidos pela empresa, *templates* de ata de reunião, modelos de documentos para registro de ocorrências, planilha para controle de horas de desenvolvimento, plano de documentação, entre outros.

O fluxo de atividades a seguir, apontado na figura 20, representa os passos para implantação dos dois processos na empresa ao longo desse período.

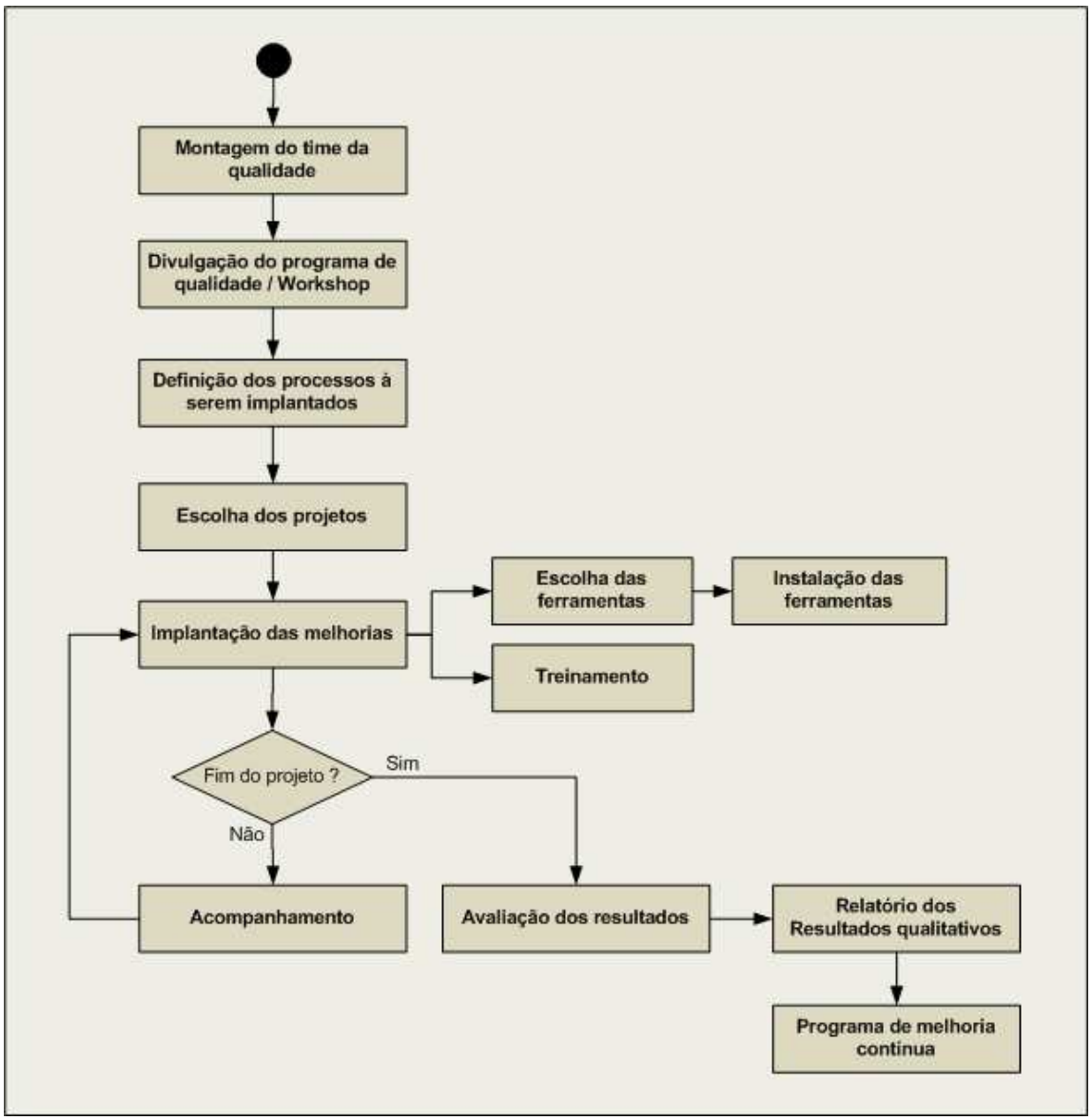

Figura 20. Fluxo de trabalho na implantação do processo.

Faz-se abaixo algumas considerações sobre a implantação dos dois processos do nível G do MPS.Br

5.2.1 Implantação do Processo de Gerência de Requisitos

O processo de Gerência de Requisitos foi o primeiro processo a ser implantado e teve como embasamento essencial a elaboração da documentação e o estabelecimento das

definições resultantes do momento inicial da implantação do nível G e da estruturação da metodologia de trabalho da empresa. A implantação desse processo levou 5 meses.

Todas as ações realizadas para a implantação desse processo foram voltadas ao atendimento de cada um dos resultados esperados na Guia Geral do MPS.Br.

É valido destacar que a empresa já possuía atividades relacionadas a elicitação de requisitos, porém não formalmente. Não existia um documento padrão nem uma forma adequada para selecionar um método de coleta de requisitos, inclusive em alguns projetos não existia evidências de aceitação dos requisitos por parte do cliente, as solicitações eram muitas vezes por telefone, de maneira verbal. Os critérios para aprovação dos requisitos eram tomados pela pressão que o cliente exercia.

A implantação do processo de Gerência de Requisitos seguiu uma metodologia gradual que consistiu em: entender os fluxos de atividades proposto pelo modelo de referência, adaptar os fluxos, elaboração dos documentos e modelos-padrão, instalação e adaptação das ferramentas de apoio para disseminação do processo, treinamento da equipe e acompanhamento da execução nos projetos estudados.

Entre as ferramentas instaladas destacam-se as ferramentas: TRAC System para rastreabilidade de ativos e controle de atividades, o Subversion (SVN) para controle de versão de código fonte e documentação e o MS-SharePoint para disseminação de documentação e informações. Outro ponto de destaque esteve relacionado a padronização dos documentos, sendo declarados no Plano de Documentação da empresa.

As dificuldades encontradas durante essa adaptação esteve relacionada ao fato do processo de gerência de requisitos estar integrado e depender de outros processos, como por exemplo, o processo de suporte, o processo de desenvolvimento de requisitos, entre outros. Sendo assim, entende-se que para melhorar o processo de gerência de requisitos outros

processos do MR-MPS, principalmente do nível F e E foram parcialmente implementados, o que ajudará futuramente no programa de melhoria contínua.

5.2.2 Implantação do Processo de Gerência de Projetos

Com o processo de gerência de requisitos já em andamento, teve início a implementação do processo de gerência de projetos.

Esta nova fase já dispunha de diversos pontos favoráveis, adquiridos nas etapas preparatórias e ao longo da implantação da fase de gerência de requisitos, ou seja, já se percebia uma predisposição e um comprometimento da equipe para avançar em sua atuação. Entretanto, mesmo assim, esta segunda fase levou 6 meses e envolveu mais diretamente a equipe de qualidade que baseada no modelo proposto, adaptou-a para as necessidades demandadas pela organização. Cabe destacar que a empresa não dispunha de nenhum gerente formado em cursos específicos de gerenciamento de projetos, o que exigiu, como parte indispensável, investimentos na capacitação de dois de seus gerentes, em curso específico de 35 horas de duração.

Assim como o processo de gerência de requisitos, todas as ações realizadas para a implantação do processo foram voltadas ao atendimento de cada um dos resultados esperados no guia geral do MPS.Br, conforme proposto no modelo. Neste aspecto cabe destacar a influência marcante do PMI e mais especificamente do PMBok (*Project Manager Body of Knowledgement*), como práticas já assimiladas pela equipe.

A melhoria do processo seguiu uma metodologia semelhante a do processo de gerência de requisitos, na qual buscou-se um entendimento dos fluxos de atividades proposto pelo modelo. Além disso procedeu-se a adaptação desses fluxos, a elaboração dos documentos e a criação de *templates.* Adicionalmente, foi feito uso e adaptação das ferramentas de apoio,

como o MS-Project e o Trac System já instaladas na empresa, e o treinamento da equipe através de curso de gerência de projetos. Para os desenvolvedores, foi realizado o treinamento através de workshop interno e finalmente, a etapa de acompanhamento da execução das atividades ocorreu, monitorando-se e tomando ações corretivas quando desvios aconteciam.

Com relação ao processo de gerência de projetos, percebeu-se que a empresa possuía boas práticas de planejamento e controle de projetos, porém essas ações eram mínimas e não atendiam às exigências do modelo. Como exemplo, cabe citar o uso do MS-Excel para definição de cronograma, o que ocasionava perda de tempo e dificuldade de acompanhamento. Não existia controle de versão dos documentos gerados, uma vez que os documentos eram feitos para cada projeto, de uma determinada forma e ficava armazenado no computador do gerente, sem a preocupação de se estabelecer sistematização.

Algumas iniciativas foram primordiais para que se pudessem alcançar as metas propostas no modelo de referência. O uso da técnica de análise de risco chamada de MOYNIHAN foi de significativa contribuição. O modelo Moynihan avalia o risco de um projeto de TI pela estimativa da intensidade de 23 fatores de risco sobre 4 fatores de sucesso. SCHMITZ (2007).

Outro ponto a ser destacado foi o fato de que após a aprovação do Plano do Projeto, a equipe se reunia para esclarecimentos dos papéis de cada membro. Dessa reunião é gerada uma Ata na qual todos assinam, concordando com sua função e responsabilidade individuais, o que formaliza seus compromissos.

Os planos de Recursos Humanos, Comunicação e de Infra-Estrutura são facilitados em sua divulgação com o uso de planilhas pré-definidas, nas quais os responsáveis apenas preenchem suas necessidades, que serão analisadas sob uma ótica realística.

Entre as dificuldades verificadas cabe destacar a necessidade de se atuar sob a tensão gerada pelas imposições de prazo impostas pelos clientes e pela já aceita, força do mercado. Essa pressão, muitas vezes percebida como “normal”, repercute no ambiente interno da empresa e passa a justificar uma série de comportamentos que levam a decisões intempestivas, justificadas pela pressa. Esse comportamento encobre falhas maiores como a não previsibilidade, a atuação em tempo e por antecipação, o que diminuiria ‘riscos’ e reduziria custos, e pior, a ‘urgência’ passa se justificar decisões, muitas delas, inadequadas ou descabidas. O pior efeito entretanto é a formação de um comportamento que passa a justificar o negligenciamento de atividades primordiais de planejamento, criando uma cultura de improvisação. No caso em estudo, essa dificuldade foi superada por força do treinamento e da difusão da importância do planejamento, e pelo indispensável apoio da alta gerência e isso transpareceu diretamente nos resultados que passaram a ser percebidos.

A disponibilização das informações para toda a equipe durante o desenrolar do processo foi um ponto importante para que todos entendessem sua participação e importância. Cabe lembrar que esta é uma das práticas bem sucedidas e amplamente conhecida do sistema Toyota de Produção, sem dúvida, uma importante referência na produção enxuta e ágil. A seguir será apresentado o procedimento de disponibilização adotado.

5.5 Disponibilização do modelo de processo na empresa

Por meio da utilização do software MS-SharePoint, foi disponibilizado o processo consensualmente definido e também, o documento oficial que define o Plano de Processo de Desenvolvimento de Software. Sendo assim, todos os colaboradores envolvidos, direta ou indiretamente no desenvolvimento passam a ter acesso a essa documentação. Como ferramenta de visualização foi adotado o MS-SharePoint, que é um meio de comunicação e de

colaboração para aplicações intranet, no qual são gerenciados conteúdos de forma rápida e segura. A figura 21, ilustra a tela do sistema na qual o processo é disponibilizado.

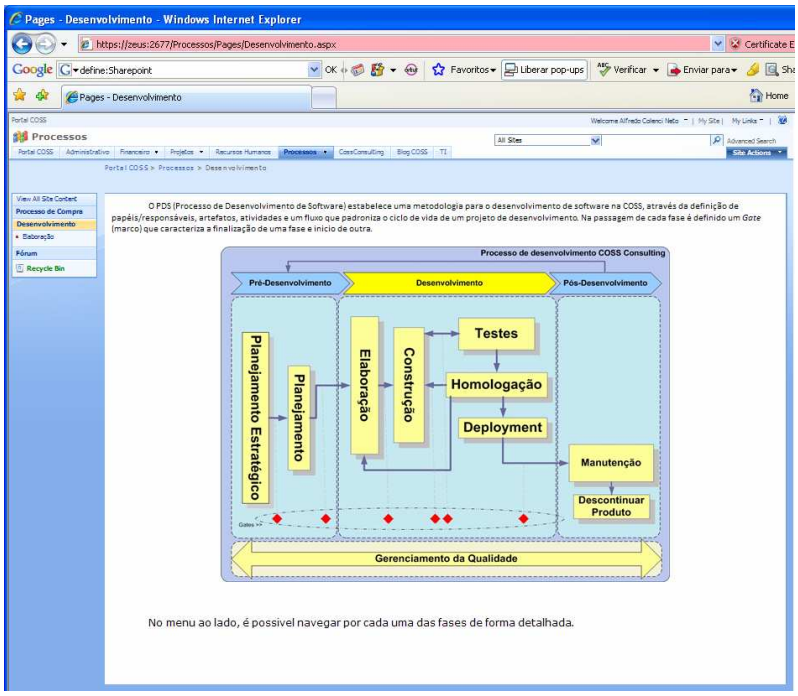

Figura 21. Tela do Share-Point.

Escolhido o meio de comunicação, foi necessário que os membros da empresa fossem treinados para seu uso. O treinamento ocorreu em dois momentos. Primeiramente, através de um workshop de 3 horas para mostrar uma visão geral do processo e seu uso dentro do SharePoint. Posteriormente foi realizado um treinamento de 4 horas, especificamente voltado para uso da ferramenta. Uma etapa simples, de baixo custo e de pouca duração mas indispensável.

Como resultado do esforço para institucionalizar o modelo de referência, a partir da implantação de dois processos: gerenciamento do projeto e gerenciamento de requisitos, partes significativas e representativas de sua aplicabilidade, foram selecionados dois projetos

desenvolvidos na empresa e que puderam posteriormente mostrar os resultados alcançados. Durante as duas aplicações foram identificadas melhorias, que imediatamente foram adaptadas no processo. Os dois projetos selecionados são descritos a seguir.

5.6 O modelo em aplicação: a mensuração dos resultados

Como se estabelecem nesta pesquisa, para corroborar a efetividade do modelo proposto, foram escolhidos dois projetos reais em uma pequena empresa produtora de software, executados nos anos 2007/2008. A escolha desses projetos se justifica, principalmente, pelos fatores: tempo de projeto relativamente curto para retorno dos dados, por terem sido os primeiros projetos da empresa depois da institucionalização do programa de melhoria e também por terem sido projetos externos, onde a exigência por qualidade é maior se comparado a projetos internos. Como metodologia de validação foi adotada a pesquisa-ação, descrita por Thiollent (1988), pelas razões já apresentadas neste trabalho.

5.6.1 Sistema iKanban

O software iKanban faz parte de um sistema de informação voltado a área de Logística e Produção para Planejamento e Controle da Produção (PCP). Trata-se de um software que controla a produção de forma automatizada utilizando a tecnologia de Identificação por Rádio Freqüência (RFID).

O projeto, encomendado por um cliente, teve um tempo aproximado de 7 meses de duração e envolveu o esforço de 9 colaboradores, entre gerentes, desenvolvedores e suporte.

O Cliente utiliza o sistema Kanban como forma de controlar seu fluxo de mercadoria e reposição de estoque, porém, quebras de cartões, perdas e falta de visão ampla do processo geravam ruptura constante no abastecimento da linha de produção.

Implantou-se então um sistema chamado WELCOSS iKanban que compreendia no uso de Tag´s RFID de 915Mhz, padrão Gen 2, acopladas aos 2.200 Kanbans da produção e antenas estrategicamente localizadas na planta da empresa. Com esse sistema pode-se rastrear e gerenciar as movimentações de material, desde o recebimento de matérias primas e componentes até a saída, para seu uso na linha de produção e conseqüentemente, visualizar a localização de cada item, em tempo real. O sistema possui interface Web, disponível na intranet da empresa e integração com o ERP (*Enterprise Resource Planning*) atualmente utilizado. A comunicação entre o ERP e o iKanban pode ser feita através da tecnologia de *Web Service*, troca de arquivos, entre outras.

A figura 22 ilustra a arquitetura da solução desenvolvida.

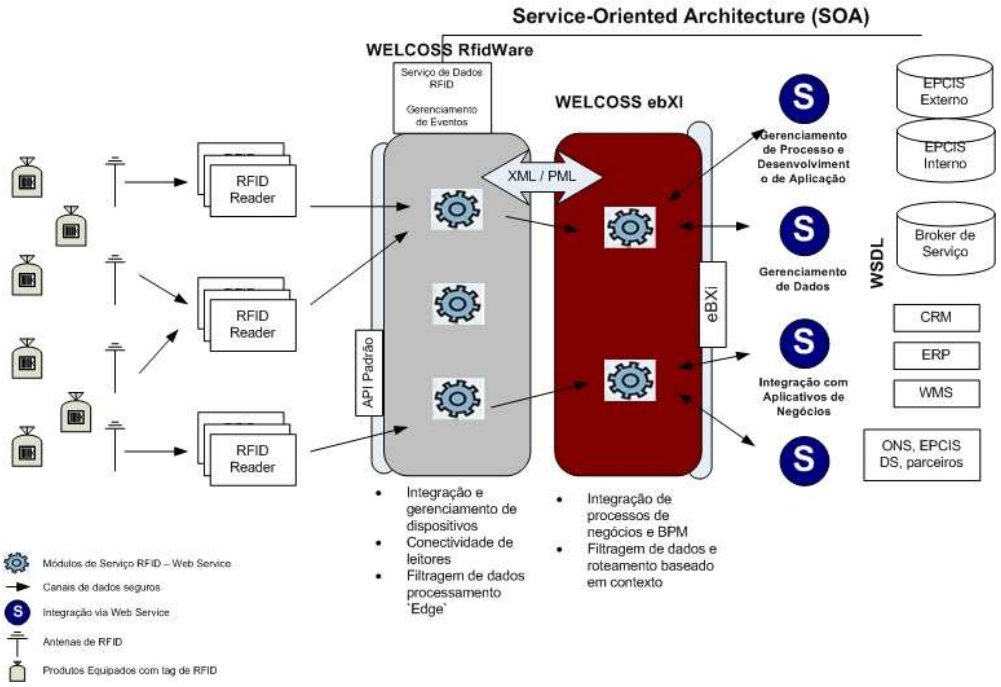

Figura 22. Arquitetura da solução WELCOSS.

Conforme relatado por COLELLA (2008), o funcionamento do sistema se dá pelo acoplamento de todos os kanbans à uma Tag de RFID. O processo se inicia quando uma mercadoria chega na empresa juntamente com a nota fiscal. No instante em que a nota fiscal é cadastrada no sistema de gestão (ERP), o Portal de RFID do recebimento é imediatamente avisado das mercadorias que irão chegar. Quando a empilhadeira passa com essas mercadorias pelo portal de recebimento, automaticamente o sistema realiza a conferência dos volumes enviando o resultado da conferência para o ERP, contendo todas as informações sobre os produtos realmente recebidos e os produtos faltantes. Quando esse produto deixa a área de recebimento para ser consumido na linha de produção a movimentação dos kanbans é também lida pelo Portal RFID, que automaticamente atualiza as informações no Banco de Dados do sistema. O mesmo ocorre quando o cartão (kanban) é levado para o quadro kanban após o item ter sido consumido na linha. Sendo assim, um usuário do sistema pode, a partir do computador remoto, visualizar de forma rápida e fácil a localização de cada item dentro da empresa e no fornecedor.

O projeto teve um cronograma aproximado de 7 meses e logo nos primeiros dias de funcionamento os benefícios foram percebidos com a imediata detecção de itens faltantes, conferência de divergências entre o que era comprado e o que de fato era recebido, antecipação do abastecimento da linha de montagem antes da parada, visualização remota do quadro kanban para todos os interessados. A partir dos dados gerados pelo sistema, torna-se possível estimar melhor a quantidade pedida de cada item, visando reduzir os estoques e minimizando o risco de rupturas na linha de produção, pela antecipação de informações proporcionada pelo sistema.

A figura 23 apresenta o fluxo de kanbans na empresa.

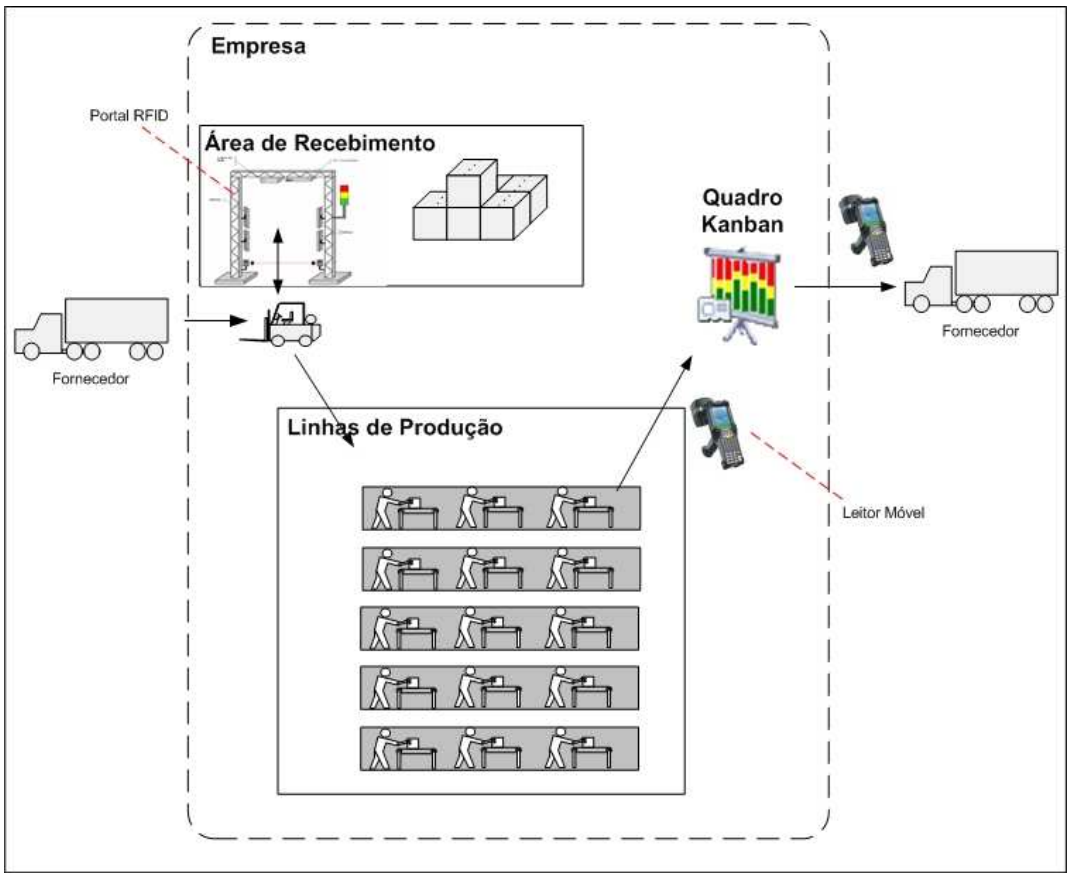

Figura 23. Fluxo de kanbans na empresa.

Os desafios do projeto estavam direcionados ao pioneirismo do uso da tecnologia de RFID no Brasil o que elevou os riscos do projeto como um todo, pois não havia referência do seu bom uso.

Por ter sido o primeiro projeto da empresa utilizando o modelo de referência alguns pontos foram marcantes, entre eles a dificuldade da equipe em seguir com disciplina, os modelos de documentos previamente estabelecidos, o que foi sendo solucionado com um acompanhamento intenso da equipe de qualidade e pela vontade manifesta da alta direção.

5.5.2 Sistema DVR

Este é o segundo caso, no qual o modelo de referência pode ser testado. Trata-se de um sistema para armazenamento de textos em gravações de vídeos. O projeto teve duração de 3 meses, sendo iniciado em junho e seu término em outubro de 2008. Ao todo envolveu 6 colaboradores.

O escopo do sistema envolvia a transferência de dados entre pontos de vendas (POS) e um sistema de gravação de vídeo digital (DVR), de forma que ao efetuar uma compra em um caixa de supermercado por exemplo, o sistema iria associar a informação da compra (texto) com a imagem capturada por uma câmera. Esse sistema está voltado para segurança e visa detectar furtos nos quais o próprio colaborador responsável pela venda efetua fraudes. Pode, adicionalmente ser ampliado para estudos de comportamento do consumidor no ato da compra.

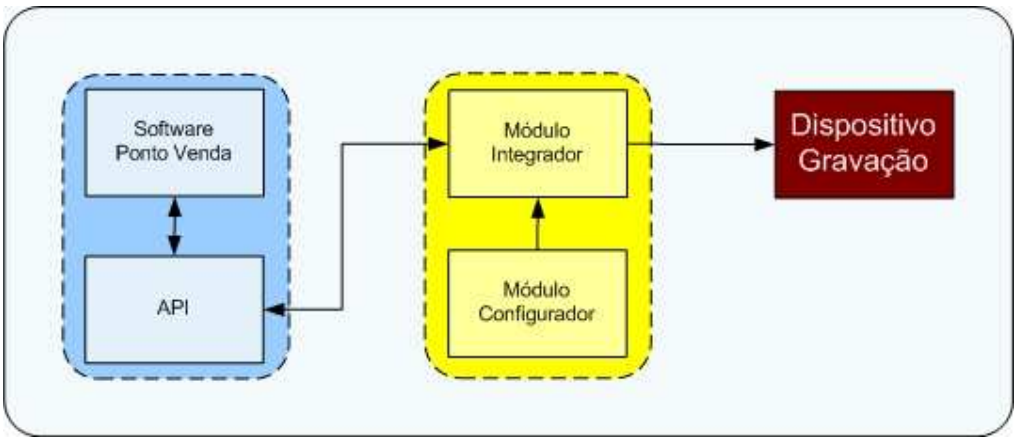

Figura 24. Arquitetura do sistema DVR.

Como característica favorável deste projeto, pode-se destacar o escopo fechado no qual o cliente conhecia exatamente as necessidades do sistema. Por outro lado a complexidade e a necessidade de desenvolvimento de código fonte em baixo nível dificultaram as

estimativas e execução do projeto. Foram desenvolvidos códigos em diversas linguagens de programação, entre elas: C#, C, C++ Builder, VB, Delphi e Kylix, tanto para rodar em plataforma Linux (Kernel 2.4) quanto Windows (2000, XP e Vista).

O quadro 9, a seguir apresenta um resumo dos projetos considerados e analisados para essa aplicação específica, bem como o tempo despedido na análise.

Quadro 9: Resumo dos projetos analisados

	PROJETO 1	PROJETO 2
Tempo	7 meses	3 meses
Pessoas Envolvidas	9	6
Área de Aplicação	Logística e Produção	Segurança
Complexidade	Alta	Média
Linguagens de Programação Utilizadas	Java, C	C#, C, Delphi, Kylix, VB, C++ Builder.
Dificuldade em implantar o modelo de referência	Alta	Baixa

5.7 Mensuração das melhorias

Descritos os dois casos escolhidos para a aplicação do modelo de referência proposto e feita sua aplicação efetiva, conforme descrito, a etapa seguinte foi a de se mensurar os resultados decorrentes, que poderiam ter aspectos positivos ou negativos e qualitativos e quantitativos. Neste sentido, prosseguiu a pesquisa através da aplicação de um questionário junto aos colaboradores envolvidos no desenvolvimento dos dois projetos. Alguns colaboradores atuaram nos dois projetos, outros só em um deles, entretanto, não se percebeu diferenças significativas em sua manifestação.

O questionário foi elaborado especificamente para este propósito e se encontra no anexo B deste trabalho. São 15 questões relacionadas a qualidade e produtividade do processo

com respostas na escala Likert, que varia de 1 a 5, sendo: 1 – discordo totalmente, 2 - discordo, 3 – indiferente, 4 - concordo e 5 - concordo totalmente.

Foram coletados dados de 9 colaboradores da empresa e o tempo de duração dessa fase da pesquisa foi de 1 dia, somando um total aproximado de 6 horas. Para a tabulação dos resultados utilizou-se do software MS-Excel e para geração dos gráficos utilizou-se do software MS-PowerPoint.

5.7.1 Resultados

De acordo com os dados levantados nesta pesquisa junto aos membros participantes dos dois projetos, pode-se perceber uma alto grau de satisfação relacionado aos processos implementados.

No início do formulário foi solicitado ao colaborador que escrevesse duas palavras que representassem o programa de melhoria como um todo. Esta foi a maneira encontrada para compreender o sentimento de cada membro participante quanto ao programa. As palavras descritas foram:

Definição	**Produtividade**
Qualidade	**Qualidade**
Qualidade	**Satisfação**
Integração	**Perfeição**
Excelência	**Melhoria**
Confiabilidade	**Documentação**
Comunicação	**Pioneirismo**
Colaboração	**Administração**

` Pode-se entender que todos os membros responderam palavras relacionadas a melhoria, satisfação, entre outras, o que nos dá uma compreensão de algo positivo.

Essa satisfação é comprovada na análise das respostas das 15 questões, visto que 86% concordaram totalmente quando perguntando se houve melhoras na execução do projeto. Outro ponto a ser destacado se relaciona a mudança na forma de trabalho, na qual 57% se mostraram satisfeitos com a melhora na documentação, 100% disseram que houve agilidade nos produtos gerados e 86% acreditam que a qualidade do produto final foi melhor se comparada com outros projetos.

Abaixo faz-se uma análise mais detalhada de cada umas das 15 perguntas feitas aos membros das equipes.

Questão 1. Houve melhoria na execução do projeto após a implantação dos processos ?

Para esta questão 13% responderam que concordam totalmente que o modelo de referência proposto proporcionou melhoria no desenvolvimento dos projetos e 87% concordam que o projeto teve melhora se comparado a outros projetos da empresa antes da implantação dos processos. Constata-se então uma concordância de 100% dos membros para a percepção de que houve melhoria, ou seja, o modelo de referência proposto foi percebido como aplicável e representa melhoria.

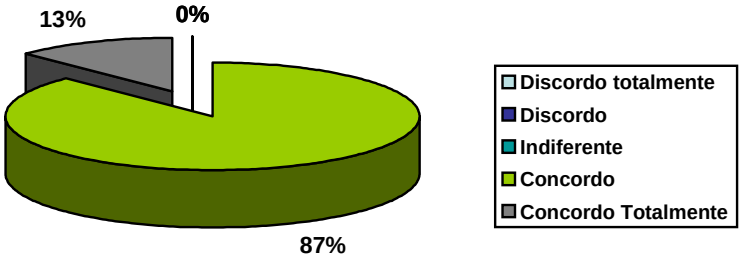

Gráfico 3. Resultados da questão 1.

Questão 2. Houve agilidade no tempo de produção de software ?

Quanto a questão relacionada a agilidade para se construir o produto 25% disseram concordar e 75% concordar totalmente que houve ganhos em termos de agilidade na construção do produto. Essa questão é particularmente interessante pois declara que apesar do controle sobre os processos a produtividade não se mostrou prejudicada.

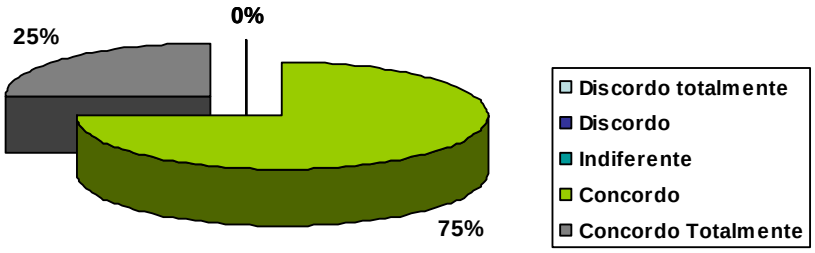

Gráfico 4. Resultados da questão 2.

Questão 3. A documentação gerada auxiliou no seu trabalho ?

Uma das preocupações com relação a melhoria do processo se deve ao fato do excesso de documentação. Nessa questão, assim como na questão 13, percebe-se que a documentação não afetou negativamente no trabalho da equipe. 87% concordaram que a documentação gerada auxilio no trabalho, apenas 13% acreditam ser indiferente.

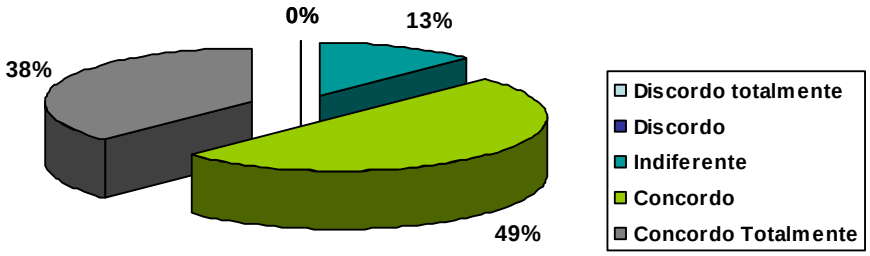

Gráfico 5. Resultados da questão 3.

Questão 4. A comunicação da equipe foi mais intensa e eficiente ?

Como ponto chave de qualquer projeto, a comunicação deve ser avaliada e em um programa de melhoria deve ser levada em consideração. Para essa questão, 62% concordam totalmente para o fato da comunicação da equipe ter sido eficiente. Isso se deve ao fato do modelo de referência proporcionar reuniões diárias, rápidas e objetivas.

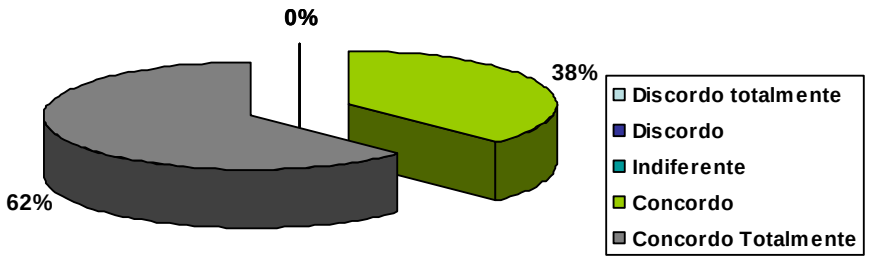

Gráfico 6. Resultados da questão 4.

Questão 5. O projeto foi executado com base em prazos mais realistas ?

As estimativas para horas de projeto representam um dos principais fatores de conflitos entre cliente e fornecedor e entre os membros da equipe. Com o modelo institucionalizado, o processo de gerência de projetos preza pelo uso de técnicas para estimativas que na visão da equipe se mostrou satisfatória visto que 87% responderam concordar ou concordar totalmente com a questão 5.

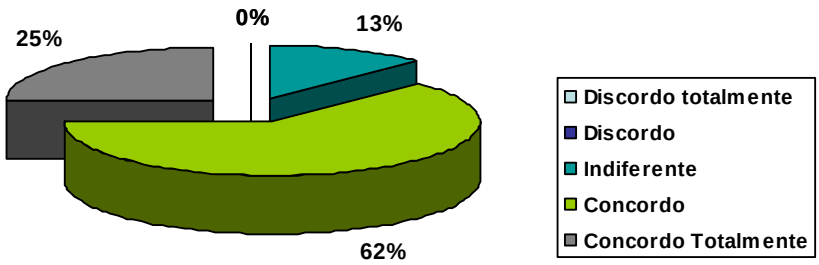

Gráfico 7. Resultados da questão 5

Questão 6. Você precisou trabalhar mais de 40 horas semanais com freqüência ?

Nenhum dos membros do projeto necessitou trabalhar, com freqüência, mais de 40 horas semanais com freqüência o que mostra um controle maior das atividades e uma estimativa mais realista das tarefas executadas, ou seja, o tempo passou a ser melhor considerado. Não se administra Tempo que é absoluto, mas as tarefas ao longo do tempo.

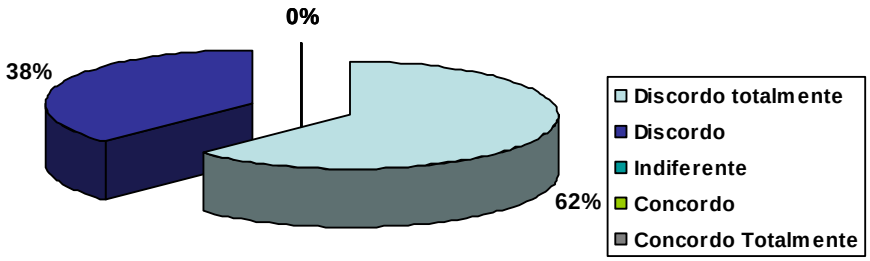

Gráfico 8. Resultados da questão 6.

Questão 7. A qualidade do produto final foi melhor que projetos anteriores ?

Quanto a questão da qualidade do produto desenvolvido, 74% concordaram totalmente com o fato da qualidade do produto ter sido melhor que de projetos anteriores, 13% concordaram e 13% disseram ter sido indiferente. Considerando a qualidade do produto final a conseqüência de um processo maduro, pode-se entender que o processo de desenvolvimento ganhou maturidade e que poderá ser melhorado ao longo de suas sucessivas aplicações (curvas de aprendizagem).

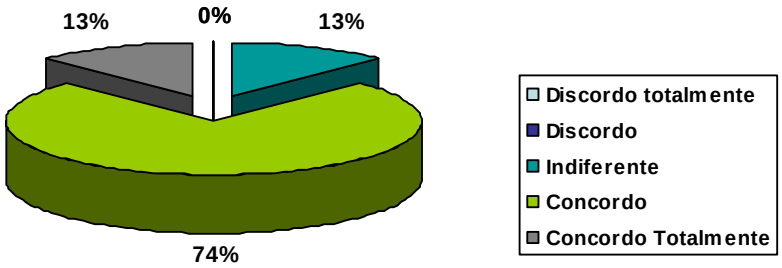

Gráfico 9. Resultados da questão 7.

Questão 8. A consulta aos documentos do projeto foi facilitada com o uso do SharePoint?

Após a disponibilização dos documentos na ferramenta SharePoint todos os membros tiveram permissão de acesso, facilitando a consulta. Nesta questão, 13% dos membros discordaram com o fato da ferramenta ter auxiliado, todos os outros julgam que a consulta aos documentos, facilitou seu trabalho.

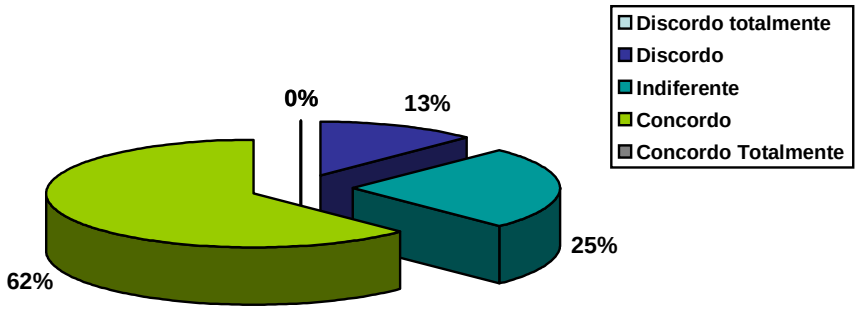

Gráfico 10. Resultados da questão 8.

Questão 9. Existe maior precisão no controle das suas atividades ?

49 % da equipe concordam totalmente que neste projetos existiu um maior controle sobre as suas atividades desempenhadas, isto se deve ao fato da gerência ter sido mais atuante. 38 % concordam e 13 % acharam indiferente.

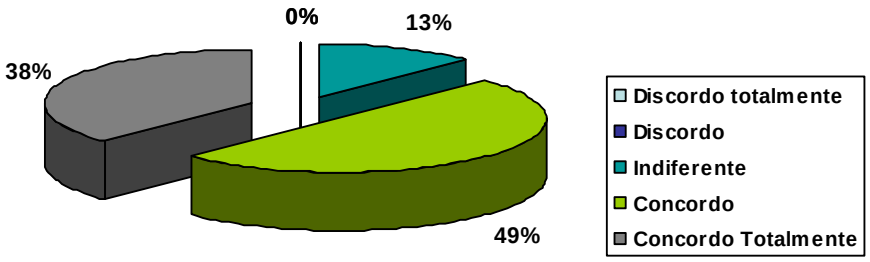

Gráfico 11. Resultados da questão 9.

Questão 10. As atividades que você executou foram passadas de forma mais clara ?

Para esta questão, 100% da equipe concordou que as atividades após terem sido definidas, foram passadas de forma mais clara. Os fatores que auxiliaram nesse quesito são:

1) reunião no início do projeto para definição de papéis e responsabilidades.
2) uso do Trac System que submete a cada desenvolvedor suas tarefas.

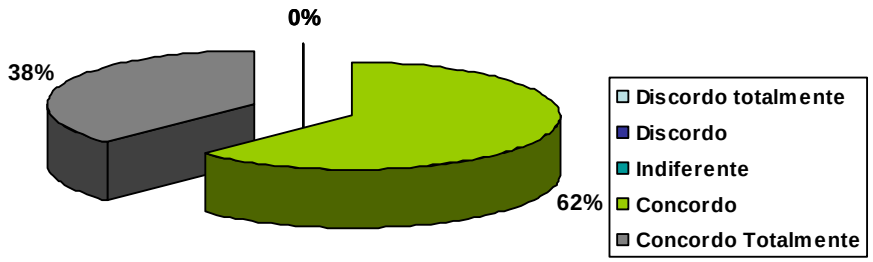

Gráfico 12. Resultados da questão 10.

Questão 11. Os papéis e responsabilidades de cada membro do projeto estão melhor definidos ?

Assim como na pergunta anterior, 100% dos membros responderam concordar que as responsabilidades foram melhor definidas. Neste modelo de referência existe uma reunião inicial onde cada membro recebe e assume suas funções, da qual é lavrada ata, na qual os participantes concordam e assinam sua participação.

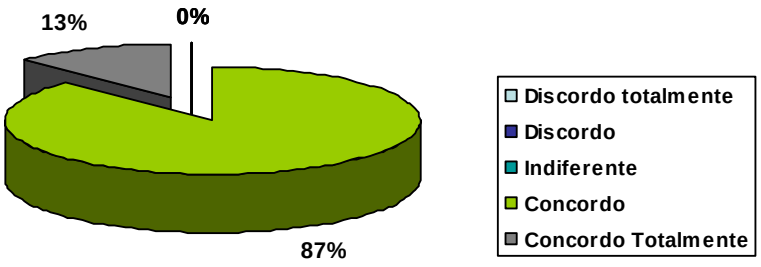

Gráfico 13. Resultados da questão 11.

Questão 12. O cliente percebeu as melhorias impostas pela empresa ?

Esta questão apesar de difícil de mensurar, principalmente para os desenvolvedores que muitas vezes não tem contato com o cliente final se mostrou satisfatória visto que 75% concordaram que a melhoria foi percebida.

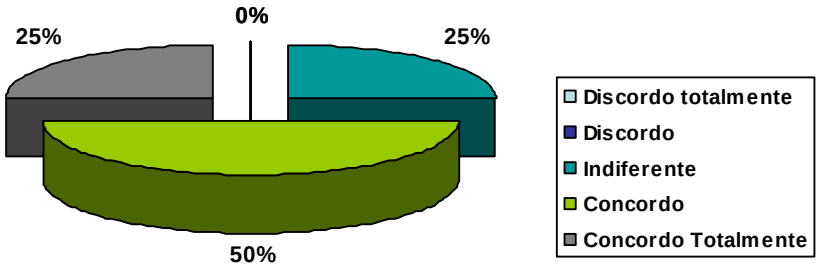

Gráfico 14. Resultados da questão 12.

Questão 13. A documentação gerada impossibilitou que seu trabalho fosse mais ágil ?

Essa questão é complementar a questão 5 que diz respeito a documentação gerada e exigida. Porém, perguntada de forma invertida. 75 % dos membros discordaram que a documentação foi um empecilho e 25 % disseram ser indiferente.

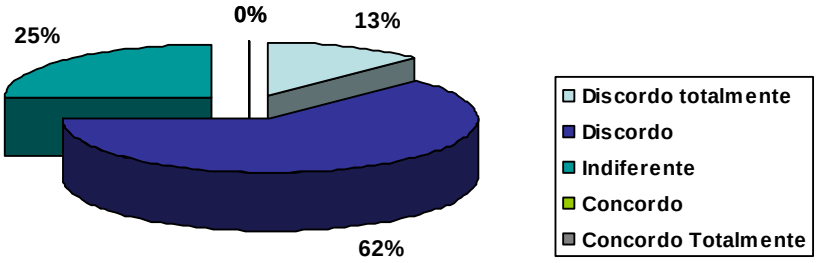

Gráfico 15. Resultados da questão 13.

Questão 14. O fato de existir uma equipe de qualidade inspecionando seu trabalho aumentou a qualidade do(s) produto(s) que você gerou ?

A grande maioria (87%) acreditam que a equipe de qualidade é fundamental para a qualidade do processo e do produto. Apenas 13% acreditam ser indiferentes.

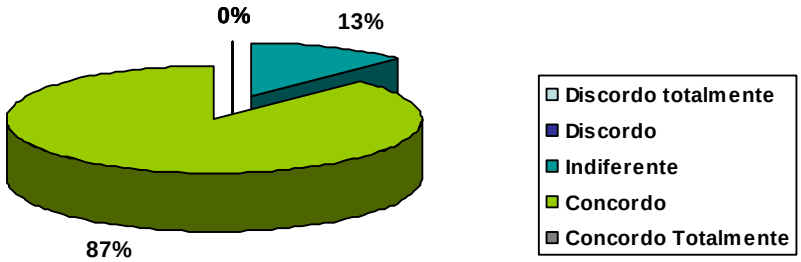

Gráfico 16. Resultados da questão 14.

Questão 15. O seu nível de satisfação aumentou após a implantação do programa de qualidade ?

Esta questão está mais relacionada ao sentimento que o colaborador tem em relação ao esforço de se melhorar os processos da empresa e conseqüentemente seu trabalho. 75% concordaram que esse programa afetou positivamente sua satisfação e 25 % se mostraram indiferentes.

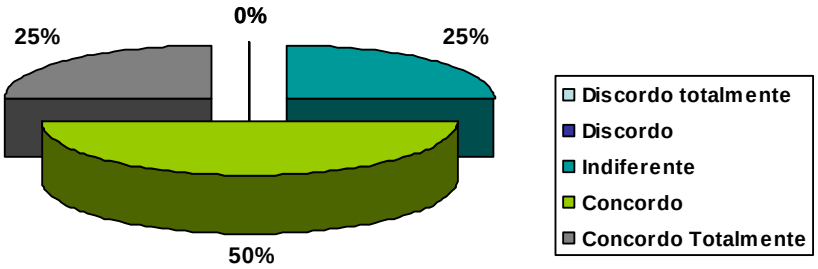

Gráfico 17. Resultados da questão 15.

Mesmo sendo uma pesquisa qualitativa, pode-se entender que os benefícios alcançados nesses dois projetos foram satisfatórios para a equipe. Entende-se também que outros benefícios, chamados de intangíveis também surtiram efeitos, como por exemplo:

- Quebra de Paradigma a medida que foram questionados processos vigentes e colocados sob a ótica de melhoria contínua;
- Diminuição da retenção do conhecimento com a transmissão do conhecimento para a organização;
- Alinhamento da organização: Planejamento Estratégico foi determinado para a institucionalização do processo;
- Aumento da Capacidade Gerencial da Empresa;
- Segurança e credibilidade: obteve-se um aumento de confiança na equipe gerencial;
- Necessidade de Visibilidade dos Resultados: implantação de GQA, avaliando os resultados da real utilização do processo

6. CONCLUSÃO

Sendo o desenvolvimento de produto um dos fatores chave para que as organizações se mantenham no mercado de forma competitiva, deve-se almejar uma gestão de PDP (Planejamento e Desenvolvimento de Produto) efetiva. Esta condição é indispensável e exige que as organizações tenham condições de garantir que o mesmo seja executado rigorosamente com qualidade e produtividade (eficiência no processo) e venha a oferecer qualidade no produto oferecido (eficácia na oferta). A conseqüência direta desta condição, que compõem a estratégia da organização vencedora é a garantia de sua competitividade.

Como apresentado neste trabalho existem desde os anos 90, esforços para auxiliar as organizações na busca desse elevado padrão de qualidade, porém, todos esse esforços ainda hoje são insuficientes para pequenas empresas. Percebendo-se esta condição, que tende a perdurar ainda por algum tempo, surge esta proposta de um modelo referencial característico para pequenas empresas produtoras de software, composto por atividades, processos e métodos próprios as suas especificas condições de operação produtiva.

O modelo proposto e aplicado com sucesso, apoiou-se em modelos de desenvolvimento de produtos industriais já disponíveis, com inovações capazes de oferecer soluções ao gerenciamento de softwares em pequenas empresas, o que representa uma significativa contribuição ao assunto, pela abordagem e sistematização oferecidas.

Como visto, esta pesquisa se justifica pela impossibilidade oferecida pelos atuais modelos de desenvolvimento de software na garantia de qualidade. Os modelos e *frameworks* atuais têm seu foco preponderantemente voltado para a fase de construção e negligenciam atividades como planejamento estratégico, projeto do produto e descontinuidade do produto, o que representa um tratamento apenas parcial da questão.

Para se chegar a este modelo de referência, partiu-se inicialmente de uma pesquisa de campo, na qual, buscou-se conhecer e entender as dificuldades de cinco empresas e os desafios que enfrentam cotidianamente na criação de seus produtos.

Para esta pesquisa foi utilizado, após um profundo estudo sobre os métodos de avaliação existentes, nascidos de uma revisão bibliográfica, onde o contexto teórico atualizado proporcionou o uso do questionário de contextualização do método MARES, criado por pesquisadores da UNIVALI e já aplicado em dezenas de avaliações.

Nesta pesquisa de campo, ficou evidente que todas as cinco organizações apresentam características semelhantes aos relatos bibliográficos de empresas imaturas. Entre as características se destacam a informalidade das ações, esforço heróico dos colaboradores, falta de documentação apropriada, falta de métodos para estimativas, entre outros. Porém, todas essas empresas almejam melhorar seus processos, mas não encontram maneira para isso, devido principalmente a falta de uma estrutura sistematizada que auxilie nessa conquista.

A idealização desse modelo de referência para desenvolvimento de software teve influência marcante do modelo de desenvolvimento de produto criado em ação conjunta de 3 universidades brasileiras (EESC/USP, UFSCar e UFSC). Esse modelo, já usado em outras áreas de aplicação, foi de fundamental importância nesta aplicação específica, pois através da definição de suas fases, atividades e marcos (*Gates*) auxiliou na criação do modelo proposto nesta tese.

Com relação a qualidade, o foco esteve direcionado ao modelo de referência chamado de MPS.Br (melhoria do processo de software brasileiro), criado em 2004 pela SOFTEX e que é visto atualmente como uma alternativa mais realista para as pequenas empresas se comparado a norma ISO 12207 e ao modelo CMMi. Atualmente o MPS.Br acha-se implantado em 104 empresas no Brasil e está sendo expandido para outros países da América Latina.

O modelo proposto foi utilizado em dois projetos em uma pequena empresa de tecnologia com o objetivo de servir de base para validar o modelo. Cabe destacar que qualquer movimento no sentido de melhorar a qualidade nos primeiros esforços é facilmente percebido, porém, a quantificação é algo trabalhoso, principalmente por não existir métodos específicos difundidos e validados. Optou-se então pelo o uso de um questionário com 15 questões que pudessem apresentar, ainda que de forma qualitativa, os resultados.

Os resultados, para as atividades do processo de gerência de requisitos e gerência de projetos, escolhidos como as etapas preponderantes, foram positivos. Entre eles, ficaram evidentes: agilidade no processo de desenvolvimento, aumento da qualidade dos produtos gerados, melhor controle das atividades e da aplicação dos recursos, estimativa mais precisa de tempo e esforço, rapidez na documentação, que uma vez sistematizada passa a ser padrão e serve de base de conhecimento. Outro ponto de destaque é que todos os nove entrevistados se mostraram satisfeitos com o programa de melhoria, uma vez que descreveram no questionário palavras positivas, como: excelência, satisfação, qualidade e melhoria.

Cabe destacar ainda que os dois projetos possuem diferenças quanto ao número de colaboradores envolvidos, escopo do projeto, tempo de duração, entre outros, ou seja o modelo pode ser aplicável a diferentes circunstâncias conceituais. Definitivamente cada projeto de software é único devido a uma série de fatores que incluem desde o tipo de aplicação a ser construída, necessidades específicas de cada cliente e até restrições de cronograma e recursos. A idealização do modelo de referência, desde o princípio, visava a possibilidade da definição de um único processo de desenvolvimento de software que fosse adaptável a qualquer situação. Sendo assim, os trabalhos tanto na empresa quanto na pesquisa desta Tese possuem possibilidades reais de continuidade.

Para a empresa em questão fica, a curto prazo, a conquista da certificação do nível G do MPS.Br e a médio e longo prazo a melhoria de outros processos, com avanços que poderão

ser atingidos em pontos como: testes, homologação, gerência de configuração, medição, treinamento, que se mostram possíveis de significativas contribuições de melhoria.

A continuidade de aplicação deste modelo e a otimização decorrente de sua difusão ensejarão o aparecimento de um modelo referencial padrão, de uso universal. Isso será atingido pelas contribuições de novas pesquisas e pela integração de pesquisadores.

Em se tratando de pesquisa existe a possibilidade de aplicação de outros processos propostos nesse trabalho, que poderão ser experimentados e permitirão avaliação da melhoria. Como conseqüência, sua adoção ensejará a aplicação referencial para que as pequenas empresas venham a amadurecer seu processo de desenvolvimento e consolidar sua atuação aos mais elevados patamares de competitividade, o que é muito significativo pois, fortalecerá um universo de cerca de 1800 empresas existentes no Brasil.

Oferece-se assim, um modelo de referência válido para aplicação disciplinada em pequenas empresas produtoras de software, no caso brasileiro.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT. Associação Brasileira de Normas Técnicas**. NBR ISO 9000:2000 – Sistemas de gestão da qualidade e garantia da qualidade – Fundamentos e Vocabulário**. Rio de Janeiro: ABNT, 2000.

_____. Associação Brasileira de Normas Técnicas. **NBR 14724: Informação e Documentação** - Trabalhos acadêmicos - Apresentação. Rio de Janeiro: ABNT, 2001.

ABES. **Mercado Brasileiro de Software: panorama e tendências**, 1º edição. Associação Brasileira das Empresas de Software. São Paulo, 2007.

ANACLETO, A. et al. **A Method for Process Assessment in Small Software Companies**. Proc. 4th Int'l Software Process Improvement and Capability Determination Conf. (SPICE 04), SPICE User Group and Critical Software SA, 2004.

BARBALHO, S. C. M. **Modelo de Referência para Desenvolvimento de Produto Mecatrônicos:** proposta e aplicações. 2006. 256f. Tese (Doutorado em Engenharia Mecânica) Escola de Engenharia de São Carlos. Universidade de São Paulo, São Carlos, 2006.

BARBALHO, S. C. M. e ROZENFELD, H. **Análise do processo de desenvolvimento de produtos de uma pequena empresa de alta tecnologia**. Anais do XXIV Encontro Nacional de Engenharia de Produção, 2004, Florianópolis. Associação Brasileira de Engenharia de Produção, Novembro, 2004.

BECK, K.; ANDRES, C**. Extreme programming explained: Embrace change**. Second Edition, Addison Wesley Professional, 2004.

BOEHM, B. W. **A spiral model of software development and enhancement**. ACM Software Engineering Notes, v. 11, 1988.

BOOCH G.; RUMBAUGH J.; JACOBSON I. **The Unified Modeling Language User Guide**, Second Edition. Addison-Wesley, 2005.

BRANDÃO, C. R. **Repensando a Pesquisa Participante**, 2º Edição, Editora Brasiliense, São Paulo. 1985.

COOPER, R. G. **Winning At New Products: Accelerating The Process From Idea To Launch,** Second Edition, Basic Books. 1993.

COLELLA, L. C. COLENCI NETO, A. VALENTE, J. F. BIANCHI, R. E. **Uso da tecnologia de identificação por rádio freqüência (RFID).** Anais do XV Simpep – Simpósio de Engenharia de Produção. UNESP. Bauru. 2008.

COUTO, A. B. **CMMI – Integração dos modelos de capacitação e maturidade de sistemas.** Rio de Janeiro: Ciência Moderna, 276 p. 2007.

CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. **CMMI: Guidelines for Process Integration and Product Improvement**. Pensilvânia, EUA: SEI Software Engineering Institute Addison-Wesley. 2003.

CLARK, K. B.; FUJIMOTO, T. **Product development performance: strategy, organization and management in the world auto industry. Boston, Mass**.: Harvard Business School Press. 1991.

CMMI-DEV. **CMMI for Development**. Carnegie Mellon University. Software Engineering Institute. Version 1.2, 2006.

COCHANGO, **Scrum for team systems**, http://www.scrumforteamsystem.com. Acesso em otubro de 2006.

DEMO, P. **Pesquisa e construção do conhecimento: metodologia científica no caminho de Habermas**. 2º Edição, Tempo Brasileiro, Rio de Janeiro, 125p, 1994,

DRUCKER, P. **Uma era de descontinuidade**: orientação para uma era de mudança. São Paulo. Círculo do Livro. 1968

FERNADES, A. A. **Fábrica de Software: implantação e gestão de operações.** 1º Edição. Atlas. São Paulo, 304p, 2004.

FIORINI, S. T., et al**. Engenharia de Software com CMM**, 1º Edição, Ed. Brasport, Rio de Janeiro, 1998.

FOWLER, M. **The New Methodology**. Disponível em http://www.martinfowler.com/articles/newMethodology.html. 2005. Acessado em maio de 2006.

GIL, A. C. **Como elaborar projetos de pesquisa**. 3º edição, Atlas, São Paulo, 2001.

HUMPHREY, W. **Managing the software process**. SEI/CMU – Addison Wesley publishing. Boston, 1989.

JEFFRIES, R. et al. **Extreme Programming Installed**. EUA, Addison-Wesley. 2001.

KAMINSKI, P. C., et al. **Evaluation of the real use of formal methodologies in the product development process in brazilian SMEs. Product: Management & Development** - Revista Brasileira de Gestão de Desenvolvimento de Produto, v.3, n.2, São Carlos - SP, 2005.

KAPLAN R. S., NORTON D.P. **Estratégia em Ação: Balanced Score Card**. 21º Edição, Editora Campus, Rio de Janeiro – RJ. 1997.

KOHAN, S. **QuickLocus: Proposta de um método de avaliação de processo de desenvolvimento de software em pequenas organizações**. 2003. Dissertação (Mestrado em Engenharia de Computação). Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, São Paulo, 2003.

KRUCHTEN, P. **The Rational Unified Process: An Introduction**, 3th Edition. Addison Wesley. 2003.

MACHADO, C. A. F. in WEBER, K. C., et al**. Qualidade e Produtividade em Software**, 4ª edição, São Paulo, Ed. Makron Books, 2001.

MAGALHAES, A. L. C. C, ROUIILER, A. C., VASCONCELOS, A. M. L. **O Gerenciamento de projetos de software desenvolvido à luz das metodologias ágeis:** uma visão comparativa. PROQUALITY – Qualidade na produção de software, v. 1, n. 1. 2005.

MCT. **Qualificação CMM e CMMI no Brasil.** Disponível em: www.mct.gov.br/upd_blob/0009/9238.pdf. Acessado em: Setembro de 2007.

_____. **Pesquisa Qualidade e Produtividade no Setor de Software Brasileiro**. Brasília: Ministério da Ciência e Tecnologia - MCT/SEPIN, 2005.

MIT. **Slicing the Knowledge-Based Economy (KBE) in India, China and Brasil**: a Tale of Three Software Industries. Massachussets Institute of Technology. Cambridge, 2003.

MPS - **Melhoria de Processo do Software Brasileiro:** Guia Geral. Versão 1.2. Campinas: SOFTEX, 2007.

MA-MPS - **Melhoria de Processo do Software Brasileiro:** Guia de Avaliação. Versão 1.1. Campinas: SOFTEX, 2006.

MOPORSOFT. **Modelo de Procesos para la Industria de Software** Versão 1.3. Cidad del México. Agosto 2005.

_____. **Software Industry Process Model.**. Versão 1.3.2. Cidad del México. April 2006.

NAKANO, D. N.; Fleury, A. C. C. **Métodos de Pesquisa na Engenharia de Produção.** In;: XVI ENEGEP – Encontro Nacional de Engenharia de Produção, Anais. Piracicaba: UNIMEP/ABEPRO. 1996 (CD ROM).

NBR ISO/ABNT, **Associação Brasileira de Normas Técnicas**. Normas NBR/ISO 8402. Rio de Janeiro, 1994.

NOGUEIRA, M. **Qual a Importância da adoção da Norma ISO 12207 nas Empresas de Desenvolvimento de Software ?** X Simpósio de Engenharia de Produção, Anais. Bauru: Simpep. 2003

PAULK M. C. **Capability Maturity Model: Guidelines for Improving the Software Process**. Addison-Wesley, 1995.

_____. **Using the Software CMM in Small Organizations**. Software Engineering Institute, CMU, Report. 1998.

_____. **Extreme Programming from a CMM Perspective.** IEEE SOFTWARE, november/december, Vol. 18, Publishing Press, p. 1-8. 2001.

PENSO, C. C. **Modelo de referência para processo de desenvolvimento de produto na indústria de alimentos**. 2003. 180f. Dissertação (Mestrado em Engenharia Mecânica) Universidade Federal de Santa Catarina (UFSC), Florianópolis – SC, 2003.

PFLEEGER S. L. **Engenharia de Software: Teoria e Prática.** 2º Edição. São Paulo, Prentice Hall, 2004.

PINTO, F. (2006). **Comparação do MPS.Br com o CMMI**. Disponível em: http://www.pontodatecnologia.com.br/2006/08/comparao-do-mpsbr-com-o-cmmi.html. Acessado em 30 jan 2007.

PMBOK. **A Guide to the Project Management Body of Knowledge.** 2000 Edition, Project Management Institute (PMI®) 2000.

PORTER, M. E. **Estratégia competitiva**. 16. ed. Rio de Janeiro: Editora Campus, 2004.

PRESSMAN, R. S. **Software Engineering: A practitioner's approach**. 6th. ed. McGraw-Hill, 2005.

PUGH, S. **Total Design: integrated methods for successful product engineering**. Wokingham, Addison Wesley. 1990.

ROYCE, W. W., **Managing the Development of Large Software Systems: concepts and techniques**, Los Angeles, Proceedings, IEEE Wescon, 1970.

ROZENFELD, H. et al. **Gestão de Desenvolvimento de Produto**. 1º Edição, São Paulo; SP. Editora Saraiva, 2006.

ROMANO, L. N. **Modelo de referência para o processo de desenvolvimento de máquinas agrícolas**. 2003. 285f. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal de Santa Catarina. Florianópolis (UFSC), Florianópolis – SC, 2003.

SALVIANO, C. F. ANACLETO, A. WANGENHEIM, C. G. **Avaliação de processos para início de programas de melhoria em micro e pequenas empresas de software**. VI Simpósio internacional de melhoria de processo de software. São Paulo, Brasil, novembro de 2004.

SCHMITZ E. A., ALENCAR A. J. VILLAR, C. B. **Modelos Qualitativos de Análise de Risco para Projetos de Tecnologia da Informação**. 1ª edição, Brasport. 196 p. 2007

SCHWABER, K.; BEEDLE, M.. **Agile Software Development with SCRUM**. Prentice Hall, 2002.

SEBRAE. Inaugura a Casa do Empreendedor. **Jornal de Negócios**. São Paulo: SEBRAE, ano VII, n.70, dez, 1998.

SEI - Software Engineering Institute. **What is CMMI ? (Capability Maturity Model Integration)** Disponível em: http://www.sei.cmu.edu/cmmi, acessado em maio de 2007.

SILVA, E. L. **Metodologia da pesquisa e elaboração de dissertação**. 3º Ed. - Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

SOMMERVILLE, I. **Software Engineering**. 6th. ed. Addison-Wesley, 2003.

SPINOLA, M. M., **Diretrizes para o desenvolvimento de software de sistemas embutidos**, 1999, Tese (Doutorado em Engenharia Elétrica), Universidade de São Paulo - São Paulo, 1999.

STANDISH - The Standish Group. **Chaos Demographics and Project Resolution**. 2004 Disponível em: <http://www.standishgroup.com/sample_research/PDFpages/q3-spotlight.pdf> Acesso em: 24 mar. 2008.

TELES, V. M. **Extreme programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade.** 1º Edição**.** São Paulo – SP, Novatec Editora, 2004.

THIOLLENT, M. **Metodologia da Pesquisa-ação**. 4ª edição, São Paulo, Editora Cortez, 1988. 108p.

VERNADAT, F. B. **Enterprise modeling and integration: principles and applications**. 1th, Chapman e Hall, London, UK, 1996.

WANGENHEIM, C. G. et al.. **Aplicando Avaliações de Contextualização em Processos de Software Alinhados ao CMMI-SE/SW.** VII Simpósio Internacional de Melhoria de Processos de Software. São Paulo/SP. 2005

_____. **Helping Small Companies Assess Software Processes**, IEEE SOFTWARE, Vol. 23, Issue 1, Page(s): 91 – 98. Jan.-Feb. 2006.

WANGENHEIM, C. G., SALVIANO, C. **Consolidação de uma metodologia para avaliação de processos de software de MPEs baseada na norma ISO/IEC 15504**. Anais do Simpósio Brasileiro de Qualidade de Software - SBQS. Porto Alegre, Brasil, 2005.

WEBER. K. C. et al. **Brazilian Software Process Reference Model and assessment method**, Computer and information sciences - ISCIS 2005. Berlin, Germany.

Anexo I

Questionário de Avaliação

Questionário de Caracterização de Empresa para Avaliação

ISO/IEC 15504

baseado no Método MARES/15504MPE (LQPS/UNIVALI e CenPRA)

A resposta a este questionário faz parte da Fase 1 da avaliação para a melhoria da qualidade dos processos de software.
A qualidade das respostas deste questionário é fundamental para o êxito deste trabalho. Assim, a empresa deve se comprometer a disponibilizar tempo adequado para que representantes da diretoria forneçam as informações solicitadas de forma precisa.
Como em todas as atividades de obtenção de informações, esta atividade segue o compromisso de confidencialidade assinado pelo Patrocinador e pela Equipe de Avaliação.

As respostas deste questionário serão utilizadas para alinhar a avaliação com o contexto e os objetivos de negócio da organização.
A avaliação não tratará necessariamente de todos os aspectos abordados no questionário. Portanto, o participante não deve esperar que todas as suas respostas tenham uma ação neste projeto e nem que aquelas que induzirem a uma ação sejam de curto prazo.
O tempo médio de preenchimento deste questionário é de uma hora, podendo variar de uma empresa para outra. No final do questionário é solicitado o tempo gasto com o preenchimento do mesmo para fins de avaliação do método utilizado.

Agradecemos a sua colaboração.

1. Caracterização da Organização

1.1 Razão Social:

1.2 Contato:

1.3 Endereço:

1.4 E-mail:

1.5 Telefone:

1.6 Website:

1.7 Início das atividades em informática:

1.8 Breve descrição da empresa:

1.9 Principal atividade da organização no desenvolvimento de software:

- ☐ Desenvolve software sob encomenda para terceiros (para cada cliente um software individual é desenvolvido para os requisitos específicos deste cliente)
- ☐ Desenvolve software-padrão sem customização (software pacote)
- ☐ Desenvolve software-padrão com customização para clientes (software que atende as necessidades de muitos clientes mas que geralmente ainda é adaptado para satisfazer completamente os requisitos específicos de um cliente)
- ☐ Outra: _____

1.10 Principal atividade da organização em Tecnologia da Informação:

- Comercialização de dados ou de bases de dados
- Consultoria e projetos em informática
- Desenvolvimento de software
- Distribuição ou editoração de software de terceiros
- Distribuição ou revenda de produtos de hardware
- Indústria de informática, telecomunicação ou automação
- Manutenção e assistência técnica em informática
- Provedor Internet
- Serviços de automação bancária
- Serviços de automação comercial
- Serviços de automação industrial
- Serviços de entrada de dados
- Serviços de processamento de dados
- Treinamento em informática
- Outras. Especifique: _____

1.11 Força de trabalho da organização:

(como total da força de trabalho da organização considere sócios, dirigentes, empregados/funcionários efetivos, incluindo terceiros prestadores de serviço, bolsistas e estagiários)

Sócios, dirigentes e empregados efetivos	____
Bolsistas e estagiários	____
Terceiros prestadores de serviço	____
Total	____

1.12 Organograma da empresa:

1.13 Porte da empresa segundo comercialização bruta anual:

- Até R$ 120 mil
- De R$ 120 mil até R$ 720 mil
- De R$ R$ 720 mil até R$ 2,5 milhões
- Mais de R$ 2,5 milhões

1.14 Indique o nível de crescimento que mais se assemelha a sua empresa atualmente:

- Pré empresa – a organização é caracterizada mais como um projeto (em pré-incubadoras, etc) do que propriamente uma empresa.
- Existência – o principal objetivo da empresa é encontrar clientes e vender produtos para viabilizar o seu negócio.
- Sobrevivência – a empresa já mostrou ser viável. Os principais objetivos são manter/buscar clientes para que o fluxo de caixa permita a empresa crescer e gerar lucro.
- Crescimento – a empresa se consolidou e tem recursos para crescer.

1.15 Faixa de mercado (mais de um item pode ser marcado):

- Atende a uma faixa de mercado que não é enfocada pelas grandes empresas.

- Produz componentes para suprir às empresas de médio e grande porte.
- Inicia o desenvolvimento de novos produtos inovadores.
- Presta serviços e manutenção para os produtos fabricados por grandes empresas.
- Outra: _____

1.16 A organização implantou programa da qualidade total, sistema da qualidade ou similar?

- Sim. Em que ano? _____
- Em estudo ou implantação
- Não

1.17 Qual o tipo de certificação obtida?

- Certificado IS0 9001
- Certificado por cliente. Especifique: _____
- "Certificado" CMM/CMMI. Em que nível?_____
- Outras certificações. Especifique: _____
- Profissionais certificados (PMP, Microsoft, etc.). Especifique: _____
- Não há nenhuma certificação

1.18 Relate até três principais oportunidades de negócio para a organização (fatores internos ou externos que podem abrir, ampliar ou melhorar o mercado da organização):

_____

_____

_____

_____

_____

1.19 Relate até três principais ameaças à organização (fatores internos ou externos que podem atrapalhar ou mesmo inviabilizar a organização):

_____

_____

_____

_____

_____

_____

PRODUTO/PROJETO 1

2.3 Breve descrição do produto/projeto:

_____

_____

_____

2.4 Quais são as principais funcionalidades do produto/projeto:

_____

_____

2.5 Quem são os principais usuários do produto/projeto:

_____

_____

2.6 Tecnologias utilizadas no desenvolvimento do produto/projeto:

_____

_____

2.7 O software é customizável? ☐ Sim ☐ Não

2.8 Sua empresa faz manutenção do software? ☐ Sim ☐ Não

2.9 Quem é responsável pela instalação do software? ☐ A empresa ☐ O cliente

2.10 Sua empresa presta suporte aos clientes do software? ☐ Sim ☐ Não

2.11 Qual a freqüência em que novos *releases* do software são gerados? (p.ex. 1 vez por ano)

_____

2.12 Duração (média) dos projetos de software: _____ meses.

2.13 Tamanho (médio) da equipe de projeto: _____ **pessoas.**

2.14 Tamanho aproximado do software: (p.ex. em Linhas de código, Pontos por Função, etc):

_____

2.15 Complexidade do software:

☐ alta: muitos algoritmos e cálculos difíceis, elementos e relações de dados muitos complexos

☐ média: alguns algoritmos e cálculos difíceis, vários arquivos e interações dos dados

☐ baixa: maioria dos algoritmos e cálculos simples, dados simples, poucas variáveis

2.16 Aspectos de qualidade relevantes para o software: Crucial (C), Importante (I), Desejável (D), Não Importante (N)

	Confiabilidade(*Capacidade do sw manter o seu nível de desempenho dentro de condições estabelecidas por um dado período de tempo.)*
	Usabilidade (*Facilidade de uso e avaliação das características dos usuários do sw.)*
	Eficiência (*Relação entre o nível de desempenho do sw e a quantidade de recursos utilizados em determinadas condições.)*
	Manutenibilidade (*Esforço necessário para executar modificações específicas.)*
	Portabilidade (*Facilidade do sw ser transferido de um ambiente para outro.)*
	Funcionalidade (*O sw tem funções para satisfazer as necessidades explícitas e implícitas.)*
	Testabilidade (*Capacidade de realizar testes no software)*
	Reusabilidade (*Capacidade de reutilizar código)*
	Outro:_____

PRODUTO/PROJETO 2

2.3 Breve descrição do produto/projeto:

_____

_____

_____

2.4 Quais são as principais funcionalidades do produto/projeto:

_____

_____

2.5 Quem são os principais usuários do produto/projeto:

_____

_____

2.6 Tecnologias utilizadas no desenvolvimento do produto/projeto:

_____

_____

2.7 O software é customizável? ☐Sim ☐Não

2.8 Sua empresa faz manutenção do software? ☐Sim ☐Não

2.9 Quem é responsável pela instalação do software? ☐ A empresa ☐ O cliente

2.10 Sua empresa presta suporte aos clientes do software? ☐Sim ☐Não

2.11 Qual a freqüência em que novos *releases* do software são gerados? (p.ex. 1 vez por ano)

_____

2.12 Duração (média) dos projetos de software: _____ meses.

2.13 Tamanho (médio) da equipe de projeto: _____ **pessoas.**

2.14 Tamanho aproximado do software: (p.ex. em Linhas de código, Pontos por Função, etc):

_____

2.13 Complexidade do software:

☐**alta: muitos algoritmos e cálculos difíceis, elementos e relações de dados muitos complexos**

☐**média: alguns algoritmos e cálculos difíceis, vários arquivos e interações dos dados**

☐**baixa: maioria dos algoritmos e cálculos simples, dados simples, poucas variáveis**

2.14 Aspectos de qualidade relevantes para o software: Crucial (C), Importante (I), Desejável (D), Não Importante (N)

	Confiabilidade (*Capacidade do sw manter o seu nível de desempenho dentro de condições estabelecidas por um dado período de tempo.)*
	Usabilidade (*Facilidade de uso e avaliação das características dos usuários do sw.)*
	Eficiência (*Relação entre o nível de desempenho do sw e a quantidade de recursos utilizados em determinadas condições.)*
	Manutenibilidade (*Esforço necessário para executar modificações específicas.)*
	Portabilidade (*Facilidade do sw ser transferido de um ambiente para outro.)*
	Funcionalidade (*O sw tem funções para satisfazer as necessidades explícitas e implícitas.)*
	Testabilidade (*Capacidade de realizar testes no software)*
	Reusabilidade (*Capacidade de reutilizar código)*

3. Qualidade dos Processos de Software

3.1 A maioria dos funcionários da empresa conhece a norma ISO/IEC 15504 (SPICE)?

Sim **Não**

3.4 Documentação Adotada:

Acompanhamento de custos
Acompanhamento de prazos
Contratos e acordos
Descrição do produto para comercialização
Documentação de marketing
Documentação de programas
Documentação do processo de software
Documentação no código
Especificação do sistema
Guia de instalação
Help on-line
Histórico do projeto
Identificação de risco
Manual de treinamento
Manual do sistema
Manual do usuário
Plano de contingência
Plano de controle da qualidade
Plano de testes
Projeto do sistema
Registro formal de revisões e testes
Relatório de teste
Outras. Especifique: _____
Não adota documentação

3.8 Cite os três pontos mais fortes no desenvolvimento de software que você observa na sua empresa:

_____

_____

3.9 Cite os três pontos mais fracos no desenvolvimento de software que você observa na sua empresa:

_____

_____

3.10 Como você caracteriza sua organização hoje, quanto aos aspectos relacionados a seguir:

	Bom	**Médio**	**Ruim**	**Não sabe**
Satisfação do cliente no atendimento às suas reais necessidades				
Tratar as solicitações de alteração dos requisitos				
Cumprir os prazos estabelecidos				
Cumprir o orçamento/custos estabelecidos				
Cumprir os requisitos estabelecidos				
Qualidade do produto entregue				
Produtividade no processo de software				
Satisfação dos funcionários no trabalho				

3.11 Selecione na tabela abaixo a meta de negócio mais importante para a sua empresa.

Metas de Negócio	**Import ante**	**Comentários**
Expandir faixa no mercado (aumentar vendas aos clientes já existentes, entrar em novos segmentes do mercado, exportar, etc.)		
Aumentar competitividade		
Aumentar valor de negócio associado ao produto ou serviço de sw		
Melhorar satisfação dos clientes		
Melhorar satisfação dos funcionários		
Outro:		

3.12 Em relação à pergunta anterior, o que se pretende melhorar para satisfazer estas metas de negócio? Selecione na tabela abaixo as 2 metas de melhoria mais importantes para a sua empresa.

Metas de Melhoria	**Import ante**	**Comentários**
Aumentar produtividade no processo de software		
Reduzir custos		
Reduzir tempo de desenvolvimento, instalação, manutenção, etc.		
Melhorar habilidade em cumprir os custos estabelecidos		
Melhorar habilidade em cumprir os prazos estabelecidos		
Melhorar habilidade de atender as necessidades reais dos clientes		
Melhorar a qualidade do produto de software		*Especifique: confiabilidade, portabilidade, eficiência, manutenibilidade, usabilidade, reusabilidade, etc.*
Obter certificação ISO 9000		
Obter certificação CMMI nível ____		
Outro:		

4. Questionário

Data do preenchimento deste questionário:

Responsável pelo preenchimento:

Cargo:

Tempo gasto com o preenchimento do questionário (em horas):

Anexo II

Formulário de Avaliação das Melhorias do Processo

Avaliação do Processo

Data: ___/ ___/ ___

Escreva duas palavras que melhor representam o programa de melhoria como um todo:

Grau de comprometimento no projeto: [ ] Baixo [ ] Médio [ ] Alto

1 - Discordo Totalmente / 2- Discordo / 3- Indiferente / 4- Concordo / 5- Concordo Totalmente

Questão	Peso				
	1	*2*	*3*	*4*	*5*
1) Houve melhoria na execução do projeto após a implantação dos processos ?					
2) Houve agilidade no tempo de produção de software ?					
3) A documentação gerada auxiliou no seu trabalho ?					
4) A comunicação da equipe foi mais intensa e eficiente ?					
5) O projeto foi executado com base em prazos mais realistas ?					
6) Você precisou trabalhar mais de 40 horas semanais com freqüência ?					
7) A qualidade do produto final foi melhor que projetos anteriores ?					
8) A consulta aos documentos do projeto foi facilitada com o uso do SharePoint ?					
9) Existe maior precisão no controle das suas atividades ?					
10) As atividades que você executou foram passadas de forma mais clara ?					
11) Os papéis e responsabilidades de cada membro do projeto estão melhor definidos ?					
12) O cliente percebeu as melhorias impostas pela empresa ?					
13) A documentação gerada impossibilitou que seu trabalho fosse mais ágil ?					
14) O fato de existir uma equipe de qualidade inspecionando seu trabalho aumentou a qualidade do(s) produto(s) que você gerou ?					
15) O seu nível de satisfação aumentou após a implantação do programa de qualidade ?					

Descreva suas impressões sobre o programa de qualidade: