

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

CARINA DA SILVA CARDOSO

**UTILIZAÇÃO DO FRAMEWORK EPF COMPOSER PARA O DESENVOLVIMENTO
DE UM METAMODELO BASEADO NO MPS.BR PARA A CRIAÇÃO DE
PROCESSOS APLICADOS À GARANTIA DA QUALIDADE EM PROJETOS DE
SOFTWARE**

CRICIÚMA

2012

CARINA DA SILVA CARDOSO

**UTILIZAÇÃO DO FRAMEWORK EPF COMPOSER PARA O DESENVOLVIMENTO
DE UM METAMODELO BASEADO NO MPS.BR PARA A CRIAÇÃO DE
PROCESSOS APLICADOS À GARANTIA DA QUALIDADE EM PROJETOS DE
SOFTWARE**

Trabalho de Conclusão do Curso, apresentado para a obtenção do grau de Bacharel no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. MSc. Gustavo Bisognin

CRICIÚMA

2012

CARINA DA SILVA CARDOSO

**UTILIZAÇÃO DO FRAMEWORK EPF COMPOSER PARA O DESENVOLVIMENTO
DE UM METAMODELO BASEADO NO MPS.BR PARA A CRIAÇÃO DE
PROCESSOS APLICADOS À GARANTIA DA QUALIDADE EM PROJETOS DE
SOFTWARE**

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Engenharia de Software.

Criciúma, 26 de Novembro de 2012.

BANCA EXAMINADORA



Prof. Gustavo Bisognin – MSc. - (UNESC) - Orientador



Prof. Ana Claudia Garcia Barbosa – MSc. - (UNESC)



Prof. Paracelso de Oliveira Caldas – MSc. - (UNESC)

Dedico este trabalho aos meus familiares, que sempre estiveram presentes durante toda a minha formação.

AGRADECIMENTOS

Em primeiro lugar gostaria de agradecer à minha família que sempre me incentivou nos momentos difíceis, me apoiando para que eu nunca desistisse de lutar pelos meus objetivos.

Agradeço ao meu orientador, Prof. MSc. Gustavo Bisognin, pela confiança depositada em mim durante toda esta jornada, sugerindo o tema para o desenvolvimento do trabalho, e por estar sempre à disposição para me auxiliar em todos os momentos, inclusive me ajudando a entrar no mercado de trabalho.

Não posso deixar de agradecer aos colegas e amigos que sempre estiveram ao meu lado.

Por fim, agradeço a todos os professores do curso de Ciência da Computação, pelo conhecimento que me foi transmitido durante todo o decorrer do curso, sem o qual não seria possível a conclusão deste trabalho.

“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.”

Charles Chaplin

RESUMO

No intuito de aumentar a qualidade dos processos, consultores e engenheiros da área, propõem a criação de metamodelos que agregam um conhecimento significativo na definição dos processos de desenvolvimento de software. Com o objetivo garantir a qualidade nos projetos, foi desenvolvido um metamodelo baseado nos principais conceitos do MPS.BR, considerando que os modelos de referência existentes atualmente nem sempre se adaptam à necessidade encontrada nas empresas brasileiras, devido à quantidade excessiva de atividades exigidas para o atendimento de cada nível de maturidade que compõe estes modelos. Foi realizada uma pesquisa durante todo o decorrer do trabalho que possibilitou a elaboração das fases e atividades do metamodelo que foi modelado com a utilização do Framework EPF Composer. A partir do metamodelo foi desenvolvido um processo como forma de demonstrar a sua aplicação e posteriormente foi realizado um modelo de avaliação através da análise de GAP, utilizando a planilha de avaliação formal do MA-MPS para demonstrar que as atividades propostas atendem os padrões de qualidade em relação ao modelo ideal. Como resultado do trabalho, temos o processo que foi elaborado com base no metamodelo, levando em consideração todos os conceitos de qualidade estudados. Com o modelo de avaliação do metamodelo foi possível realizar a comparação entre o que foi proposto e o modelo ideal, no qual foi constatado que o metamodelo atende 60% dos itens de qualidade estabelecidos no modelo ideal, demonstrando que as atividades sugeridas agregam qualidade aos projetos aos quais forem submetidos. Dentro do contexto abordado nesta pesquisa, conclui-se que o fato de implantar processos eficientes no desenvolvimento de projetos de software permite que os projetos sejam gerenciados com maior rigor, não havendo dúvidas sobre o que deve ou não ser realizado. O plano de Garantia da Qualidade deve ser aplicado em todo o ciclo de vida dos projetos de software, com a intenção de prevenir o retrabalho ocasionado pela correção de falhas, levando ao acúmulo de trabalho que pode resultar no abandono de procedimentos essenciais para o bom andamento do projeto.

Palavras-chave: Framework EPF Composer. Metamodelo. Modelo MPS.BR. Garantia da Qualidade em Projetos de Software.

ABSTRACT

In order to improve the quality of processes, consultants and engineers in the area, proposing the creation of metamodels that add significant knowledge in defining the processes of software development. With the aim of ensuring the quality of projects, a metamodel was developed based on the main concepts of MPS.BR, whereas the reference models that currently exist do not always fit the needs found in Brazilian companies, due to the excessive amount of activities required for the servicing each level of maturity that makes these models. Research was carried out throughout the course of the work that led to the elaboration of the phases and activities of the metamodel that was modeled using the EPF Composer Framework. From a process metamodel was developed as a way to demonstrate its application and was subsequently conducted an evaluation model by analyzing GAP, using the spreadsheet formal evaluation of the MA-MPS to demonstrate that the proposed activities meet the quality standards compared to the ideal model. As a result of the work we have the process that was based on meta-model, taking into account all quality concepts studied. With the evaluation model of the metamodel was possible to make a comparison between what was proposed and the ideal model, in which it was found that 60% of metamodel meets established quality items in ideal model, demonstrating that the suggested activities add quality to the projects which are submitted. Within the context discussed in this study, it is concluded that the fact implement efficient processes in the development of software projects allows projects to be managed more rigorously, with no doubts about what should or should not be done. The Quality Assurance Plan should be applied throughout the lifecycle of software projects, with the intention of preventing rework caused by the correction of faults, leading to the accumulation of work that can result in abandonment procedures essential to the smooth running the project.

Palavras-chave: Framework EPF Composer. Metamodel. Model MPS.BR. Quality Assurance in Software Projects.

LISTA DE ILUSTRAÇÕES

Figura 1- Modelo Cascata	24
Figura 2- Modelo Incremental.....	25
Figura 3- Modelo de Prototipação	27
Figura 4- Modelo Espiral	28
Figura 5- Um ciclo com suas fases e iterações.....	31
Figura 6- Componentes do modelo MPS.BR	40
Figura 7- Níveis de maturidade do modelo CMMI	49
Figura 8- Pirâmide com níveis de maturidade dos modelos CMMI e MPS.BR	52
Figura 9- Definição do conteúdo do método versus a aplicação do conteúdo em um Processo	59
Figura 10- Estereótipos da Linguagem	60
Figura 11- Modelo de Atividades baseado no SPEM.....	62
Figura12- Fases de desenvolvimento do trabalho proposto.....	73
Figura13- Metamodelo	75
Figura 14- Fase de Entrada.....	77
Figura 15- Fase de Análise	78
Figura 16- Fase de Modelo	79
Figura 17- Fase de Desenvolvimento.....	80
Figura 18- Fase de Teste	81
Figura 19- Fase de Liberação	83
Figura 20- Fase de Avaliação.....	84
Figura 21- Processo - Fase de Entrada	85
Figura 22- Processo - Fase de Análise	86
Figura 23- Processo - Fase de Modelo	87
Figura 24- Processo - Fase de Desenvolvimento.....	89
Figura 25- Processo - Fase de Teste	90
Figura 26- Processo - Fase de Liberação	91
Figura 27- Processo - Fase de Avaliação	92
Figura 28 – Planilha de Avaliação – Processo GRE	93
Figura 30- Evidências – GRE 4 e GRE 5	95
Figura 31- Evidências – AP 1.1 e AP 2.1	96
Figura 32- Planilha de Avaliação – Processo GRP	97

Figura 33- Evidências – GPR 1 a GPR 4	98
Figura 34- Evidências – GRE 4 a GRE 6	99
Figura 35- Evidências – GRE 7 a GRE 9	99
Figura 36- Evidências GRP 10 a GRP 12	101
Figura 37- Evidências – GPR 13 a GPR 15	102
Figura 38- Evidências – GPR 16 e GPR 17	103
Figura 39- Evidências – AP 1.1 e AP 2.1	104
Figura 40- Evidências – AP 2.1	105

LISTA DE TABELAS

Tabela 1- Comparação entre os modelos para processo de software	47
Tabela 2- Níveis de maturidade do modelo MPS.BR.....	51
Tabela 3- Processo de Avaliação.....	65
Tabela 4- Escala de caracterização do grau de implementação de resultados esperados.....	67

LISTA DE ABREVIATURAS E SIGLAS

AD	Artefato Direto
AF	Afirmação
AI	Artefato Indireto
AMP	Avaliação e Melhoria do Processo Organizacional
AQU	Aquisição
CMMI	Capability Maturity Model Integration
DFP	Definição do Processo Organizacional
DRE	Desenvolvimento de Requisitos
DRU	Desenvolvimento para Reutilização
EPF	Eclipse Process Framework
FDD	Feature Driven Development
GCO	Gerência de Configuração
GDE	Gerência de Decisões
GPP	Gerência de Portfólio de Projetos
GQA	Gerência da Qualidade
GRE	Gerência de Requisitos
GRP	Gerência de Projetos
GRH	Gerência de Recursos Humanos
GRI	Gerência de Riscos
GRU	Gerência de Reutilização
HTML	Hyper Text Markup Language
IA	Instituição Avaliadora
IBM	Internacional Business Machines
ITP	Integração do Produto
ISO/IEC	Organização Internacional para Padronização/ Comissão Eletrotécnica Internacional
MA-MPS	Método de Avaliação para Melhoria de Processo de Software
MED	Medição
MOF	Meta-Object Facility
MPS.BR	Melhoria de Processo do Software Brasileiro
MR-MPS	Modelo de Referência para Melhoria de Processo de Software
NBR	Norma Brasileira

OMG	Object Management Group
PA	Área de Processo
PCP	Projeto e Construção do Produto
PMBOK	Project Management Body of Knowledge
SCAMPI	Standard CMMI Appraisal Method for Process
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro
SPEM	Software Process Engineering Metamodel
UML	Linguagem de Modelagem Unificada
VAL	Validação
VER	Verificação
XP	Extreme Programming

SUMÁRIO

1 INTRODUÇÃO	16
1.1 OBJETIVO GERAL	18
1.2 OBJETIVOS ESPECÍFICOS	18
1.3 JUSTIFICATIVA	19
1.4 ESTRUTURA DO TRABALHO	21
2 ENGENHARIA DE SOFTWARE	23
2.1 CICLOS DE VIDA DE DESENVOLVIMENTO	23
2.1.1 Modelo Cascata	24
2.1.2 Modelo Incremental	25
2.1.3 Modelo De Prototipação	26
2.1.4 Modelo Espiral	27
3 QUALIDADE DE SOFTWARE	30
3.1 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	30
3.1.1 Processo Unificado	30
3.2 DESENVOLVIMENTO ÁGIL	31
3.2.1 Extreme Programming	34
3.2.2 Scrum	36
3.2.3 Crystal	37
4 MODELO MPS.BR	39
4.1 MODELO DE REFERÊNCIA	40
4.1.1 Nível G – Parcialmente Gerenciado	41
4.1.2 Nível F – Gerenciado	42
4.1.3 Nível E – Parcialmente Definido	43
4.1.4 Nível D - Largamente Definido	44
4.1.5 Nível C – Definido	45
4.1.6 Nível B – Gerenciado Quantitativamente	45
4.1.7 Nível A – Em Otimização	46
4.2 COMPARATIVO ENTRE modelos de processo	46
4.3 COMPARATIVO ENTRE OS MODELOS DE PROCESSO CMMI E MPS.BR	48
4.4 ÁREA DE PROCESSO Garantia da QQualidade (GQA)	53
4.4.1 GQA1	54
4.4.2 GQA2	55
4.4.3 GQA3	56

4.4.4 GQA4	56
5 FRAMEWORK EPF COMPOSER	58
5.1 FORMATO DO MODELO.....	60
5.2 METAMODELO	62
5.3 MÉTODO DE AVALIAÇÃO (MA-MPS).....	64
5.3.1 Metodologia do MA-MPS	66
6 TRABALHOS CORRELATOS	68
6.1 MPS.BR - UM ESTUDO DO MODELO MPS.BR COMO BENEFÍCIO PARA PEQUENAS E MÉDIAS EMPRESAS	68
6.2 DESENVOLVIMENTO DE UMA METODOLOGIA PARA O GERENCIAMENTO DE PROJETOS BASEADO NA ÁREA DE PROCESSO: GERÊNCIA DE PROJETOS (GPR) DO MODELO DE MATURIDADE MPS.BR	69
6.3 PROPOSTA DE UM PROCESSO DE GERENCIAMENTO DE REQUISITOS PARA UMA EMPRESA DESENVOLVEDORA DE SOFTWARE COM FOCO NA SATISFAÇÃO DO NÍVEL G DO MPS.BR	69
6.4 AVALIAÇÃO DO PROCESSO DE GERÊNCIA DE CONFIGURAÇÃO UTILIZANDO O MODELO MPS.BR	70
6.5 UMA PROPOSTA DE APOIO SISTEMATIZADO À GERÊNCIA DE PORTFÓLIO DO MPS.BR	70
7 DESENVOLVIMENTO DO METAMODELO	71
7.1 METODOLOGIA.....	72
7.1.1 Identificação de modelos de referência	73
7.1.2 Sintetização dos conceitos de metamodelos	74
7.1.3 Construção do metamodelo	74
7.1.4 Construção do modelo de processo	85
7.1.5 Modelo de avaliação baseado no metamodelo proposto	92
7.2 RESULTADOS OBTIDOS	105
8 CONCLUSÃO	109

1 INTRODUÇÃO

A utilização de sistemas computacionais tem demonstrado um crescimento significativo no decorrer dos anos, não somente no Brasil, mas no mundo todo.

Cada vez mais, empresários de software e desenvolvedores demonstram a preocupação em atender de forma satisfatória as necessidades do mercado. Devido à crescente demanda e demais fatores envolvidos no processo, os níveis de qualidade acabam não sendo atingidos.

A falta de planejamento na estimativa do esforço necessário em projetos é um problema que ocorre com frequência, prejudicando o nível de previsibilidade não somente do esforço a ser realizado, como também dos custos envolvidos, dificultando a comparação entre o que foi estimado e o que foi realizado.

Estudos sobre a qualidade no setor de software brasileiro mostraram a necessidade de um esforço significativo capaz de aumentar a maturidade dos processos de software das empresas. (GUERRA, 2009, p.41).

Identificou-se também que a falta de qualidade dos sistemas computacionais presentes atualmente no mercado, está diretamente relacionada com a qualidade do processo a que é submetido. No intuito de aumentar a qualidade dos processos de desenvolvimento de software, consultores e engenheiros da área, propõem a criação de metamodelos que agregam um *know how* significativo na definição dos processos de desenvolvimento de software.

Com base em modelos de referência existentes, o modelo MPS.BR foi criado buscando o aproveitamento de modelos e padrões de melhoria de processo, atendendo à necessidade da implantação de princípios da engenharia de software adequado ao contexto das empresas e aos padrões internacionais de melhoria de processos de software.

O modelo MPS.BR é subdividido em sete níveis de maturidade (nível G a nível A), sendo que será abordado nesta pesquisa somente a área de

processo GQA do nível F, onde o foco principal é apoiar a gestão do projeto de software, agregando processos que preservem a Garantia da Qualidade.

Segundo Softex (2011), o propósito do Processo GQA é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os recursos e planos predefinidos.

Para o desenvolvimento de um novo processo que garanta a qualidade nos projetos de software, será utilizado o Framework EPF Composer, permitindo que o processo seja representado graficamente de maneira eficaz, descrevendo as atividades e definindo os papéis responsáveis por cada tarefa.

Este processo é baseado no metamodelo desenvolvido a partir dos conceitos do MPS.BR, utilizando como fonte de conhecimento modelos como o CMMI e Project Management Body of Knowledge (PMBOK) para comparar os modelos existentes ao metamodelo proposto.

Apesar de todo o estudo que tem sido realizado ao longo dos anos, diversos problemas relacionados à falta de qualidade dos sistemas desenvolvidos, continuam ocorrendo em muitas organizações desenvolvedoras de software.

Devido ao mau planejamento, ocorre um grande acúmulo de trabalho, que acaba resultando no abandono de planos e procedimentos essenciais para o bom andamento do projeto, o que resulta muitas vezes em produtos que funcionam, mas ultrapassam os prazos e custos estabelecidos previamente.

De acordo com Softex (2011), durante a avaliação de Garantia da Qualidade são identificados problemas e não conformidades relacionadas à correção ou melhoria de processos e trabalhos executados que se originam quando há desvios entre o esperado e o que foi realizado.

O plano de Garantia da Qualidade deve ser aplicado em todo ciclo de vida dos projetos de software, com a intenção de evitar possíveis falhas e por consequência, prevenir o retrabalho ocasionado pela correção dessas falhas.

O fato de implantar processos eficientes no desenvolvimento de

projetos de software permite que os projetos sejam gerenciados com maior rigor, sendo que não haverá dúvidas sobre o que deve ou não ser feito.

Esta pesquisa foi executada com a finalidade de realizar um estudo detalhado sobre aspectos da Engenharia de Software, voltada para a melhoria da qualidade em processos de desenvolvimento de software, tendo como base o modelo MPS.BR, que é um modelo de maturidade nacional, podendo ser aplicado à organizações de diversos tamanhos, utilizando padrões de qualidade internacional.

Dentro do contexto apresentado, encontrou-se a possibilidade de desenvolver um metamodelo baseado no modelo MPS.BR, utilizando conceitos e técnicas que garantam a qualidade no desenvolvimento dos projetos de software.

Com isso, foi elaborado um processo baseado no metamodelo proposto, no qual foi aplicado um método de avaliação com o intuito de determinar o percentual de aderência do processo em relação a um modelo ideal, neste caso, o modelo MPS.BR.

1.1 OBJETIVO GERAL

Desenvolver um metamodelo, baseado no modelo MPS.BR para a criação de processos aplicados à garantia da qualidade no desenvolvimento de projetos de software.

1.2 OBJETIVOS ESPECÍFICOS

Entre os objetivos específicos desta pesquisa estão:

- a) compreender o conceito de Metamodelos;
- b) compreender o modelo MPS.BR;
- c) compreender o funcionamento do Framework EPF Composer para a modelagem do metamodelo;
- d) desenvolver um modelo de processo baseado na área de processo GQA do MPS.BR;

- e) definir métodos, técnicas e padrões para o desenvolvimento de um metamodelo para garantir a qualidade em projetos de software;
- f) definir um método de avaliação do metamodelo proposto.

1.3 JUSTIFICATIVA

As atividades empregadas com objetivo de garantir a qualidade em projetos de software possibilitam maior visibilidade do projeto para toda a organização, de uma forma independente levando em conta o processo e o produto (SOFTEX, 2011).

As atividades de Garantia da Qualidade devem ser empregadas nas organizações com a finalidade de ajudar a preparar e rever procedimentos que devem ser seguidos do início até o final do projeto, servindo de apoio à gerência de projetos.

O responsável por desempenhar as atividades de garantia da qualidade nas organizações, deve ter autoridade e independência, o bastante para executar suas funções, sendo que tem o compromisso de fiscalizar se todas as tarefas estão sendo desempenhadas conforme os procedimentos estabelecidos.

O processo de Garantia da Qualidade deve ser empregado com o objetivo de avaliar objetivamente os processos executados, produtos de trabalho e serviços em relação à descrição de processos aplicáveis, padrões e procedimentos (SOFTEX, 2011).

As não conformidades devem ser identificadas, documentadas e corrigidas. E tanto a equipe quanto a gerência do projeto deve receber *feedback* com os resultados das atividades que foram desempenhadas para manter a qualidade nos projetos.

A melhoria do processo de desenvolvimento de software resulta no aumento da qualidade e da produtividade, sendo também um diferencial da empresa.

Com a melhoria de processos é possível demonstrar a aceleração da curva de aprendizado dos desenvolvedores, aumentando a moral da equipe, elevando a produtividade da empresa, diminuindo custos do desenvolvimento e principalmente aumentando a qualidade do software produzido (DIAS, 2008).

O Modelo de Melhoria de Processo de Software Brasileiro – MPS.BR, foi escolhido como objeto de estudo por ser um modelo mais acessível e que engloba regras e padrões de desenvolvimento de software utilizados internacionalmente.

O MPS.BR é um repositório de várias práticas que vêm sendo utilizadas com sucesso ao longo dos anos, permitindo a avaliação dos processos de forma objetiva, detectando os pontos fortes e fracos empregados no desenvolvimento de softwares.

Seria conveniente avaliar os processos, técnicas e métodos de maneira eficiente, prevenindo esforços significativos sem a possibilidade de retorno.

Segundo Rocha (2001), além de conhecer os métodos e processos de engenharia de software, é necessário saber como avaliá-los para propiciar uma base sólida à evolução do processo. A disponibilidade de uma base histórica de dados é um elemento indispensável, que abrange as diversas perspectivas e fatores que afetam o desenvolvimento de software.

A meta da garantia da qualidade é fornecer à gerência os dados necessários para que fique informada sobre a qualidade do produto, ganhando assim compreensão e confiança de que a qualidade do produto está satisfazendo suas metas (PRESSMAN, 2006).

A qualidade é um aspecto que deve ser tratado simultaneamente ao processo de desenvolvimento do software, pois ela não pode ser imposta depois que o produto já está finalizado (ROCHA, 2001).

Um dos grandes motivos que levaram ao desenvolvimento desta pesquisa é a busca da melhoria contínua na criação de processos que garantam a qualidade no desenvolvimento de projetos de software, estabelecendo técnicas e métodos de identificação, documentação e tratamento de não conformidades.

1.4 ESTRUTURA DO TRABALHO

A estrutura do trabalho divide-se em dez capítulos.

No primeiro capítulo é contextualizada a introdução, apresentando o objetivo geral, os objetivos específicos e a justificativa do trabalho realizado.

No segundo capítulo são tratados os Ciclos de vida de desenvolvimento, onde são descritos os modelos Cascata, Incremental, Prototipação e Espiral.

O terceiro capítulo trata da Qualidade de Software demonstrando os principais conceitos de Processo de desenvolvimento de software, no qual é apresentado o Processo Unificado, o Desenvolvimento Ágil, o Extreme Programming, o Scrum e o Crystal.

No quarto capítulo é contextualizado o modelo MPS.BR, onde são detalhados os seus conceitos e modelos de referência, apresentando todos os seus níveis de maturidade.

Neste capítulo também é realizado o comparativo entre os modelos de processo CMMI e MPS.BR, detalhando também a área de processo GQA do MPS.BR.

O capítulo cinco trata do Framework EPF Composer, demonstrando o seu formato baseado no SPEM.

No capítulo 6 são descritos detalhadamente os conceitos de Metamodelos.

No capítulo 7 é tratado o Método de Avaliação MA-MPS, demonstrando a sua metodologia de aplicação.

No capítulo 8 são apresentados os trabalhos correlatos, onde são demonstrados os trabalhos relacionados à área do trabalho desenvolvido.

No capítulo 9 é demonstrado o trabalho prático desenvolvido, sendo descrito de forma detalhada a metodologia utilizada para o desenvolvimento da pesquisa, da elaboração do metamodelo e do desenvolvimento do processo, além da metodologia utilizada para a realização da avaliação do metamodelo, apresentando os resultados obtidos.

Por fim, no capítulo 10 é descrita a conclusão ao qual se chegou com a finalização do trabalho desenvolvido.

2 ENGENHARIA DE SOFTWARE

A Engenharia de Software emprega métodos que permitem que seja feito o planejamento de todas as fases de desenvolvimento do projeto de software de forma que seja utilizado um conjunto de critérios que garantam a qualidade do software (PRESSMAN, 2006).

Para empregarmos os conceitos da Engenharia de Software nos projetos de desenvolvimento é necessário definir uma sequência de procedimentos a serem aplicados, relatórios e documentos a serem redigidos com o propósito de manter a organização em todas as fases de desenvolvimento do software, permitindo que sejam feitas mudanças ao longo do trabalho e o devido acompanhamento e avaliação do que está sendo produzido.

A Engenharia de Software possui três fases principais: a definição, o desenvolvimento e a manutenção, onde é desenvolvida a análise do sistema e dos requisitos, a codificação, o teste de software, no qual, permite que seja feita uma manutenção corretiva, incluindo melhoramentos e adaptações.

Além de conhecer os métodos, ferramentas e processos envolvidos no desenvolvimento do projeto, deve-se levar em consideração, principalmente, o método de avaliação que será empregado para garantir que o processo evolua de acordo com o esperado (ROCHA; MALDONADO; WEBER, 2001).

Com a evolução da Engenharia de Software houve a necessidade de adequar os métodos utilizados de acordo com as necessidades encontradas e as situações vivenciadas no cotidiano.

Com a finalidade de suprir essas necessidades, surgiram novos paradigmas de engenharia de software, ou também chamados de Ciclos de vida de Desenvolvimento, que são definidos de acordo com os métodos, ferramentas e as funcionalidades previamente estabelecidas no projeto de desenvolvimento do software.

2.1 CICLOS DE VIDA DE DESENVOLVIMENTO

Ciclos de vida são várias atividades realizadas durante o desenvolvimento do software, determinando se a proposta é viável, qual a estrutura e as funcionalidades, quais as estimativas de custo, como deve ser planejado o cronograma das atividades de desenvolvimento, adotando práticas que ajudarão a garantir a qualidade do software (GUSTAFSON, 2003).

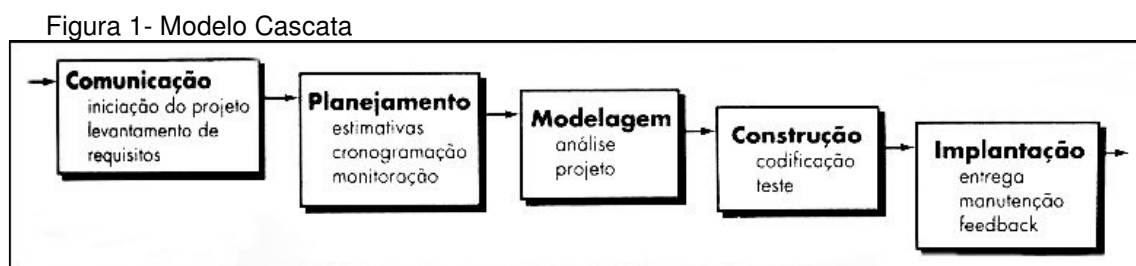
De acordo com o tipo de software que será desenvolvido, é possível determinar qual o paradigma adequado, levando em consideração aspectos que definem se o projeto deverá ser desenvolvido de maneira a ser aprimorado posteriormente de acordo com o surgimento das necessidades.

Com isso se torna possível prever o processo de desenvolvimento do trabalho, acompanhando os resultados apresentados em cada etapa.

2.1.1 Modelo Cascata

Um dos paradigmas mais antigos da Engenharia de Software é o modelo cascata, também chamado de ciclo de vida clássico, que segue uma linha de desenvolvimento sequencial, onde cada fase de desenvolvimento do software deve ser concluída antes que se inicie a próxima etapa (PRESSMAN, 2006).

Conforme a Figura 1, o desenvolvimento inicia-se com o levantamento dos requisitos, partindo para o planejamento, modelagem, análise e projeto, construção através da codificação e execução de testes, finalizando com a implantação e manutenção do software.



Fonte: Adaptado de Pressman (2006, p.39).

A aplicação desse modelo nos projetos de desenvolvimento de software deve ser feita de forma cautelosa, tendo consciência de que o seu

desenvolvimento será de forma linear, não permitindo que posteriormente sejam feitas modificações nos requisitos estabelecidos inicialmente.

O desenvolvimento sequencial traz uma desvantagem, sendo que não permite que seja feita a revisão das etapas desenvolvidas anteriormente, prejudicando a correção de problemas detectados na fase de testes (SOMMERVILLE, 2007).

O trabalho deve seguir de forma sequencial, o que muitas vezes pode ocasionar em estados de bloqueio, sendo que antes de iniciar uma determinada etapa do modelo, é necessário aguardar a finalização da etapa anterior, fazendo com que exceda o tempo gasto para o desenvolvimento do software.

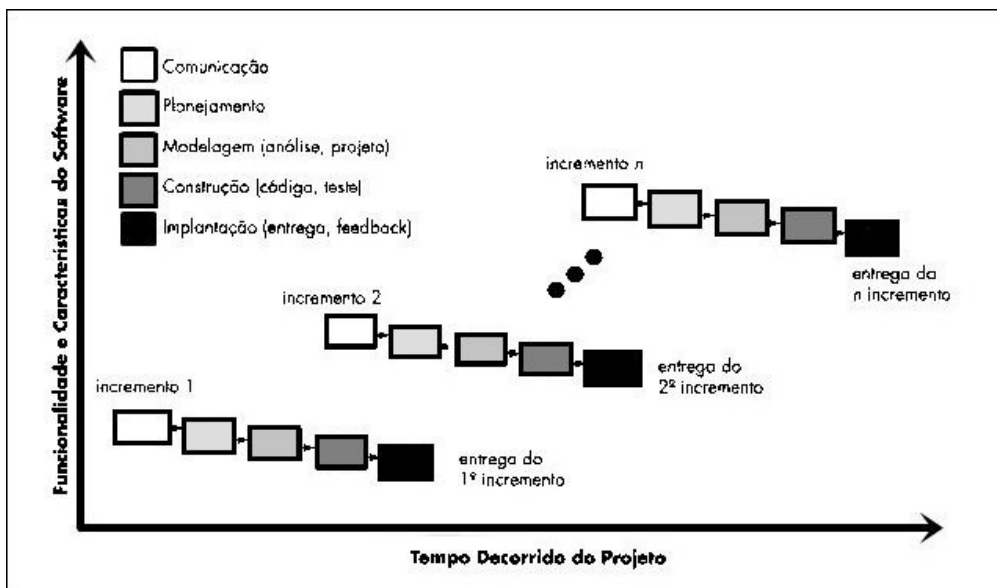
2.1.2 Modelo Incremental

O objetivo do modelo incremental é projetar um sistema com funcionalidades mínimas e entregá-lo ao cliente de forma que sejam fornecidas posteriormente novas versões do software de acordo com o qual é desenvolvido e aperfeiçoado (GUSTAFSON, 2003).

O modelo incremental utiliza uma série de incrementos, ou seja, versões do software, que fornecem mais funcionalidades à medida que cada versão é desenvolvida e entregue ao cliente.

De acordo com a figura 2, o processo inicia-se com planejamento do primeiro incremento, onde são atendidas as funcionalidades básicas, que serão revisadas detalhadamente ou avaliadas pelo cliente com a finalidade de melhor satisfazer as necessidades no próximo incremento a ser desenvolvido, repetindo esse processo até que o software esteja completamente desenvolvido.

Figura 2- Modelo Incremental



Fonte: Adaptado de Pressman (2006, p.40).

A aplicação do modelo incremental é útil quando não há mão de obra disponível e prazo suficiente para o desenvolvimento completo do software, dentro do prazo de entrega estabelecido (PRESSMAN, 2006).

Além disso, o desenvolvimento através de incrementos permite que a cada incremento desenvolvido sejam realizados testes detalhados possibilitando a detecção de falhas no funcionamento, permitindo que sejam feitas as correções de uma forma mais eficiente. Tendo em vista, que realizando testes somente no produto final, algumas falhas podem passar despercebidas.

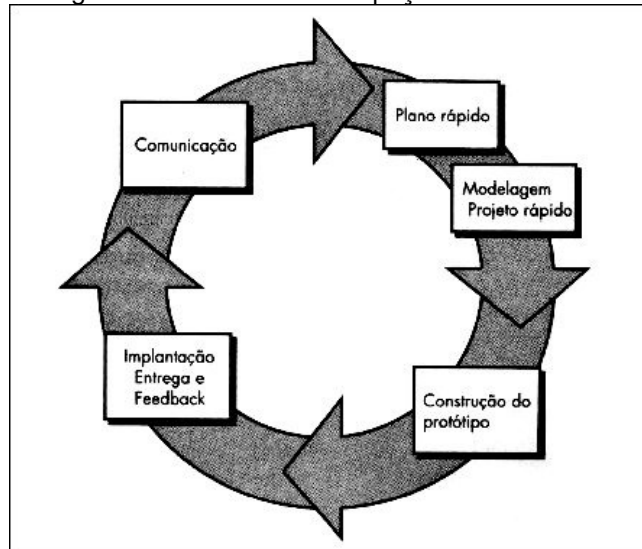
2.1.3 Modelo De Prototipação

A aplicação do paradigma de prototipação pode oferecer uma melhor abordagem quando os requisitos e funcionalidades do sistema não são claramente especificados, gerando dúvidas no processo de desenvolvimento e na aplicação de técnicas para a prevenção e resolução de problemas (PRESSMAN, 2006).

Conforme a Figura 3, através da comunicação entre o cliente e o engenheiro de software são definidas as necessidades e os objetivos do software a ser desenvolvido. A partir destas especificações é criado um projeto rápido dando origem a um protótipo que será implementado e depois avaliado

pelo cliente com a finalidade de aprimorar os requisitos do software e adequá-lo às necessidades do cliente.

Figura 3- Modelo de Prototipação



Fonte: Adaptado de Pressman (2006, p.43).

Levando em consideração que o protótipo pode servir como o primeiro sistema, posteriormente devem ser aprimoradas as funcionalidades que funcionam precariamente, com a finalidade de garantir a qualidade do sistema.

Com isso, deve-se ter consciência de que o desenvolvimento do protótipo surge somente como uma ferramenta de definição de requisitos, sendo descartado parcialmente, para que o software real seja submetido à engenharia utilizando as técnicas de qualidade de software em seu desenvolvimento.

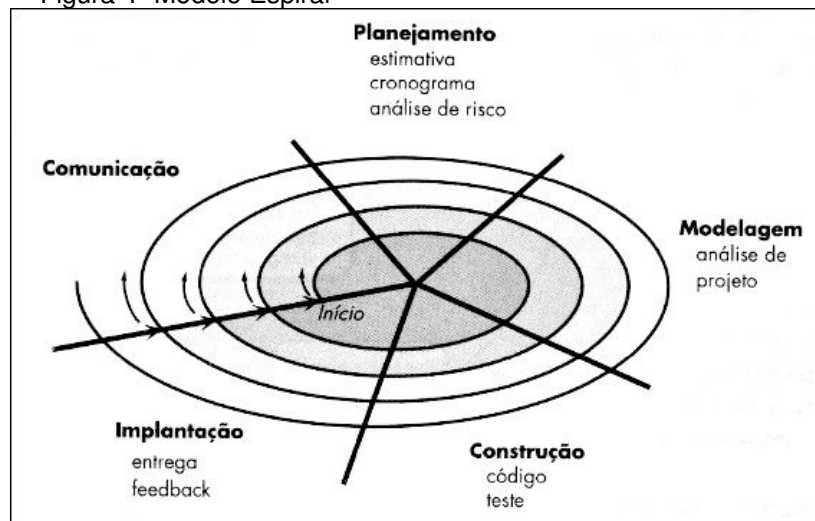
2.1.4 Modelo Espiral

A imagem do modelo proposto por B. Boehm é de uma espiral que começa no meio e continuamente revisa as tarefas básicas do usuário: comunicação, planejamento, análise de risco, engenharia, construção e implantação, e avaliação do usuário (GUSTAFSON, 2003).

O modelo Espiral engloba as características mais eficientes dos ciclos de desenvolvimento citados anteriormente, agregando a ele a execução de análises de risco durante a aplicação do modelo (PRESSMAN, 2006).

Conforme a Figura 4, o modelo Espiral é dividido em um conjunto de atividades definidas pela equipe de engenharia de software, seguindo um caminho espiral no sentido horário, onde cada loop na espiral representa uma fase de desenvolvimento do projeto de software.

Figura 4- Modelo Espiral



Fonte: Adaptado de Pressman (2006, p.45).

O desenvolvimento inicia com a definição dos objetivos e restrições do projeto através da elaboração de um plano detalhado de gerenciamento com a finalidade de identificar os riscos.

Para cada risco identificado, é realizada uma análise detalhada com o propósito de apontar providências a serem tomadas para diminuir os riscos, selecionando um modelo de desenvolvimento para o sistema.

Após este procedimento, o projeto é revisado e a partir disso é tomada a decisão de prosseguir ou não ao próximo loop da espiral. Dando início à elaboração de planos para a próxima fase de desenvolvimento.

O modelo Espiral pode ser implantado com sucesso em grandes aplicações, mas também pode apresentar problemas (ALVES; VANALLE, 2001).

Dentre os problemas que podem ocorrer na aplicação do modelo Espiral, pode ser citada a limitação do modelo em relação às possíveis alterações necessárias para a sua adequação em sistemas terceirizados.

Com a implementação do modelo Espiral, são definidos os objetivos a serem atendidos e as restrições que devem ser consideradas no decorrer do desenvolvimento.

A análise de riscos é efetuada com o objetivo de identificar dúvidas e incertezas que podem influenciar negativamente, sendo aplicados outros modelos para simular os problemas encontrados buscando a forma mais adequada de refinar os requisitos do sistema.

Com base nos conceitos apresentados até o momento, é possível perceber que a Engenharia de Software, ao longo dos anos tem buscado alternativas de adequação dos modelos de desenvolvimento de software na tentativa de manter e aprimorar a Qualidade de Software nas organizações, preservando conceitos utilizados atualmente e inserindo outros, para suprir as falhas encontradas.

3 QUALIDADE DE SOFTWARE

A qualidade no desenvolvimento de software está ligada diretamente à conformidade dos requisitos funcionais e critérios de desenvolvimento documentados, determinando a maneira com que o software é submetido à engenharia (PRESSMAN, 2006).

Com o passar dos anos, e a crescente demanda na área de desenvolvimento de software, as técnicas de implementação que antes atendiam às necessidades, agora já não são satisfatórias, ocasionando em falhas no sistema.

Devido à complexidade cada vez maior dos sistemas, a diminuição do tempo previsto para o desenvolvimento, e a falta de planejamento adequado, os desenvolvedores de software acabam enfrentando dificuldades.

Na tentativa de sanar estes problemas, foram desenvolvidas técnicas de desenvolvimento de software melhor estruturadas.

3.1 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Conforme Rocha e Weber (1999) é necessário que as organizações desenvolvedoras de software aprendam a entregar produtos de qualidade, respeitando o prazo estabelecido e os custos planejados.

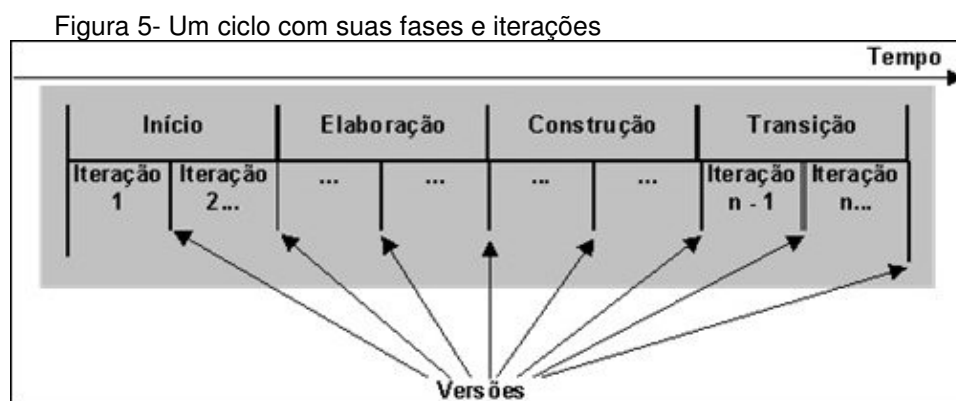
Devido aos diversos problemas encontrados durante o processo de desenvolvimento de software, os profissionais da área buscam cada vez mais, aprimorar as técnicas utilizadas com a finalidade de garantir a qualidade nos sistemas desenvolvidos, através de técnicas eficazes tanto para a melhoria do trabalho da equipe, quanto para a qualidade do produto final.

Para isso, foram concebidos modelos especializados de processo que apresentam características de alguns modelos convencionais, que utilizados isoladamente não atendem às necessidades de determinado projeto.

3.1.1 Processo Unificado

O Processo Unificado é uma técnica que utiliza algumas características e os melhores recursos dos modelos de desenvolvimento de software convencionais com o objetivo de implementar os melhores princípios de desenvolvimento ágil (PRESSMAN, 2006).

A Figura 5 demonstra um ciclo composto pelas fases de início, elaboração, construção e transição e são subdivididas em iterações.



Fonte: Martins (2009).

O processo unificado é formado por ciclos que são repetidos durante a vida de um sistema. A cada conclusão dos ciclos, uma versão do sistema é gerada e distribuída interna, ou externamente (MARTINS, 2009).

O Processo Unificado é composto por um conjunto de atividades utilizadas para transformar requisitos do usuário em um sistema de software iterativo e incremental, dando a sensação evolucionária no desenvolvimento, direcionado a casos de uso, ou seja, descrevendo a visão do cliente em um sistema centrado na arquitetura, mantendo o foco na compreensibilidade e na execução das metas adequadas.

3.2 DESENVOLVIMENTO ÁGIL

No ano de 2001, dezesseis notáveis desenvolvedores de software, dentre eles, Kent Beck, assinaram o manifesto para o desenvolvimento ágil de software com o objetivo de solucionar os problemas reais da engenharia de software, proporcionando benefícios aos projetos no qual são aplicados.

A agilidade no desenvolvimento de software é mais do que uma resposta à modificação. Enfatiza a entrega rápida do software e reconhece que o planejamento é importante, mas que um plano de projeto deve ser flexível (PRESSMAN, 2006).

Para alcançar a agilidade no desenvolvimento de software, foram apontados doze pontos principais:

- a) procura satisfazer o cliente de forma contínua, tendo-o como a maior prioridade;
- b) aceita modificações no sistema mesmo que tardiamente, sendo que estas modificações são consideradas como um diferencial competitivo;
- c) entrega de softwares funcionando no menor espaço de tempo;
- d) trabalho em equipe entre os desenvolvedores e o pessoal de negócio durante todo o projeto;
- e) construção de projetos em torno de indivíduos motivados, oferecendo estrutura e o apoio necessário para o desenvolvimento do trabalho;
- f) conversa entre a equipe para transmitir a informação de forma mais eficiente;
- g) a principal medida de progresso é manter um bom funcionamento do software desenvolvido;
- h) manter um ritmo constante de desenvolvimento sustentável através de processos ágeis;
- i) atenção contínua à excelência técnica e ao bom projeto para facilitar a agilidade;
- j) maximizar a quantidade de trabalho não efetuado;
- k) manter uma equipe auto-organizada para obter melhores arquiteturas e requisitos de projetos;
- l) tomada de decisões em equipe sobre como se tornar mais efetiva, ajustando adequadamente seu comportamento em situações de intervalos regulares.

Para aplicar a agilidade em processos de software é de extrema importância que o processo seja projetado de forma que possibilite alterações no decorrer das atividades aperfeiçoando-as e eliminando tarefas desnecessárias.

É fundamental que os processos sejam adaptáveis de forma incremental. Para isso, é importante que sejam desenvolvidos e disponibilizados protótipos do software em curtos períodos de tempo para que sejam avaliados pelo cliente ou equipe de testes, permitindo que sejam feitas as alterações e correções necessárias de forma rápida e eficaz.

De acordo com Pressman (2006), para a aplicação de técnicas de desenvolvimento ágil, é muito importante que a equipe de desenvolvimento esteja comprometida com o trabalho, sendo assim algumas características são fundamentais como:

- a) competência: manter conhecimento global do processo e das técnicas que a equipe decidiu aplicar;
- b) foco comum: Toda a equipe de desenvolvimento deve estar comprometida e focada em uma meta comum a todos, como concentrar-se em entregar incrementos nos prazos especificados e realizar adaptações de acordo com a necessidade do cliente;
- c) colaboração: Os membros da equipe devem colaborar uns com os outros para criar informações que facilitarão na compreensão do trabalho realizado pela equipe e construir informações que forneçam valor de negócio para o cliente;
- d) capacidade de tomada de decisão: A equipe deve ter autonomia para tomar decisões sobre aspectos técnicos e de projeto;
- e) habilidade de resolver problemas vagos: Continuamente a equipe se depara com problemas que aparecem no decorrer do processo. Com isso devem reconhecer que cada modificação efetuada e cada resolução de problemas é uma lição aprendida

que pode ser benéfica para a equipe mais adiante no projeto ou em projetos futuros;

f) respeito e confiança mútua: Uma equipe ágil deve manter o respeito entre os envolvidos no trabalho e a confiança necessária para que cada um saiba do cumprimento de suas tarefas com a finalidade de manter o processo num todo e não em partes isoladas;

g) auto-organização: A colaboração de todos para a organização da equipe é um fator muito importante no andamento do processo. A equipe se torna mais motivada aceitando a responsabilidade de cumprir os compromissos estabelecidos previamente.

Atualmente existem diversos modelos ágeis de processo destacando-se: Extreme Programming (XP), Scrum e Crystal.

Cada um dos processos mencionados anteriormente possuem características distintas, sendo que o sucesso de sua aplicação depende do tipo de estrutura a ser desenvolvida e da configuração do projeto em questão.

3.2.1 Extreme Programming

Segundo Beck (2004), a XP é uma disciplina de desenvolvimento de software eficiente, flexível, previsível e de baixo risco que se destaca por diferenciais como:

- a) feedback antecipado, concreto e contínuo derivado de ciclos curtos;
- b) utiliza uma abordagem incremental de planejamento, gerando um plano que evoluirá no decorrer do projeto;
- c) possui habilidade de agendar a implementação das funcionalidades permitindo que sejam feitas modificações de acordo com as necessidades de negócio;
- d) confiança nos testes escritos por programadores e clientes com a finalidade de monitorar o andamento e a evolução do

desenvolvimento, permitindo que seja feita a detecção de erros de forma rápida;

- e) confiança na comunicação oral, testes e código-fonte para comunicar a estrutura e o objetivo do sistema;
- f) utilização e confiança em um processo de desenvolvimento evolutivo;
- g) confiança na colaboração dos programadores envolvidos no projeto;
- h) Confiança nas práticas aplicadas a curto e longo prazo.

A XP é uma técnica de desenvolvimento que aborda o risco em todas as etapas do processo, gerando softwares de alta qualidade de forma produtiva, exigindo ciclos curtos de entrega com iterações de funções requisitadas pelo cliente com o objetivo de obter feedback sobre o processo. Em cada iteração são planejadas as tarefas que devem ser executadas para que possam ser solucionados os problemas encontrados.

Na metodologia XP o processo se torna mais ágil devido ao desenvolvimento iterativo, onde o progresso do desenvolvimento é avaliado durante cada iteração, o que simplifica o planejamento, tornando-o viável (MALLMANN, 2011).

São desenvolvidas primeiramente as funcionalidades com maior prioridade, com a intenção de satisfazer as necessidades mais importantes do cliente. Desta forma é criado e mantido um conjunto de testes que são executados a cada modificação com o objetivo de assegurar a qualidade do software desenvolvido.

A XP foi desenvolvida para abordar as necessidades de pequenas equipes de desenvolvimento diante de requisitos vagos e instáveis (BECK, 2004).

Segundo Mallmann (2011), a XP foi criada com o intuito de satisfazer plenamente as necessidades do cliente, entregando o software de acordo com o que foi solicitado e na data em que o cliente desejar.

Com isso, conclui-se que aplicando o desenvolvimento através de iterações é possível detectar as falhas existentes antes mesmo da finalização

do software. Levando em consideração que é mais vantajoso resolver pequenos problemas em um curto espaço de tempo do que resolver grandes problemas perto da data de entrega do software.

3.2.2 Scrum

Segundo Pressman (2006), o Scrum é um modelo de processo de desenvolvimento ágil que foi desenvolvido na década de 90 por Jeff Sutherland e sua equipe, com a finalidade de incorporar atividades de requisitos, análise, projeto, evolução e entrega.

O método Scrum utiliza técnicas incrementais e iterativas de desenvolvimento focando em situações onde os requisitos podem mudar constantemente. Sendo assim, são criadas iterações que são chamadas de Sprints. A quantidade de Sprints pode variar de acordo com as atividades a serem realizadas, o tamanho e a complexidade do produto a ser desenvolvido, podendo ser inspecionados, ajustados, testados, documentados e expandidos.

Segundo Mallmann (2011), o método Scrum foi desenvolvido com a finalidade de permitir que seja adaptado durante qualquer fase do projeto de desenvolvimento de acordo com o surgimento das necessidades.

Pequenas equipes são organizadas com o objetivo de maximizar a comunicação através de reuniões diárias que duram geralmente 15 minutos, onde são tratadas questões pendentes, os obstáculos encontrados na execução das tarefas e é feita a especificação do que será realizado até a próxima reunião. Todas as respostas dos membros da equipe são avaliadas pelo líder da equipe onde são detectados problemas potenciais possibilitando que estes problemas sejam solucionados o mais rápido possível.

Após o desenvolvimento de cada iteração, são disponibilizadas demos, ou seja, a entrega de incrementos com algumas funcionalidades para que possam ser utilizadas e avaliadas pelo cliente dentro de um tempo previamente estipulado, com o objetivo de detectar melhorias a serem realizadas e falhas a serem corrigidas (PRESSMAN, 2006).

O método Scrum permite que a equipe de desenvolvimento trabalhe de modo bem sucedido mesmo tendo que lidar com todas as incertezas que geralmente são encontradas.

Este método possibilita o monitoramento constante das atividades desenvolvidas pela equipe, assegurando que as técnicas e práticas aplicadas estão sendo postas em prática da maneira adequada.

3.2.3 Crystal

O Crystal é uma família de modelos ágeis de processos de desenvolvimento que adotam técnicas iterativas visando à adaptação do processo utilizado de acordo com o tamanho e a complexidade do projeto a ser desenvolvido (PRESSMAN, 2006).

A família Crystal permite que equipes ágeis possam selecionar dentre os processos existentes, aquele que seja mais apropriado ao projeto que será desenvolvido.

A melhoria no desenvolvimento de software é conquistada com a integração efetiva da equipe, desenvolvendo as habilidades individuais das pessoas envolvidas no projeto (COCKBURN, 2005).

Os projetos de desenvolvimento que utilizam metodologias da família Crystal, usam ciclos de desenvolvimento com um incremento que geralmente dura entre um a três meses, e permite que sejam incorporados ao projeto práticas de outras metodologias (LEITÃO, 2010).

Cada projeto de software é desenvolvido por equipes de tamanhos e habilidades diferentes, cada um com prioridades distintas. Com o intuito de proporcionar a adequação necessária a cada processo de desenvolvimento, a família Crystal foi criada, destacando técnicas de desenvolvimento de software orientadas a humanos, levando em consideração a cooperação das pessoas e a utilização de técnicas selecionadas de acordo com a necessidade de cada projeto.

A família Crystal é composta de diferentes metodologias de desenvolvimento ágil possibilitando que seja feita a adequação necessária de acordo com o tamanho da equipe e suas necessidades (AMORIN, 2008).

O desenvolvimento de software no modelo Crystal, adota uma metodologia de jogo cooperativo, instanciando uma coleção de técnicas ágeis como estimativas de tamanho por Story Points, gerenciamento de projetos com Kanban, Pair Programming etc.. Que foca diretamente no aumento da maturidade e na troca de experiências de equipes num mundo ágil, fomentando o desenvolvimento de software com qualidade e agilidade.

4 MODELO MPS.BR

O MPS.BR, ou seja, Melhoria de Processo do Software Brasileiro é um programa criado em dezembro de 2003, coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) com o objetivo de melhorar a qualidade dos processos de softwares brasileiros, com duas importantes metas a serem alcançadas a médio e longo prazo.

De acordo com Softex (2011), as metas do MPS.BR são:

- a) meta técnica: Visando a criação e o aprimoramento do modelo MPS;
- b) meta de mercado: Visando a adoção e disseminação do modelo MPS em todo o país, focando principalmente nas pequenas e médias empresas, com um custo razoável e em um intervalo de tempo justo.

Com as mudanças que vêm ocorrendo no mundo dos negócios, as empresas buscam cada vez mais a adequação de seus processos produtivos, e para a isso as empresas desenvolvedoras de software se vêm obrigadas a adaptar-se de acordo com essa necessidade para atender ao mercado que é tão competitivo.

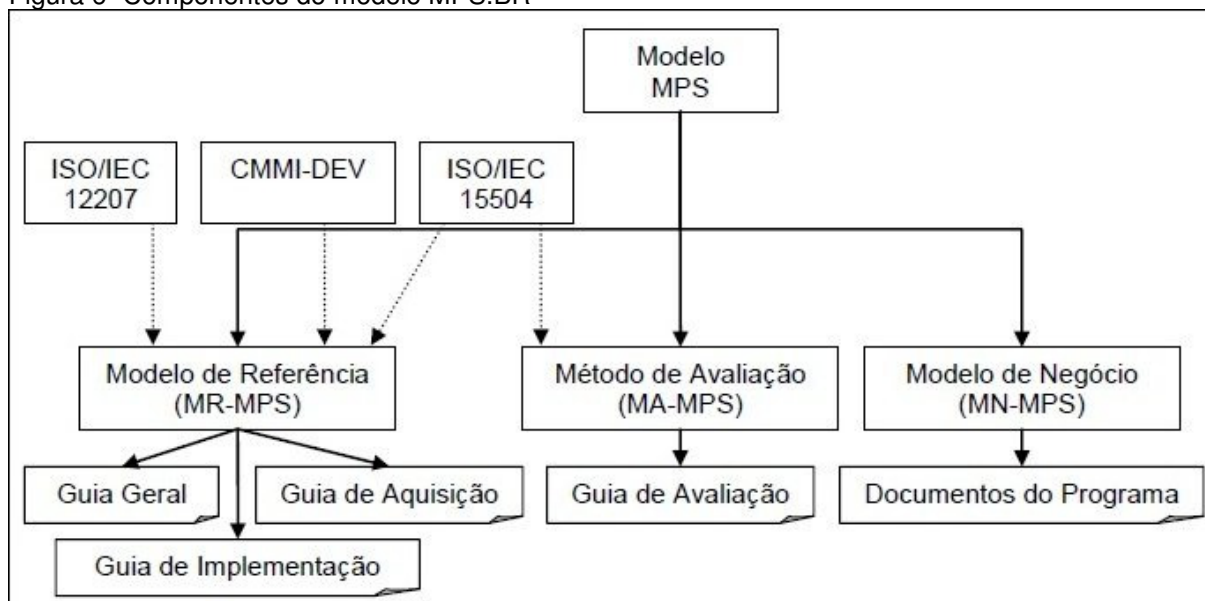
Para suprir as necessidades do mercado e com o intuito de competir nacional e internacionalmente, as empresas de software estão focando principalmente na qualidade de seus produtos e serviços correlatos, através de modelos de processo eficazes, buscando cada vez mais a produção de softwares eficientes de acordo com os padrões internacionais de qualidade.

O modelo MPS.BR, foi desenvolvido e é constantemente adaptado com a finalidade de adequar seus processos de desenvolvimento, aplicando conceitos da engenharia de software já existentes, buscando a adequação ao perfil de empresas com características diferentes, atingindo de micro à médias empresas, sejam elas públicas ou privadas.

Segundo Softex (2011), o MPS.BR é formado por três componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS).

Estes componentes são baseados nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade em produtos de software, que são descritos por meio de guias e documentos, conforme a figura 6:

Figura 6- Componentes do modelo MPS.BR



Fonte: Softex (2011, p.13).

Conforme mostra a figura 6, cada componente do MPS.BR é documentado na forma de Guias, que servem para demonstrar de forma detalhada todas as técnicas utilizadas no MPS.BR.

No Guia de Aquisição são demonstradas as boas práticas necessárias para a aquisição de software e serviços.

No Guia de Implementação, são sugeridas algumas formas de como implementar o MR-MPS nas organizações.

O Guia de Avaliação é voltado para as Instituições Avaliadoras (IA), contendo os requisitos e métodos de avaliação de acordo com a Norma Internacional.

No Guia Geral é feita uma explanação resumida de todos os Guias citados anteriormente.

4.1 MODELO DE REFERÊNCIA

O Modelo de Referência (MR-MPS) define os níveis de maturidade, processos e atributos do processo, declarando os objetivos e os resultados

esperados em sua execução, permitindo que seja feita a avaliação da capacidade dos processos, identificando a habilidade de alcançar os objetivos de negócio, atuais e futuros, de acordo com os atributos de processo associados a cada nível de maturidade, descritos no Guia Geral.

No Guia de Aquisição são descritas as boas práticas para a aquisição de software e serviços correlatos. No Guia de Implementação são sugeridas algumas formas de como proceder na implantação de cada um dos níveis do MR-MPS.

O MR-MPS estabelece níveis de maturidade que servem para analisar a evolução de processos através de estágios que definem qual o desempenho dos processos em uma organização, podendo prever o seu desempenho futuro a partir dos resultados apresentados no decorrer de cada estágio de melhoria da implementação dos processos.

Segundo Softex (2011), a escala de maturidade é dividida em sete níveis, que tem início no nível G, progredindo até o nível A, onde cada um destes níveis de maturidade possuem propósitos de melhoria da qualidade com foco em uma determinada área. Sendo assim, para chegar ao próximo nível, é necessário que todos os resultados esperados para aquele nível sejam atingidos.

A divisão do processo em níveis, proporciona uma melhor implementação e avaliação, permitindo uma melhor visibilidade dos resultados obtidos para a melhoria dos processos em prazos mais curtos, mesmo sendo para micros, pequenas ou médias empresas.

4.1.1 Nível G – Parcialmente Gerenciado

De acordo com Softex (2011), o nível de maturidade G é composto pelos processos de Gerência de Projetos (GRP) e Gerência de Requisitos (GRE).

O Processo GRP estabelece planos que definem os recursos a serem utilizados e as atividades a serem desenvolvidas, bem como a

disponibilização de informações que permitam que sejam efetuadas correções relacionadas a problemas no desempenho do projeto.

O Processo GRE é utilizado para gerenciar os requisitos e os componentes do produto com o objetivo de identificar requisitos e planos inconsistentes no projeto.

Através do comprometimento da equipe técnica são avaliados os requisitos baseando-se em critérios objetivos, permitindo que mudanças nos requisitos sejam gerenciadas ao longo do projeto.

4.1.2 Nível F – Gerenciado

Segundo Softex (2011), o nível de maturidade F é formado pelos processos do nível G juntamente com os Processos de Aquisição (AQU), Gerência de Configuração (GCO), Garantia da Qualidade (GQA), Gerência de Portfólio de Projetos (GPP) e Medição (MED).

O Processo AQU permite que seja gerenciada a aquisição de produtos ou serviços que satisfaçam as necessidades do cliente, definindo estratégias de aquisição, critérios de aceitação do produto e metas a serem cumpridas.

Com isso, são estabelecidos critérios de seleção do fornecedor através da avaliação das propostas em busca de um produto que satisfaça a necessidade do cliente, mantendo um contrato que expresse claramente as responsabilidades e obrigações acordadas durante a negociação, incorporando o produto ao projeto, caso seja conveniente.

O Processo GCO tem a finalidade de estabelecer e manter a integridade de um processo ou projeto e disponibilizá-los aos envolvidos (SOFTEX, 2011).

Os itens de configuração são identificados através dos requisitos e colocados em baselines¹ que são registradas ao longo do tempo e disponibilizadas de forma controlada. Para garantir a integridade e a

¹ Consistem em repositórios do projeto, onde são armazenadas versões de diretórios e arquivos. À medida que o tempo passa, a baseline incorpora o trabalho concluído pelos desenvolvedores desde a criação da última baseline (NUNES, 2005).

consistência das informações contidas nas baselines, são realizadas auditorias de configuração.

O propósito do Processo GQA é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os recursos e planos predefinidos (SOFTEX, 2011).

O Processo GPP inicia e mantém projetos necessários, e sustentáveis com a finalidade de atender os objetivos e estratégias da organização. Estabelece a autoridade necessária para a execução e qualificação de projetos, determinando e justificando se é necessário dar continuidade aos investimentos em determinado projeto.

O Processo MED tem a função de coletar e armazenar os dados relativos dos produtos para que sejam analisados e relatados todos os processos implementados nos projetos com a finalidade de apoiar os objetivos da organização, informando aos interessados os resultados das análises para que sejam tomadas as decisões cabíveis.

4.1.3 Nível E – Parcialmente Definido

Segundo Softex (2011), o nível de maturidade E é composto por: Avaliação e Melhoria do Processo Organizacional (AMP), Definição do Processo Organizacional (DFP), Gerência de Recursos Humanos (GRH) e Gerência de Reutilização (GRU).

O Processo AMP tem a finalidade de determinar se os processos aplicados estão contribuindo para atingir as metas e objetivos da organização e para planejar a implantação de melhorias contínuas nos processos conforme as análises realizadas.

O Processo DFP tem o propósito de definir um conjunto de processos e padrões de trabalho que serão aplicados de acordo com as necessidades de negócio (SOFTEX, 2011).

O Processo GRH, tem como objetivo, dispor os projetos com os recursos humanos necessários, mantendo a idoneidade adequada às necessidades da organização.

O processo GRU, é utilizado para gerenciar o ciclo de vida dos ativos reutilizáveis através de mecanismos de armazenamento e recuperação de ativos reutilizáveis que são periodicamente mantidos e tem suas modificações controladas ao longo de seu ciclo de vida.

4.1.4 Nível D - Largamente Definido

De acordo com Softex (2011), o Nível de Maturidade D é formado pelos processos dos níveis G e E, juntamente com processos de Desenvolvimento de Requisitos (DRE), Integração do Produto (ITP), Projeto e Construção do Produto (PCP), Validação (VAL), e Verificação (VER).

O Processo DRE tem o objetivo de definir os requisitos do sistema a ser desenvolvido, atendendo às necessidades de todos os envolvidos no projeto. Efetuando o levantamento dos requisitos do cliente que posteriormente serão transformados em requisitos funcionais e não-funcionais que devem ser analisados e gerenciados ao longo do desenvolvimento do produto.

O Processo ITP tem como objetivo proporcionar a integração e consistência do projeto com o que será produzido, através da criação de um ambiente operacional satisfatório, documentando os procedimentos de integração do produto assegurando que tudo ocorra da maneira correta até a entrega do produto.

O Processo PCP tem a finalidade de projetar a solução mais adequada com o objetivo de satisfazer aos requisitos do sistema, desenvolvê-la e posteriormente dar início à implementação.

O Processo VAL é utilizado para garantir que sejam avaliados todos os componentes do produto, verificando se todas as necessidades do usuário serão atendidas, garantindo que será desenvolvido o produto correto.

O Processo VER tem o objetivo de avaliar os serviços e o produto de trabalho com a finalidade de garantir que os requisitos sejam atendidos, identificando os itens a serem analisados de acordo com o planejamento efetuado ao longo do desenvolvimento do produto.

4.1.5 Nível C – Definido

Segundo Softex (2011), o nível de maturidade C é composto por processos dos níveis G ao D, juntamente com processos de Desenvolvimento para Reutilização (DRU), Gerência de Decisões (GDE), e Gerência de Riscos (GRI).

O Processo DRU identifica formas de Reutilização de Software com o objetivo de aplicar técnicas da engenharia de software para o desenvolvimento de sistemas que atendam a domínios de conhecimento específicos.

O Processo GDE é aplicado com o objetivo de fazer a análise de situações críticas que requerem a tomada de decisões tanto em questões organizacionais quanto de projeto. Para isso, utiliza um processo formal que analisa soluções alternativas com o objetivo de encontrar a solução adequada na resolução do problema.

O Processo GRI utiliza atividades de identificação de riscos organizacionais e de projeto com o objetivo de reduzir estes riscos, através de ações de contingência utilizadas para minimizar o impacto na organização e nos projetos.

4.1.6 Nível B – Gerenciado Quantitativamente

O nível de maturidade B possui uma evolução do processo Gerência de Projetos, juntamente com Atributos de Processo previsíveis, estáveis e com capacidade de processo (SOFTEX, 2009).

O nível de maturidade B é composto pelos níveis de maturidade do G ao C, evoluindo através do acréscimo de novos resultados de gerenciamento quantitativo (SOFTEX, 2011).

Neste nível os processos são gerenciados de forma quantitativa, através do acompanhamento do processo utilizando uma base de referência, com a qual é possível diagnosticar a instabilidade do processo de forma imediata.

4.1.7 Nível A – Em Otimização

O nível de maturidade A não possui processos específicos, sendo composto pelos níveis de maturidade do G ao B, onde a implementação dos processos de análise de desempenho devem satisfazer plenamente os atributos de pelo menos um dos processos selecionados na análise de desempenho (SOFTEX, 2011).

O nível A tem a proposta de otimizar os processos que apresentam variações e que são controlados constantemente, sendo que através da identificação destas variações podem ser tomadas medidas de correção e melhorias nos processos.

4.2 COMPARATIVO ENTRE MODELOS DE PROCESSO

O modelo MPS.BR foi construído para definir e aprimorar técnicas de melhoria e avaliação de processos de software nas organizações e foi desenvolvido com base técnica composta pelas normas NBR ISO/IEC 12207:2008 e NBR ISO/IEC 15504-2 e definido em conformidade com o modelo CMMI (SOFTEX, 2011).

De acordo com Guerra e Colombo (2009), o modelo Capability Maturity Model Integration (CMMI) define um modelo de maturidade para o desenvolvimento de software, utilizando seu método de avaliação Standard CMMI Appraisal Method for Process Improvement (SCAMPI), definido pela norma NBR ISO/IEC 12207.

O modelo MPS.BR também possui seu próprio método de avaliação com base na norma NBR ISO/IEC 15504-2:2003, o MA-MPS que posteriormente será descrito com mais detalhes.

A norma NBR ISO/IEC 12207 estabelece uma série de atividades de aquisição, desenvolvimento, operação e manutenção de software, orientando na adaptação de sua utilização nos projetos de desenvolvimento de software.

A Norma NBR ISO/IEC 15504 pode ser utilizada para o planejamento, gerenciamento, monitoramento, melhoria da aquisição, fornecimento, desenvolvimento, evolução e suporte de software nas organizações (GERRA; COLOMBO, 2009).

A comparação entre os modelos para processo de desenvolvimento de software é demonstrada abaixo, na tabela 1, onde os modelos citados são: NBR ISO/IEC 12207, NBR ISO/IEC 15504, CMMI e MPS.

Tabela 1- Comparação entre os modelos para processo de software

Aspectos abordados	NBR ISO/IEC 12207	CMMI	NBR ISO/IEC 15504	MPS
Objetivo	Estabelecer um entendimento comum para os processos entre todos os envolvidos com software	Determinar a capacitação da apoiar sua evolução de acordo com os níveis estabelecidos	Conhecer e avaliar os processos, determinar a capacitação e promover a melhoria	Melhorar os processos de software nas micro, pequenas e médias empresas, a um custo acessível, em diversos locais do país
Abordagem	Definição dos processos para aquisição, fornecimento, desenvolvimento, operação e manutenção de software	Avaliação dos processos e enquadramento da organização em um dos níveis de maturidade	Avaliação dos processos da organização em relação a níveis de capacitação	Avaliação dos processos em relação aos níveis de maturidade que são uma combinação entre processos e sua capacidade
Organizações-alvo	Organizações em geral	Organizações que necessitam de uma comprovação formal de sua capacidade	Organizações em geral	Organizações em geral
Definição de processos	Estabelece 43 processos, organizados em 7 áreas de processo	Estabelece 22 áreas de processos, organizados em 5 níveis de maturidade	Estabelece 35 processos, organizados em 5 categorias	Estabelece 22 processos, organizados em 7 níveis crescentes de maturidade
Flexibilidade nos aspectos definidos pelo modelo	Classificação de processos pode ser utilizada conforme os objetivos da organização	Níveis e áreas-chave de processo são à base do modelo e não podem ser alterados	Define perfis de processo e práticas de acordo com os objetivos da organização	Níveis com menor conteúdo de implementação para motivar a adesão ao modelo para as empresas que desejam melhorar seus processos

Fonte: Adaptado de Guerra e Colombo (2009, p.80).

Analisando a tabela 1, podemos perceber que os modelos citados, são divididos em processos que são subdivididos em áreas, organizadas de formas diferenciadas.

Comparando os modelos em questão, é possível perceber que todos buscam a disseminação do conhecimento em relação aos processos de desenvolvimento aplicados nas organizações, preocupando-se com a evolução das técnicas aplicadas e com a melhoria contínua dos processos que são definidos e adaptados de acordo com os objetivos da organização, com exceção do modelo CMMI que não permite que sejam feitas alterações por considerar que as divisões e subdivisões do processo são a base do modelo e devem ser preservados.

4.3 COMPARATIVO ENTRE OS MODELOS DE PROCESSO CMMI E MPS.BR

Segundo Pressman (2006), o modelo CMMI representa um metamodelo de processo contínuo e em estágios, ambos definidos em cinco PAs, que são avaliadas de acordo com metas e práticas estabelecidas.

As PAs citadas anteriormente são:

- a) PP: planejamento de projeto;
- b) GREQ: gestão de requisitos;
- c) MA: medição de análise;
- d) GC: gestão de configuração;
- e) GQPP: gesto de qualidade de processo e de produto.

De acordo com Pressman (2006), o modelo contínuo utiliza níveis de capacitação para avaliar as PAs, subdividindo-os em níveis de 0 a 5 que são:

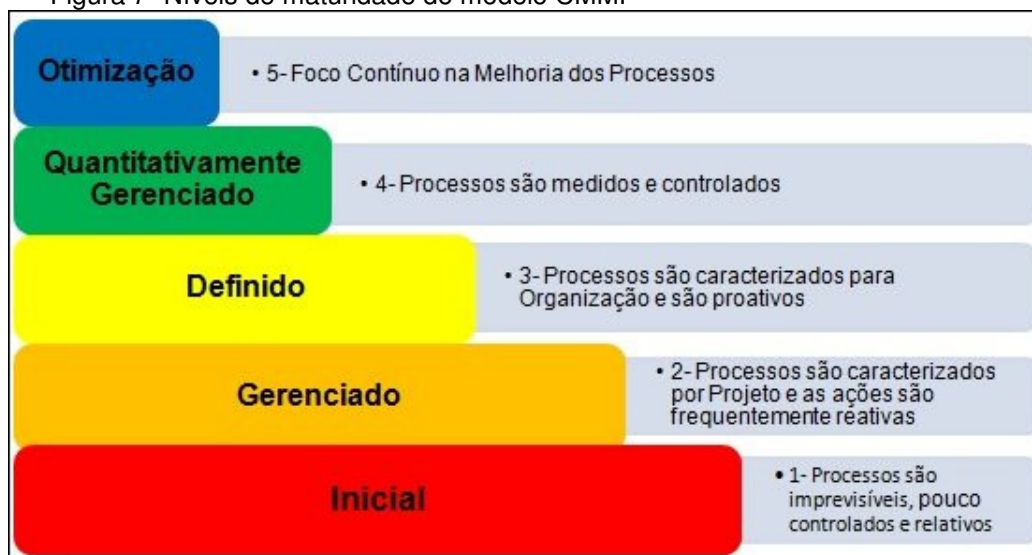
- a) Nível 0 (Incompleto): não atinge todas as metas e objetivos definidos para o nível 1;
- b) Nível 1 (Realizado): todas as metas específicas da PA foram satisfeitas;
- c) Nível 2 (Gerido): todos os critérios do nível 1 foram satisfeitos atendendo a política definida pela organização, onde todas as atividades estão sendo monitoradas e avaliadas;
- d) Nível 3 (Definido): todos os critérios do nível 2 foram satisfeitos, onde o processo é feito de acordo com os padrões da

organização com o objetivo de alcançar o aperfeiçoamento contínuo;

- e) Nível 4 (Quantitativamente gerido): todos os critérios do nível 3 foram satisfeitos, utilizando medições e avaliação quantitativa para o controle das PAs;
- f) Nível 5 (Otimizado): todos os critérios do nível 4 foram satisfeitos e são utilizados meios quantitativos para otimizar a PA, atendendo às necessidades do cliente e tornando o processo mais eficiente.

O modelo em estágios é aplicado com o objetivo de proporcionar a melhoria de processos atingindo cinco níveis de maturidade conforme mostra a figura 7 (GEVAERD, 2009).

Figura 7- Níveis de maturidade do modelo CMMI



Fonte: Adaptado de Gevaerd (2009, p.17).

De acordo com a Figura 7, no nível inicial os processos são imprevisíveis, e pouco controlados, o que tende ao excesso no comprometimento quanto ao que será desenvolvido.

No nível gerenciado os processos são planejados, executados e controlados com o objetivo de assegurar que os projetos estão sendo executados de acordo com o que foi planejado e documentado (GEVAERD, 2009).

No nível definido é feita a caracterização dos processos, descrevendo procedimentos, métodos e ferramentas de acordo com o padrão da organização com o objetivo de melhorar continuamente os processos aplicados (COUTO, 2007).

No nível quantitativamente gerenciado os processos são medidos e controlados com base em objetivos quantitativos para garantir a qualidade e o bom desempenho dos processos, levando em consideração as necessidades do cliente e dos demais envolvidos no processo (COUTO, 2007).

No nível otimização a melhoria contínua de desempenho dos processos é o foco principal, onde os processos são continuamente revisados com base em critérios quantitativos com o objetivo de gerenciar a melhoria dos processos (GEVAERD, 2009).

Considerando todos os aspectos citados anteriormente, podemos concluir que o CMMI é um modelo no qual sua implantação deve ser muito bem planejada, levando em consideração que é um processo a longo prazo.

O grande desafio na implantação do modelo CMMI é o dia-a-dia da empresa que acaba absorvendo todos os recursos disponíveis, o que muitas vezes acaba prejudicando a aplicação do modelo (ROCHA; MALDONADO; WEBER, 2001).

Para que a implantação seja bem sucedida, devem ser planejadas questões que envolvem as pessoas envolvidas no projeto, suas habilidades e seu comprometimento com as mudanças, sendo esclarecidas todas as dúvidas para que não haja acomodação durante a execução do processo, o que pode causar na desistência da aplicação do modelo.

O CMMI foi desenvolvido inicialmente para aplicação em organizações de grande porte, que desenvolvem projetos complexos e de longa duração, obedecendo a modelos-padrão, exigindo que sejam controladas e monitoradas precisamente todas as metas e práticas do projeto em busca da melhoria contínua.

Pequenas organizações também podem utilizar os princípios do modelo CMMI, desde que sejam interpretados cuidadosamente, para adaptá-los da melhor maneira possível à realidade e às necessidades da empresa.

Mas deve-se analisar a viabilidade do processo, sendo que para a sua implantação é necessário um investimento de alto custo.

O Modelo MPS.BR define sete níveis de maturidade de processo de desenvolvimento de software, que possuem um conjunto de processos e atributos de processo que identificam onde a organização deve concentrar esforços para a melhoria da qualidade no desenvolvimento (GEVAERD, 2009).

Os sete níveis de maturidade do modelo MPS.BR abordados no capítulo 4 desta pesquisa, são demonstrados também na figura 8 juntamente com os atributos de processo referente a cada nível.

“A adoção de um nível de maturidade significa a obtenção de um grau de melhoria nas competências para a realização de um conjunto de tarefas ou atividades onde todos os objetivos são atendidos.” (GEVAERD, 2009, p.27).

O modelo MPS.BR possui um grande potencial em relação a sua aplicabilidade em diferentes regiões, considerando as características das empresas brasileiras através da aplicação de técnicas, padrões e modelos de processo já conhecidos, mas que vêm sendo aperfeiçoados de acordo com as necessidades encontradas nas organizações.

Tabela 2- Níveis de maturidade do modelo MPS.BR

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI Desenvolvimento para Reutilização – DRU Gerência de Decisões – GDE	AP 1.1, AP 2.1, AP 2.2, AP 3.1, e AP 3.2
D	Verificação – VER Validação – VAL Projeto e Construção do Produto – PCP Integração do Produto – ITP Desenvolvimento de Requisitos - DRE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
E	Gerência de Projetos – GPR (evolução) Gerência de Reutilização – GRU Gerência de Recursos Humanos – GRH Definição do Processo Organizacional – DFP Avaliação e Melhoria do Processo Organizacional - AMP	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2

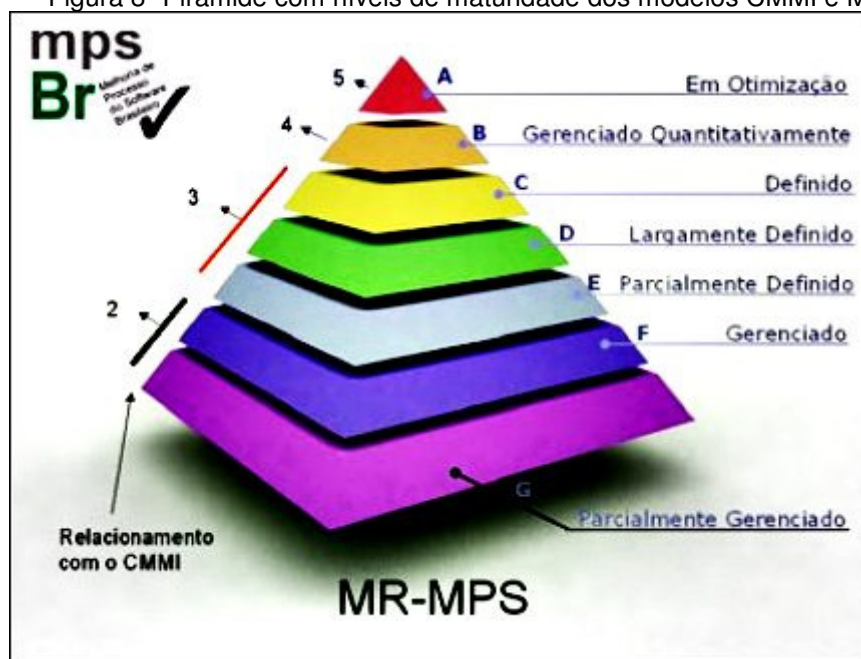
F	Medição – MED Garantia da Qualidade – GQA Gerência de Portfólio de Projetos – GPP Gerência de Configuração – GQO Aquisição – AQU	AP 1.1, AP 2.1 e AP 2.2
G	Gerência de Requisitos – GRE Gerência de Projetos – GPR	AP 1.1 e AP 2.1

Fonte: Softex (2011, p.22).

De acordo com Rocha, Montoni, Santos e Mafra (2005), processos foram atribuídos a cada um dos níveis de maturidade demonstrados na tabela 2, conforme a norma ISO/IEC 12207.

Com o objetivo de comparar os modelos CMMI e MPR.BR, abaixo temos a figura 8 que demonstra a divisão dos níveis de maturidade existentes nos dois modelos. Desta forma podemos observar que o modelo MPS.BR, apesar de ser baseado nos princípios do CMMI, possui uma divisão diferente em relação aos estágios do CMMI, sendo subdividido em mais níveis.

Figura 8- Pirâmide com níveis de maturidade dos modelos CMMI e MPS.BR



Fonte: Softex (2009).

Analisando a figura 8 podemos perceber que o nível 2 do modelo CMMI é representado pelos modelos F e G do MPS.BR. O nível 3, é representado pelos modelos C, D e E do modelo MPS.BR.

Esta subdivisão permite que seja feita a implantação do modelo em empresas de pequeno e médio porte de forma gradual, diminuindo os esforços empregados para atingir cada nível e conseqüentemente reduzindo o custo.

A visibilidade dos resultados obtidos também pode ser vista com maior clareza em relação ao CMMI, sendo que os resultados podem ser obtidos em um curto espaço de tempo, o que não acontece no modelo CMMI.

Tanto o modelo CMMI quanto o modelo MPS.BR possuem seus próprios métodos de avaliação, sendo eles o SCAMPI e o MA-MPS, que será descrito no capítulo 7 de forma detalhada.

4.4 ÁREA DE PROCESSO GARANTIA DA QUALIDADE (GQA)

Nos parágrafos a seguir serão referenciados conceitos estabelecidos pela Softex (2011), desenvolvedora e coordenadora do modelo de processo MPS.BR.

A PA Garantia da Qualidade é responsável por manter a conformidade entre os processos executados e os produtos de trabalho predefinidos.

Para assegurar que os processos estejam em conformidade são avaliados os padrões e procedimentos com o objetivo de identificar problemas e não conformidades, registrando-os para que sejam aplicadas ações corretivas que são acompanhadas até que sejam concluídas.

A GQA serve de apoio ao gerente, ajudando a mantê-lo informado sobre a realização das atividades do projeto, fiscalizando de forma eficiente todas as tarefas executadas desde o início do projeto até a sua conclusão, auxiliando inclusive, na criação e revisão de procedimentos e padrões de desenvolvimento.

Todo o trabalho de fiscalização do cumprimento das atividades deve ser executado com autoridade, comprometimento e responsabilidade por parte do grupo que executa esta tarefa, sendo que qualquer falha pode ocasionar em sérias conseqüências negativas.

Esta PA tem como objetivos principais:

- a) Avaliar a execução de produtos de trabalho e processos e serviços de acordo com procedimentos e padrões predefinidos;
- b) Identificar e documentar as não conformidades encontradas;
- c) Proporcionar à equipe de trabalho e à gerência o feedback com o resultado das atividades de garantia da qualidade, assegurando que as não conformidades sejam corrigidas.

As atividades do processo devem ser avaliadas de forma objetiva através de um profissional externo, que não esteja envolvido no desenvolvimento da documentação, nas atividades que serão auditadas e em nenhuma outras atividade relacionada ao projeto em questão, permitindo que a avaliação seja feita de forma séria e eficiente.

A utilização de checklist pode ser uma forma de fiscalizar as atividades com eficiência, sendo que serão especificadas todas as atividades que deverão ser realizadas pela equipe de desenvolvimento, controlando a execução de todas elas, através da apresentação de evidências que comprovem a sua execução. Caso haja alguma não conformidade, tudo será registrado, exigindo que sejam apresentadas as razões para o descumprimento da tarefa, estabelecendo um prazo para a execução de sua correção.

No prazo estabelecido para a correção das não conformidades encontradas, deve ser efetuada uma nova auditoria com o objetivo de verificar se foram feitas as correções necessárias. Caso a não conformidade persista, deve ser estabelecido um novo prazo para a correção, informando ao superior da equipe para que tome conhecimento do assunto e exija o cumprimento desta atividade.

Os resultados esperados com a aplicação da GQA serão apresentados nos itens GQA1, GQA2, GQA3 e GQA4 das seções a seguir.

4.4.1 GQA1

Nos parágrafos a seguir, serão citados conceitos estabelecidos pela Softex (2011).

Para garantir a qualidade nos projetos de software todos os produtos de trabalho devem ser submetidos a uma avaliação rigorosa, sendo previamente selecionados grupos de processos de maior importância ou de maior valor agregado ao projeto para que sejam avaliados no caso de existir no projeto uma quantidade muito grande de processos e produtos de trabalho que devem levar em consideração todos os processos utilizados na organização, focando no todo e não somente nos processos definidos no MR-MPS.

A GQA1 deve ser aplicada de acordo com o planejamento efetuado no início do projeto, através da documentação das atividades de garantia da qualidade que deverão ser executadas ao longo do ciclo de vida do projeto, avaliando as atividades de forma objetiva, utilizando ferramentas de verificação, checklists, e ferramentas automatizadas, identificando as possíveis falhas para que sejam devidamente corrigidas no mínimo espaço de tempo.

Para verificar a conformidade dos produtos de trabalho com os padrões de processo, é aconselhável que sejam feitas auditorias antes que sejam repassados a clientes internos e externos, ou quando o produto de trabalho irá servir de referência para alguma outra atividade ligada ao projeto.

Para avaliar os produtos de trabalho é verificado se foram atendidos todos os padrões predefinidos, a consistência interna e externa, a clareza e a completeza em relação ao template², levando em consideração o conteúdo do documento, incluindo critérios rígidos para aumentar a eficácia da avaliação.

4.4.2 GQA2

Nos parágrafos a seguir, serão citados conceitos estabelecidos pela Softex (2011).

Na GQA2 são selecionados os processos e atividades de apoio ao processo que serão avaliados na auditoria de garantia da qualidade com o objetivo de verificar se a execução está em conformidade com o que foi pré-estabelecido.

² Ambiente estabelecido como modelo, permitindo criar conteúdos de uma forma rápida. (INFOPÉDIA, 2012).

Esta avaliação é feita através da verificação da descrição do processo ou da realização de entrevistas com os responsáveis pela execução das atividades que serão auditadas, sendo necessário que sejam auditados todos os processos do projeto que foram implementados a partir do MR-MPS.

4.4.3 GQA3

Nos parágrafos a seguir, serão citados conceitos estabelecidos pela Softex (2011).

Na GQA3 todas as não conformidades são identificadas e registradas através de documentos, onde os desvios entre o esperado e o que foi efetivamente realizado, são informados às partes interessadas para que sejam tomadas as medidas necessárias para que sejam efetuadas as correções necessárias.

Deve ser efetuado um registro onde são armazenadas todas as não conformidades encontradas, possibilitando que sejam analisadas as circunstâncias nas quais elas ocorreram, podendo identificar métodos de prevenção das falhas para que sejam realizadas alterações nos procedimentos, quando necessário, para a melhoria da qualidade, evitando ao máximo que ocorram novamente estas não conformidades.

4.4.4 GQA4

Nos parágrafos a seguir, serão citados conceitos estabelecidos pela Softex (2011).

Através da identificação e documentação das não conformidades encontradas no projeto, é realizada uma análise com a finalidade de encontrar a solução mais apropriada para a resolução das falhas.

A resolução das não conformidades pode ser realizada exigindo que o processo executado satisfaça aos padrões e procedimentos, mas permite também que sejam feitas alterações nos procedimentos com o objetivo de torná-lo utilizável.

Os aspectos citados acima podem ser desconsiderados se necessário, caso seja tomada a decisão de não atender aos critérios estabelecidos no processo desde que sejam arcadas as consequências.

É necessário que seja estabelecida uma estratégia de acompanhamento das não conformidades, estabelecendo as causas que levaram a esta não conformidade, o responsável por resolvê-la, as medidas a serem tomadas, e o prazo estimado para a resolução do problema, para que seja monitorado frequentemente se foram tomadas as providências especificadas.

Caso expire o prazo para a resolução das não conformidades, devem ser tomadas algumas medidas corretivas, informando aos respectivos níveis hierárquicos, estabelecendo um novo prazo para a conclusão das ações corretivas.

Caso os níveis hierárquicos não se importem com as medidas de correção, será registrado o fechamento desta não conformidade, o que caracteriza a falta de comprometimento da organização em relação às medidas de garantia da qualidade nos projetos desenvolvidos.

Com base nos resultados apresentados nas áreas de processo mencionadas acima, o presente trabalho tem como objetivo a utilização de alguns destes conceitos para a modelagem de um metamodelo utilizando o framework de processos EPF Composer.

5 FRAMEWORK EPF COMPOSER

O Eclipse Process Framework Composer (EPF Composer), é uma ferramenta de criação de processos de desenvolvimento de software que permite a criação, configuração e publicação de métodos (HAUMER, 2007, tradução nossa).

Segundo Oliveira (2009), o EPF Composer é um projeto Open Source da Eclipse Foundation, apoiado pela Internacional Business Machines (IBM) e é uma evolução do Software Process Engineering Metamodel (SPEM).

O EPF Composer permite que seja estruturada a promoção do reuso de frameworks de processos em pequenas unidades modulares (KULESZA, 2011).

Cada unidade modular utiliza guias específicos e artefatos de entrada e saída para a realização de tarefas que são definidas pelas unidades modulares.

Através da utilização de esquemas pré-definidos é possível criar processos de desenvolvimento de software e adaptá-los de acordo com as necessidades da organização desenvolvedora de software, proporcionando um amplo conhecimento ao profissional que o utiliza.

A administração de bibliotecas de métodos reutilizáveis é usada para montar processos com métodos reutilizáveis que podem ser reconfigurados para suprir as necessidades do projeto, podendo ser publicados pela promulgação de planos de projeto ou processo documentação (Haumer, 2007: tradução nossa).

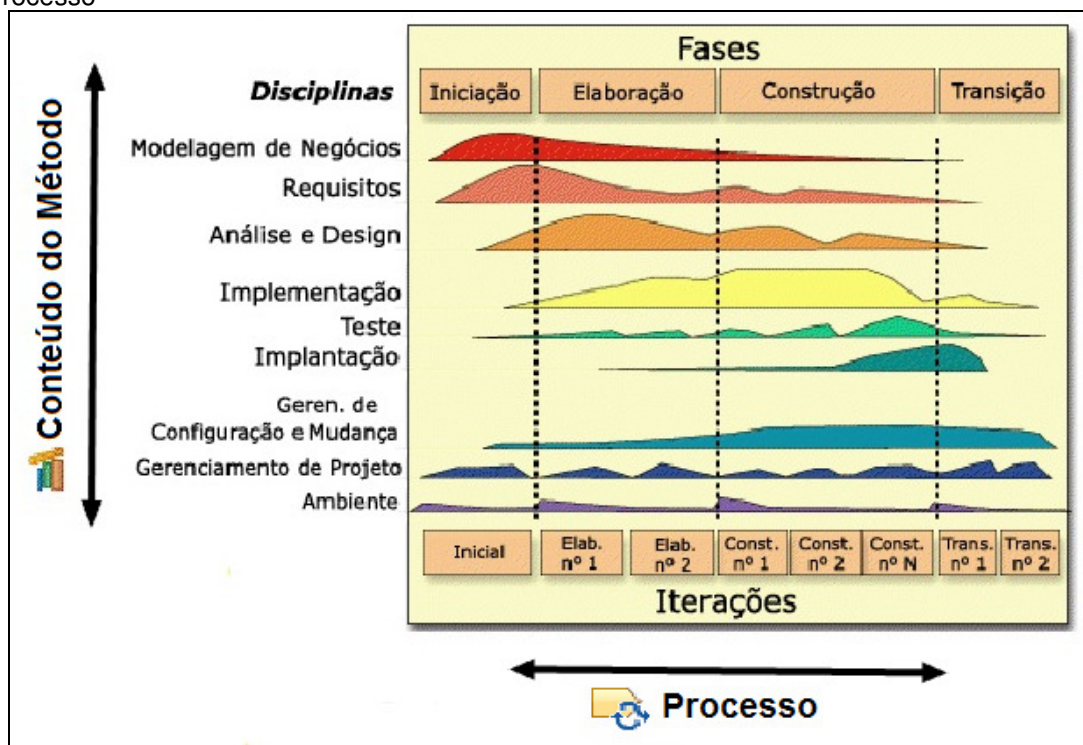
Além de ser uma ferramenta de uso gratuito, o EPF Composer permite que sejam disponibilizados facilmente todos os processos desenvolvidos na linguagem HTML.

Esta ferramenta oferece soluções aos problemas que geralmente são encontrados pela equipe de desenvolvimento na aquisição e gestão dos métodos de processo, mantendo a centralização das informações e fornecendo a documentação necessária enquanto o trabalho é realizado.

O EPF Composer tem como base para o seu desenvolvimento o SPEM, que é um metamodelo utilizado para a modelagem de processos de software, através da definição de estereótipos da Linguagem de Modelagem Unificada (UML) e provê um metamodelo baseado em Meta-Object Facility (MOF) que é a tecnologia adotada pela Object Management Group (OMG) para a definição de metadados através de um conjunto de elementos de modelagem que permitem a aplicação de regras para a construção de metamodelos.

O método utilizado pelo SPEM é composto por disciplinas que são compostas por definições de tarefas que possibilitam a classificação e descrição de como o trabalho de desenvolvimento está sendo realizado. Como por exemplo, o ciclo de vida iterativo incremental.

Figura 9- Definição do conteúdo do método versus a aplicação do conteúdo em um Processo



Fonte: Adaptado de OMG (2008).

Conforme demonstrado na Figura 9, o processo é dividido em fases que são compostas por iterações, onde são instanciados os recursos para a realização do trabalho através da atribuição de produtos reais de trabalho,










como as entradas e saídas de tarefas. O gráfico demonstra o esforço de trabalho empregado em cada disciplina, ao longo de cada fase do processo.

5.1 FORMATO DO MODELO

O SPEM é um metamodelo que permite descrever de forma concreta os processos de desenvolvimento de software, através de uma proposta de unificação das metodologias para modelagem de processos existentes.

Para a representação das fases e atividades do processo são utilizados os estereótipos estabelecidos na UML, conforme demonstrado na Figura 10.

Figura 10- Estereótipos da Linguagem

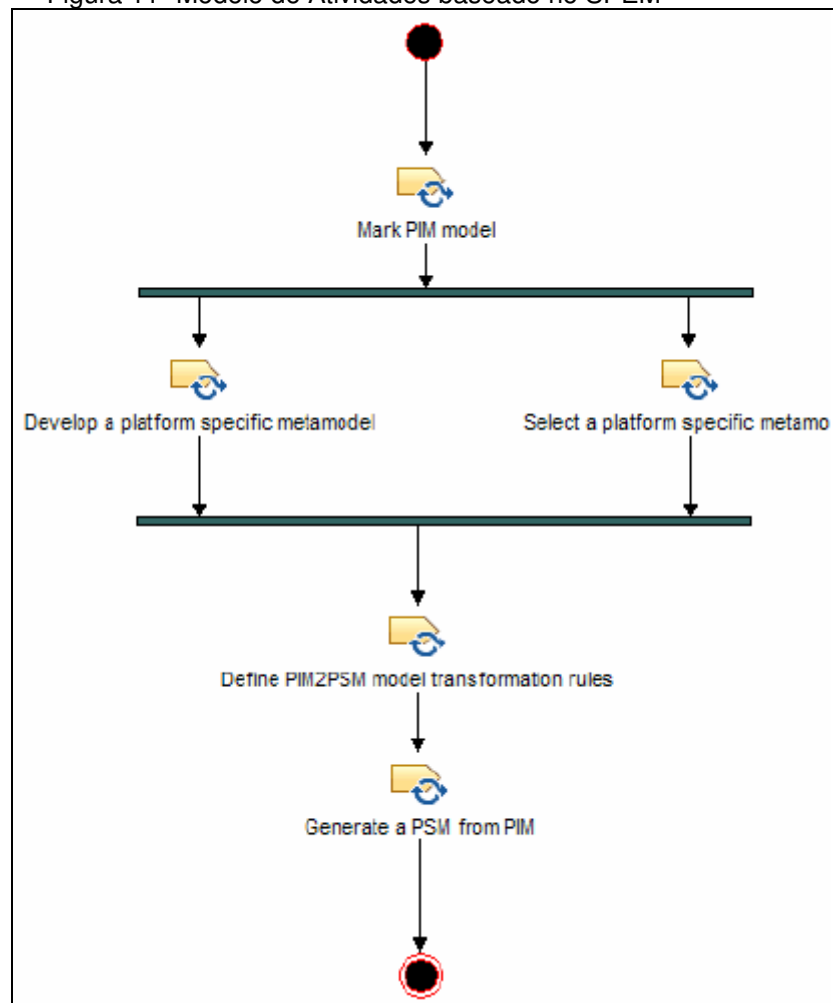
Estereótipo	Notação	Descrição
Processo		Uma instância do Processo Padrão para um projeto específico.
Disciplina		Um conjunto de definições de tarefa que contribuem para o atendimento de um mesmo objetivo do processo.
Fase		Um período significativo de um Processo que possui um objetivo específico.
Definição de Produto de Trabalho		Um conjunto de características desejadas para um produto de trabalho que pode ser reutilizado em diferentes processos.
Atividade		Um trabalho que deve ser realizado no decorrer de um Processo e que pode ser decomposto em outras atividades ou tarefas.
Definição de Tarefa		Um trabalho reutilizável e que não pode ser decomposto.
Definição de Função		Um conjunto de capacidades desejadas para um determinado papel que pode ser reutilizado em diferentes processos.
Orientação		Uma conduta que deve ser adotada durante a realização de uma tarefa do Processo.
Passo		É utilizado para organizar a realização de uma atividade, através de etapas ou subunidades de trabalho.

Fonte: Adaptado de OMG (2008).

Com a utilização dos estereótipos apresentados acima, é possível criar metamodelos de processo, onde são detalhadas as etapas do desenvolvimento com a representação das atividades a serem realizadas, demonstrando os produtos de trabalhos, definindo funções e os passos para a realização das tarefas, permitindo também que sejam estabelecidas as fases do processo através da definição de objetivos específicos.

Na Figura 11, é demonstrado um exemplo de modelo de atividades baseado no SPEM, com a aplicação dos estereótipos para a definição dos passos necessários para o desenvolvimento das atividades do processo.

Figura 11- Modelo de Atividades baseado no SPEM



Fonte: OMG (2008).

Com base no que foi apresentado anteriormente, o presente trabalho visa demonstrar a utilização de todos os conceitos de processo de software para o desenvolvimento de um Metamodelo, onde serão aplicadas as técnicas utilizadas no modelo MPS.BR para proporcionar qualidade durante o desenvolvimento de projetos de software.

5.2 METAMODELO

Metamodelos são conjuntos de regras e construtores utilizados na modelagem de processos (OLIVEIRA, 2009).

Através dos metamodelos são aplicados conceitos de modelos específicos, com o objetivo de executá-los através da definição de atividades, prazos e regras determinando de que forma será executado o processo de desenvolvimento.

Os metamodelos de metodologias ágeis estão despertando o interesse das organizações desenvolvedoras de software que buscam cada vez mais a qualidade e agilidade em seus processos (MALLMANN, 2011).

Dentre os metamodelos de metodologias ágeis os que mais se completam são o metamodelo XP, o metamodelo Scrum e o metamodelo Feature Driven Development (FDD).

O metamodelo XP suporta múltiplos projetos, armazena o histórico das decisões, e é desenvolvido de forma iterativa, o que torna o processo mais ágil (MALLMANN, 2011).

Através do metamodelo Scrum é possível planejar a entrega do produto e monitorá-la constantemente, permitindo que seja verificado se todas as regras do modelo estão sendo praticadas.

O metamodelo da metodologia FDD utiliza pequenas funcionalidades que devem ser implementadas somente em uma iteração, na qual são definidas a data de criação e a data prevista para a conclusão das atividades (MALLMANN, 2011).

Na aplicação do metamodelo FDD existe a preocupação em planejar todas as atividades a serem executadas, mas não é feita a documentação de controle dos processos.

Com o objetivo de aplicar os conceitos de metamodelo apresentados neste projeto de pesquisa, propõe-se o desenvolvimento de um metamodelo de processo, instanciando elementos que são utilizados para representar cada passo do processo.

Cada vez mais a aplicação de metamodelos é utilizada com o objetivo de agregar eficiência na definição dos processos aplicados no meio organizacional, buscando satisfazer as normas e os padrões de qualidade durante a execução das atividades, tornando o processo mais confiável.

Com a utilização dos elementos de processo são estabelecidas as regras e a sequência na qual o processo será formado (FONTOURA; HARTMANN; PRICE, 2006).

O SPEM é um metamodelo que serve como base para a especificação de qualquer processo, através da ideia de que o processo é uma colaboração entre papéis de processo utilizados para a realização de atividades em entidades chamadas de produtos de trabalho, com a finalidade de atingir um objetivo específico.

Um metamodelo de processo é utilizado para a representação dos objetos e atividades envolvidas no processo de software, onde cada fase do processo é demonstrada de forma clara e simples permitindo que seja realizado o gerenciamento de todo o processo.

O metamodelo para Fluxo de Trabalho é utilizado para identificar os atores e as atividades realizadas dentro do processo, representando-os através de fluxos de trabalho, tarefas, objetos manipulados (dados, documentos, etc.), papéis e atores.

Os metamodelos permitem que sejam apresentadas as atividades das organizações, através da representação das entradas e saídas existentes no processo, permitindo que sejam identificados os objetivos e as necessidades da organização na qual será aplicado.

A partir dos conceitos de metamodelo que foram apresentados, conclui-se que utilizando metamodelos é possível aplicar regras de desenvolvimento ágil adaptando-as de acordo com a necessidade, permitindo que através do método de avaliação utilizado, seja monitorado todo o andamento do processo.

A seguir, serão abordados os conceitos e a metodologia do MA-MPS que é o método que será utilizado para a avaliação do metamodelo proposto.

5.3 MÉTODO DE AVALIAÇÃO (MA-MPS)

Nos parágrafos a seguir, serão citados conceitos estabelecidos pela Softex (2012).

Processo e o Método de Avaliação (MA-MPS) foram definidos para permitir a avaliação objetiva dos processos de software nas organizações, atribuindo por meio dos resultados desta avaliação, um nível de maturidade do MR-MPS, podendo ser aplicado a qualquer indústria de software e a organizações de qualquer tamanho.

O MA-MPS foi desenvolvido com base na Norma ISO/IEC 15504 tendo como objetivo, verificar a maturidade das organizações na execução de seus processos de software. Para isso, são descritas uma série de tarefas que devem ser realizadas, tendo como ponto de partida a seleção de uma Instituição Avaliadora (IA), e posteriormente serão registrados na base de dados da Softex os resultados da avaliação.

O processo de avaliação tem validade de três anos a partir da data da última avaliação efetuada na organização e é composto por quatro subprocessos, onde é contratada a avaliação, a sua realização é preparada e aplicada finalizando com a documentação dos resultados obtidos.

Na Tabela 3 podemos observar detalhadamente os quatro subprocessos citados anteriormente.

Tabela 3- Processo de Avaliação

PROCESSO DE AVALIAÇÃO	
SUBPROCESSO	ATIVIDADE
Contratar a avaliação	Pesquisar Instituições Avaliadoras Estabelecer a avaliação
Preparar a realização da avaliação	Viabilizar a avaliação Planejar a avaliação Preparar a avaliação Conduzir a avaliação inicial Completar a preparação da avaliação
Realizar a avaliação final	Conduzir a avaliação final Avaliar a execução do processo de avaliação
Documentar os resultados da avaliação	Relatar resultados Registrar resultados

Fonte: Softex (2012, p.11).

Através da aplicação deste processo são obtidas as informações que identificam quais os processos de software da organização, determinando

o grau de execução dos resultados esperados e a partir disso, é determinado o nível de maturidade da organização.

O coordenador da IA comunica à Softex sobre a contratação da avaliação, definindo a equipe de avaliação e o coordenador local que acompanhará todo o processo juntamente com o Auditor da avaliação.

Para que seja aplicado o processo de avaliação nas organizações, é efetuado um planejamento de todos os passos a serem seguidos durante a avaliação. É preenchida a planilha com os indicadores, onde é demonstrado que a implementação dos processos está sendo executada.

A demonstração dos indicadores de implementação podem ser demonstradas através do produto principal da realização de uma tarefa, da apresentação de documentos e de entrevistas que confirmem a implementação do processo.

De acordo com as evidências apresentadas, é caracterizado o grau de implantação dos atributos do processo, onde é identificado se o processo foi totalmente implementado, largamente implementado, parcialmente implementado, ou não foi implementado. O processo é satisfeito somente quando todos os resultados esperados foram totalmente implementados ou largamente implementados, sendo não satisfeito quando estiver em qualquer outra situação.

A partir disso, são apontados os pontos fortes e fracos e as oportunidades de melhoria para o processo, sendo atribuído à organização o nível de maturidade de acordo com a avaliação efetuada, informando à Softex todos os resultados obtidos na avaliação.

5.3.1 Metodologia do MA-MPS

Para a realização da avaliação dos processos aplicados nas organizações, o MA-MPS utiliza de uma série de atividades e tarefas que ao serem executadas permitem que seja identificado se os processos estão sendo aplicados de acordo com os padrões de qualidade estabelecidos.

Para que a avaliação seja realizada de forma eficiente deve ser elaborado o plano de avaliação. Para preparação da avaliação é efetuado o preenchimento de uma Planilha de Indicadores, onde são relatadas as evidências que vão demonstrar se a implementação dos processos está em conformidade com os procedimentos e se foram gerados os resultados esperados.

Os indicadores são considerados *diretos*, quando representam o produto da realização de uma determinada tarefa. Os indicadores *indiretos* são artefatos gerados durante a execução dos projetos, mas não é o objetivo da realização da tarefa, já as afirmações confirmam a implementação do processo e são obtidas através de entrevistas ou apresentações.

Após a avaliação é iniciado o período de ajustes, onde é documentada toda a avaliação realizada até o momento e posteriormente são realizadas auditorias para garantir que os ajustes recomendados estão sendo executados.

Para a execução da avaliação final são executadas entrevistas com os colaboradores com o objetivo de questionar a implementação dos processos e atributos. Deve haver pelo menos uma afirmação para cada resultado esperado, e um projeto concluído deve ter afirmações para 50% dos resultados, aproximadamente.

Todas as informações coletadas durante a entrevista devem ser registradas na planilha de indicadores e o grau de implementação dos resultados esperados deve ser caracterizado conforme a Tabela 4.

Tabela 4- Escala de caracterização do grau de implementação de resultados esperados

GRAU DE IMPLEMENTAÇÃO	CARACTERIZAÇÃO
Totalmente implementado (T)	<ul style="list-style-type: none"> - O indicador direto está presente e é julgado adequado - Existe pelo menos um indicador indireto e/ou afirmação confirmando a implementação - Não foi notado nenhum ponto fraco substancial
Largamente implementado (L)	<ul style="list-style-type: none"> - O indicador direto está presente e é julgado adequado - Existe pelo menos um indicador indireto e/ ou afirmação confirmando a implementação - Foi notado um ou mais pontos fracos substanciais

Parcialmente implementado (P)	- O indicador direto não está presente ou é julgado inadequado - Artefatos/afirmações sugerem que alguns aspectos do resultado esperado estão implementados - Pontos fracos foram documentados
Não implementado (N)	- Qualquer situação diferente das acima
Não avaliado (NA)	- O projeto não está na fase de desenvolvimento que permite atender ao resultado ou não faz parte do escopo do projeto atender ao resultado.
Fora do escopo (F)	- O resultado esperado está fora do escopo da avaliação, conforme documentado no plano da avaliação.

Fonte: Softex (2012, p.60).

A caracterização do grau de implementação dos resultados esperados é realizada com a finalidade de identificar os pontos fracos e fortes encontrados no processo durante a avaliação, possibilitando que sejam apontadas oportunidades de melhoria de acordo com a importância dos projetos.

Após a caracterização do grau de implementação dos resultados esperados, e dos atributos de processo, são apresentados os resultados obtidos na avaliação como forma de fornecer *feedback* em relação ao método de avaliação aplicado para posteriormente, serem documentados os resultados da avaliação realizada.

6 TRABALHOS CORRELATOS

Durante o desenvolvimento desta pesquisa, foram analisados diversos trabalhos com abordagem no desenvolvimento e aplicação de processos, semelhantes ao trabalho apresentado, mas com foco em outras áreas específicas.

A seguir serão descritos alguns destes trabalhos.

6.1 MPS.BR - UM ESTUDO DO MODELO MPS.BR COMO BENEFÍCIO PARA PEQUENAS E MÉDIAS EMPRESAS

Trabalho de Conclusão de Curso de Wilker Felix Ferreira para obtenção do grau de Bacharel em Sistemas de Informação, em 2009, pelo

Departamento de Sistemas de Informação, da Universidade Estadual de Goiás
- Unidade Universitária de Ceres.

O trabalho apresenta um estudo sobre o modelo de melhoria de processo de software brasileiro MPS.BR e a implantação do nível G em uma empresa de pequeno porte, com o objetivo de promover a conscientização de que processos bem planejados e definidos são fundamentais para o sucesso no desenvolvimento de projetos.

Como resultado obtido da aplicação do MPS.BR em uma organização, foi realizada a comparação dos resultados alcançados comparando-os com o modelo do Guia de Implementação.

6.2 DESENVOLVIMENTO DE UMA METODOLOGIA PARA O GERENCIAMENTO DE PROJETOS BASEADO NA ÁREA DE PROCESSO: GERÊNCIA DE PROJETOS (GPR) DO MODELO DE MATURIDADE MPS.BR

Trabalho de Conclusão de Curso de Maridiane Corrêa Plácido para a obtenção do grau de Bacharel em Ciência da Computação, em 2010, pelo Departamento de Ciência da Computação, da Universidade do Extremo Sul Catarinense.

O trabalho tem como objetivo, desenvolver uma metodologia para o gerenciamento de projetos baseada na área de processo Gerência de Projetos (GPR) do modelo de maturidade MPS.BR.

Como resultado, foi desenvolvida uma modelagem de processo baseado nos resultados estabelecidos no nível G de maturidade do modelo MPS.BR, onde foram descritas as fases do processo.

6.3 PROPOSTA DE UM PROCESSO DE GERENCIAMENTO DE REQUISITOS PARA UMA EMPRESA DESENVOLVEDORA DE SOFTWARE COM FOCO NA SATISFAÇÃO DO NÍVEL G DO MPS.BR

Trabalho de Conclusão de Curso de Marcelo Rodrigues Gomes para a obtenção do grau de Bacharel em Ciência da Computação, em 2010, pelo Departamento de Ciência da Computação, do Centro de Ciências Exatas e

Tecnológicas da Universidade Estadual do Oeste do Paraná – Campus de Cascavel.

O trabalho tem como objetivo, propor uma abordagem para a definição do processo de gerenciamento de requisitos em uma empresa desenvolvedora de software de pequeno porte, para auxiliá-la na melhoria da qualidade dos serviços prestados.

Como resultado obtido foi aplicada uma estratégia de implantação da Gerência de Requisitos, tendo como base a qualidade, os requisitos e os processos.

6.4 AVALIAÇÃO DO PROCESSO DE GERÊNCIA DE CONFIGURAÇÃO UTILIZANDO O MODELO MPS.BR

Trabalho de Conclusão de Curso de Márcio Assis Miranda para a obtenção do título de especialização da Pós-Graduação Lato Sensu em Melhoria de Processo de Software, em 2008, da Universidade Federal de Lavras, do Estado de Minas Gerais.

O trabalho propõe uma abordagem para a implantação do processo de gerência da configuração, baseado no modelo MPS.BR, apresentando os resultados da avaliação do processo.

6.5 UMA PROPOSTA DE APOIO SISTEMATIZADO À GERÊNCIA DE PORTFÓLIO DO MPS.BR

Trabalho de Conclusão de Curso de Paulo Fernando Souza Rodrigues Júnior para obtenção do grau de Bacharel em Ciência da Computação, em 2009, pelo Departamento de Ciência da Computação, da Universidade Federal do Pará, Instituto de Ciências Exatas e Naturais – Faculdade de Computação.

O trabalho propõe uma abordagem para definir uma metodologia eficaz para a implementação da Gerência de Portfólio, definindo ajustes necessários para atender os requisitos exigidos e elaborar um protótipo que

contemple os ajustes propostos com base nos resultados esperados pelo MPS.BR.

Como resultado foi realizada uma análise do resultado esperado do processo GPP, e das ferramentas utilizadas, mapeando as características destas ferramentas. Foi proposto a realização de ajustes na ferramenta dotProject e elaborada uma metodologia de uso sobre esta ferramenta.

7 DESENVOLVIMENTO DO METAMODELO

Com a realização desta pesquisa foi desenvolvido um metamodelo, tendo como base o modelo MPS.BR e a área de processo GQA do nível F de maturidade, utilizando também como fonte de conhecimento os modelos CMMI e PMBOK, possibilitando a identificação dos pontos considerados críticos em relação à realidade das empresas de software brasileiras.

Com a identificação dos pontos críticos, foi possível elaborar um metamodelo adequado às empresas brasileiras, eliminando atividades que geram esforço e investimento desnecessário, podendo causar sérios problemas para as organizações.

Levando em consideração os pontos citados anteriormente, foi desenvolvido um metamodelo genérico, ou seja, onde são definidas as fases e as atividades, mas a forma na qual as atividades serão executadas fica a critério do usuário, podendo ser aplicado para projetos e produtos. Para o desenvolvimento do metamodelo foi utilizada a ferramenta EPF Composer.

A seguir será demonstrada a metodologia utilizada para o desenvolvimento do trabalho proposto.

7.1 METODOLOGIA

A metodologia utilizada para o desenvolvimento deste trabalho teve início através do levantamento bibliográfico, onde foram estudados os conceitos da Engenharia de Software, abordando os principais ciclos de vida de desenvolvimento, conceitos relacionados à Garantia da Qualidade, incluindo métodos e processos de desenvolvimento de software e estudo sobre o EPF Composer.

O estudo sobre o modelo MPS.BR foi realizado utilizando os guias e documentos disponibilizados no site da Softex, onde foi possível compreender os conceitos relacionados ao modelo e ao método de avaliação MA-MPS.

Para o desenvolvimento do metamodelo foram realizadas as etapas demonstradas na Figura 12.

Figura12- Fases de desenvolvimento do trabalho proposto



Fonte: Do autor.

A seguir serão descritas de forma detalhada, todas as fases de desenvolvimento do trabalho proposto.

7.1.1 Identificação de modelos de referência

Na primeira etapa de elaboração do metamodelo foi realizada a identificação de modelos de referência, no qual foi utilizado o MPS.BR como principal fonte de pesquisa, mas para obter uma visão geral em relação aos modelos de referência existentes, foram abordados no estudo os modelos CMMI e PMBOK.

Com os estudos realizados em relação ao modelo MPS.BR foi possível compreender os principais conceitos de Qualidade de Software que envolvem este modelo, no qual foram agregados padrões e técnicas utilizadas internacionalmente.

Para entender melhor a composição e a base do MPS.BR, foi realizado o estudo sobre o modelo CMMI, possibilitando a identificação dos pontos positivos e negativos em relação à sua aplicação, e a identificação do motivo no qual foi tomado como base para a elaboração de um modelo voltado à realidade brasileira.

Para obter conhecimento em relação ao desenvolvimento de metamodelos e modelos de processo, foi realizado o estudo do PMBOK, que possibilitou uma visão geral dos ciclos de vida de desenvolvimento e sua aplicação nas empresas.

7.1.2 Sintetização dos conceitos de metamodelos

Na segunda etapa de desenvolvimento do trabalho, foram sintetizados os conceitos dos metamodelos existentes, no qual foi possível identificar e comparar os padrões estabelecidos em cada modelo de referência, levando em consideração as atividades propostas e avaliando a sua aplicação de acordo com a necessidade das empresas.

Com a comparação realizada entre os modelos estudados, percebeu-se que o modelo CMMI, apesar de possuir técnicas e padrões conhecidos no mundo todo, é voltado principalmente para empresas de grande porte, dificultando a aplicação em empresas de médio e pequeno porte devido à enorme quantidade de atividades necessárias para a obtenção de cada nível de maturidade.

Com a identificação desta necessidade foi criado o MPS.BR, tornando mais acessível a obtenção dos níveis de maturidade nas empresas de menor porte.

De acordo com os estudos realizados, a criação do MPS.BR não foi suficiente para suprir esta necessidade. Com isso, foi proposto o desenvolvimento de um metamodelo baseado no MPS.BR para a criação de processos aplicados à garantia da qualidade em projetos de software, conforme será demonstrado a seguir.

7.1.3 Construção do metamodelo

Na terceira etapa, foram reunidos os conceitos dos modelos de referência estudados anteriormente, para a elaboração de um metamodelo composto por 7 fases com suas respectivas atividades, ficando a critério do usuário a forma na qual cada atividade será executada, podendo ser aplicado para projetos e produtos (demanda).

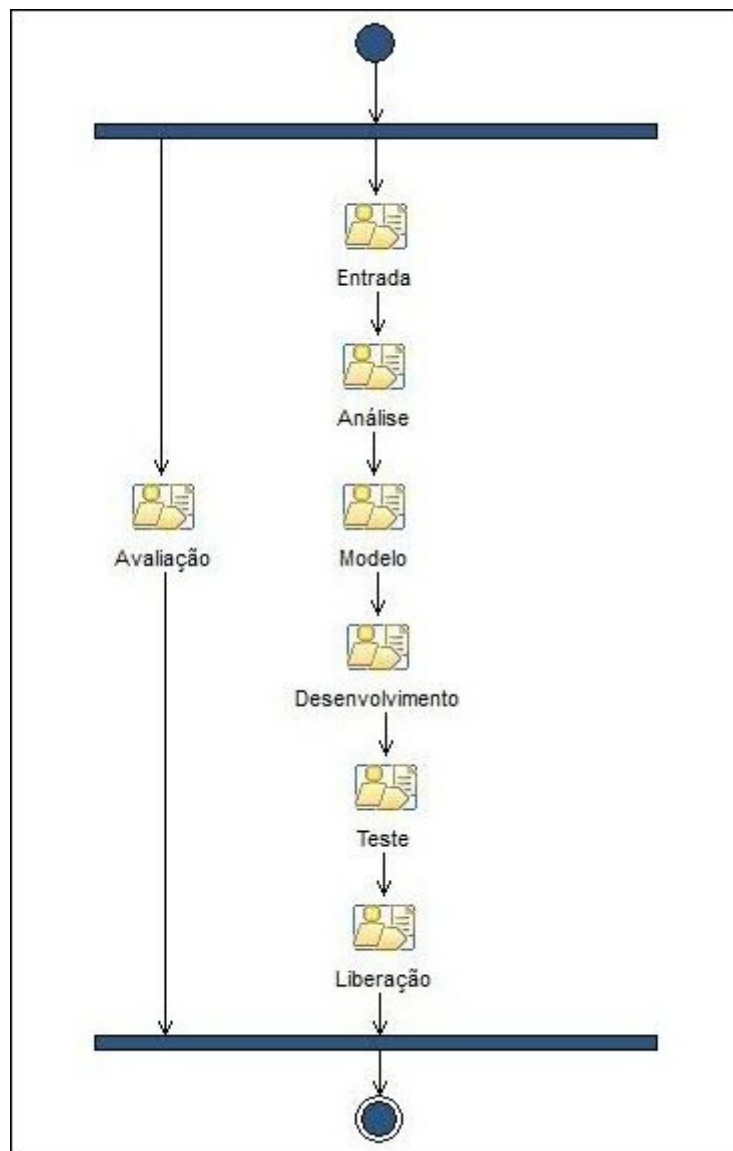
Para o desenvolvimento do metamodelo foram utilizados os conceitos do metamodelo SPEM, que utiliza a unificação de várias

metodologias para modelagem de processos, representando as fases e atividades através de estereótipos estabelecidos pela UML.

Através dos estereótipos estudados, foi possível identificar as fases de um processo, que são períodos que possuem um objetivo específico, e são compostos por atividades, que por sua vez, podem ser subdivididas em tarefas, resultando em artefatos, ou seja, produtos de uma atividade.

Para a representação do metamodelo foi utilizado como base o modelo de atividades, no qual foram aplicados alguns dos estereótipos citados anteriormente. A Figura 13 demonstra todas as fases do metamodelo desenvolvido.

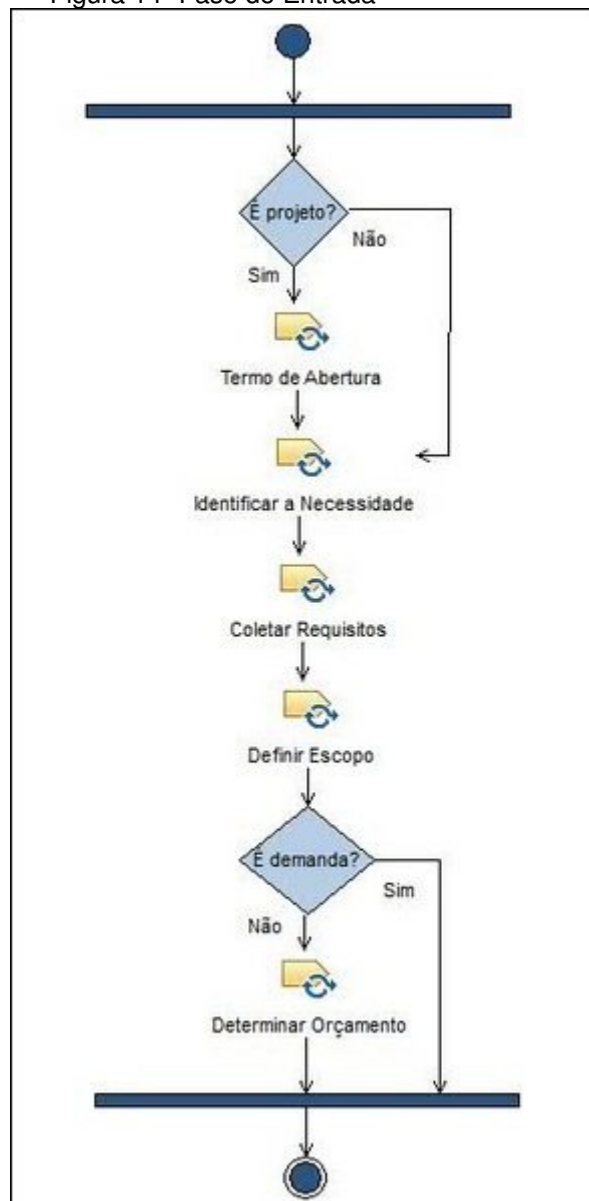
Figura13- Metamodelo



Fonte: Do autor.

Para a modelagem do metamodelo foi utilizado o Framework EPF Composer. A primeira fase do metamodelo a ser desenvolvida e modelada foi a fase de Entrada, conforme mostra a Figura 14.

Figura 14- Fase de Entrada

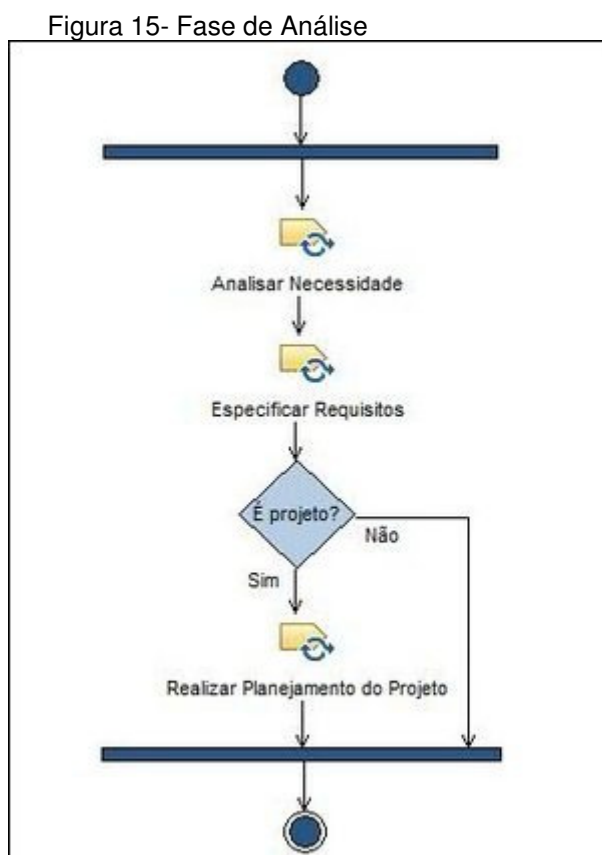


Fonte: Do autor.

A fase de Entrada foi elaborada com base na comparação realizada entre os modelos de referência estudados, e no conhecimento de especialista adquirido no decorrer deste trabalho e no meio profissional no qual foi possível vivenciar a realidade e as necessidades encontradas nas empresas desenvolvedoras de software.

Levando em consideração todo o conhecimento adquirido, esta fase foi criada para atender tanto à demanda quanto ao desenvolvimento de novos projetos nas organizações.

Em seguida, foi elaborada a fase de Análise, conforme demonstrado na Figura 15.



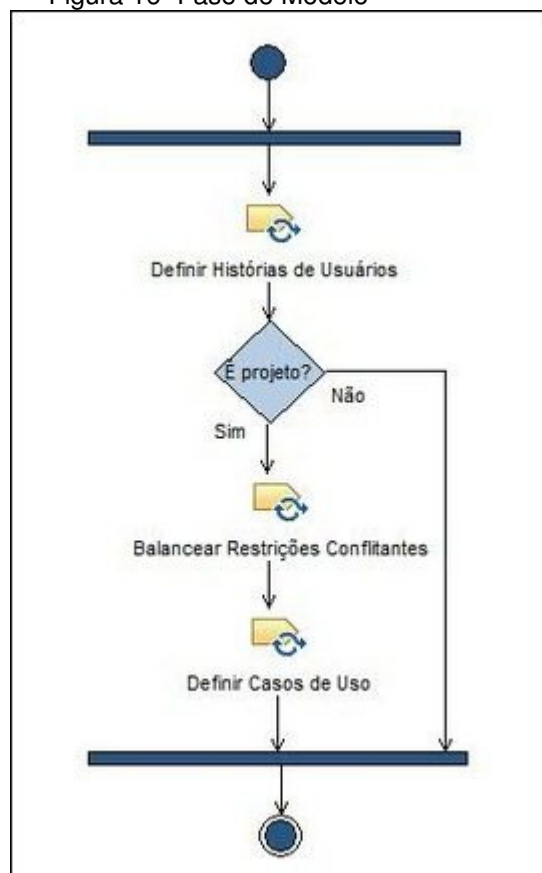
Fonte: Do autor.

A fase de Análise foi desenvolvida, com a utilização de atividades que visam o entendimento da real necessidade do cliente, definindo as regras de negócio envolvidas, estabelecendo a prática de efetuar o planejamento antes de dar início ao desenvolvimento dos projetos ou de alterações solicitadas pelo cliente (demanda).

Durante a atividade de realização do planejamento, existem vários pontos que devem ser analisados, como a definição de recursos, estimativa e controle de riscos, dentre outros. Mas a forma na qual será realizado o planejamento das atividades e os critérios que deverão ser planejados, serão decididos pelo usuário, de acordo com a necessidade da organização.

A fase de Modelo foi elaborada com base na análise dos requisitos realizada na fase de Análise, conforme é exibido na Figura 16.

Figura 16- Fase de Modelo



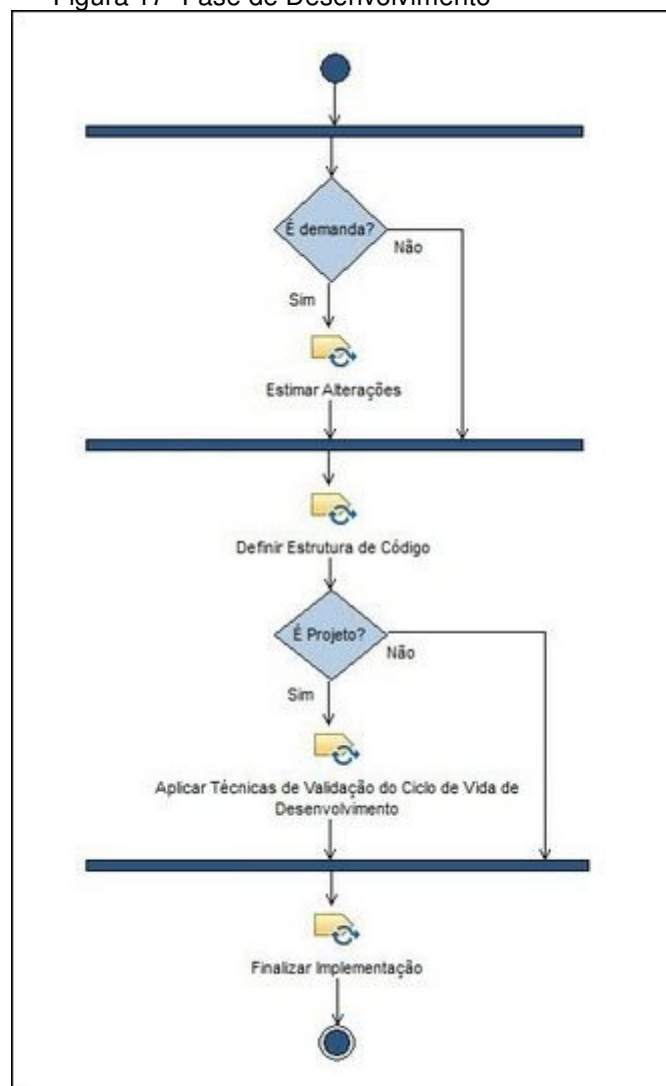
Fonte: Do autor.

Na fase de Modelo são definidas as histórias de usuários, que devem ser realizadas de acordo com os requisitos e as regras de negócio estabelecidas na fase anterior, possibilitando a realização da rastreabilidade e permitindo que sejam identificadas as situações conflitantes entre os requisitos do sistema. Posteriormente foi estabelecida a atividade de definição dos casos de uso, onde serão descritas todas as funcionalidades que deverão existir no sistema, no ponto de vista do usuário.

Após o término da fase de Modelo, todo o cenário das funcionalidades deve estar definido, possibilitando que seja iniciada a próxima fase, na qual foram abordadas as principais atividades a serem executadas para o desenvolvimento do projeto ou produto, levando em consideração todos os conceitos de qualidade estudados durante este trabalho.

Desta forma, foi iniciada a fase de desenvolvimento, conforme mostra a Figura 17.

Figura 17- Fase de Desenvolvimento



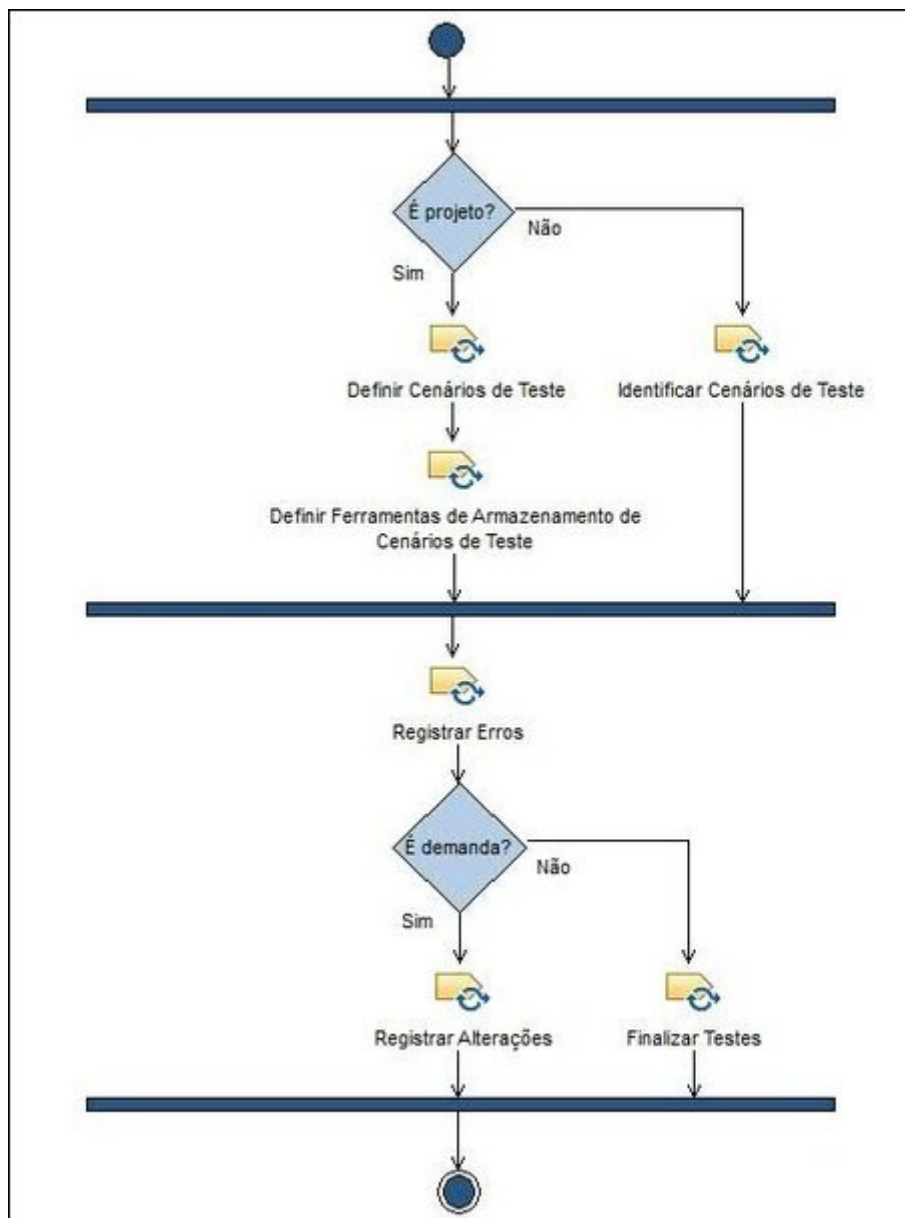
Fonte: Do autor.

Na fase de Desenvolvimento deve ser definida a estrutura do código que será implementado para atender às necessidades e funcionalidades descritas nas fases anteriores. Após a definição da estrutura de código, devem ser aplicadas técnicas de validação do ciclo de vida de desenvolvimento para garantir que o trabalho está sendo executado da forma correta, evitando o retrabalho para a correção de erros que podem ser identificados e corrigidos logo na fase de desenvolvimento.

Para atender a demanda, é fundamental que seja realizada a estimativa das alterações que serão realizadas no sistema, para que sejam identificadas as situações em que possa ocorrer conflito entre funcionalidades.

A próxima fase a ser desenvolvida foi a fase de Teste, conforme demonstra a Figura 18.

Figura 18- Fase de Teste



Fonte: Do autor.

Após o desenvolvimento, foi definida a fase de Teste, no qual foram estabelecidas as atividades de definição de cenários de testes e definição de ferramenta para armazenamento de cenários de teste, com objetivo de tornar os testes automatizados, através do reaproveitamento de cenários, economizando tempo durante a execução dos testes.

O registro de erros encontrados também é fundamental, como forma de controlar as correções que devem ser efetuadas e como forma de garantir que novos testes sejam realizados após a efetuação das devidas correções no sistema.

Para demanda também é muito importante que seja realizado o registro das alterações, como forma de disseminar a informação dentro da própria empresa e principalmente para os usuários do sistema.

Em seguida foi elaborada a fase de liberação, conforme demonstrado na Figura 19.

Figura 19- Fase de Liberação



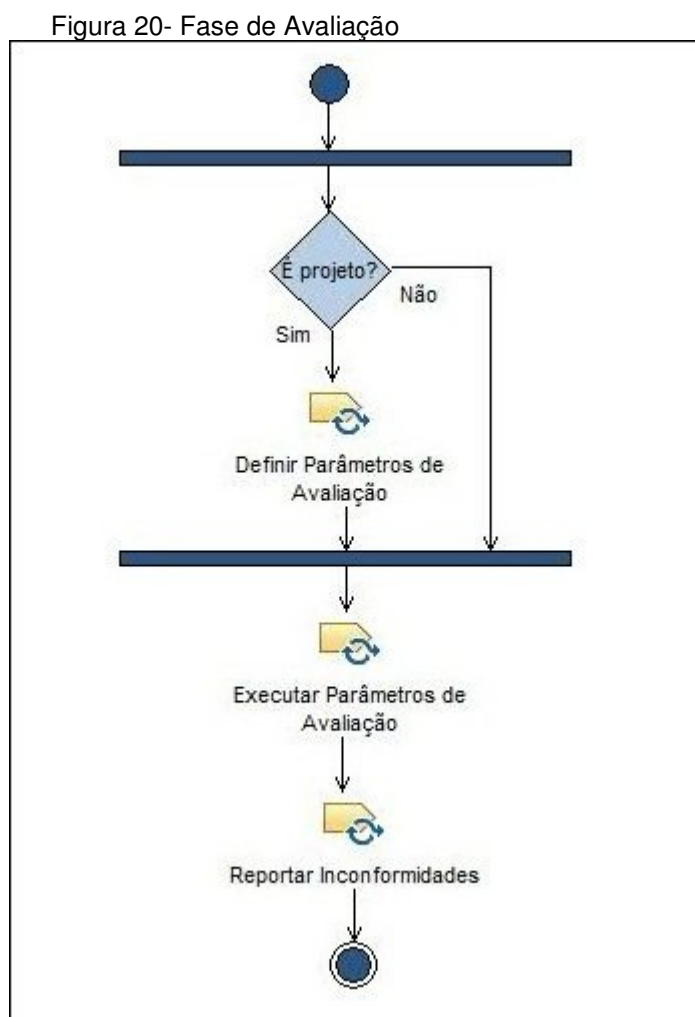
Fonte: Do autor

Na fase de Liberação, deve ser executado o teste de aceitação mínima no *release*, como forma de verificar se o sistema está de acordo com os requisitos estabelecidos e às necessidades do usuário final.

Para garantir que os sistemas serão desenvolvidos com qualidade, deve ser aplicada a pesquisa de satisfação, permitindo que seja medida a satisfação do usuário em relação ao atendimento de suas necessidades, em relação ao cumprimento dos prazos estabelecidos, dentre várias questões que podem ser analisadas, permitindo que se tenha uma ampla visão em relação às melhorias que devem ser efetuadas para garantir a qualidade nos projetos e

produtos desenvolvidos nas organizações, finalizando com uma reunião de encerramento.

Em seguida, foi desenvolvida a fase de Avaliação, conforme exibido na Figura 20.



Fonte: Do autor.

Após o desenvolvimento de todas as fases do metamodelo, foi elaborada a fase de Avaliação, na qual são definidos os parâmetros de avaliação do projeto, que podem variar de acordo com a organização. É realizada a execução dos parâmetros estabelecidos, com o objetivo de identificar os pontos falhos, reportando as inconformidades e apontando as melhorias que devem ser efetuadas para a correção destas falhas.

Para realizar a avaliação do metamodelo desenvolvido, foi elaborado um modelo de processo, conforme será demonstrado o capítulo a seguir.

7.1.4 Construção do modelo de processo

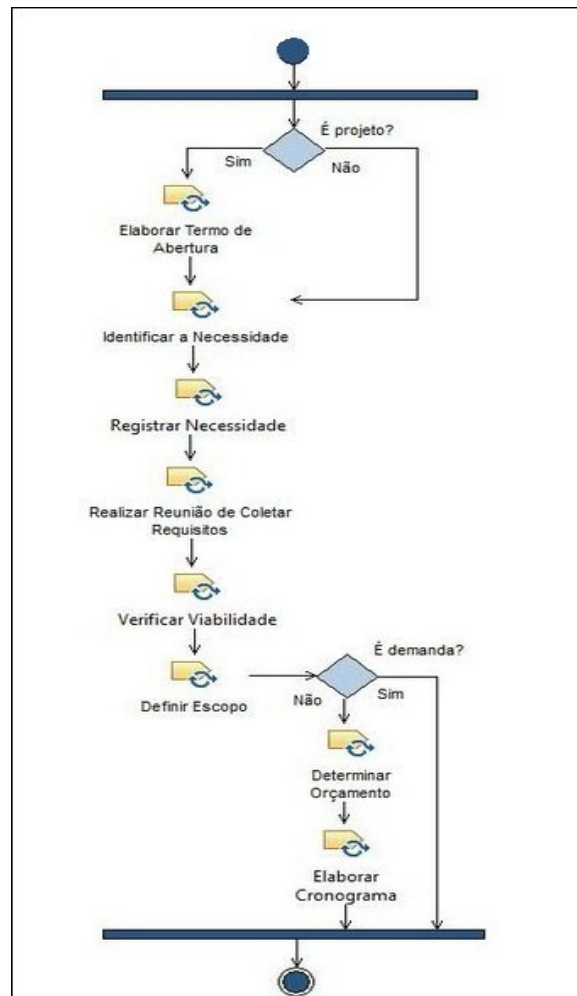
Com base no metamodelo desenvolvido foi elaborado um modelo de processo, pensando na realidade encontrada atualmente nas empresas.

Para a criação do processo foram reunidas todas as atividades contidas no metamodelo, onde foram incluídas algumas tarefas que são opcionais durante a aplicação do processo, mas que de acordo com a pesquisa realizada durante todo o decorrer do trabalho, agregam qualidade ao processo quando aplicadas.

Identificou-se a necessidade da elaboração de um modelo de processo, como forma de demonstrar a aplicação dos conceitos de qualidade empregados no metamodelo, sendo que todas as atividades propostas tem o objetivo de garantir a qualidade desde a coleta de requisitos, a análise e desenvolvimento do produto, até a conclusão o projeto.

Na Figura 21 é demonstrada a fase de Entrada do modelo de processo.

Figura 21- Processo - Fase de Entrada



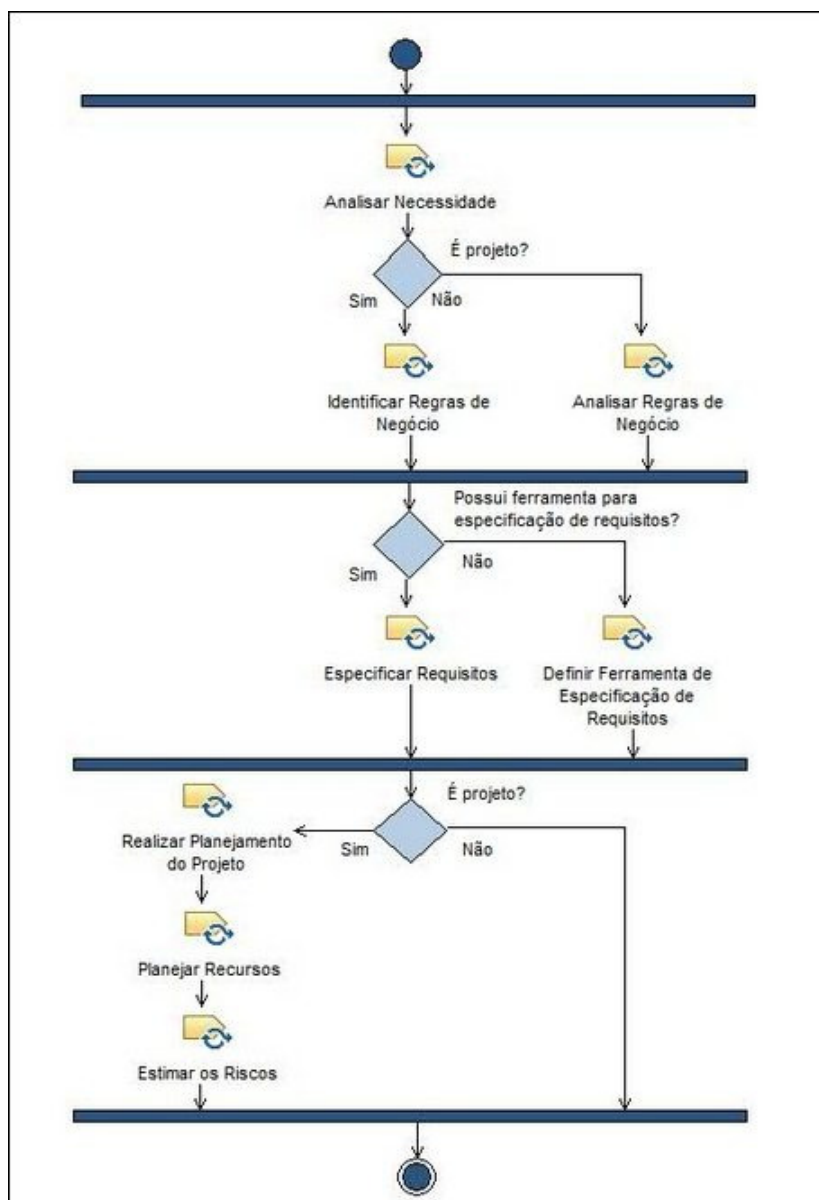
Fonte: Do autor.

Na Figura 22 é demonstrada a fase de Análise do processo, onde foram incluídas atividades que permitem a análise detalhada da necessidade identificada na fase de entrada.

Nesta fase é realizada a análise de negócio e a especificação dos requisitos, como forma de documentar as funcionalidades que serão desenvolvidas posteriormente, garantindo que a implementação será realizada de acordo com o que foi estabelecido pelo analista.

Nesta fase também é realizado o planejamento do projeto, onde são estimados os riscos, definidos os recursos, dentre outras atividades que podem ser incluídas de acordo com a necessidade da organização.

Figura 22- Processo - Fase de Análise

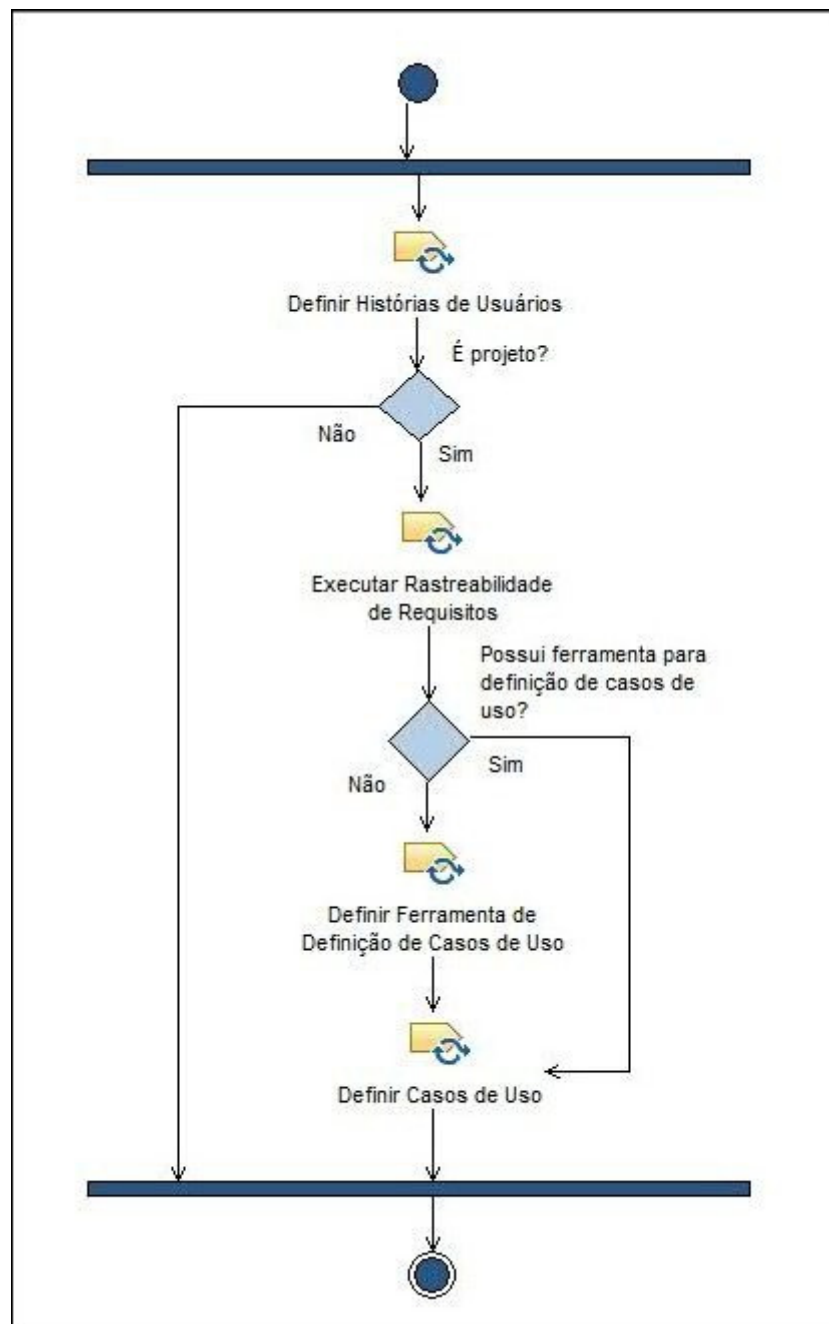


Fonte: Do autor.

Na Figura 23 é exibida a fase de Modelo do processo, onde são definidas as histórias de usuário, que é a definição das funcionalidades de acordo com a visão do cliente.

Nesta fase é realizada a rastreabilidade de requisitos, onde é verificado o relacionamento entre as funcionalidades, permitindo que seja feita a prevenção de conflitos no sistema e posteriormente são definidos os casos de uso.

Figura 23- Processo - Fase de Modelo



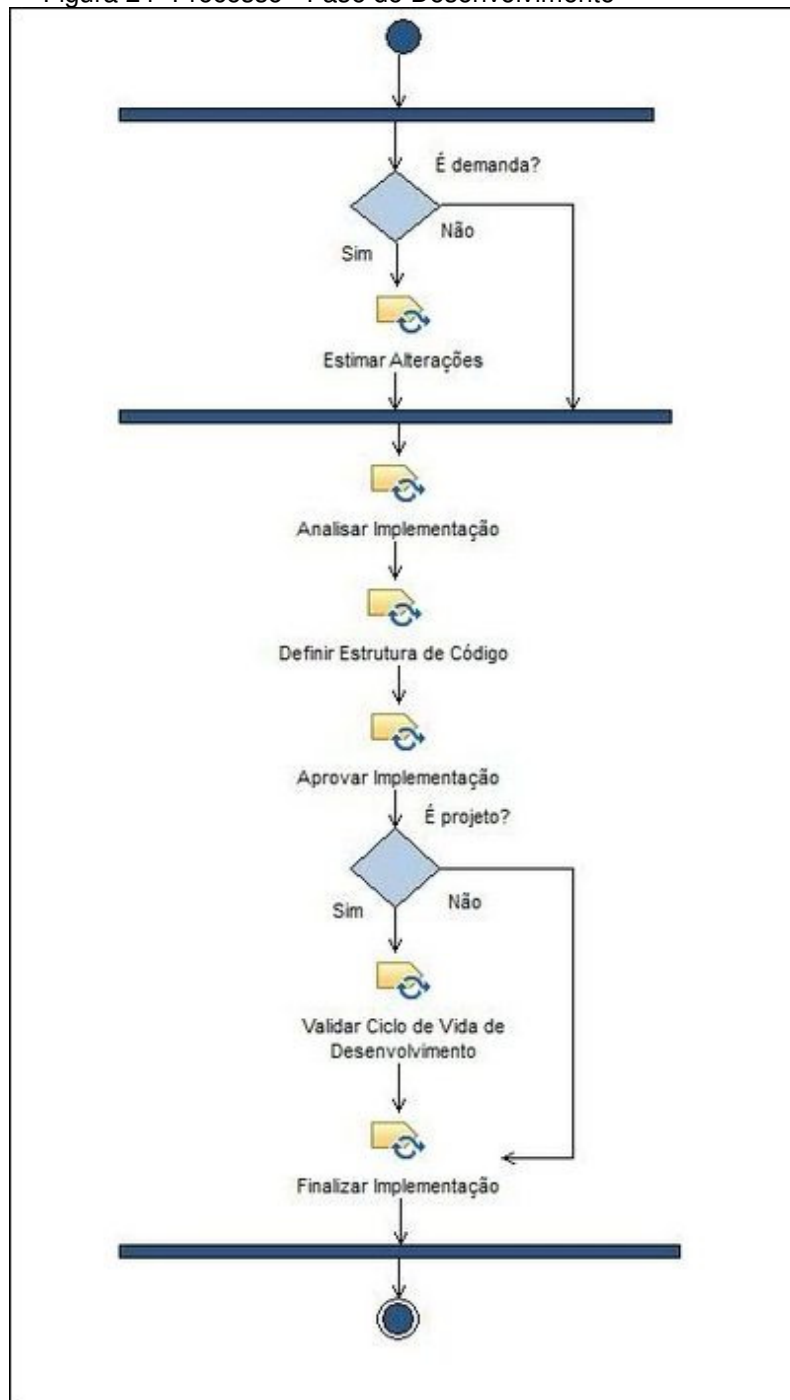
Fonte: Do autor.

Na Figura 24 é demonstrada a fase de Desenvolvimento, onde é realizada a estimativa das alterações a serem realizadas, para o caso de atendimento da demanda.

Nesta fase é efetuada a análise de implementação, onde é definida a estrutura de código e realizada a validação do ciclo de vida de

desenvolvimento, para garantir que as atividades estão sendo executadas da forma adequada.

Figura 24- Processo - Fase de Desenvolvimento



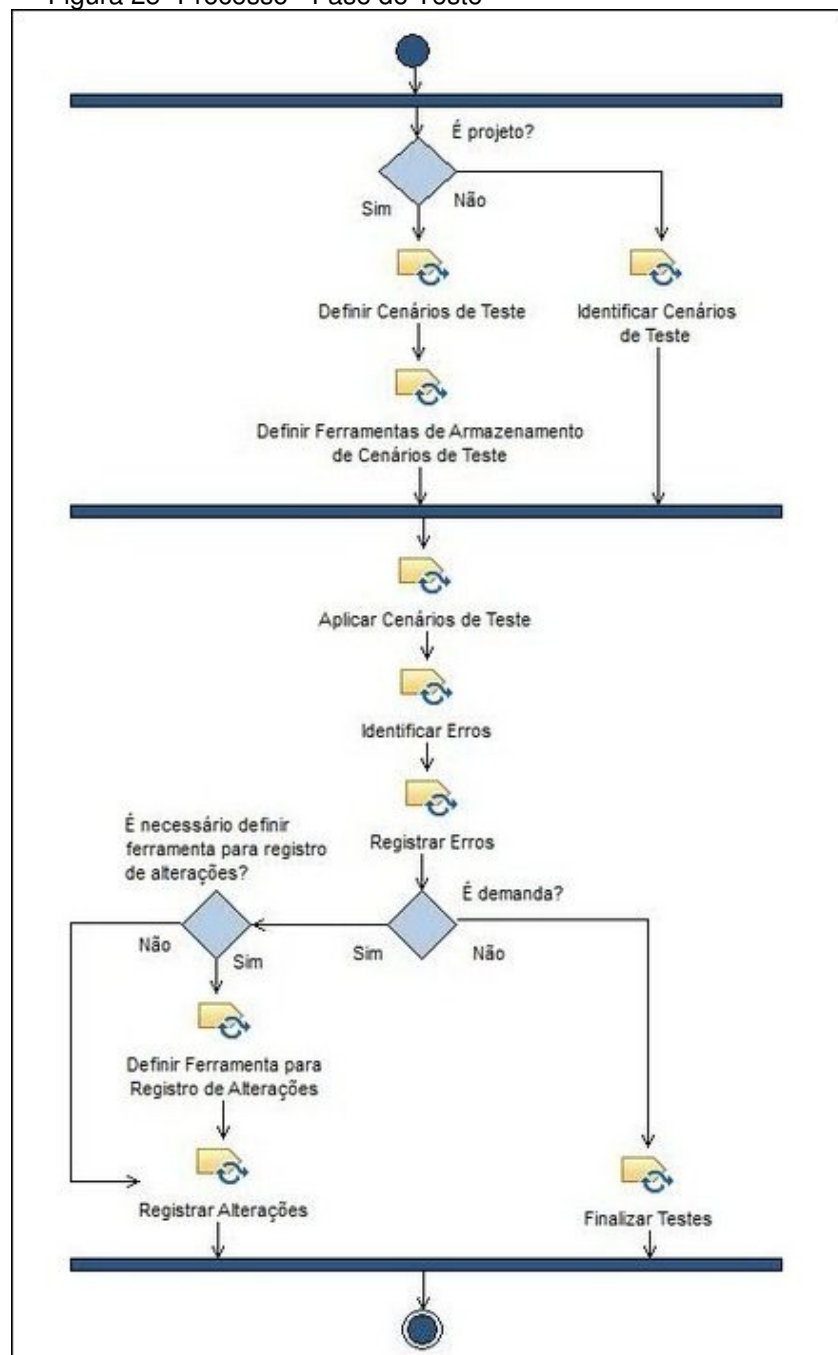
Fonte: Do autor.

Na Figura 25 é exibida a fase de Teste do modelo de processo, permitindo a definição de ferramentas de armazenamento de cenários de teste,

possibilitando o reaproveitamento de cenários, evitando o retrabalho de desenvolvê-los a cada teste realizado.

Os erros são registrados quando identificados e no caso de demanda, as alterações realizadas são registradas como forma de informação para o cliente e para a própria organização.

Figura 25- Processo - Fase de Teste

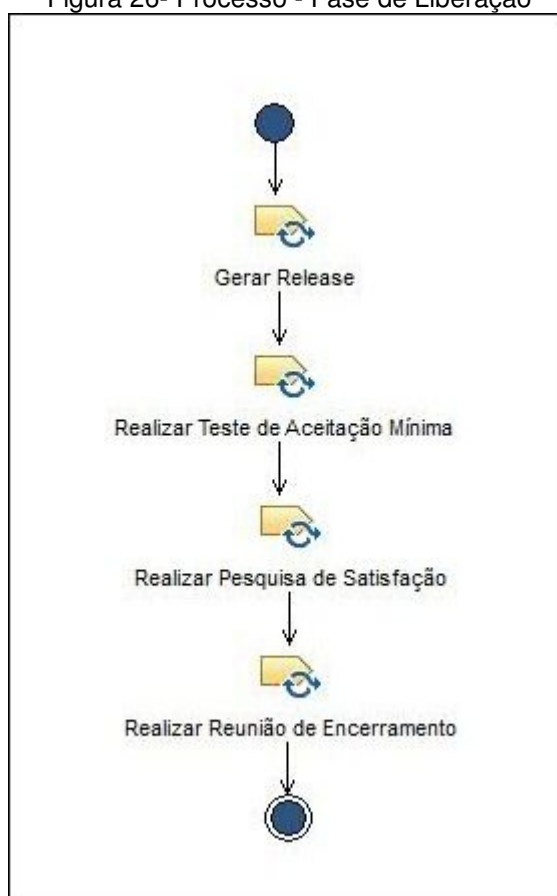


Fonte: Do autor.

A Figura 26 demonstra a fase de Liberação, onde é efetuado o teste de aceitação mínima para garantir que o sistema está de acordo com a solicitação do cliente e conforme o planejamento.

Nesta fase é realizada uma pesquisa de satisfação como forma de medir a qualidade do produto final, identificando os pontos que devem ser melhorados em relação ao atendimento e ao produto disponibilizado para o cliente, concluindo a fase com uma reunião de encerramento.

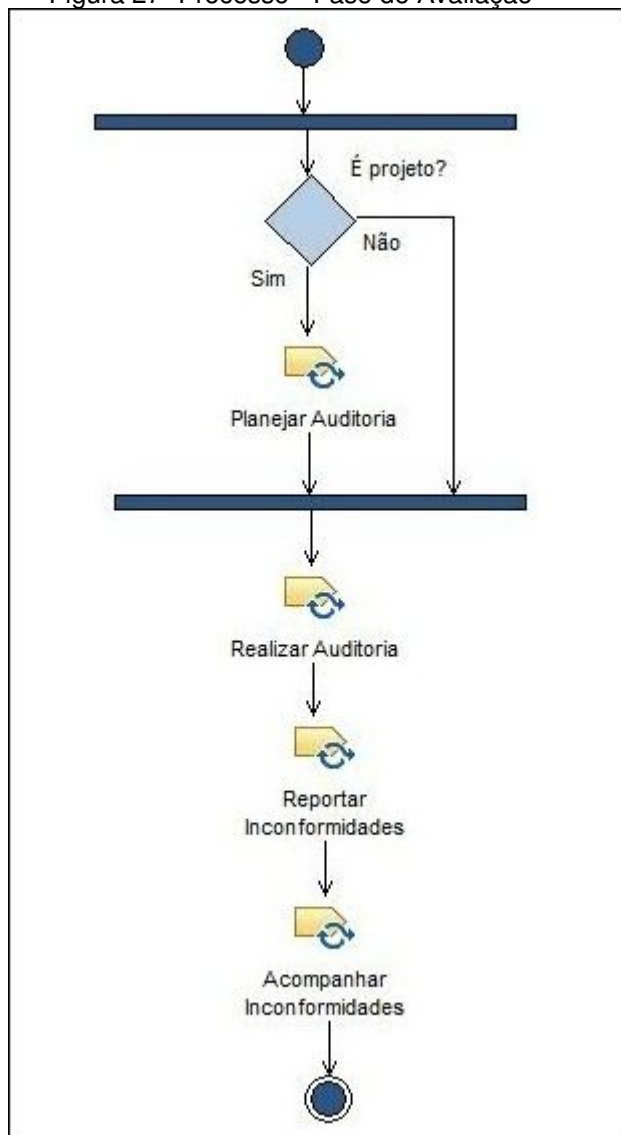
Figura 26- Processo - Fase de Liberação



Fonte: Do autor.

A Figura 27 mostra a fase de Avaliação, onde é realizado o planejamento das auditorias que serão realizadas como forma de avaliar as atividades desenvolvidas durante todo o projeto, com isso é possível identificar e reportar as anomalias e acompanhar a resolução das inconformidades.

Figura 27- Processo - Fase de Avaliação



Fonte: Do autor.

No próximo capítulo será demonstrado o método de avaliação utilizado para avaliar o metamodelo desenvolvido neste trabalho.

7.1.5 Modelo de avaliação baseado no metamodelo proposto

Para efetuar a avaliação do metamodelo, primeiramente foi realizado o estudo do método de avaliação MA-MPS para compreender o seus principais

conceitos, a forma como é aplicado e como são gerados os resultados da avaliação.

Com base nos estudos percebeu-se que para realizar uma avaliação formal, seria essencial a aplicação do metamodelo em uma organização. Mas como o tempo disponível não seria suficiente e para que essa aplicação fosse possível seriam necessários muitos recursos e um esforço significativo, optou-se por realizar como forma de avaliação a análise de GAP utilizando a planilha de avaliação formal do MA-MPS.

Com a utilização desta planilha foi comparado o metamodelo desenvolvido, em relação ao modelo ideal, analisando os processos GRE e GRP, que compõe o nível de maturidade G do MPS.BR, e foram analisados também os atributos de processo.

Realizando a avaliação através da análise do processo GRE é possível identificar se os requisitos do produto do projeto são gerenciados, permitindo a identificação de inconsistências em relação aos requisitos, ao planejamento do projeto e aos produtos de trabalho.

A análise do processo GRP permite identificar se foi realizado o planejamento do projeto em relação ao controle do andamento das atividades, permitindo a realização de correções quando há desvios no decorrer do projeto.

O preenchimento da planilha foi realizado através do resultado esperado e das evidências, sendo que, para cada questionamento foram apresentados Artefatos Diretos (AD) que demonstram o cumprimento de cada item. Os Artefatos Indiretos (AI) e as Afirmações (AF) não foram apresentados, sendo que seria necessário a aplicação do metamodelo em uma organização.

Para demonstrar a forma como foi realizada a avaliação, serão detalhados os itens contidos na planilha.

Figura 28 – Planilha de Avaliação – Processo GRE

Resultado esperado / evidências	Fonte da evidência	ORG	Projeto 1	Projeto 2	Projeto 3	Projeto 4	Final
O propósito do processo Gerência de Requisitos é gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.							

Fonte: Softex (2009).

A Figura 28 demonstra o Resultado esperado com a avaliação do processo GRE que são demonstrados através da apresentação de evidências.

Na coluna Fonte da evidência é identificado o grau de implementação de cada evidência apresentada, sendo classificada como Totalmente implementada (T), Largamente implementada (L), Parcialmente implementada (P), Não implementada (N), e Não avaliada (NA).

Nesta avaliação as evidências foram classificadas apenas como T ou L, sendo que para utilizar as demais classificações seria necessária a aplicação do processo em uma organização. As colunas ORG, Projeto1 a Projeto 4, e Final, também não foram utilizadas pelo mesmo motivo.

Na Figura 29 são demonstradas as evidências relacionadas ao entendimento, aceitação e avaliação dos requisitos.

No GRE 1 foram apresentadas evidências que demonstram que o modelo avaliado permite que seja identificada a necessidade do cliente e coletados os requisitos para a análise, onde serão definidas as regras de negócio e armazenamento dos requisitos em uma ferramenta de armazenamento como forma de documentação.

No GRE 2 não foram apresentadas evidências, devido a este item não ter sido implementado.

No GRE 3 foi demonstrada a atividade *Balancear Requisitos Conflitantes* como forma de demonstrar a execução da rastreabilidade dos requisitos em relação às funcionalidades que serão desenvolvidas.

Figura 29- Evidências – GRE 1 a GRE 3

GRE 1. Os requisitos são entendidos, avaliados e aceitos junto ao fornecedor de requisitos, utilizando critérios objetivos. As evidências apresentadas para este resultado permitem assegurar: (ii) que as pessoas autorizadas a definir e a alterar requisitos foram identificadas? (ii) que existe um documento de requisitos que represente seu entendimento? (iii) que foram definidos critérios para análise de requisitos e se estes foram usados como base para a avaliação e a aceitação dos requisitos do projeto?	
AD - Na fase de Entrada é identificada a necessidade junto ao cliente e coletados os requisitos que posteriormente são analisados	T
AD - Na fase de Análise são definidas as regras de negócio e realizada a documentação através da especificação de requisitos	T
	(T,L,P,N,NA)
GRE 2. O comprometimento da equipe técnica com os requisitos aprovados é obtido. As evidências apresentadas para este resultado permitem assegurar: (i) que foi obtido e registrado um comprometimento formal da equipe técnica com os requisitos aprovados? (ii) que um novo comprometimento da equipe técnica com os requisitos foi obtido e registrado quando houve mudanças nos requisitos?	
Este item não foi implementado	N
	(T,L,P,N,NA)
GRE 3. A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida. As evidências apresentadas para este resultado permitem assegurar que foi criada e mantida, ao longo do projeto, a rastreabilidade bidirecional entre os requisitos e demais produtos de trabalho, incluindo os planos do projeto e as unidades de código?	
AD - Na fase de Modelo, a atividade "Balancear Requisitos Conflitantes" permite que seja realizada a rastreabilidade para verificar se as funcionalidades estão de acordo com os requisitos	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Na Figura 30 foi demonstrado nos processos GRE 4 e GRE 5 a atividade *Definir Histórias de Usuários* e *Definir Casos de Uso*, da fase de Modelo como evidências que demonstram a existência da revisão dos requisitos e funcionalidades estabelecidas de acordo com a visão do cliente, e a rastreabilidade como forma de garantir que não haja conflito entre as funcionalidades após a alteração.

Figura 30- Evidências – GRE 4 e GRE 5

GRE 4. Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos. As evidências apresentadas para este resultado permitem assegurar: (i) que foram executadas revisões para identificar inconsistências em planos e demais produtos de trabalho do projeto, com base nos requisitos? (ii) que foram executadas ações para corrigir inconsistências identificadas ao longo do projeto?	
AD - Na fase de Modelo, a atividade "Definir Histórias de Usuário" possibilita que sejam revistos os requisitos e definidas as funcionalidades de acordo com a visão do cliente	T
	(T,L,P,N,NA)
GRE 5. Mudanças nos requisitos são gerenciadas ao longo do projeto. As evidências apresentadas para este resultado permitem assegurar: (i) que existe um histórico das solicitações de mudança em requisitos do projeto, disponível para a equipe do projeto? (ii) que foi realizada uma análise do impacto destas mudanças antes de sua implementação?	
AD - Na fase de Modelo é possível realizar mudanças nos requisitos através das atividades "Definir Histórias de Usuário" e "Definir Casos de Uso"	T
AD - Através da rastreabilidade é possível garantir que não haja conflito entre funcionalidades após a alteração	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Na Figura 31 são exibidos os atributos de processo no qual é avaliado se o modelo apresenta atividades de execução, gerenciamento e planejamento do processo.

No AP 1.1, foram apresentadas como evidências as atividades de *Coleta de Requisitos*, da fase de Entrada, permitindo que seja realizada a *Especificação dos Requisitos* na fase de Análise e posteriormente que sejam modeladas as funcionalidades na fase de Modelo, através da *Definição de Histórias de Usuários e Definição de Casos de Uso*.

Após a execução das fases de Desenvolvimento e Teste, é executada a fase de Liberação, onde é executado o *Teste de Aceitação Mínima* na *Release* gerada, como forma de garantir que o sistema foi desenvolvido de acordo com o planejamento.

No AP 2.1 não foram apresentadas evidências relacionadas ao gerenciamento, pois não foi implementada uma atividade relacionada à política organizacional, mas como se trata de um modelo genérico, pode ser incluída esta atividade durante o planejamento do projeto, de acordo com a necessidade da organização.

Em relação ao planejamento, foi apresentada como evidência a atividade *Realizar Planejamento do Projeto* da fase de Análise, onde é possível realizar todo o planejamento através da inclusão ou retirada de itens que não sejam necessários ao controle da organização ao qual será aplicado.

A fase de Avaliação foi apontada como evidência para a demonstração de que o modelo permite que seja realizado o monitoramento da realização das atividades do projeto, como forma de garantir que o planejamento está sendo cumprido.

Figura 31- Evidências – AP 1.1 e AP 2.1

Atributos de Processo	
Resultado esperado / evidências	
AP 1.1 O processo é executado	
RAP 1. O processo atinge seus resultados definidos. As evidências apresentadas para este resultado permitem assegurar que o processo transforma produtos de trabalho de entrada identificáveis em produtos de trabalho de saída, também identificáveis, permitindo atingir o propósito do processo?	
AD - Através da fase de Entrada são coletados os requisitos que são especificados na fase de Análise e tem as funcionalidades definidas na fase de Modelo.	T
AD - Após o desenvolvimento e teste do produto, na fase de Liberação é executado o teste de aceitação mínima na release gerada para garantir que o produto foi desenvolvido de acordo com o planejamento.	T
	(T,L,P,N,NA)
AP 2.1 O processo é gerenciado	
RAP 2. Existe uma política organizacional estabelecida e mantida para o processo. As evidências apresentadas para este resultado permitem assegurar: (i) que foi definida uma política organizacional estabelecendo as expectativas da organização para a execução do processo e se esta política é conhecida e de fácil acesso aos interessados? (ii) que a política organizacional foi atualizada, quando necessário? (iii) que a política organizacional tem respaldo da alta administração (por exemplo, por meio de aprovação da alta administração)? Este item não foi implementado	N
	(T,L,P,N,NA)
RAP 3. A execução do processo é planejada. As evidências apresentadas para este resultado permitem assegurar que existe um plano para a execução do processo?	
AD - Na fase de Análise, através da atividade "Realizar o Planejamento do Projeto" é permitindo que seja planejada toda a execução do processo.	T
	(T,L,P,N,NA)
RAP 4. (Para o nível G). A execução do processo é monitorada e ajustes são realizados. As evidências apresentadas para este resultado permitem assegurar que o processo foi executado conforme planejado e se ações corretivas foram tomadas quando a execução do processo se desviou dos planos?	
AD - Na fase de Avaliação é realizado o monitoramento da execução das atividades realizadas o projeto	T
	(T,L,P,N,NA)
RAP 10. (Para o nível G). O processo planejado para o projeto é executado. As evidências apresentadas para este resultado permitem assegurar que o projeto foi conduzido a partir da execução do seu processo planejado, bem como se existem registros de execução das atividades do processo com base no seu planejamento?	
AD - Na fase de Análise é executada a atividade "Definir Planejamento do Projeto"	T
AD - Na fase de Avaliação é realizado o acompanhamento das atividades de acordo com o que foi estabelecido no planejamento	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

A Figura 32 exibe os resultados esperados com a avaliação do processo GPR, considerando a existência do planejamento das atividades, e recursos e controle do andamento do projeto.

Figura 32- Planilha de Avaliação – Processo GRP

Resultado esperado / evidências	Fonte da evidência	ORG	Projeto 1	Projeto 2	Projeto 3	Projeto 4	Final
O propósito do processo Gerência de Projetos é estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto. O propósito deste processo evolui à medida que a organização cresce em maturidade. Assim, a partir do nível E, alguns resultados evoluem e outros são incorporados, de forma que a gerência de projetos passe a ser realizada com base no processo definido para o projeto e nos planos integrados. No nível B, a gerência de projetos passa a ter um enfoque quantitativo, refletindo a alta maturidade que se espera da organização. Novamente, alguns resultados evoluem e outros são incorporados.							

Fonte: Softex (2009).

Na Figura 33 é exibido o GRP 1, onde é avaliada a definição do escopo do projeto. Neste caso foi apresentada a fase de Entrada como evidência, sendo que nesta fase é executada a atividade de *Definir Escopo*, no qual é definido o objetivo do projeto que será desenvolvido posteriormente.

Na fase de Análise, onde é realizada a especificação dos requisitos em uma ferramenta de armazenamento definida pela organização.

No GRP 2 foi apresentada como evidência a fase de Análise, na qual foi definida a atividade de *Realizar Planejamento do Projeto* que permite que seja planejada e estimada a complexidade das tarefas que serão executadas.

No GPR 3 foram apresentadas as sete fases do metamodelo, como forma de demonstrar a definição do ciclo de vida de desenvolvimento e interdependência entre todas as fases.

Figura 33- Evidências – GPR 1 a GPR 4

GPR 1. O escopo do trabalho para o projeto é definido. As evidências apresentadas para este resultado permitem assegurar que o escopo do projeto foi definido?	
AD - Na Fase de Entrada existe a atividade "Definir Escopo"	T
	(T,L,P,N,NA)
GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados. As evidências apresentadas para este resultado permitem assegurar que o tamanho e/ou a complexidade das tarefas e dos artefatos gerados no projeto foram estimados utilizando métodos adequados (ex: baseados na EAP ou estrutura equivalente, em técnicas de estimativa ou em dados históricos)?	
AD - Na fase de Análise existe a atividade "Realizar Planejamento do Projeto"	T
	(T,L,P,N,NA)
GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos. As evidências apresentadas para este resultado permitem assegurar que o modelo do ciclo de vida do projeto foi definido, indicando suas fases, as relações de sequência e interdependência entre elas?	
AD - O modelo possui 7 fases interdependentes: Entrada, Análise, Modelo, Desenvolvimento, Teste, Liberação e Avaliação	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Na Figura 34 é demonstrado o GPR 4, onde tem como evidência a fase de Entrada com a atividade *Determinar Orçamento*, onde é definido o cronograma das atividades e estimado o custo para o desenvolvimento.

A fase de Análise também foi apresentada como forma de demonstrar a realização da estimativa de recursos, riscos e esforço através da atividade *Realizar Planejamento do Projeto*.

Nos processos GPR 5 e GPR 6 também foi apresentada a fase de Análise como evidência, sendo que esta fase permite que seja executado todo o planejamento das atividades que serão aplicadas no decorrer do projeto.

Figura 34- Evidências – GRE 4 a GRE 6

GPR 4. (Até o nível F) 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.	
As evidências apresentadas para este resultado permitem assegurar que foram realizadas estimativas de custo e esforço para tarefas e produtos de trabalho com base em dados históricos ou métodos de estimativas e se foram documentadas as suas justificativas?	
AD - Na fase de Entrada existe a atividade "Determinar Orçamento" onde é definido o cronograma e estimado o custo. Na fase de Análise existe a atividade "Realizar Planejamento do Projeto" onde é realizada a estimativa de recursos, riscos e esforço	T
	(T,L,P,N,NA)
GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos.	
As evidências apresentadas para este resultado permitem assegurar que: (i) o orçamento e o cronograma foram definidos, revistos e atualizados ao longo do desenvolvimento, conforme necessário?; (ii) o cronograma possui marcos e/ou pontos de controle?; (iii) o cronograma estabelece as dependências entre tarefas?	
AD - Na fase de Entrada a atividade "Determinar Orçamento" permite a definição do cronograma das atividades e o custo estimado para a execução destas atividades	T
	(T,L,P,N,NA)
GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados.	
As evidências apresentadas para este resultado permitem assegurar que: (i) existe uma lista dos riscos identificados para o projeto? (ii) foi realizada uma análise para determinar o impacto, o grau de importância, a probabilidade e a prioridade de cada risco?	
AD - Na fase de Análise, a atividade "Realizar Planejamento do Projeto" permite a realização da estimativa e definição de riscos envolvidos no projeto	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Conforme mostra a Figura 35, no GPR 7, GPR 8 e GPR 9 foi apresentada a fase de Análise como artefato direto, pois nesta fase é possível planejar os recursos humanos e a estrutura física, como também determinar mecanismos de armazenamento documentações referente ao projeto, incluindo políticas de segurança da informação.

Figura 35- Evidências – GRE 7 a GRE 9

GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo. As evidências apresentadas para este resultado permitem assegurar que: (i) a equipe do projeto foi selecionada a partir das competências requeridas para realizar as atividades do projeto e considerando o perfil dos candidatos?; (ii) foi planejado treinamento, quando necessário?	
AD - Na fase de Análise, a atividade "Realizar Planejamento do Projeto" permite que sejam planejados os recursos necessários para a execução das atividades	T
	(T,L,P,N,NA)
GPR 8. Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados. As evidências apresentadas para este resultado permitem assegurar que foram planejados os recursos e o ambiente de trabalho necessários? (obs: aqui trata-se de outros recursos que não recursos humanos).	
AD - Na fase de Análise, a atividade "Realizar Planejamento do Projeto" permite que sejam planejados os recursos físicos necessários para o desenvolvimento do projeto	T
	(T,L,P,N,NA)
GPR 9. 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. As evidências apresentadas para este resultado permitem assegurar que existe um plano para gerência de dados, que relacione todos os documentos gerados no projeto, sua distribuição, mídia para armazenamento, forma de proteção (segurança e sigilo) e recuperação dos dados?	
AD - Na fase de Análise, a atividade "Realizar Planejamento do Projeto" permite que seja planejado o armazenamento dos dados, possibilitando a definição de políticas de segurança da informação.	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Na Figura 36 são exibidos o GRP 10, GPR 11 e GPR 12, nestes itens são avaliadas questões relacionadas à organização e planejamento do projeto através da documentação das atividades, verificação da viabilidade do projeto e revisão do plano do projeto.

Para demonstrar a existência destes itens foi apresentada a atividade *Realizar Planejamento do Projeto*, onde é possível documentar todo o planejamento realizado para que posteriormente seja realizada a avaliação com auditorias que garantem que as ações planejadas foram executadas da forma adequada.

Como forma de verificar a viabilidade do projeto, foram apresentadas as atividades *Identificar Necessidade* e *Coletar Requisitos*.

Com a atividade *Realizar Planejamento do Projeto* é possível realizar a revisão e adequação do planejamento de acordo com a necessidade e comunicar as alterações às partes interessadas para que seja firmado o compromisso com o que foi estabelecido.

Figura 36- Evidências GRP 10 a GRP 12

GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos. As evidências apresentadas para este resultado permitem assegurar que as informações de planejamento do projeto foram documentadas, organizadas e relacionadas entre si, de forma a comporem o plano de projeto?	
AD - Na fase de Análise, a atividade "Realizar Planejamento do Projeto" permite que seja realizado todo o planejamento do projeto de acordo com a necessidade da organização na qual será aplicado, possibilitando a documentação de todas as ações que posteriormente serão avaliadas (fase de Avaliação) com auditorias para garantir que todas as ações planejadas foram executadas	T
	(T,L,P,N,NA)
GPR 11. A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados. As evidências apresentadas para este resultado permitem assegurar que a viabilidade do projeto foi avaliada, a partir dos objetivos do projeto e dos recursos financeiros, técnicos, humanos, bem como das restrições impostas pelo	
AD - Na fase de Entrada, durante as atividades "Identificar a Necessidade" e "Coletar Requisitos" é possível verificar a viabilidade do projeto de acordo com as informações coletadas com o cliente.	T
	(T,L,P,N,NA)
GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido. As evidências apresentadas para este resultado permitem assegurar que há registro de que todos os interessados tomaram conhecimento, revisaram e se comprometeram com o planejamento do projeto?	
AD - Na fase de Análise, durante a atividade "Realizar Planejamento do Projeto" é possível realizar a revisão e adequação do que foi planejado e comunicar as partes interessadas para que seja firmado o compromisso com o que foi definido no planejamento	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Na Figura 37 é exibido o GPR 13, onde é avaliado se existe gerenciamento ao longo do ciclo de vida do projeto, comparando o que foi planejado com o realizado.

Na Fase de Desenvolvimento existe a atividade *Aplicar Técnicas de Validação do Ciclo de Vida de Desenvolvimento* que permite o controle das atividades realizadas durante o desenvolvimento do projeto.

Na fase de Teste é realizada a verificação do produto desenvolvido com a finalidade de garantir que a implementação foi executada conforme planejado, identificando e registrando as inconformidades através da atividade *Registrar Erros*.

Na Fase de Liberação é realizada a atividade *Teste de Aceitação Mínima* como forma de garantir a qualidade da *Release* que será disponibilizada para o cliente.

Na fase de Avaliação são executadas as atividades *Definir Parâmetros de Avaliação*, *Executar Parâmetros de Avaliação* e *Reportar*

Inconformidades, permitindo que seja realizado o acompanhamento das atividades do planejamento e da correção das inconformidades reportadas.

No GPR 14 foi demonstrada a fase de Entrada como evidência, através da atividade *Termo de Abertura* que permite que sejam definidas as partes interessadas, informando as atividades acordadas para o desenvolvimento do projeto.

No GPR 15 não foram apresentadas evidências, pois este item não foi implementado no modelo.

Figura 37- Evidências – GPR 13 a GPR 15

GPR 13. O projeto é gerenciado utilizando-se o Plano do Projeto e outros planos que afetam o projeto e os resultados são documentados. As evidências apresentadas para este resultado permitem assegurar que o projeto foi gerenciado ao longo de seu ciclo de vida, comparando o planejado e o realizado?	
AD - Na fase de Desenvolvimento, a atividade "Aplicar Técnicas de Validação do Ciclo de Vida de Desenvolvimento" permite que seja verificado se o desenvolvimento está de acordo com o esperado	T
AD - Na fase de Teste é verificado se o produto desenvolvido está de acordo com o que foi especificado a atividade "Registrar Erros" permite que seja efetuado o registro das inconformidades que serão corrigidas antes de chegar ao cliente final	T
AD - Na fase de Liberação, a atividade "Teste de Aceitação Mínima" permite que seja realizada a verificação da <i>release</i> para identificar se os requisitos foram atendidos, se as funcionalidades estão de acordo com o planejamento.	T
AD - Na fase de Avaliação, as atividades "Definir Parâmetros de Avaliação", "Executar Parâmetros de Avaliação" e "Reportar Inconformidades", permitem que seja realizado o acompanhamento das atividades do planejamento, acompanhando a correção das inconformidades identificadas	T
	(T,L,P,N,NA)
GPR 14. O envolvimento das partes interessadas no projeto é gerenciado. As evidências apresentadas para este resultado permitem assegurar que o que foi planejado em relação ao envolvimento das partes interessadas foi seguido e se existe evidência de que os compromissos assumidos foram	
AD - Na fase de Entrada, a atividade "Termo de Abertura" é utilizada para definir as partes interessadas, informando também as atividades acordadas para o desenvolvimento do projeto	T
	(T,L,P,N,NA)
GPR 15. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento. As evidências apresentadas para este resultado permitem assegurar que ocorreram revisões nos marcos do projeto e em outros pontos estabelecidos no planejamento?	
Este item não foi implementado	N
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Na Figura 38 são exibidos o GPR 16 e GPR 17, no qual foi demonstrada a fase de Avaliação como forma de identificar as inconformidades e reportá-las aos responsáveis, acompanhando a correção até a sua conclusão.

Para corrigir os desvios em relação ao projeto foi apresentada a fase de Liberação na qual é realizada a atividade *Pesquisa de Satisfação* como forma de medir a qualidade do produto e do atendimento ao cliente, com isso

são identificadas e estabelecidas as melhorias necessárias para a correção e prevenção dos problemas.

Figura 38- Evidências – GPR 16 e GPR 17

GPR 16. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas.	
As evidências apresentadas para este resultado permitem assegurar que existem registros de identificação e análise dos problemas ocorridos no projeto e de que estes problemas foram tratados com os interessados?	
AD - Na fase de Avaliação é realizado o acompanhamento do projeto onde são identificadas as inconformidades através da aplicação de parâmetros de avaliação, onde os erros são reportados aos responsáveis e a correção destas falhas é acompanhada até a sua conclusão	T
	(T,L,P,N,NA)
GPR 17. 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.	
As evidências apresentadas para este resultado permitem assegurar que: (i) na monitoração do projeto foram identificadas ações corretivas, tanto para corrigir desvios em relação ao planejado, quanto para prevenir a repetição dos problemas identificados? (ii) estas ações foram acompanhadas e investigadas quanto à efetividade, antes de serem consideradas concluídas? (iii) os problemas e as ações corretivas foram repassados para níveis hierárquicos superiores, quando necessário, para garantir sua conclusão?	
AD - Na fase de Liberação, a atividade "Pesquisa de Satisfação" permite que seja realizada a medição da qualidade do produto e do atendimento ao cliente, com isso podem ser estabelecidas ações de melhoria para corrigir os problemas	T
AD - Na fase de Avaliação, as atividades "Definir Parâmetros de Avaliação", "Executar Parâmetros de Avaliação" e "Reportar Inconformidades" garantem o monitoramento das atividades realizadas em todo o projeto, identificando as inconformidades e reportando aos responsáveis para que sejam corrigidas. A correção é acompanhada para garantir que seja executada de acordo com o esperado.	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Na Figura 39 são avaliados os atributos de processo em relação ao processo GRP.

No AP 1.1 foram apresentadas como evidência todas as fases do processo, sendo que o produto de cada fase do modelo é utilizado na fase seguinte.

Na fase de Entrada é realizada a coleta dos requisitos que serão analisados, modelados para serem implementados e posteriormente são testados para garantir que o desenvolvimento foi realizado da forma adequada.

A realização do teste de aceitação mínima foi definida para garantir que o produto foi desenvolvido de acordo com o planejamento e de acordo com a necessidade do cliente.

Na fase de Avaliação é realizado todo o monitoramento da execução de todas as fases do projeto.

No AP 2.1 foi apresentada como evidência a fase de Análise, onde é realizado o planejamento das atividades a serem executadas.

Em relação à política organizacional do processo, não foram apresentadas evidências, pois este item não foi implementado no modelo.

Figura 39- Evidências – AP 1.1 e AP 2.1

Atributos de Processo	
Resultado esperado / evidências	
AP 1.1 O processo é executado	
RAP 1. O processo atinge seus resultados definidos.	
As evidências apresentadas para este resultado permitem assegurar que o processo transforma produtos de trabalho de entrada identificáveis em produtos de trabalho de saída, também identificáveis, permitindo atingir o propósito do processo?	
AD - Através da fase de Entrada são coletados os requisitos e definido o escopo do projeto.	T
AD - Nas fases de Análise e Modelo, é realizado o planejamento das atividades e definidas as funcionalidades no qual o sistema deve possuir	T
AD - Na fase de Desenvolvimento é definida a estrutura de código e realizada a implementação	T
AD - Na fase de Teste são definidos e aplicados os cenários de teste para a identificação de erros	T
AD - Na fase de Liberação é realizado o teste de aceitação mínima para garantir que o produto do projeto está de acordo com o esperado	T
AD - Na fase de Avaliação é realizado o monitoramento da execução das atividades do projeto	T
	(T,L,P,N,NA)
AP 2.1 O processo é gerenciado	
RAP 2. Existe uma política organizacional estabelecida e mantida para o processo.	
As evidências apresentadas para este resultado permitem assegurar: (i) que foi definida uma política organizacional estabelecendo as expectativas da organização para a execução do processo e se esta política é conhecida e de fácil acesso aos interessados? (ii) que a política organizacional foi atualizada, quando necessário? (iii) que a política organizacional tem respaldo da alta administração (por exemplo, por meio de aprovação da alta administração)?	
Este item não foi implementado	N
	(T,L,P,N,NA)
RAP 3. A execução do processo é planejada.	
As evidências apresentadas para este resultado permitem assegurar que existe um plano para a execução do processo?	
AD - Na fase de Análise é realizado o planejamento do projeto através da atividade "Realizar o Planejamento do Projeto"	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

Na Figura 40 foi apresentada a fase de Avaliação como forma de realizar o monitoramento da execução das atividades planejadas e apontar ações corretivas quando identificadas as falhas.

A fase de Análise também foi apontada como artefato direto através da atividade *Definir Planejamento do Projeto*, permitindo assegurar que o projeto foi executado conforme o planejado, através do registro de execução das atividades.

Figura 40- Evidências – AP 2.1

RAP 4. (Para o nível G). A execução do processo é monitorada e ajustes são realizados. As evidências apresentadas para este resultado permitem assegurar que o processo foi executado conforme planejado e se ações corretivas foram tomadas quando a execução do processo se desviou dos planos?	
AD - Na fase de Avaliação é realizado o monitoramento da execução das atividades realizadas o projeto	T
	(T,L,P,N,NA)
RAP 10. (Para o nível G). O processo planejado para o projeto é executado. As evidências apresentadas para este resultado permitem assegurar que o projeto foi conduzido a partir da execução do seu processo planejado, bem como se existem registros de execução das atividades do processo com base no seu planejamento?	
AD - Na fase de Análise é executada a atividade "Definir Planejamento do Projeto"	T
AD - Na fase de Avaliação é realizado o acompanhamento das atividades de acordo com o planejamento	T
	(T,L,P,N,NA)

Fonte: Adaptado de Softex (2009).

7.2 RESULTADOS OBTIDOS

Um dos resultados obtidos com este trabalho foi o processo desenvolvido com base no metamodelo proposto.

A elaboração deste processo permitiu a demonstração dos conceitos de qualidade empregados no metamodelo, adquiridos durante todo levantamento bibliográfico realizado desde o início do trabalho.

O processo foi desenvolvido para demonstrar a aplicação de todas as fases e atividades do metamodelo, podendo ser adaptadas de acordo com a necessidade da organização ao qual será aplicado.

No processo foram mantidas as sete fases definidas no metamodelo, e foram agregadas novas tarefas que auxiliam na execução de cada atividade proposta.

Todas as fases do processo atendem à demanda e ao desenvolvimento de projetos desde a sua concepção. Para isso foram definidas atividades e tarefas que permitem a identificação da necessidade transformando-a em requisitos que serão analisados e especificados.

Com a definição das regras de negócio são definidas as funcionalidades de acordo com a necessidade descrita pelo cliente, utilizando a rastreabilidade para verificar o relacionamento entre as funcionalidades,

garantindo que não exista conflito de requisitos que possa acarretar em problemas futuros, que por consequência gerem retrabalho.

Todo o levantamento inicial de requisitos e as atividades relacionadas anteriormente servem como uma preparação para a fase de desenvolvimento, sendo que para iniciar a implementação é necessário que as funcionalidades estejam bem definidas e a partir disso é iniciada a definição da estrutura de código, ou a estimativa das alterações a serem realizadas no código, caso se trate do atendimento da demanda.

Para garantir que o desenvolvimento será executado de acordo com o planejamento inicial, é essencial a aplicação de técnicas de validação do ciclo de vida de desenvolvimento, ficando a critério da organização a utilização da técnica que melhor se adéqua à sua necessidade.

Após a finalização da implementação é iniciada a fase de teste, onde são definidos os cenários de teste. Para agilizar esta atividade foi proposta a definição de uma ferramenta de armazenamento para que os cenários de teste sejam reaproveitados, tornando o teste automatizado e evitando o retrabalho, sendo que sem a utilização de uma ferramenta de armazenamento, os cenários teriam que ser criados a cada teste realizado.

O registro de erros é uma forma encontrada para garantir que todos os erros encontrados sejam repassados ao desenvolvimento para serem corrigidos e testados novamente após a efetuação das correções necessárias. Desta forma, diminui significativamente a possibilidade de que o sistema chegue ao cliente contendo erros.

Foi proposto o registro de alterações para o atendimento da demanda, considerando que desta forma o cliente pode ser informado das alterações no produto ao qual utiliza, servindo também de disseminação da informação na própria organização desenvolvedora.

Após a finalização desta etapa, é disponibilizada a *release*, na qual é realizado o teste de aceitação mínima para garantir que o produto está de acordo com os requisitos e as funcionalidades estabelecidas inicialmente.

Ao constatar que o produto está de acordo com o planejamento é liberado ao cliente e posteriormente é realizada uma pesquisa de satisfação

para identificar as possíveis falhas e os pontos que devem ser corrigidos ou melhorados.

Para finalizar o projeto é proposta a realização de uma reunião de encerramento para informar à partes interessadas a finalização do projeto e apontar algumas considerações que se julgue necessário.

Por fim, foi proposta a fase de Avaliação, que deve ser executada no decorrer de todas as fases do projeto como forma de garantir, através de auditorias periódicas, que todas as atividades do projeto estão sendo executadas.

Com isso são identificadas as falhas que devem ser reportadas ao responsável para que sejam corrigidas as inconformidades. O acompanhamento deve ser realizado para garantir que a correção será efetuada, e melhorias devem ser estipuladas para garantir que estas inconformidades não ocorram novamente.

Para a avaliação do metamodelo desenvolvido foi realizada uma comparação em relação ao modelo ideal. Para isso foi utilizada a planilha oficial do MA-MPS no qual foram avaliados os processos GRE e GRP, juntamente com os atributos de processo, ambos do nível de maturidade G do modelo MPS.BR.

Com essa avaliação percebeu-se que o metamodelo desenvolvido atende cerca de 60% dos itens estabelecidos no modelo ideal, sendo que foi atendida maior parte das atividades que agregam qualidade no desenvolvimento dos projetos de software.

Para chegar a este resultado foram considerados os requisitos de avaliação do nível G, sendo que para realizar uma avaliação em relação a todos os níveis do MPS.BR seria necessário a implantação do metamodelo e do processo em uma organização, realizando o acompanhamento periódico de todo o andamento do trabalho para chegar ao resultado ideal, utilizando todos os padrões de avaliação e todos os graus de classificação estabelecidos no MA-MPS.

Considerando que este seria um trabalho que tomaria um tempo muito grande, envolvendo vários recursos e um enorme esforço, sugere-se que seja abordado em trabalhos futuros.

A elaboração deste trabalho resultou na publicação de dois artigos no VI Congresso Sul Brasileiro de Computação (SULCOMP), sendo uma deles relacionado à automação de testes, cujo título é *Desenvolvimento de uma metodologia*

baseada na automação do processo de testes de software como estratégia para a garantia da cobertura funcional, elaborado juntamente com a acadêmica Camilla Damiani, e o outro artigo é relacionado ao desenvolvimento deste trabalho, agregando conhecimento em diversas áreas voltadas à Qualidade de Software.

8 CONCLUSÃO

Com a realização deste trabalho foi possível agregar um vasto conhecimento em relação aos conceitos abordados na Engenharia de Software, principalmente voltado para a Qualidade de Software, conceito este, que vêm se tornando cada vez mais presente na realidade das organizações brasileiras.

Com o decorrer dos anos, o setor de desenvolvimento de software tem crescido significativamente, e por consequência, aumentando a concorrência e os esforços organizacionais para manter a qualidade dos produtos desenvolvidos.

A falta de planejamento e de processos confiáveis que garantam a qualidade durante todo o ciclo de vida do projeto é um fator crítico que motivou o desenvolvimento deste trabalho, permitindo a elaboração de um metamodelo baseado no modelo MPS.BR, como forma de garantir a qualidade em projetos de software.

Com os estudos realizados percebeu-se que os modelos de referência existentes atualmente não se adequam à realidade das empresas brasileiras, sendo que estes modelos apresentam uma quantidade excessiva de atividades e tarefas que devem ser cumpridas para o atendimento de cada nível de maturidade.

Para a aplicação de modelos como o CMMI é necessário o envolvimento de vários recursos, além de um esforço enorme e um custo elevado, considerando que sua aplicação é voltada para projetos complexos e de longa duração.

O modelo MPS.BR foi desenvolvido a partir da necessidade de aplicar um modelo de processo eficiente mas que fosse adequado às pequenas e médias empresas, mas com a pesquisa realizada identificou-se que este modelo ainda não está de acordo com a realidade das empresas à qual se destina.

Levando em consideração o conhecimento de especialista adquirido vivenciando a realidade de uma empresa desenvolvedora de software,

constatou-se que a aplicação do MPS.BR envolve um esforço muito grande para que seja atingido cada nível de maturidade, onde todos os processos e grupos de trabalho são submetidos à uma rigorosa avaliação.

Com isso o processo de aplicação se torna longo e cansativo, podendo inclusive prejudicar a organização caso não seja aplicado adequadamente.

Pensando nesta realidade, foi desenvolvido um metamodelo baseado no MPS.BR, utilizando os principais conceitos de qualidade estudados, como forma de atender a necessidade das empresas brasileiras através da eliminação de atividades julgadas como desnecessárias e da inclusão de novas atividades e tarefas que tornam o processo adaptável de acordo com a organização na qual será aplicado.

Para garantir a qualidade na execução dos processos, foram incluídas no metamodelo, sete fases interdependentes que propõe a realização do planejamento do projeto, entrada e processamento de requisitos, aplicação de técnicas de validação do ciclo de vida de desenvolvimento, definição de ferramentas para automatização de testes como forma de evitar o retrabalho, realização de teste de aceitação mínima, pesquisa de satisfação para medir a qualidade do produto desenvolvido, e avaliação de todas as atividades com o objetivo de identificar e corrigir as falhas, prevenindo a ocorrência de erros futuros.

Para demonstrar a aplicação do metamodelo desenvolvido, foi elaborado um processo, no qual foram utilizadas todas as fases e atividades citadas anteriormente, sendo que a execução de todas estas fases agrega qualidade ao produto e ao projeto desenvolvido.

Através da análise de GAP realizada com a utilização da planilha de avaliação formal do MA-MPS, foi constatado que o metamodelo proposto atende 60% dos processos e atributos de processo avaliados, demonstrando a qualidade nas atividades e tarefas empregadas para o seu desenvolvimento.

Conclui-se que os objetivos propostos foram atendidos, levando em consideração todos os pontos apresentados anteriormente e os resultados obtidos após a realização do trabalho, sendo que a criação de um metamodelo

de qualidade foi motivada pela busca da melhoria contínua nos processos aplicados nos projetos de desenvolvimento de software.

REFERÊNCIAS

ALVES, Rafael Ferreira; VANALLE, Rosângela Maria. **Ciclo de vida de desenvolvimento de sistemas – Visão conceitual dos modelos clássicos, Espiral e Prototipação**. UNIMEP, 2001.

Disponível em:

<http://www.abepro.org.br/biblioteca/ENEGEP2001_TR93_0290.pdf>. Acesso em: 6 mar. 2012.

AMORIN, Fernanda Rodrigues dos Santos d'. **Engenharia de requisitos para métodos ágeis**. Recife: UFPE, 2008.

Disponível em: <<http://www.cin.ufpe.br/~in1020/arquivos/monografias/2008-1/monografiaFernandaDamorim.pdf>>. Acesso em: 05 abr. 2012.

BECK, Kent. **Programação extrema (XP) explicada: acolha as mudanças**. Porto Alegre: Bookman, 2004.

COCKBURN, Alistair. **Escrevendo casos de uso eficazes**. Porto Alegre: Bookman, 2005.

COUTO, Ana Brasil. **CMMI – Integração dos modelos de capacitação e maturidade de sistemas “Capability Maturity Integration”**. Ciência Moderna, 2007.

Disponível em:

<<http://www.conhecaolivreiro.com.br/pdf/livros/cultura/1861326.pdf>>. Acesso em: 03 mai. 2012.

DIAS, André Felipe. **Por que Investir em Melhoria de Processos?** PRONUS Engenharia de Software, 2008. 5p.

Disponível em:

<http://www.pronus.eng.br/artigos_tutoriais/processo_desenvolvimento/melhoria_processo.php?pagNum=4>. Acesso em: 20 set. 2011.

FERREIRA, Wilker Felix. **MPS.BR – Um estudo do Modelo MPS.BR como benefício para as pequenas e médias empresas.** Ceres, 2009. 69p.

Disponível em:

<http://www.softwarepublico.gov.br/file/17234990/MONOGRRAFIA_WILKER_>. Acesso em: 15 out. 2011.

FONTOURA, Lisandra Manzoni; HARTMANN, Júlio; PRICE, Roberto Tom. **Metamodelo para Adaptação de Processos de Software com base em Riscos.** Rio Grande do Sul: UFRGS, 2006.

Disponível em:

<http://www.inf.ufrgs.br/amadeus/download/fontoura_ideas2006.pdf>. Acesso em: 27 mai. 2012.

GEVAERD, Guilherme de Sá. **Uma Ferramenta de Avaliação de Maturidade de Processo de Software no CMMI e MPS.BR.** Joinville: UDESC, 2009.

Disponível em: <<http://www.pergamumweb.udesc.br/dados-bu/000000/000000000000D/00000D6D.pdf>>.

Acesso em: 03 mai. 2012.

GOMES, Marcelo Rodrigues. **Proposta de um processo de gerenciamento de requisitos para uma empresa desenvolvedora de software com foco na satisfação do Nível G do MPS.BR.** Cascavel: UNIOESTE, 2010.

Disponível em: <[http://www.inf.unioeste.br/~tcc/2010/TCC-](http://www.inf.unioeste.br/~tcc/2010/TCC-Marcelo%20Gomes.pdf)

Marcelo%20Gomes.pdf>. Acesso em: 12 abr. 2012.

GUERRA, Ana Cervigni; COLOMBO, Regina Maria Thienne. . **Tecnologia da informação: qualidade de produto de software.** Brasília: Ministério da Ciência e Tecnologia, 2009.

GUSTAFSON, David A. **Teoria e problemas de engenharia de software.** Porto Alegre: Bookman, 2003.

HAUMER, Peter. **Eclipse Framework Composer.** 2007.

Disponível em:

<<http://www.eclipse.org/epf/general/EPFComposerOverviewPart1.pdf><http://homepage.dcc.ufmg.br/~genaina/ES/EPF/OverviewPart1.pdf>>. Acesso em: 20 mai. 2012.

INFOPÉDIA. **Enciclopédia e Dicionários Porto Editora.** Portugal, 2012.

Disponível em: <<http://www.infopedia.pt/contactos/contactos.jsp>>. Acesso em: 10 dez. 2012.

JUNIOR, Paulo Fernando Souza Rodrigues. **Uma proposta de apoio sistematizado à gerência de portfólio do MPS.BR**. Belém: UFPA, 2009. Disponível em: <http://www.spider.ufpa.br/publicacoes/tcc/tcc_paulojunior_2009.pdf>. Acesso em: 19 abr. 2012.

KULESZA, Uirá. **Linhas de processo de software: conceitos, técnicas e ferramentas**. Natal: UFRN, 2011. Disponível em: <<http://www.slideshare.net/uirakulesza/linhas-de-processos-de-software-minicurso-sbqs-2011>>. Acesso em: 25 mai. 2012.

LEITÃO, Michele Vasconcelos. **Aplicação de Scrum em ambiente de desenvolvimento de software educativo**. Recife: Escola Politécnica de Pernambuco – Universidade de Pernambuco , 2010. Disponível em: <http://scholar.googleusercontent.com/scholar?q=cache:A6hRdGm2Xo0J:scholar.google.com/+fam%C3%ADlia+crystal&hl=pt-BR&as_sdt=0>. Acesso em: 02 abr. 2012.

MALLMANN, Paulo Roberto. **Um modelo abstrato de gerência de software para metodologias ágeis**. São Leopoldo: UNISINOS, 2011. Disponível em: <http://btd.unisinis.br/tde_arquivos/1/TDE-2011-07-01T152637Z-1454/Publico/PauloRobertoMallmannComputacao.pdf>. Acesso em: 8 abr. 2012.

MARTINS, Paula Ventura; SILVA, Alberto Rodrigues – **Comparação de Metamodelos de Processo de Desenvolvimento de Software**. INESC-ID 2004. Disponível em: <http://scholar.googleusercontent.com/scholar?q=cache:E3gOKIITvhgJ:scholar.google.com/+metamodelo&hl=pt-BR&as_sdt=0,5>. Acesso em: 28 mai. 2012.

MARTINS, Vidal – **Processo unificado de desenvolvimento de software**. Companhia de Informática do Paraná-CELEPAR , 2009. Disponível em: <<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1227>>. Acesso em: 27 mar. 2012.

MIRANDA, Márcio Assis. **Avaliação do processo de gerência de configuração utilizando o modelo MPS.BR**. Minas Gerais: UFLA, 2008.

Disponível em:

<http://scholar.googleusercontent.com/scholar?q=cache:AyWDVf_NzrEJ:scholar.google.com/+monografia+mps.br&hl=pt-BR&as_sdt=0>. Acesso em: 22 abr. 2012.

NUNES, Vanessa Battestin. **Integrando gerência de configuração de Software, documentação e gerência de conhecimento em um ambiente de desenvolvimento de software**. Vitória, 2005.

Disponível em:

<<http://www.inf.ufes.br/~falbo/files/DissertacaoNunesVanessa.pdf>>. Acesso em: 10 nov. 2012.

OLIVEIRA, Paulo César. **EPF Composer – Eclipse Process Framework Composer**. Pernambuco : UFPE, 2009. 39p.

Disponível em:

<http://docs.google.com/viewer?a=v&q=cache:DXq4HoR1IjMJ:www.cin.ufpe.br/~if720/slides/pco_EPFCOMposer.ppt+epf+composer+em+portugu%C3%AAs&hl=pt-BR&gl=br&pid=bl&srcid=ADGEESHtN2WTaVQsQbBTGkjl5rSg6xeGJZWwDolBjHMfXhK1IM5PglzIF2F_EJSOqwyBI7sGOH_xtyAENRFPgk1olxRBpv2IUvnVmyBk-LfqRsqWafHoCQAj0YI8Uk01923RAgvvBQC-&sig=AHIEtbRrwHRYjuiJNrZJpq2jXU3mPflv1w>. Acesso em: 17 mai. 2012.

OMG – Object Management Group. **Software & System Process Engineering Meta-Model Specification**. 2008.

Disponível em: <<http://www.omg.org/spec/SPEM/2.0/PDF>>. Acesso em: 20 ago. 2012.

PLACIDO, Maridiane Correa. **Desenvolvimento de uma metodologia para o gerenciamento de projetos baseado na área de processo: Gerência de Projetos (GPR) do modelo de maturidade MPS.BR**. Criciúma: UNESC, 2010.

Disponível em: <<http://www.unesc.net/portal/capa/index/90>>. Acesso em: 04 jun. 2012.

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed São Paulo: McGraw-Hill, 2006.

PMI. **Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (Guia PMBOK)**. Pennsylvania, 2008.

Disponível em: <<http://www.pmi.org/en/PMBOK-Guide-and-Standards/Standards-Library-of-PMI-Global-Standards.aspx>>. Acesso em: 16 out. 2012.

ROCHA, Ana Regina Cavalcanti da; MALDONADO, José Carlos; WEBER, Kival Chaves. . **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001.

ROCHA, Ana Regina; MONTONI, Mariano; SANTOS, Gleison; MAFRA. **Estação TABA: Uma infra-estrutura para implantação do modelo de referência para melhoria de processo de software**. Rio de Janeiro: COPPE/EFRJ, 2005.

Disponível em: <http://www.cos.ufrj.br/~savio/Arquivos/SBQS-2005/sbqs2005_tabo_mps.pdf>. Acesso em: 11 mai. 2012.

SOFTEX. **MPS.BR - Guia de Avaliação**. 2012.

Disponível em:

<http://www.softex.br/mpsbr/_guias/guias/MPSBR_Guia_de_Avaliacao_2012.pdf>. Acesso em: 23 abr. 2012.

_____. **MPS.BR - Guia de Avaliação**. 2011.

Disponível em:

<http://www.softex.br/mpsbr/_guias/guias/MPSBR_Guia_de_Avaliacao_2011.pdf>. Acesso em: 10 nov. 2011.

_____. **MPS.BR - Guia de Implementação – Parte 4: Fundamentos para Implementação do Nível D do MR-MPS**. 2011.

Disponível em:

<http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_4_2011.pdf>. Acesso em: 17 nov. 2011.

_____. **MPS.BR - Guia de Implementação – Parte 5: Fundamentos para Implementação do Nível C do MR-MPS**. 2011.

Disponível em:

<http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_5_2011.pdf>. Acesso em: 24 nov. 2011.

_____. **WAMPS 2009 – V Workshop Anual do MPS**. 2009.

Disponível em:

<<http://www.softex.br/portal/softexweb/uploadDocuments/Softex%20WAMPS%202009%20Web.pdf#page=53>>. Acesso em: 24 nov. 2012.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Pearson Addison-Wesley, 2007.

WEBER, Kival Chaves; ROCHA, Ana Regina Cavalcanti da. **Qualidade e produtividade em software**. São Paulo: MAKRON Books, 1999.

APÉNDICE(S)

APÊNDICE A – ARTIGO CIENTÍFICO

Utilização do Framework EPF Composer para o desenvolvimento de um metamodelo baseado no MPS.BR para a criação de processos aplicados à garantia da qualidade em projetos de software

Carina da Silva Cardoso¹, Gustavo Bisognin²

¹Universidade do Extremo Sul Catarinense (UNESC)
Caixa Postal 3167 – 88806-000 – Criciúma – SC – Brasil

²MSc. Professor do Curso de Ciência da Computação - Universidade do Extremo Sul Catarinense (UNESC)

Caixa Postal 3167 – 88806-000 – Criciúma – SC – Brasil

cary_cardoso@hotmail.com, gbisog@gmail.com

Abstract. *The lack of quality computer systems currently in the market is directly related to the quality of the processes they undergo. In order to improve the quality of software development, consultants and engineers in the area, proposing the creation of metamodels that add significant knowledge in defining the processes of software development. The development of metamodels aims to ensure quality in software projects. The implementation of the Quality Assurance Plan permits the realization of prevention of rework to correct faults, the backlog and abandonment activities essential to the smooth running of projects.*

Resumo. *A falta de qualidade dos sistemas computacionais presentes atualmente no mercado, está diretamente relacionada com a qualidade dos processos a que são submetidos. No intuito de aumentar a qualidade dos no desenvolvimento de software, consultores e engenheiros da área, propõem a criação de metamodelos que agregam um conhecimento significativo na definição dos processos de desenvolvimento de software. A elaboração de metamodelos tem o objetivo de garantir a qualidade nos projetos de software. A aplicação do plano de Garantia da Qualidade permite a realização da prevenção do retrabalho para a correção de falhas, do acúmulo de trabalho e do abandono de atividades essenciais para o bom andamento dos projetos.*

1. Introdução

A utilização de sistemas computacionais tem demonstrado um crescimento significativo no decorrer dos anos, não somente no Brasil, mas no mundo todo.

A falta de planejamento na estimativa do esforço necessário em projetos é um problema que ocorre com frequência, prejudicando o nível de previsibilidade não

somente do esforço a ser realizado, como também dos custos envolvidos, dificultando a comparação entre o que foi estimado e o que foi realizado.

Estudos sobre a qualidade no setor de software brasileiro mostraram a necessidade de um esforço significativo capaz de aumentar a maturidade dos processos de software nas empresas. (GUERRA, 2009, p.41).

A falta de qualidade dos sistemas computacionais presentes atualmente no mercado, está diretamente relacionada com a qualidade do processo a que é submetido.

No intuito de aumentar a qualidade dos processos de desenvolvimento de software, consultores e engenheiros da área, propõem a criação de metamodelos que agregam um conhecimento significativo na definição dos processos de desenvolvimento de software.

De acordo com Softex (2011), durante a avaliação de Garantia da Qualidade são identificados problemas e não conformidades relacionadas à correção ou melhoria de processos e trabalhos executados que se originam quando há desvios entre o esperado e o que foi realizado.

O plano de Garantia da Qualidade deve ser aplicado em todo ciclo de vida dos projetos de software, com a intenção de evitar possíveis falhas e por consequência, prevenir o retrabalho ocasionado pela correção dessas falhas.

O fato de implantar processos eficientes no desenvolvimento de projetos de software permite que os projetos sejam gerenciados com maior rigor, sendo que não haverá dúvidas sobre o que deve ou não ser feito.

Dentro do contexto apresentado, encontrou-se a possibilidade de desenvolver um metamodelo baseado no modelo MPS.BR, utilizando conceitos e técnicas que garantam a qualidade no desenvolvimento dos projetos de software.

Com isso, foi elaborado um processo baseado no metamodelo proposto, no qual foi aplicado um método de avaliação com o intuito de determinar o percentual de aderência do processo em relação a um modelo ideal, neste caso, o modelo MPS.BR.

Para garantir a qualidade na execução dos processos, foram incluídas no metamodelo, sete fases interdependentes que propõe a realização de atividades que agregam qualidade aos projetos quando aplicadas.

Para demonstrar a aplicação do metamodelo desenvolvido, foi elaborado um processo, no qual foram utilizadas todas as fases e atividades citadas anteriormente, sendo que a execução de todas estas fases agrega qualidade ao produto e ao projeto desenvolvido.

Através da análise de GAP realizada com a utilização da planilha de avaliação formal do MA-MPS, foi constatado que o metamodelo proposto atende 60% dos processos e atributos de processo avaliados, demonstrando a qualidade nas atividades e tarefas empregadas para o seu desenvolvimento.

Considerando todos os pontos apresentados anteriormente e os resultados obtidos após a realização do trabalho, sendo que a criação de um metamodelo de qualidade foi motivada pela busca da melhoria contínua nos processos aplicados nos projetos de desenvolvimento de software.

2. Modelo MPS.BR

O MPS.BR, ou seja, Melhoria de Processo do Software Brasileiro é um programa criado em dezembro de 2003, coordenado pela Associação para Promoção da

Excelência do Software Brasileiro (SOFTEX) com o objetivo de melhorar a qualidade dos processos de softwares brasileiros, com duas importantes metas a serem alcançadas a médio e longo prazo.

O modelo MPS.BR, foi desenvolvido e é constantemente adaptado com a finalidade de adequar seus processos de desenvolvimento, aplicando conceitos da engenharia de software já existentes, buscando a adequação ao perfil de empresas com características diferentes, atingindo de micro à médias empresas, sejam elas públicas ou privadas.

Segundo Softex (2011), o MPS.BR é formado por três componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS). Estes componentes são baseados nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade em produtos de software, que são descritos por meio de guias e documentos.

A escala de maturidade é dividida em sete níveis, que tem início no nível G, progredindo até o nível A, onde cada um destes níveis de maturidade possuem propósitos de melhoria da qualidade com foco em uma determinada área. (SOFTEX, 2011).

Para chegar ao próximo nível, é necessário que todos os resultados esperados para aquele nível sejam atingidos. A divisão do processo em níveis, proporciona uma melhor implementação e avaliação, permitindo uma melhor visibilidade dos resultados obtidos para a melhoria dos processos em prazos mais curtos, mesmo sendo para micros, pequenas ou médias empresas.

“A adoção de um nível de maturidade significa a obtenção de um grau de melhoria nas competências para a realização de um conjunto de tarefas ou atividades onde todos os objetivos são atendidos.” (GEVAERD, 2009, p.27).

O modelo MPS.BR possui um grande potencial em relação a sua aplicabilidade em diferentes regiões, considerando as características das empresas brasileiras através da aplicação de técnicas, padrões e modelos de processo já conhecidos, mas que vêm sendo aperfeiçoados de acordo com as necessidades encontradas nas organizações.

3. Framework EPF Composer

O Eclipse Process Framework Composer (EPF Composer), é uma ferramenta de criação de processos de desenvolvimento de software que permite a criação, configuração e publicação de métodos (HAUMER, 2007, tradução nossa).

Segundo Oliveira (2009), o EPF Composer é um projeto Open Source da Eclipse Foundation, apoiado pela Internacional Business Machines (IBM) e é uma evolução do Software Process Engineering Metamodel (SPEM).

O EPF Composer permite que seja estruturada a promoção do reuso de frameworks de processos em pequenas unidades modulares (KULESZA, 2011).

Cada unidade modular utiliza guias específicos e artefatos de entrada e saída para a realização de tarefas que são definidas pelas unidades modulares.

Através da utilização de esquemas pré-definidos é possível criar processos de desenvolvimento de software e adaptá-los de acordo com as necessidades da organização desenvolvedora de software, proporcionando um amplo conhecimento ao profissional que o utiliza.

Esta ferramenta oferece soluções aos problemas que geralmente são encontrados pela equipe de desenvolvimento na aquisição e gestão dos métodos de processo, mantendo a centralização das informações e fornecendo a documentação necessária enquanto o trabalho é realizado.

O EPF Composer tem como base para o seu desenvolvimento o SPEM, que é um metamodelo utilizado para a modelagem de processos de software, através da definição de estereótipos da Linguagem de Modelagem Unificada (UML) e provê um metamodelo baseado em Meta-Object Facility (MOF) que é a tecnologia adotada pela Object Management Group (OMG) para a definição de metadados através de um conjunto de elementos de modelagem que permitem a aplicação de regras para a construção de metamodelos.

Com a utilização de estereótipos da linguagem UML foi desenvolvido um metamodelo representado através do diagrama de atividades, no qual é possível detalhar as etapas de desenvolvimento dos projetos, demonstrando suas tarefas e atividades.

3.1. Metamodelos

Metamodelos são conjuntos de regras e construtores utilizados na modelagem de processos (OLIVEIRA, 2009).

Através dos metamodelos são aplicados conceitos de modelos específicos, com o objetivo de executá-los através da definição de atividades, prazos e regras determinando de que forma será executado o processo de desenvolvimento.

Os metamodelos de metodologias ágeis estão despertando o interesse das organizações desenvolvedoras de software que buscam cada vez mais a qualidade e agilidade em seus processos (MALLMANN, 2011).

Cada vez mais a aplicação de metamodelos é utilizada com o objetivo de agregar eficiência na definição dos processos aplicados no meio organizacional, buscando satisfazer as normas e os padrões de qualidade durante a execução das atividades, tornando o processo mais confiável.

Com a utilização dos elementos de processo são estabelecidas as regras e a sequência na qual o processo será formado (FONTOURA; HARTMANN; PRICE, 2006).

O SPEM é um metamodelo que serve como base para a especificação de qualquer processo, através da ideia de que o processo é uma colaboração entre papéis de processo utilizados para a realização de atividades em entidades chamadas de produtos de trabalho, com a finalidade de atingir um objetivo específico.

Um metamodelo de processo é utilizado para a representação dos objetos e atividades envolvidas no processo de software, onde cada fase do processo é demonstrada de forma clara e simples permitindo que seja realizado o gerenciamento de todo o processo.

O metamodelo para Fluxo de Trabalho é utilizado para identificar os atores e as atividades realizadas dentro do processo, representando-os através de fluxos de trabalho, tarefas, objetos manipulados (dados, documentos, etc.), papéis e atores.

Com a utilização de metamodelos é possível aplicar regras de desenvolvimento ágil adaptando-as de acordo com a necessidade, permitindo que através do método de avaliação utilizado, seja monitorado todo o andamento do processo.

4. Desenvolvimento do metamodelo

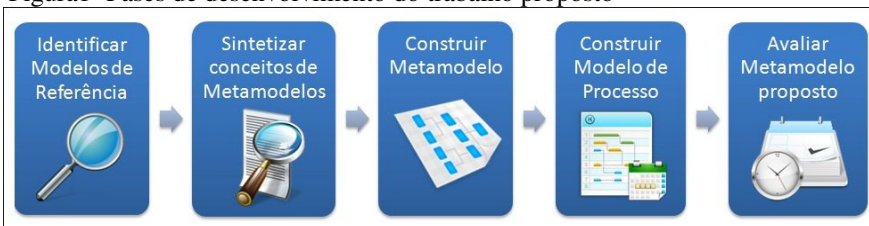
O metamodelo foi elaborado, tendo como base o modelo MPS.BR e a área de processo GQA do nível F de maturidade, utilizando também como fonte de conhecimento os modelos CMMI e PMBOK, possibilitando a identificação dos pontos considerados críticos em relação à realidade das empresas de software brasileiras.

Com a identificação dos pontos críticos, foi possível elaborar um metamodelo adequado às empresas brasileiras, eliminando atividades que geram esforço e investimento desnecessário, podendo causar sérios problemas para as organizações.

Levando em consideração os pontos citados anteriormente, foi desenvolvido um metamodelo genérico, ou seja, onde são definidas as fases e as atividades, mas a forma na qual as atividades serão executadas fica a critério do usuário, podendo ser aplicado para projetos e produtos. Para o desenvolvimento do metamodelo foi utilizada a ferramenta EPF Composer.

Para o desenvolvimento do metamodelo foram realizadas as etapas demonstradas na Figura 1.

Figura1- Fases de desenvolvimento do trabalho proposto



Fonte: Do autor.

Na primeira etapa de elaboração do metamodelo foi realizada a identificação de modelos de referência, no qual foi utilizado o MPS.BR como principal fonte de pesquisa, mas para obter uma visão geral em relação aos modelos de referência existentes, foram abordados no estudo os modelos CMMI e PMBOK.

Na segunda etapa de desenvolvimento do trabalho, foram sintetizados os conceitos dos metamodelos existentes, no qual foi possível identificar e comparar os padrões estabelecidos em cada modelo de referência, levando em consideração as atividades propostas e avaliando a sua aplicação de acordo com a necessidade das empresas.

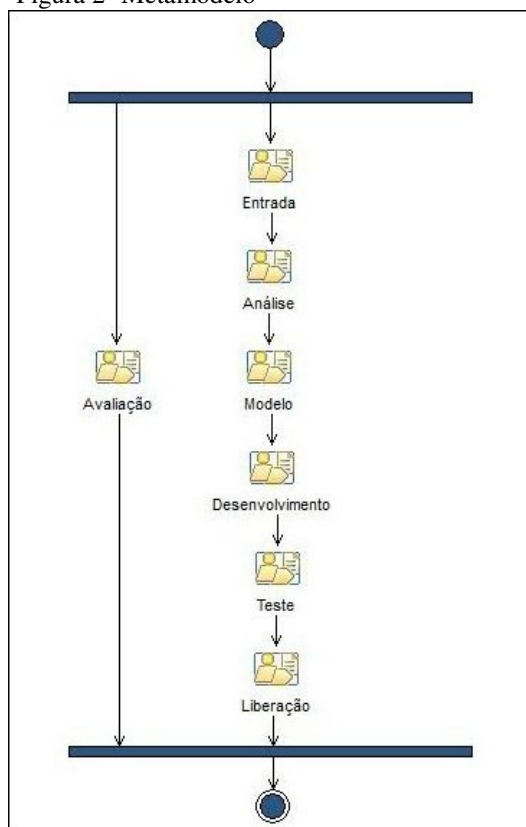
Na terceira etapa, foram reunidos os conceitos dos modelos de referência estudados anteriormente, para a elaboração de um metamodelo composto por 7 fases com suas respectivas atividades, ficando a critério do usuário a forma na qual cada atividade será executada, podendo ser aplicado para projetos e produtos (demanda).

Para o desenvolvimento do metamodelo foram utilizados os conceitos do metamodelo SPEM, que utiliza a unificação de várias metodologias para modelagem de processos, representando as fases e atividades através de estereótipos estabelecidos pela UML.

Através dos estereótipos estudados, foi possível identificar as fases de um processo, que são períodos que possuem um objetivo específico, e são compostos por

atividades, que por sua vez, podem ser subdivididas em tarefas, resultando em artefatos, ou seja, produtos de uma atividade.

Figura 2- Metamodelo



Fonte: Do autor.

Após o desenvolvimento de todas as fases do metamodelo, foi elaborada a fase de Avaliação, na qual são definidos os parâmetros de avaliação do projeto, que podem variar de acordo com a organização. É realizada a execução dos parâmetros estabelecidos, com o objetivo de identificar os pontos falhos, reportando as inconformidades e apontando as melhorias que devem ser efetuadas para a correção destas falhas.

Com base no metamodelo desenvolvido foi elaborado um modelo de processo, pensando na realidade encontrada atualmente nas empresas.

Para a criação do processo foram reunidas todas as atividades contidas no metamodelo, onde foram incluídas algumas tarefas que são opcionais durante a aplicação do processo, mas que de acordo com a pesquisa realizada durante todo o decorrer do trabalho, agregam qualidade ao processo quando aplicadas.

Identificou-se a necessidade da elaboração de um modelo de processo, como forma de demonstrar a aplicação dos conceitos de qualidade empregados no metamodelo, sendo que todas as atividades propostas tem o objetivo de garantir a

qualidade desde a coleta de requisitos, a análise e desenvolvimento do produto, até a conclusão o projeto.

4.1. Modelo de Avaliação baseado no Metamodelo proposto

Para efetuar a avaliação do metamodelo, primeiramente foi realizado o estudo do método de avaliação MA-MPS para compreender o seus principais conceitos, a forma como é aplicado e como são gerados os resultados da avaliação.

Com base nos estudos percebeu-se que para realizar uma avaliação formal, seria essencial a aplicação do metamodelo em uma organização. Mas como o tempo disponível não seria suficiente e para que essa aplicação fosse possível seriam necessários muitos recursos e um esforço significativo, optou-se por realizar como forma de avaliação a análise de GAP utilizando a planilha de avaliação formal do MA-MPS.

Com a utilização desta planilha foi comparado o metamodelo desenvolvido, em relação ao modelo ideal, analisando os processos GRE e GRP, que compõe o nível de maturidade G do MPS.BR, e foram analisados também os atributos de processo.

Realizando a avaliação através da análise do processo GRE é possível identificar se os requisitos do produto do projeto são gerenciados, permitindo a identificação de inconsistências em relação aos requisitos, ao planejamento do projeto e aos produtos de trabalho.

A análise do processo GRP permite identificar se foi realizado o planejamento do projeto em relação ao controle do andamento das atividades, permitindo a realização de correções quando há desvios no decorrer do projeto.

O preenchimento da planilha foi realizado através do resultado esperado e das evidências, sendo que, para cada questionamento foram apresentados Artefatos Diretos (AD) que comprovam o cumprimento de cada item. Os Artefatos Indiretos (AI) e as Afirmções (AF) não foram apresentados, sendo que seria necessário a aplicação do metamodelo em uma organização.

5. Conclusão

Com o decorrer dos anos, o setor de desenvolvimento de software tem crescido significativamente, e por consequência, aumentando a concorrência e os esforços organizacionais para manter a qualidade dos produtos desenvolvidos.

Os modelos de referência existentes atualmente não se adéquam à realidade das empresas brasileiras, sendo que estes modelos apresentam uma quantidade excessiva de atividades e tarefas que devem ser cumpridas para o atendimento de cada nível de maturidade.

Para a aplicação de modelos como o CMMI é necessário o envolvimento de vários recursos, além de um esforço enorme e um custo elevado, considerando que sua aplicação é voltada para projetos complexos e de longa duração.

O modelo MPS.BR foi desenvolvido a partir da necessidade de aplicar um modelo de processo eficiente mas que fosse adequado às pequenas e médias empresas,

mas com a pesquisa realizada identificou-se que este modelo ainda não está de acordo com a realidade das empresas à qual se destina.

Levando em consideração o conhecimento de especialista adquirido vivenciando a realidade de uma empresa desenvolvedora de software, constatou-se que a aplicação do MPS.BR envolve um esforço muito grande para que seja atingido cada nível de maturidade, onde todos os processos e grupos de trabalho são submetidos à uma rigorosa avaliação.

Com isso o processo de aplicação se torna longo e cansativo, podendo inclusive prejudicar a organização caso não seja aplicado adequadamente.

Pensando nesta realidade, foi desenvolvido um metamodelo baseado no MPS.BR, utilizando os principais conceitos de qualidade estudados, como forma de atender a necessidade das empresas brasileiras através da eliminação de atividades julgadas como desnecessárias e da inclusão de novas atividades e tarefas que tornam o processo adaptável de acordo com a organização na qual será aplicado.

Através da análise de GAP realizada com a utilização da planilha de avaliação formal do MA-MPS, foi constatado que o metamodelo proposto atende 60% dos processos e atributos de processo avaliados, comprovando a qualidade nas atividades e tarefas empregadas para o seu desenvolvimento.

Conclui-se que os objetivos propostos foram atendidos, levando em consideração todos os pontos apresentados anteriormente e os resultados obtidos após a realização do trabalho, sendo que a criação de um metamodelo de qualidade foi motivada pela busca da melhoria contínua nos processos aplicados nos projetos de desenvolvimento de software.

Referências

FONTOURA, Lisandra Manzoni; HARTMANN, Júlio; PRICE, Roberto Tom. Metamodelo para Adaptação de Processos de Software com base em Riscos. Rio Grande do Sul: UFRGS, 2006.
Disponível em: <http://www.inf.ufrgs.br/amadeus/download/fontoura_ideas2006.pdf>. Acesso em: 27 mai. 2012.

GEVAERD, Guilherme de Sá. Uma Ferramenta de Avaliação de Maturidade de Processo de Software no CMMI e MPS.BR. Joinville: UDESC, 2009.
Disponível em: <<http://www.pergamumweb.udesc.br/dados-bu/000000/00000000000D/00000D6D.pdf>>. Acesso em: 03 mai. 2012.

GUERRA, Ana Cervigni; COLOMBO, Regina Maria Thienne. . Tecnologia da informação: qualidade de produto de software. Brasília: Ministério da Ciência e Tecnologia, 2009.

HAUMER, Peter. Eclipse Framework Composer. 2007.
Disponível em:
<<http://www.eclipse.org/epf/general/EPFComposerOverviewPart1.pdf><http://homepages.dcc.ufmg.br/~genaina/ES/EPF/OverviewPart1.pdf>>. Acesso em: 20 mai. 2012.

KULESZA, Uirá. Linhas de processo de software: conceitos, técnicas e ferramentas. Natal: UFRN, 2011.

Disponível em: <<http://www.slideshare.net/uirakulesza/linhas-de-processos-de-software-minicurso-sbqs-2011>>. Acesso em: 25 mai. 2012.

MALLMANN, Paulo Roberto. Um modelo abstrato de gerência de software para metodologias ágeis. São Leopoldo: UNISINOS, 2011.

Disponível em: <http://bdtd.unisinos.br/tde_arquivos/1/TDE-2011-07-01T152637Z-1454/Publico/PauloRoberoMallmannComputacao.pdf>. Acesso em: 8 abr. 2012.

OLIVEIRA, Paulo César. EPF Composer – Eclipse Process Framework Composer. Pernambuco : UFPE, 2009. 39p.

Disponível em:

<http://docs.google.com/viewer?a=v&q=cache:DXq4HoR1IjMJ:www.cin.ufpe.br/~if720/slides/pco_EPFComposer.ppt+epf+composer+em+portugu%C3%AAs&hl=pt-BR&gl=br&pid=bl&srcid=ADGEESHtN2WTaVQsQbBTGkjl5rSg6xeGJZWwDolBjH MfXhK11M5PgIzIF2F_EJSOqwyB17sGOH_xtyAENRFPgk1olxRBpv2IUvnVmyBk-LfqRsqWafHoCQAj0YI8Uk01923RAgvvBQC-&sig=AHIEtbRrwhRYjuiJNrZJpq2jXU3mPflv1w>. Acesso em: 17 mai. 2012.

_____. MPS.BR - Guia de Avaliação. 2011.

Disponível em:

<http://www.softex.br/mpsbr/_guias/guias/MPSBR_Guia_de_Avaliacao_2011.pdf>. Acesso em: 10 nov. 2011.

_____. MPS.BR - Guia de Implementação – Parte 4: Fundamentos para Implementação do Nível D do MR-MPS. 2011.

Disponível em:

<http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_4_2011.pdf>. Acesso em: 17 nov. 2011.

_____. MPS.BR - Guia de Implementação – Parte 5: Fundamentos para Implementação do Nível C do MR-MPS. 2011.

Disponível em:

<http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_5_2011.pdf>. Acesso em: 24 nov. 2011