

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

ARIELA DAL TOÉ DA SILVA

**MODELAGEM DE UM PROCESSO PARA O GERENCIAMENTO E
DESENVOLVIMENTO DE REQUISITOS BASEADO NO MODELO BRASILEIRO
DE QUALIDADE DE PROCESSO MPS.BR**

CRICIÚMA, DEZEMBRO DE 2010

ARIELA DAL TOÉ DA SILVA

**MODELAGEM DE UM PROCESSO PARA O GERENCIAMENTO E
DESENVOLVIMENTO DE REQUISITOS BASEADO NO MODELO BRASILEIRO
DE QUALIDADE DE PROCESSO MPS.BR**

Trabalho de Conclusão de Curso apresentado
para obtenção do Grau de Bacharel em Ciência
da Computação da Universidade do Extremo
Sul Catarinense.


Orientador: Prof. MSc. Gustavo Bisognin

CRICIÚMA, DEZEMBRO DE 2010

ARIELA DAL TOÊ DA SILVA

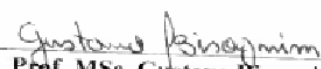
**Modelagem de um Processo para o Gerenciamento e Desenvolvimento de
Requisitos Baseado no Modelo Brasileiro de Qualidade de Processo
MPS.BR**

Submetido ao corpo docente do Curso de Ciência da Computação da
Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau
de Bacharel em Ciência da Computação.



Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:



Prof. MSc. Gustavo Bisognin (UNESC)
Orientador



Profa. MSc. Ana Claudia Garcia Barbosa(UNESC)

Dedico este trabalho a minha família,
especialmente a minha mãe, Gorette Dal Toé
da Silva e ao meu pai, Luiz Carlos da Silva; e,
também, ao meu irmão Alison Dal Toé da
Silva, minha cunhada, meu namorado e a
todos os meus amigos.

AGRADECIMENTOS

Agradeço primeiramente a Deus por iluminar o meu caminho durante esta caminhada e a minha família que esteve presente em todos os momentos difíceis dando força para a concretização dos meus objetivos.

Agradeço também ao meu orientador Gustavo Bisognin, por ter-me aceitado como orientanda e por toda a atenção e dedicação oferecida a este trabalho de conclusão de curso.

Ao meu irmão Alison e a minha cunhada Leila, que me apoiaram durante esta trajetória e ao meu namorado Elton, e todos os meus amigos e colegas, pelo inventivo e companheirismo concedidos.

“Compaixão significa não somente um pensamento de generosidade diante de quem sofre, mas representa também a inteligência na bondade, isto é, a compreensão da dor alheia.” **Bernardo Bertolucci.**

RESUMO

A presente pesquisa refere-se ao desenvolvimento de uma modelagem de processo para o Gerenciamento e Desenvolvimento de Requisitos baseado no modelo de referência para a melhoria do processo de software brasileiro MR-MPS.BR. Para a realização da pesquisa foi necessário o estudo das disciplinas de Engenharia de Software e Engenharia de Requisitos, além do entendimento dos modelos de qualidade CMMI, ISO e MPS.BR. De acordo com os estudos realizados, foram abordados processos, conceitos e finalidade das principais disciplinas que envolvem a engenharia requisitos. Dentro de um conjunto de resultados esperados, definidos pela área de processo de gerenciamento de requisitos do modelo MPS.BR, que foi escolhido por ser um modelo inovador baseando-se em normas consagradas e largamente validadas da área da engenharia de software. A composição do processo e sua publicação foi realizada com a utilização do *Eclipse Process Framework* (Epf Composer).

Palavras – Chave: Engenharia de Requisitos, Modelo de Qualidade MR-MPS.BR, Engenharia de Software.

ABSTRACT

This research is related to the development of a modeling process for the Management and Development Requirements based on the Brazilian reference model for software process improvement MR-MPS.BR. To fulfill the survey, it was necessary to study the subjects of Software Engineering and Requirements Engineering, and also the understanding of the CMMI quality models, ISO and MPS.BR. According to studies, were discussed processes, concepts and purpose of the main disciplines that involve the engineering requirements. Within a set of expected results as defined by the area of process management requirements MPS.BR model, which was chosen because it has been being an innovative model based on established and widely validated standards in the field of software engineering. The composition of the process and its publication were made using the Eclipse Process Framework (EPF Composer).

Key - words: Requirements Engineering, quality model MR-MPS.BR, Software Engineering

LISTA DE ILUSTRAÇÕES

Figura 1. Modelo de Caso de Uso.....	40
Figura 2. Componentes do MPS.BR.....	46
Figura 3. Composição do Trabalho Desenvolvido.....	61
Figura 4. Processo Desenvolvido.....	62
Figura 5. Fase de Iniciação	63
Figura 6. Fase de Elaboração.....	65
Figura 7. Fase de Construção.....	66
Figura 8. Fase de Entrega de Requisitos.....	67

LISTA DE SIGLAS

BID	Banco Interamericano de Desenvolvimento
CA	Consultoria de Aquisição
EA	Enterprise Architect
ER	Engenharia de Requisitos
FINEP	Financiadora de Estudos e Projetos
GR	Gerenciamento de Requisitos
GRE	Gerência de Requisitos
GRP	Gerência de Projetos
IA	Instituições Avaliadoras
IOGE	Instituição Organizadora de Grupos de Empresas
ISO	Organização Internacional para Padronização
MCT	Ministério da Ciência e Tecnologia
MPS.BR	Melhorias de Processo de Software Brasileiro
PA	Área de Processo
SEBRAE	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVO GERAL.....	13
1.2 OBJETIVOS ESPECIFICOS	13
1.3 JUSTIFICATIVA	14
2 ENGENHARIA DE SOFTWARE	18
2.1 Conceito de Software	18
2.2 História da engenharia de Software.....	20
2.3 MODELOS DE PROCESSO DE SOFTWARE	21
2.3.1 Modelo em Cascata.....	22
2.3.2 Modelo Desenvolvimento Evolucionário	23
2.3.3 Engenharia de Software Baseada em Componentes	24
2.3.4 Modelo Iterativo Incremental.....	26
2.3.5 Modelo Espiral.....	27
3 ENGENHARIA DE REQUISITOS	30
3.1 ELICITAÇÃO DE REQUISITOS	31
3.2 ESTUDO DA VIABILIDADE	34
3.3 VALIDAÇÃO DE REQUISITOS.....	35
3.4 DESENVOLVIMENTO DE REQUISITOS	36
3.4.1 Visão dos Modelos Tradicionais.....	37
3.4.2 Visão dos Modelos Ágeis	40
3.4.3 Histórias dos Usuários.....	41
4 MODELOS DE PROCESSO DE QUALIDADE.....	42
4.1 ISO.....	42
4.2 CMMI.....	43
4.3 MPS.BR.....	44
5 MODELO DE QUALIDADE MPS.BR	46
5.1 NÍVEIS DE MATURIDADE DO PROCESSO.....	48
5.2 ÁREAS DE PROCESSO DO MODELO	49
5.2.1 Gerência de Projetos.....	50
5.2.2 Gerência de Requisitos	51
6 TRABALHOS CORRELATOS	58
6.1 APLICAÇÃO DE MÉTRICAS DE SOFTWARE NO MÉTODO SCRUM DA METODOLOGIA ÁGIL	58
6.2 ENGENHARIA DE SOFTWARE ORIENTADA AOS PROCESSOS	58
6.3 DEFINIÇÃO DE PROCESSOS EM UM AMBIENTE DE DESENVOLVIMENTO DE SOFTWARE.....	59
6.4 ENSINO DA ENGENHARIA DE REQUISITOS POR MEIO DE UM AMBIENTE COLABORATIVO.....	60

7 DESENVOLVIMENTO DE UMA MOTODOLOGIA BASEADA NO NÍVEL G DO MPS.BR PARA A ÀREA DE GRE.....	61
7.1 METODOLOGIA DESENVOLVIDA	61
CONCLUSÃO.....	68
REFERÊNCIAS	71
APÊNDICE A – VISUALIZAÇÃO DO PROCESSO PRINCIPAL NO FRAMEWORK EPF COMPOSER	74
APÊNDICE B – VISUALIZAÇÃO DA COMPOSIÇÃO DA FASE DE INICIAÇÃO ...	75
APÊNDICE C - ARTIGO.....	76

1 INTRODUÇÃO

O processo de engenharia de requisitos é o responsável por transformar requisitos de usuário em requisitos de software. Estes são a base a partir da qual a qualidade é medida, sendo possível afirmar, portanto, que a inconformidade apresentada por este tipo de requisito denota falta de qualidade.

Atualmente, a indústria de software tem demonstrado um interesse notável na engenharia de requisitos, fato que evidencia claramente a necessidade de se entender o que se deseja construir antes de começar a fazê-lo.

O alto custo do esforço despendido para o entendimento do problema é um excelente indicador para a abordagem de uma gerência efetiva, controlada e adequada à rápida compreensão dos requisitos de usuários de software.

O modelo de avaliação de maturidade do processo de desenvolvimento Modelo de Processo de Software Brasileiro (MPS.BR), considera o gerenciamento de requisitos como uma das principais etapas para a aquisição da qualidade em desenvolvimento de sistemas de software. Para isso, é de fundamental importância que a empresa que deseja aumentar a qualidade de seus produtos esteja com um processo de engenharia de requisitos largamente implementado e institucionalizado nas equipes de desenvolvimento.

Para se alcançar aderência à área de processo Gerência de Requisitos (GRE) é necessário que os requisitos tenham sido definidos e que a empresa também possua seus processos de desenvolvimento de requisitos definidos.

Um dos principais problemas das empresas que desenvolvem software é a falta de processos que abordem o gerenciamento adequado dos requisitos de usuário e a sua rastreabilidade com os requisitos de software. Em virtude da falta de conhecimento e de entendimento dos requisitos existentes no software, muitos usuários acabam solicitando alterações e criações que não estão de acordo com os requisitos do software.

Essa abordagem inadequada ocasiona um desperdício de mão de obra, pois a empresa perde tempo ao disponibilizar uma pessoa para fazer uma análise dos requisitos do software a fim de verificar a possibilidade e a viabilidade de se desenvolver a solicitação do cliente. Ao investir esse tempo, muitas alterações necessárias ficam pendentes, visto que o tempo se torna escasso para a execução de rotinas necessárias no sistema.

Esse problema tende a ser potencializado no decorrer dos anos em que ocorre um aumento considerável do acúmulo de trabalho (*backlog*), gerado em razão do elevado número de alterações e atividades, fazendo com que muitas delas, fiquem na fila de atendimento.

Motivado pelos problemas acima apresentados, o presente estudo propõe, para o gerenciamento e desenvolvimento de requisitos de software, um modelo de processo aderente ao modelo de referência MPS.BR visando à minimização do impacto causado pela falta de entendimento dos papéis responsáveis pela elicitação dos requisitos a nível de usuário no processo produtivo de software.

1.1 OBJETIVO GERAL

Modelar um processo para o gerenciamento e desenvolvimento de requisitos baseado na área de processo GRE do Modelo Brasileiro MPS.BR

1.2 OBJETIVOS ESPECIFICOS

Os objetivos específicos do projeto são baseados nos seguintes itens:

- a) Estudar a disciplina Requisitos;
- b) Compreender o modelo de referência MPS.BR;

- c) Desenvolver um modelo de processo baseado na área de processo GRE do MPS.BR;
- d) Definir métodos, técnicas e padrões para o desenvolvimento e gerenciamento de requisitos de software.

1.3 JUSTIFICATIVA

Segundo Pressman (2005) a qualidade de um software é a conformidade em que os requisitos funcionais e de desempenho são declarados. Quando um software é desenvolvido por profissionais, espera-se que ele siga os padrões de desenvolvimento já documentados. Logo, para se ter um software de qualidade seus requisitos deverão estar bem definidos e documentados. Sendo assim, quando estes não estão de acordo com os requisitos é sinônimo de falta de qualidade.

O processo de gestão de requisitos de software é formado basicamente por dois importantes processos no decorrer de sua criação, sendo eles: o processo de Engenharia de Requisitos (ER), fundamentado pela coleta e documentação dos requisitos do software; e o de Gerenciamento de Requisitos (GR), responsável pelo gerenciamento da evolução desses requisitos.

Ainda segundo Pressman (2005) a ER se divide em cinco etapas: o estudo da viabilidade do software, a obtenção e a análise de requisitos, a especificação e documentação de requisitos, a validação e, ainda, o gerenciamento de requisitos.

Onde o estudo da viabilidade do software é fundamentado na análise do sucesso do projeto, nos benefícios que trarão a organização, na viabilidade de criar integração entre os softwares que já estão em uso, à questão financeira e os prazos.

Na etapa de obtenção e análise dos requisitos, é preciso coletar todas as necessidades do usuário, pois, dessa forma, ao iniciar o projeto, o engenheiro já terá previamente definido todas as necessidades e os requisitos do software. É importante ainda que já se conheça os interessados no software e que participarão da execução do projeto.

De acordo com Sommerville (2007) os requisitos são divididos em dois tipos, sendo esses: Requisitos do Usuário e Requisitos do Software. Os primeiros são claramente representados, sem a utilização de termos técnicos por meio de diagramas; eles podem ser compreendidos por gerentes de cliente, por usuários finais do software e por pessoas que possuem conhecimento na área do negócio. Já as funções, as restrições e os serviços dos requisitos de software são descritos detalhadamente, sendo especificadas as funcionalidades do software, e definindo-se exatamente o que será desenvolvido.

Na fase de especificação e documentação, os requisitos são documentados de forma clara e objetiva. As especificações resultantes desse processo podem ser escritas em linguagem natural ou, até mesmo, em linguagem estruturada. Após realizar-se a documentação do projeto, inicia-se a fase de validação dos requisitos, na qual é verificado se eles estão de acordo com o solicitado pelo cliente, pois, caso algum deles não esteja de acordo, posteriormente isso poderá causar transtornos, além do alto custo a ser investido na correção. A fim de evitar-se esse tipo de problema, utilizam-se algumas técnicas de validação, tais como a revisão de todos os requisitos e a criação de um protótipo do software. No processo de Gestão de Requisitos existe outra fase muito importante, que é a fase do Gerenciamento de Requisitos.

Segundo Sommerville (2007) o GRE é o processo responsável por compreender e controlar as mudanças a serem feitas nos requisitos do software. Para que esse gerenciamento seja feito, faz-se necessário, muitas vezes, que haja um acompanhamento individual dos

requisitos e das ligações entre os requisitos dependentes, sendo que tal gerenciamento deve ser iniciado assim que a primeira versão do documento de requisitos for disponibilizada.

Para facilitar o processo de identificação, de controle e de rastreamento dos requisitos, utiliza-se um conjunto de atividade que compõe o processo de Gerência de Requisitos, no qual estão definidos no modelo Brasileiro de Qualidade de Software (MPS.BR).

Atualmente, no Brasil, 73% das empresas que desenvolvem software são conhecidas como pequenas e médias empresas; por esse motivo, poucas delas têm adotado um modelo de referência para o desenvolvimento de seus projetos de software. Já, após a criação do Modelo de Processo de Software Brasileiro (MPS.BR), muitas empresas começaram a aderi-lo.

Em 2009, foi realizada uma pesquisa com 174 empresas que adotaram o modelo MPS.BR eo fator que mais despertou a atenção dos empresários foi o fato de que, após a adesão, o nível de satisfação dos clientes aumentou em até 50%; atribui-se esse aumento à comunicação de estimativas de prazos para a realização do projeto solicitado. Após a avaliação, a perspectiva é de que, até dezembro de 2010, aumente, para 300, o número de empresas que adotarão o modelo mencionado.

Por existir mais de um nível de processo no Modelo MPS.BR, será abordado, no presente estudo, apenas um dos processos referente ao nível de maturidade G: a Gerência de Requisitos; sendo este formado também pela Gerência de Projetos, temática que não será trabalhada.

A partir do Modelo MPS.BR, propõe-se a criação da uma modelagem de processo para o gerenciamento e desenvolvimento de requisitos de software, a qual tende a diminuir o número de solicitações feitas pelo cliente e assim evitando o cancelamento e reprovação das

solicitações. Esse modelo tem o intuito de efetuar uma reciclagem das solicitações recebidas, minimizando, dessa forma, o desperdício de tempo e dinheiro.

2 ENGENHARIA DE SOFTWARE

Além de abordar a evolução e a criação de software, o presente capítulo apresentará algumas das dificuldades encontradas no decorrer do desenvolvimento de software bem como demonstrar o quão importante foi a criação da disciplina de engenharia de software, por meio da qual os softwares passaram a ter maior reconhecimento de mercado e, assim, diminuindo gradativamente a ocorrência de inconsistência no decorrer do processo de utilização dos softwares.

2.1 CONCEITO DE SOFTWARE

Segundo Pressman (2007), os softwares assumem um papel duplo: ora é o produto, ora é o que veicula o produto. Enquanto produto, o software disponibiliza um potencial de computação que pode ser freqüentemente encontrado tanto quanto em telefones celulares ou, até mesmo, em computadores de grande porte. O software é o transformador da informação. Quando analisado por veículo de entrega do produto, o software age como uma base de controle do computador, possibilitando a comunicação da informação (rede) e, também, a criação e o controle de ferramentas e ambientes desenvolvedores de software.

Deve-se projetar e implementar um software de modo que ele possa ser reutilizado no desenvolvimento de outros, como ocorre na reutilização das interfaces gráficas que podem ser reutilizadas pelo engenheiro de software na criação de novas aplicações.

Pressman (2007) afirma que atualmente os engenheiros de software estão encontrando dificuldades nas sete categorias de softwares de computadores listadas abaixo:

- a) Sistemas: são os programas responsáveis por servir a outros programas.

- b) Aplicação: são as aplicações responsáveis pela especificidade de determinado negócio.
- c) Científico e de engenharia: são aplicações modernas utilizadas na área científica da engenharia que estão se afastando os algoritmos numéricos convencionais, ou seja, estão adotando um novo formato de desenvolvimento.
- d) Embutidos reside dentro de um sistema ou produto usado para implementar e controlar as características e as funções específicas de determinado produto ou sistema;
- e) Linhas de produtos: são os programas ou aplicações projetados para fornecer capacidade específica para clientes diferentes. Um exemplo deste tipo software são os sistemas de controle de estoque.
- f) Aplicações da web: são os softwares acessados via web e que compreendem uma gama de amplas aplicações.
- g) Inteligência artificial: são os programas que fazem uso de algoritmos não numéricos, responsáveis por resolver programas complexos que não são compreendidos facilmente por computadores ou à análise direta.

Um projeto de software é iniciado em razão da necessidade de determinado negócio, sendo que ela pode ser proveniente de várias situações, tais como: correção de defeito de aplicações existentes, adaptação de um sistema ligado para mudar o ambiente de negócio, extensão das funções e características de uma aplicação existente ou desenvolvimento de um produto novo. Em muitas situações, as necessidades de um novo projeto de software são expressas informalmente, partindo de uma simples conversa (PRESSMAN, 2007).

2.2 HISTÓRIA DA ENGENHARIA DE SOFTWARE

De acordo com Sommerville (1997), a engenharia de software é uma área de conhecimento composta por teorias, ferramentas, métodos e conjuntos necessários à geração de um produto de software. Ele afirma também que trata-se de um ramo da engenharia cujo foco principal é o desenvolvimento de software adequando assim os custos com sistemas de alta qualidade.

Segundo o mesmo autor, o conceito de engenharia de software surgiu durante a conferência da “Crise de Software”, ocorrida em 1968 e resultante da direta introdução de novos hardwares de computadores baseados em circuitos integrados, assim, os softwares que vinham sendo desenvolvidos não estavam de acordo com os recursos fornecidos pelos novos hardwares.

De acordo com Pressman (2007) a prática da engenharia de software foi adotada em virtude da preocupação existente na indústria de software, pela maneira em que os softwares estão sendo desenvolvidos.

A engenharia de software, além de ampliar as garantias relacionadas à qualidade do produto, é a área de conhecimento específico que auxilia o desenho de soluções de software, utilizado para responder a crescente complexidade e diversidade dos sistemas bem como as alterações nos requisitos durante o seu processo de desenvolvimento.

Laitinem (2000) cita que historicamente a engenharia de software começou a ser utilizada em projetos de desenvolvimento de grandes sistemas para as organizações estatais ou industriais.

Sendo a engenharia de software uma disciplina que ainda não atingiu a maturidade de outros ramos da engenharia, é importante a existência de processos que garantam qualidade e controle de desempenho de atividades nessa área. Sendo assim, o papel

dos recursos humanos intervenientes nesse processo é um fator decisivo para o sucesso dos projetos. Existem características desejáveis na sua gestão (BOOT, 1995):

- a) Planejamento estratégico para longo prazo, assegurando-se assim uma plataforma de orientação para todos os intervenientes.
- b) Empenho na organização, passando-se do mero cumprimento de regras e prazos para uma atitude de empenho e automotivação.
- c) Autogestão pelos intervenientes, cabendo, à gestão, a delegação de tarefas e a anulação de um controle sufocante.
- d) Perspectiva unitária, garantindo-se que os esforços individuais se conjuguem para um melhor desempenho em equipe e premiando-se o mérito individual.
- e) Promoção da existência de intervenientes flexíveis.

Em suma, pode-se afirmar que não existem processos de desenvolvimento de software mais ou menos corretos, há os processos de desenvolvimento mais ou menos adequados à complexidade do projeto pedido, ao tipo de equipe que o prática e ao tipo de utilização pretendido para o produto de software resultante.

2.3 MODELOS DE PROCESSO DE SOFTWARE

Os modelos são representações abstratas de um processo de software e cada um deles tem uma forma diferente de representar um processo, pois são obtidas perspectivas diferentes de software; assim as informações do processo são fornecidas de forma parcial. Eles também são chamados de modelos de processos genéricos ou, ainda, de paradigmas de processo: Modelo em Cascata, Modelo de Desenvolvimento Evolucionário e a Engenharia de Software Baseada em Componentes.

2.3.1 Modelo em Cascata

Segundo Pressman (2007) o modelo em cascata também é chamado de Ciclo de Vida Clássico do Software.

De acordo Sommerville (2007) o modelo em Cascata ou Ciclo de Vida do Software, proveniente da unificação das fases do processo de desenvolvimento de Software, foi o primeiro modelo a ser criado mediante os processos mais comuns da Engenharia de Software. Os principais estágios desse modelo são:

- a) **Análise e definição de requisitos:** são os serviços, as restrições e os objetivos do sistema, definidos por meio de uma pesquisa com os futuros usuários do software.
- b) **Projeto de sistema e software:** é o processo que envolve a definição dos requisitos que o software terá e também define os requisitos de hardware que serão necessários. Nessa fase, tem-se uma arquitetura geral do sistema envolvendo a identificação e a descrição das abstrações fundamentais do sistema de software.
- c) **Implementação e teste de unidade:** nesse estágio, o projeto de sistema é realizado como conjuntos de unidade. Após essas separações, realizam-se os testes para que seja verificado se as unidades estão de acordo com sua especificação.
- d) **Integração e teste de sistema:** nessa fase, as unidades do programa são integradas e testadas como um sistema completo para, assim, garantir que os requisitos do software sejam atendidos. O sistema é liberado para o cliente após serem feitas a realização de todos os testes e a verificação dos requisitos.

- e) Operação e manutenção: é considerada a etapa mais longa do ciclo de vida de um software, pois nela o sistema já está em uso pelo cliente e, ao ser utilizado, passam a surgir as solicitações de correções dos erros não identificados nas etapas anteriores. Nesse momento, são criados novos requisitos para o sistema, fazendo com que novas versões e/ou atualizações sejam liberadas para o cliente.

No modelo em cascata, todas as etapas devem ser executadas individualmente, sem que seja dado início a uma das etapas antes de a anterior ser concluída e devidamente documentada e certificada pelos membros do projeto de software.

2.3.2 Modelo Desenvolvimento Evolucionário

Segundo Pressman (2007) o modelo de desenvolvimento evolucionário é aquele em que o sistema evolui enquanto é desenvolvido, permitindo que os requisitos de negócio e de produto sejam mudados frequentemente à medida que o desenvolvimento prossegue, dificultando, dessa forma, a finalização do produto.

Sommerville (2007) afirma que, neste processo, ocorre o desenvolvimento de uma implementação inicial, pois os resultados são expostos aos usuários e, diante dos comentários destes, são geradas várias versões até que o sistema esteja, de fato, adequado

De acordo com o mesmo autor, existem dois tipos fundamentais de desenvolvimento evolucionário:

- a) Desenvolvimento exploratório: é o processo que trabalha com o cliente para que haja exploração dos requisitos antes da entrega do produto final; assim, são desenvolvidas primeiramente as partes do sistema que já foram compreendidas,

permitindo que o sistema evolua de acordo com as novas características designadas pelo cliente.

- b) Prototipação: tem por objetivo compreender os requisitos do cliente e, a partir disso, desenvolver os melhores requisitos para o sistema; dessa forma, o protótipo se concentrará na experimentação dos requisitos mal compreendidos.

Sommerville (2007) afirma que o desenvolvimento evolucionário torna-se mais eficaz que o modelo em cascata, pois atendendo as necessidades do cliente de forma imediata. A principal vantagem está na possibilidade de se desenvolver de forma incremental, ou seja, à medida em que o usuário compreender melhor seu problema, e isso refletirá diretamente no software.

O autor acrescenta ainda que a engenharia e o gerenciamento encontram-se dois pontos negativos em relação a este processo de desenvolvimento: O processo não é visível, porque os gerentes não conseguem medir o progresso do produto e como os produtos são desenvolvidos de forma rápida, torna-se inviável a produção da documentação. O outro problema se refere à freqüente má estruturação do sistema em razão das mudanças e incorporações contínuas.

2.3.3 Engenharia de Software Baseada em Componentes

Sommerville (2007) afirma que a Engenharia de Software Baseada em Componentes é reuso de programas que depende de componentes de software reutilizáveis e ainda dos *frameworks* de integração deles estes, os quais, por vezes, são sistemas comerciais independentes.

Segundo Sommerville (2007), os estágios de especificação dos requisitos e o estágio de validação do modelo em questão são semelhantes aos dos demais, porém o que o difere é a utilização do reuso. Os estágios deste modelo são:

- a) **Análise de componentes:** nesta fase, é feita uma busca pelos componentes para implementar as especificações dos requisitos. Na maioria das vezes, não existe correspondências exatas e, com isso, os componentes acabam fornecendo somente parte das funcionalidades necessárias.
- b) **Modificação de requisitos:** durante o estágio de modificação, os requisitos são analisados com base nas informações obtidas dos componentes. Após são realizadas as modificações nos componentes para a posterior liberação. Mas, caso as modificações não sejam necessárias, os componentes retornam ao estágio de análise.
- c) **Projeto de sistema com reuso:** neste estágio, é projetado um novo *framework* ou é reusado um já existente para o desenvolvimento do projeto do sistema. Os *frameworks* freqüentemente são baseados nos componentes reusados e nos projetos já desenvolvidos, mas, em alguns momentos, há a necessidade de se desenvolver um novo *framework* ainda que os componentes sejam reusados.
- d) **Desenvolvimento e integração:** nesta fase, quando um software não pode ser adquirido externamente ele é desenvolvido e seus componentes e sistemas COTS são integrados para criar-se o novo software. Neste modelo a integração dos sistemas pode ser parte do processo de desenvolvimento.

2.3.4 Modelo Iterativo Incremental

Segundo Pressmam (2007), a aplicação do Modelo Iterativo Incremental ocorre basicamente da mesma forma que no modelo em Cascata, ressaltando-se como diferença apenas o fato de a aplicabilidade ser feita de forma iterativa no primeiro. São aplicadas racionalmente seqüência lineares à medida que o tempo decorre; cada uma delas produz os incrementos do software. Quando é realizada uma incrementação no software, ao primeiro incremento dá-se o nome de núcleo do produto, ou seja, muitos requisitos básicos são satisfeitos, mas muitas características suplementares não são elaboradas.

Sommerville (2007) afirma que, após a identificação dos incrementos do sistema, os requisitos dos serviços a serem entregues no primeiro incremento são definidos detalhadamente e depois são desenvolvidos. No momento em que o primeiro incremento está sendo desenvolvido, pode ser dado início ao processo de análise do segundo incremento a ser desenvolvido, porém, o incremento em processo de desenvolvimento não poderá ser alterado durante o seu desenvolvimento.

De acordo com Pressmam (2007) o desenvolvimento incremental é útil quando não há mão de obra disponível para a implementação completa dentro do prazo comercial estabelecido para a entrega do projeto.

Para Sommerville (2007) o processo de desenvolvimento incremental tem as seguintes vantagens:

- a) O cliente não precisa esperar o término do desenvolvimento do sistema para poder utilizá-lo. Assim que os requisitos são desenvolvidos dentro de cada incremento, eles já podem ser disponibilizados ao cliente.
- b) O cliente pode usar os incrementos como protótipo e, assim, ganhar experiência, e conhecendo os incrementos posteriores.

- c) O risco de falha no projeto de sistema é relativamente inferior quando comparado a outros modelos.
- d) Os serviços de maior prioridade são desenvolvidos e entregues primeiramente; com isso, a probabilidade de falhas é pequena. Os incrementos posteriores são integrados aos principais e, assim, o processo de teste dos principais incrementos é mais intenso a fim de que falhas sejam evitadas.

Uma desvantagem do modelo incremental que a entrega de um incremento não pode conter mais de vinte mil linhas de código. Onde pode dificultar o mapeamento dos requisitos nos incrementos de tamanho adequado.

No modelo incremental, os requisitos não podem ser definidos detalhadamente enquanto o incremento ainda está em processo de desenvolvimento, pois isso pode dificultar a identificação dos recursos exigidos por todos os incrementos.

2.3.5 Modelo Espiral

Segundo Pressman (2007), o modelo espiral foi proposto originalmente por Boehm em 1988 como um modelo evolucionário de processo de software, pois combina a natureza iterativa da prototipagem com os aspectos controlados e sistemáticos do modelo em cascata.

O modelo espiral de desenvolvimento é um gerador de modelo de processo guiado por risco usado para guiar a engenharia de sistemas intensivos em software com vários interessados concorrentes. Ele tem duas principais características distintas. A primeira é uma abordagem cíclica, para aumentar incrementalmente o grau de definição e implantação de um sistema enquanto diminui seu grau de risco. A outra é um conjunto de marcos de ancoragem, para garantir o comprometimento dos interessados com soluções exequíveis e mutuamente satisfatórias para o sistema. (BOEHM, 2001, p. 45)

Pressman (2007) afirma que, neste modelo, o software é desenvolvido numa série de versões, sendo que, nas primeiras versões, o software pode ser classificado como um

modelo de papel ou como um protótipo. O modelo em questão é dividido em um conjunto de atividades de arcabouços definidas pela equipe de engenharia de software. Cada atividade de arcabouço representa um segmento do caminho da espiral. Na medida em que o processo é iniciado, a equipe de engenharia de software realiza as atividades indicadas, por um circuito em volta da espiral, em sentido horário, partindo-se do centro.

Segundo Pressman (2007), o primeiro circuito em volta da espiral pode resultar no desenvolvimento da especificação do produto. A passagem em torno da espiral pode ser usada para desenvolver um protótipo. Quando é concluída a base da espiral, onde consiste o processo de planejamento, muitas vezes podem-se ocorrer ajustes no plano do projeto, assim o custo e o cronograma são ajustados de acordo com o *feedback* recebido após a entrega ao cliente. E o gerente de projetos pode ajustar o número de interações planejadas de acordo com a necessidade para completar o software. O modelo em questão é adaptado para aplicações ao longo da vida do software.

Segundo Sommerville (2007), no decorrer do processo de desenvolvimento, podem ocorrer *loops*. A correção de cada loop do modelo espiral é dividida em quatro setores:

a) Definição de objetivos: é, nesta fase, que os objetivos do projeto são definidos.

Também são identificadas as restrições do processo e do produto; assim, cria-se um plano de gerenciamento e os riscos são relacionados.

b) Avaliação e redução de riscos: com a identificação dos riscos, é feita uma análise detalhada a fim de que providências sejam tomadas em relação à redução dos riscos.

c) Desenvolvimento e validação: após a realização da análise dos riscos, um modelo de desenvolvimento é escolhido com base nos riscos a serem encontrados durante o processo de desenvolvimento.

d) Planejamento: neste ponto, é realizada a revisão do projeto e caso haja *loop* uma decisão é tomada. Caso a decisão seja prosseguir ao próximo loop, são criados os planos para a fase seguinte do projeto.

Sommerville (2007) afirma que os processos de desenvolvimento de software podem ser aprimorados por meio de padronizações. Com isso, a diversidade de processos de software é reduzida ao longo da organização, fazendo com que a comunicação e a redução do tempo de treinamento sejam aprimoradas e o apoio, ao processo automatizado, torne-se mais econômico.

3 ENGENHARIA DE REQUISITOS

Neste capítulo, além de tratar-se da definição do requisito de software, serão apresentadas informações relevantes acerca da temática, tais como: sua evolução e no que se baseia sua criação, a necessidade de obtê-lo formalizada e detalhadamente e também as formas de representá-lo dentro de um projeto de software bem definido. Os requisitos de software são todas as necessidades funcionais e não funcionais que o software terá que constar para atender a necessidade do cliente final levando-se sempre em consideração o foco do negócio.

Segundo Pressman (2007) compreender os requisitos de um problema é um das tarefas mais difíceis de um engenheiro de software, mas, quando se pensa e se age, torna-se mais fácil o seu entendimento.

Pressman (2007) afirma que construir um projeto de software é desafiador e, ao mesmo tempo, criativo e prazeroso, tanto que, ainda que não possuam todas as informações necessárias, os desenvolvedores anseiam por dar início ao desenvolvimento.

Assim como as demais atividades do processo, a engenharia de requisitos precisa ser adaptada às necessidades do processo, do projeto, das pessoas envolvidas e, principalmente, do produto. Pois estas atividades são de suma importância o desenvolvimento do projeto.

A engenharia de requisitos é responsável por definir as funcionalidades, as características, o escopo e os requisitos envolvidos bem como documentar inteligivelmente tais informações, possibilitando, dessa forma, que os interessados no sistema possam acessá-las e compreendê-las.

As atividades desenvolvidas pela engenharia de requisitos não são realizadas apenas com o intuito de contribuir no atingimento de um nível considerável de qualidade de desenvolvimento de software; elas são primordiais na posterior mensuração da qualidade do produto, pois permitem que as características inicialmente esperadas sejam comparadas ao resultado do desenvolvimento, atestando, assim, o atendimento, ou não, das expectativas iniciais.

A engenharia de requisitos constrói uma ponte entre o projeto e a construção do software, por meio da qual a equipe analisa o contexto do software a ser realizado, identificando as necessidades específicas e o que a construção do projeto precisará para satisfazê-las.

Sommerville (2007) afirma que o objetivo da engenharia de requisitos é criar e manter os documentos dos requisitos do sistema. O processo geral da engenharia de requisitos é subdividido em quatro sub-processos de alto nível, sendo eles: a obtenção dos requisitos (elicitação), o estudo da viabilidade, a conversão dos requisitos em alguma forma padrão (especificação) e a verificação dos que realmente definem o sistema que o cliente necessita (validação).

3.1 ELICITAÇÃO DE REQUISITOS

Segundo Sommerville (2007), é no processo de elicitação e de análise dos requisitos que os engenheiros de software trabalham com a participação do cliente e com os usuários finais do sistema. Nesse momento, são identificados o domínio da aplicação, os serviços a serem fornecidos, o desempenho esperado para o sistema e as restrições de *hardware*.

O autor afirma também que este processo envolve várias pessoas, ou grupo de pessoas, da organização solicitadora do *software*, conhecidas como *stakeholders*, os quais podem comprometer a compreensão e a elicitação dos requisitos quando:

- a) não definirem quais processos o sistema de computador deverá conter, analisando apenas os termos gerais; e tiverem dificuldades de articular o que desejam, fazendo, por vezes, solicitações surreais, ignorando os custos de seus requisitos.
- b) utilizarem-se de termos próprios para expressar os requisitos, dificultando o trabalho do engenheiro caso ele não conheça o domínio do cliente. Por isso, para que os requisitos solicitados sejam atendidos, é de suma importância que o engenheiro responsável conheça a área de negócio do cliente.
- c) expressarem diferentemente requisito. Quando isso ocorre, o engenheiro deve conhecer todas as fontes potenciais de requisitos e descobrir os pontos em comum e conflituosos.
- d) os requisitos do sistema sofrerem influências dos fatores políticos.
- e) houver dinamismo na forma de análise do ambiente econômico e de negócio, sendo este alterado inevitavelmente de acordo com a evolução do processo de análise, podendo surgir, dessa forma, novos requisitos solicitados por *stakeholders* diferentes.

Cada organização terá seu próprio modelo geral que dependerá, sobretudo, do fator de negócio e de dados como nível de conhecimento da equipe, tipo de sistema que será desenvolvido e os padrões que este terá. As atividades de processo são tratadas de forma espiral, ou seja, são intercaladas à medida que o processo progride da parte interna para a externa da espiral.

Segundo Sommerville (2007), são consideradas atividades de processo, os itens apresentados abaixo:

a) **Obtenção de requisitos:** neste processo, são reunidas todas as informações do sistema proposto. Com elas, obtêm-se os requisitos de usuário e de sistema. Tais informações são coletadas por meio de entrevistas com os *stakeholders* (usuários), nas quais conta-se com o auxílio de um cenário e do protótipo na obtenção dos requisitos. A abordagem oriunda da orientação feita na utilização do ponto de vista organiza o processo de elicitación e os próprios requisitos. Um ponto forte dessa abordagem é a possibilidade de se gerar um *framework* que venha a descobrir todos os conflitos entre os requisitos propostos por diferentes *stakeholders*. Os tipos genéricos de ponto de vista são:

- i. Ponto de vista de interação - é representado por pessoas ou por sistemas que irão interagir diretamente com o sistema a ser desenvolvido.
- ii. Pontos de vista indiretos - são representados pelos *stakeholders* que não utilizam o sistema, mas que influenciam diretamente na criação dos requisitos.
- iii. Pontos de vista de domínio - são as características e restrições de domínio que têm influência nos requisitos de sistema.

Esses pontos de vista identificam requisitos diferentes: o ponto de vista de interação fornece os requisitos detalhados de característica e de interface do sistema, enquanto ponto de vista indireto concede os requisitos e as restrições da organização de alto nível e o ponto de vista de domínio fornece os requisitos de restrições de domínio que se aplica no sistema.

Sommerville (2007) afirma que, além dos pontos de vista acima citados, devem ser identificados também os pontos de vista mais específicos, tais como:

- a) fornecedor e receptor de serviços do sistema; sistemas que devem *interfacear* diretamente com o sistema especificado; regulamentos e padrões que se aplicam ao sistema; fontes de requisitos de negócio e não funcionais do sistema; pontos de vista de engenharia que refletem os requisitos de pessoas que devem desenvolver, gerenciar, e manter o sistema; pontos de vista de marketing; outros pontos de vista que geram requisitos de características do produto, esperadas pelos clientes; e como o sistema deve refletir a imagem externa da organização.
- b) Classificação e organização de requisitos, atividade que envolve os requisitos não estruturados, agrupando os requisitos relacionados e os organizando em conjuntos coerentes.
- c) Priorização e negociação de requisitos, atividade responsável por resolver os conflitos de requisitos existentes, pois, quando vários *stakeholders* participam do processo, conflitos de requisitos são gerados, então são relacionados os requisitos por ordem de prioridade e entra-se em negociação.
- d) Documentação de requisitos formal ou informalmente.

3.2 ESTUDO DA VIABILIDADE

Sommerville (2007) afirma que o processo de engenharia de requisito deve ser iniciado com o estudo da viabilidade, o qual consiste em um conjunto preliminar dos requisitos de negócio ou em um esboço da descrição do sistema e de como este pretende apoiar os processos de negócios. O estudo da viabilidade resultará num relatório em que constará se é válido dar continuidade aos projetos de requisitos e ao processo de

desenvolvimento. Este um estudo é realizado de forma breve e focado para se obter uma série de respostas de questões ilegíveis.

3.3 VALIDAÇÃO DE REQUISITOS

Segundo Pressman (2007), a validação de requisitos examina a especificação para garantir que todos os requisitos do software sejam declarados sem que haja ambigüidade, para que haja correção das inconsistências, omissões e erros que o produto apresente, adequando-o às normas estabelecidas para o processo, para o projeto e, sobretudo, para ele próprio.

Sommerville (2007) ressalta que a validação sobrepõe à análise, pois a primeira é designada para a descoberta de problemas, erros ou inconsistências, a fim de se evitar retrabalho no momento do desenvolvimento ou na operacionalização do sistema, fator que geraria custos excessivos. O custo de correção de um requisito é mais elevado que a correção de erros de um projeto e ou de uma codificação.

Segundo o mesmo autor, durante o processo de validação, devem ser realizadas algumas verificações, as quais serão descritas abaixo:

- a) Validade, na qual são verificadas as funções adicionais e diferentes que são necessárias.
- b) Consistência, ponto em que são verificadas as restrições ou descrições contrárias para a mesma função do sistema.
- c) Completeza, a qual consiste na verificação da documentação a fim de certificar-se de que todos os requisitos das funções e das restrições desejadas pelo usuário estejam especificados.

- d) Realismo, realiza com o intuito de se verificar a real possibilidade dos requisitos serem implementados de acordo com o conhecimento da tecnologia existente, levando-se em conta o orçamento e os prazos de desenvolvimento do sistema.
- e) Facilidade de verificação, na qual é analisado se os requisitos estão bem descritos a fim de que sejam diminuídas as divergências entre cliente e fornecedor, ou seja, devem-se descrever os conjuntos de testes que possam demonstrar que o sistema entregue atende a cada requisito.

Para a realização das verificações, existem duas técnicas de revisões de requisitos, as quais são responsáveis pela análise sistemática de cada requisito. A prototipação é a criação de um modelo executável do sistema, que é apresentado e liberado aos usuários finais e aos clientes, para verificar se ele atende às necessidades reais do cliente. E a geração de casos de teste é fase de teste de todos os requisitos.

3.4 DESENVOLVIMENTO DE REQUISITOS

O processo de desenvolvimento de requisito é o um dos mais importantes dentro de um projeto de software, pois, quando um requisito for mal desenvolvido, muitos problemas futuros podem ser gerados no decorrer do processo de desenvolvimento. Para que os requisitos sejam bem desenvolvidos, é preciso seguir alguns passos básicos, os quais serão descritos abaixo:

- a) O workshop inicial é primeiro contato, por meio do qual se obtém a primeira visão que a organização tem.

- b) As entrevistas com *stakeholders* (usuários) são realizadas com funcionários e/ou usuários de cada setor da organização, documentando-se todas informações obtidas.
- c) A validação do processo de negócio por setor consiste na avaliação que os usuários realizam acerca do modelo de processo de negócio de seu setor. Caso haja correções, estas serão realizadas. Após a aprovação dos usuários, as assinaturas são coletadas, servindo de comprovação de que os modelos estão de acordo com o solicitado.
- d) A análise e a validação do processo de negócio ocorre após a aprovação dos modelos de negócio dos setores. Os modelos são unificados e, juntamente com a diretoria da organização, é realizada a última avaliação.
- e) A elicitación dos requisitos consiste na identificação dos requisitos funcionais e não funcionais, registrando-os na ferramenta *Enterprise Architect* (EA).
- f) A modelagem dos casos de uso é o agrupamento dos requisitos de acordo com a funcionalidade, originando-se, dessa forma, o modelo de caso de uso.
- g) A criação do protótipo é uma ação paralela à modelagem dos casos de uso. Por meio dela, são criados os protótipos das telas dos sistemas, para a posterior validação do usuário.

3.4.1 Visão dos Modelos Tradicionais

Segundo Sommerville (2007), os requisitos devem ser escritos em linguagem natural, pois também devem ser compreendidos por pessoas que não possuem conhecimento técnico. Quanto mais bem detalhado eles estiverem, a compreensão será facilitada. Para mais bem exemplificar, existem alguns modelos de sistemas que são baseados em diferentes

abordagens durante o processo de análise dos requisitos. Para as diferentes abordagens são descritos seguintes modelos:

- a) Fluxo de dados: neste modelo, é mostrada a maneira como os dados serão processados nas diferentes etapas do sistema.
- b) Composição: é o modelo que representa a forma como são compostas a composição ou a agregação das entidades do sistema.
- c) Arquitetura: este modelo apresenta os principais subsistemas do projeto.
- d) Classificação: neste modelo, os diagramas de objeto/herança mostram as características comuns das entidades.
- e) Estímulo-resposta: por meio deste modelo, é apresentado como o sistema reage aos eventos externos e internos.

3.4.1.1 Lista de Requisitos

A lista de requisitos é a documentação em que constam todos os requisitos dos sistemas, levantados pela equipe responsável, conforme explanado anteriormente. A qual realiza as entrevistas, o acompanhamento e as reuniões na organização, contando com a participação dos usuários e funcionários.

No documento, são listados todos os requisitos funcionais e não funcionais a serem serão implementados no projeto; considerando que um requisito funcional define o que o sistema faz, descrevem as ações de transformação que os componentes de hardware ou software do sistema devem executar sobre as entradas a fim de produzir as saídas. Já os requisitos não funcionais definem os atributos do sistema durante a execução deste e determinam também como os atributos por eles definidos podem ser considerados de qualidade ou de restrições de sistemas de softwares ou de processos de software.

3.4.1.2 Modelo de Casos de Uso

Segundo Ivar Jacobson (2002), o caso de uso é definido como um documento narrativo que descreve a seqüência de eventos de um ator que usa um sistema para completar um processo.

O modelo de caso de uso é utilizado para a criação de manuais dos sistemas, pois neles as funcionalidades do sistema são descritas de forma natural e numa linguagem. Abaixo serão apresentados os elementos que compõem o caso de uso bem como suas respectivas descrições.

- a) Ator: é a representação de papéis desempenhados pelos diversos usuários; os atores não são parte integrada do sistema, tratam-se da parte que interage com o sistema.
- b) Caso de uso: são as representações dos serviços, tarefas e/ou funções que poderão ser utilizadas pelos usuários. Cada caso de uso possui atributos que devem ser claramente definidos, sendo assim, devem ser estabelecidos o nome, a descrição e o cenário (diálogo entre o ator e o sistema) do caso de uso
- c) Associações: são as ligações realizadas entre o ator e o caso de uso, sendo executada somente após a definição de ambos.

Os itens acima descritos podem ser observados na figura 1.

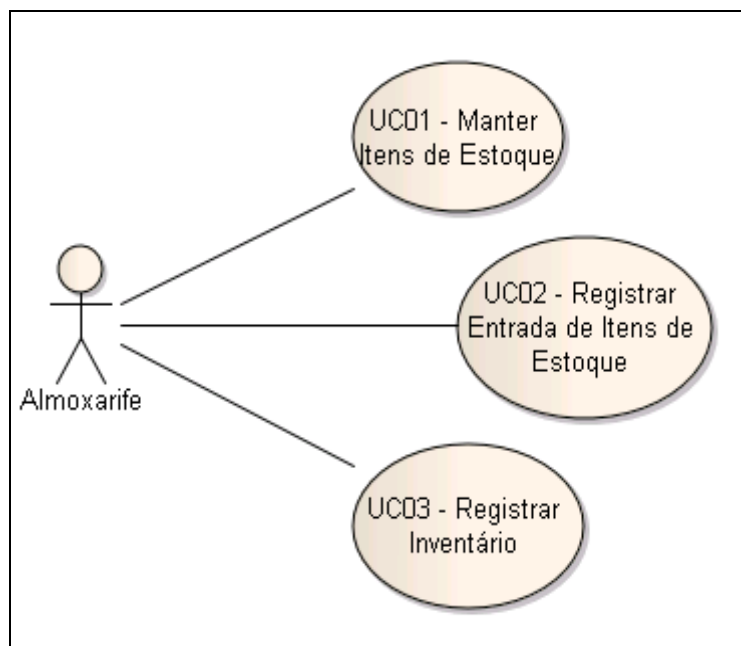


Figura 1. Modelo de Caso de Uso
Fonte: Criação própria.

3.4.2 Visão dos Modelos Ágeis

Segundo Sommerville (2007), os métodos ágeis tiveram início nos anos 80 e início da década de 90. A visão do modelo era qualificar a maneira de obtenção do melhoramento dos softwares. Para isso defendia a utilização de cuidadosos planejamentos de projeto, garantindo a qualidade formalizada com o uso de métodos de análise e projetos apoiados por ferramentas de CASE e por rigorosos controles de processo de desenvolvimento.

Sommerville (2007) afirma que os métodos ágeis têm princípios básicos: o envolvimento do cliente no processo de desenvolvimento, pois o sucesso depende da participação dos usuários no processo de desenvolvimento; outro é a participação de membros individuais que não estão compondo a equipe de desenvolvimento. Quando há a participação de usuários no processo de desenvolvimento pode ocasionar um desconforto para a equipe

pois começa-se a pressão para a entrega do sistemas no prazo determinado e assim a comprometi-se a simplicidades do sistema desenvolvido.

3.4.3 Histórias dos Usuários

Segundo Ress (2002), as Histórias dos Usuários (*Users Storys*) são desenvolvidas em reuniões entre os usuários e a equipe técnica, nas quais é realizado o planejamento de cada interação. Nelas são, portanto, documentados todos os requisitos para o desenvolvimento do projeto.

Segundo o mesmo ator, as *Users Story* têm como principal objetivo pôr em prática as funcionalidades mais valiosas do desenvolvimento. Elas especificam a numeração das atividades e funcionalidades que o software suportará, ou seja, por meio deste recurso que o cliente, em outras palavras, escreve histórias com suas atividades.

Cohn (2004) afirma que uma boa história depende de seis atributos fundamentais: ser independente, negociável, estimável, testável e valorizar os usuários e clientes. Quando uma história possui os seis atributos, pode-se elicitar corretamente a necessidade do cliente e dos usuários.

4 MODELOS DE PROCESSO DE QUALIDADE

Neste capítulo, serão apresentados os principais modelos de qualidade de software, suas respectivas histórias e seus principais fundamentos. Um dos grandes desafios para a Engenharia de Software tem sido desenvolver software de qualidade, capaz de atender o prazo, o esforço e os custos estabelecidos. Ao mesmo tempo, têm sido requeridos softwares cada vez mais complexos, pois o mercado anseia por produtos de maior qualidade. Mediante a esses desafios, as empresas desenvolvedoras de software têm mostrado crescente preocupação quanto à qualidade dos seus produtos, criando novos mecanismos que propiciem o planejamento, o controle, a avaliação e o alcance da qualidade.

A qualidade do processo interfere direta e significativamente na qualidade do produto resultante, tornando-se indispensável incluí-la no processo de obtenção do produto, como meio de avaliação das características de qualidade dos produtos de acordo com o ciclo de desenvolvimento.

4.1 ISO

A Organização Internacional para Padronização (ISO) tem como principal objetivo definir as normas e as atividades técnicas que visam cobrir as diferentes áreas da tecnologia. As organizações só conseguem a obtenção da certificação ISO após a especificação dos requisitos, e a adequação dos processos e serviços, de acordo com as exigências da Norma. Assim que adequado todo o processo nas normas a certificação é emitida pelos órgãos competentes.

Crespo (2005) afirma que a norma IEEE 829-1998 descreve os documentos para as atividades utilizadas a fim de testar um produto de software. São definidos oito documentos para que sejam cobertas as tarefas de planejamento, especificação e relato dos testes. Os artefatos sugeridos na norma devem ser seguidos a fim de que seja possível adaptar-se o processo de teste de acordo com as necessidades e as disponibilidades dos recursos.

Segundo Rocha (2001) na ISO, existe duas normas específicas para a área de software: ISO/IEC 12207 e a ISO/IEC 15504. A primeira delas é responsável por estabelecer os processos, as atividades e as tarefas aplicadas durante o processo de aquisição, fornecimento, desenvolvimento, operação e manutenção do software. Apesar de descrever o processo de vida de um software, ela não especifica detalhadamente como programar ou realizar atividades e tarefas que estejam envolvidas no processo. Já a ISO/IEC 15504 objetiva ser um padrão internacional de avaliação de melhorias do processo de software bem como ser a determinadora da maturidade de processos de uma organização.

4.2 CMMI

CMMI é um modelo de referência que fornece orientação para o desenvolvimento de processos de software. Tem por objetivo eliminar as inconsistências, aumentar a clareza e o entendimento do processo, e, assim, fornecer tecnologia comum e estilo consistente. Com isso, estabelece regras de construção uniformes que venham a assegurar as consistências com a ISO/IEC 15504.

Segundo Sommerville (2002), o CMMI baseia-se em um conjunto de capacidades da engenharia de software que permite medir se as organizações conseguem alcançar os diferentes níveis de maturidade.

A ISO abrange as áreas que são relacionadas ao desenvolvimento do projeto, enquanto o CMMI abrange as áreas de processos de software.

O CMMI possui as seguintes representações: contínua, a qual permite que numa organização seja selecionada a área, ou grupo de áreas, de processos e assim melhorar os processos relacionados, sendo que está usa-se de um nível de capacidade para caracterizar as melhorias relativas a uma determinada área de processo individual; e por estágio, em que se utiliza conjuntos pré-definidos de áreas de processos (PA's) para definir os caminhos que a organização percorrerá, caracterizando-as por nível de maturidade, sendo que cada nível de maturidade tem um conjunto de áreas de processo que caracterizam diferentes comportamentos organizacionais.

4.3 MPS.BR

De acordo com a SOFTEX (2009), o programa Melhoria de Processo do Software Brasileiro - MPS.BR, criado em dezembro de 2003, é um mobilizador de longo prazo. Ele é coordenado pela Associação para Promoção da Excelência do Software Brasileiro - SOFTEX, a qual conta com o apoio do Ministério da Ciência e Tecnologia - MCT, a Financiadora de Estudos e Projetos - FINEP, o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas - SEBRAE e o Banco Interamericano de Desenvolvimento - BID

O propósito deste modelo é melhorar os processos do software de forma que a mobilização vise à criação e ao aprimoramento dele em conformidade com as Normas Internacionais ISO/IEC 12207 [ISO/IEC, 2008a] e na ISO/IEC 15504 [ISO/IEC, 2003]. Possibilitando, dessa forma, que ele seja compatível com as melhores práticas da engenharia de software, e adequado à realidade das empresas brasileiras. O custo da adoção deste modelo é razoável para todas as regiões brasileiras.

SOFTEX (2009) afirma que o MPS.BR é baseado em duas metas:

- a) Técnicas: visa à criação e ao aprimoramento do modelo MPS esperando os seguintes resultados: guias do modelo MPS; Instituições Implementadoras credenciadas para prestar serviços de avaliação seguindo o método de avaliação MA-MPS; consultores de aquisição (CA), certificados para prestar serviços de consultoria de aquisição de software; e serviços relacionados
- b) De Mercado: visa à disseminação e à adoção do modelo MPS em todas as regiões do país, num intervalo de tempo justo, a um custo razoável. Sendo esse custo tanto em PME (foco principal) quanto em grandes organizações públicas e privadas, esperando os seguintes resultados: criação e aprimoramento do modelo de negócio MN-MPS; cursos, provas e *workshops*; organizações que implementaram o modelo MPS; organizações com a avaliação MPS publicada tem um prazo de validade de três anos.

5 MODELO DE QUALIDADE MPS.BR

SOFTEX (2009) afirma que o modelo Qualidade de Software (MPS.BR) é fundamentada a partir das normas ISO/IEC 12207 2 e ISO/IEC 15504, levando-se sempre em consideração a importância do modelo CMMI para as empresas brasileiras que atuam no mercado internacional. Conforme pode ser observado na Figura 2, o modelo MPS possui três componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA – MPS) e o Modelo de Negócio (MN-MPS).

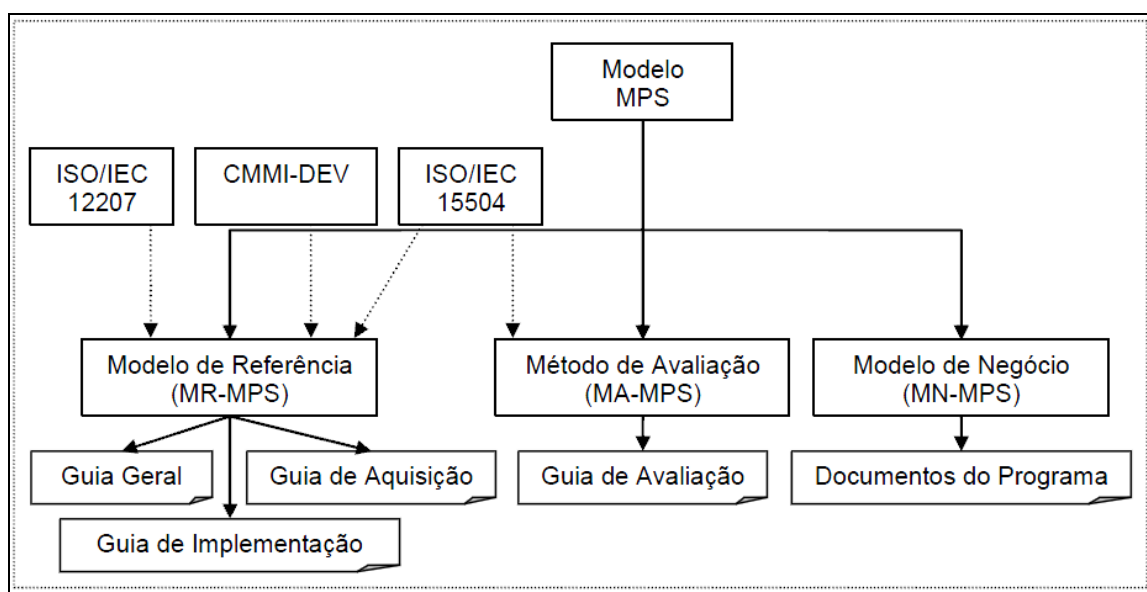


Figura 2. Componentes do MPS.BR
Fonte: MPS.BR (2009, p. 13)

Segundo a SOFTEX (2009), o Modelo de Referência (MR-MPS) contém todos os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o MR-MPS, o qual possui as definições dos níveis de maturidade, de processos e de atributo de processo. O MR-MPS está em conformidade com os requisitos de modelos de referência de processo da Norma Internacional ISO/IEC 15504-2 [ISO/IEC, 2003] cujo guia de implementação sugere formas de se implementar cada um dos níveis do MR-MPS. Há também o guia de aquisição, que é um documento complementar destinado a

organizações que pretendem adquirir software e serviços correlatos;, este guia, no entanto, não contém requisitos do MR-MPS.

SOFTEX (2009) afirma que o modelo de avaliação (MA-MPS) foi definido de acordo com a Norma Internacional ISO/IEC 15504. O processo e o Método de Avaliação MA-MPS foram definidos de forma a: permitir a avaliação objetiva dos processos de software de uma organização e/ou unidade organizacional; permitindo a atribuição de um nível de maturidade do MR-MPS com base no resultado da avaliação; ser aplicável a qualquer domínio na indústria de software e ser aplicável a organizações /unidades organizacionais de qualquer tamanho.

O Modelo - MN-MPS descreve regras de negócio para a implementação do MR-MPS pelas Instituições Implementadora, avaliação seguindo o MA-MPS pelas Instituições Avaliadoras (IA), organização de grupos de empresas pelas Instituições Organizadoras de Grupos de Empresas (IOGE). Para a implementação do MR-MPS e avaliação MA-MPS, e certificação de Consultores de Aquisição (CA) e programas há calendários de treinamento do MPS.BR, sendo que são realizados por meio de cursos, provas e workshops.

No presente estudo, será abordada apenas parte do modelo de referência (MR-MPS), o qual define os níveis de maturidade que, por sua vez, são uma combinação entre processo e capacidade.

SOFTEX (2009), a capacidade do processo é a caracterização da habilidade dele a fim de alcançar os objetivos de negócio, atuais e futuros; estando relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade.

5.1 NÍVEIS DE MATURIDADE DO PROCESSO

No modelo de referência MR-MPS, define-se os níveis de maturidade formados pela combinação dos processos e pela capacidade.

SOFTEX (2009) afirma que, para a definição dos processos, foram utilizados os requisitos apresentados pela ISO/IEC 15504-2, declarando o propósito e os resultados esperados de sua execução. Assim, permitindo-se a avaliação e a atribuição dos graus de efetividade na execução dos processos.

De acordo com a SOFTEX (2009), os níveis de maturidade estabelecem patamares na evolução dos processos, caracterizando, assim, os estágios de melhoria da implementação de processo nas organizações, que, conforme o seu nível de maturidade permitirá prever o seu desempenho futuro ao executar um ou mais processos.

O MR-MPS estabelece sete níveis de maturidade, sendo eles: A (em otimização), B (gerenciado quantitativamente), C (definido), D (largamente definido), E (parcialmente definido), F (gerenciado) e G (parcialmente gerenciado). De acordo com escala de maturidade, inicia-se no nível G, progredindo-se até o nível A.

A cada nível de maturidade é atribuído um perfil de processo que indica em que ponto a organização deve ampliar os esforços para que haja melhoria.

A fim de que haja progressão e o alcance de determinado nível de maturidade do MR-MPS, é preciso obter-se os resultados esperados dos respectivos processos e também que os atributos estejam aderentes ao nível que se deseja obter.

5.2 ÁREAS DE PROCESSO DO MODELO

Dentre as áreas de processos do modelo de referência MR-MPS citadas acima, neste projeto de pesquisa, será abordada apenas uma das áreas do nível G, sendo este o primeiro nível de maturidade do modelo de referência MR-MPS, conforme explanado anteriormente.

Segundo a SOFTEX (2009) o processo de implantação deve ser desenvolvido com muita cautela, pois consiste no início da implantação de melhoria dos processos de software de uma organização. Ao final da implantação deste nível, a organização deve gerenciar parcialmente seus projetos de desenvolvimento de software.

O nível de maturidade G possui dois pontos importantes que acabam tornando sua implementação um tanto quanto desafiadora: a mudança de cultura organizacional que orientará na definição e na melhoria dos processos de desenvolvimento de software e, também, a conceituação do termo “projeto” para a organização, ou seja, o que ela entende, ou passará a entender, por projeto.

De acordo com a SOFTEX (2009), o projeto usará os próprios padrões e procedimentos, não sendo necessários os padrões em nível organizacional. Se existirem, na organização, processos que deverão ser adaptados ao projeto, estes fatores serão declarados durante o seu planejamento. As adaptações podem ser: de inclusão de alteração em processos, de atividades, de ferramentas, de técnicas, de procedimentos, de padrões, de medidas, dentre outras.

O nível de maturidade G é constituído pelas áreas de Gerência de Projetos (GPR) e Gerência de Requisitos (GRE).

5.2.1 Gerência de Projetos

Segundo a SOFTEX (2009), o processo de Gerência de Projetos tem como propósito determinar e manter os planos que definem as atividades, os recursos e as responsabilidades do projeto. Tem ainda o dever de promover as informações sobre o andamento do projeto, tornando possível a realização de correções quando, no decorrer do processo, ocorrer algum desvio significativo no desempenho do projeto. Os níveis de maturidade crescem de acordo com a evolução dos processos.

SOFTEX (2009) afirma que o processo de Gerência de Projetos envolve várias atividades, entre elas destacam-se: o desenvolvimento de um plano de controle do projeto; a obtenção de um comprometimento de manter o controle durante toda a execução do projeto; e conhecer o processo do projeto de modo que ações corretivas possam ser aplicadas no decorrer da execução quando houver um desvio do projeto planejado.

Fazem parte do desenvolvimento do plano de projeto: a identificação e estimulação do escopo, dos produtos de trabalho e das tarefas do projeto e a definição dos compromissos bem como do cronograma de execução, baseando-se sempre no ciclo de vida estabelecido para o projeto.

O plano do projeto é que estabelece a base de execução e de controle das atividades, podendo também determinar o nível de progresso da execução do projeto, por meio de comparações dos atributos reais do produto de trabalho e de tarefas, pelo esforço, pelo custo e pelo cronograma planejado.

É feita a revisão do início ao fim de cada fase do projeto ou de algumas das atividades de fundamental importância para o seu sucesso. Esta revisão faz-se necessária para que haja a verificação das condições de ingresso a uma nova fase, ou seja, apenas será dada continuidade ao processo se a fase anterior já tiver obtido sucesso no processo de revisão.

5.2.2 Gerência de Requisitos

SOFTEX (2009) afirma que a Gerência de Requisitos tem como principal propósito gerenciar os requisitos do produto e dos componentes do projeto e também de identificar as inconsistências existentes entre os requisitos, o plano do projeto e os produtos de trabalho do projeto.

A Gerência de Requisitos tem como principal objetivo controlar a evolução dos requisitos, sendo responsável pelo gerenciamento de todos os requisitos recebidos ou gerados pelo projeto, incluindo ainda os funcionais e os não funcionais, bem como os impostos no projeto pela organização. Para que haja um controle dos requisitos acordados, a organização deverá executar um conjunto de passos definidos e apropriados. Quando houver a criação de um novo requisito no projeto, ele deverá ser revidado antes de ser acrescentado ao escopo do projeto a fim de evitar-se que o entendimento seja comprometido.

De acordo com SOFTEX (2009), o efetivo e seguro entendimento das necessidades do cliente, bem como dos requisitos do projeto, dependem da comunicação dos fornecedores com o elicitador de requisitos. Quando as necessidades e os requisitos são bem compreendidos, a probabilidade de o projeto obter sucesso é consideravelmente maior.

Os principais assuntos ligados aos requisitos devem ser bem estudados junto aos fornecedores de requisitos, entre eles destacam-se os seguintes: definição e aprovação de requisitos, solicitação de mudanças nos requisitos, entre outros.

De acordo com Dorfmann e Thayer (1990), os requisitos de software são a representação da necessidade requerida pelo usuário, podendo esta ser encontrada ou adquirida em determinado produto, ou componente de produto, para resolver um problema, alcançar um objetivo ou para satisfazer um contrato. Já a Gerência de Requisitos envolve o processo de identificação dos requisitos do produto e dos componentes do produto do projeto.

É necessário estabelecer e manter um acordo entre cliente e equipe de projeto sobre tais requisitos, que possibilite o controle e o tratamento das mudanças de requisitos ocasionadas no decorrer do processo de desenvolvimento. Tal controle pode ser gerenciado por meio da rastreabilidade dos requisitos.

Segundo a SOFTEX (2009) quando os requisitos são bem gerenciados, o processo de rastreabilidade pode ser estabelecido desde um requisito fonte, passando por todos os níveis de decomposição do produto até os requisitos de mais baixo nível e, assim, até o requisito fonte.

De acordo com SOFTEX (2009), quando é utilizado, no projeto, o processo de rastreabilidade dos requisitos, alguns resultados são esperados, dentro os quais destacam-se:

- a) GRE1 – Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando-se critérios objetivos. Este resultado tem como principal objetivo garantir que os requisitos estejam claramente definidos de acordo com o entendimento - realizado junto ao fornecedor de requisito, . As informações dos fornecedores podem ser identificadas no plano do projeto e a forma de comunicação entre eles também poderá constar nele. , Registrando-se sempre por meio de atas, e-mail, ferramentas de comunicação, entre outros meios. Em relação à comprovação do entendimento dos requisitos, estes podem ser documentados de diferentes formas, tais como a lista de requisitos, as especificações de caso de uso ou o detalhamento conforme a metodologia da organização, etc.. Depois de realizada a documentação dos requisitos, deve-ser identificar se eles, bem como os componentes do produto do projeto, estão de acordo com a expectativa do cliente e dos usuários. Para isso, os requisitos são avaliados de acordo com a base de critérios e os objetivos previamente estabelecidos. Para que haja essa verificação, pode ser usado um *checklist*, ou

seja, um documento com todos os tópicos tratados no decorrer da elaboração do projeto.

- b) GRE2 - O comprometimento da equipe técnica com os requisitos aprovados é obtido. Ao aprovarem-se os requisitos formaliza-se o comprometimento da equipe técnica para com os requisitos aprovados. O registro formal pode ser feito por meio de ata de reunião, e-mail ou outro mecanismo. Quando há mudanças de requisitos aprovados pelos fornecedores, tais como alterações ou criações, o compromisso já estabelecido pela equipe técnica pode ser afetado. Assim, quando isso ocorrer, a equipe técnica deverá obter e registrar os requisitos após terem sido novamente aprovados a partir dos critérios estabelecidos na GRE1.
- c) GRE3 – A Rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida. Este resultado indicará se há necessidade de se estabelecer um mecanismo que permita a rastreabilidade e a dependência dos requisitos e os produtos de trabalho. Quando já definida a rastreabilidade, está facilitará o processo de avaliação de impacto das mudanças que os requisitos podem ocasionar. Caso não haja um sistema de rastreamento, deve ser criado um mecanismo (sistema) que possibilite a realização da rastreabilidade bidirecional entre os requisitos e os produtos de trabalho.
- d) GRE4 – Revisões em planos e produtos de trabalho do projeto são realizadas visando a identificar e corrigir inconsistências em relação aos requisitos. Este resultado tem por intuito a realização de revisões, por meio de algum mecanismo equivalente, para identificar inconsistências entre os requisitos e os demais elementos do projeto, tais como, planos, atividades e produtos de trabalho. Quando inconsistências forem identificadas, elas devem ser

registradas, executando-se ações corretivas a fim de resolvê-las. Quando houver mudanças nos requisitos, é importante examinar se os demais artefatos estão consistentes com as alterações realizadas. As ações criadas para a correção das inconsistências devem ser acompanhadas até que sejam totalmente resolvidas.

- a) GRE5 – Mudanças nos requisitos são gerenciadas ao longo do projeto. Durante a execução do projeto, os requisitos podem ser modificados por uma série de motivos. Assim, requisitos adicionais podem ser incorporados ao projeto, enquanto outros podem ser retirados ou sofrerem alterações. . Quando ocorrerem mudanças, elas devem ser registradas e os requisitos podem ser revistos conforme definido na GRE. Um histórico de decisões acerca dos requisitos deverá ser disponibilizado. As decisões deverão ser tomadas após a análise de impacto que inclui aspectos como a influência em outros requisitos, a expectativa dos interessados, o esforço, o cronograma, os riscos e o custo.

Dentro do processo de Gerência de Requisitos, devem ser respeitados alguns atributos. Segundo o Guia Geral do MR-MPS, definido pela SOFTEX (2009), a capacidade do processo é representada por um conjunto de atributos de processo, onde está expresso o grau de refinamento e a institucionalização em que o processo é executado na organização ou unidade organizacional.

Segundo a SOFTEX (2009) os níveis de maturidade são níveis acumulativos, ou seja, se a organização está no nível F, ela já possui a certificação do nível de capacidade e os atributos de processo exigidos pelos níveis G e F. Já no que se refere aos atributos de processo, para atingir-se o nível G do MR-MPS, uma organização deverá atender aos resultados esperados pelos RAPs de 1 a 10. Para que o processo seja considerado satisfatório ao nível G, é preciso que o atributo de processo AP 1.1 seja caracterizado como T (totalmente

implementado) e o que o atributo de processo AP 2.1 seja caracterizado como T (totalmente implementado) ou L (largamente implementado). Os atributos de processo AP 1.1 e AP 2.1 serão posteriormente descritos bem como a numeração de RAPs mencionada..

- a) AP 1.1 – O processo é executado. Este atributo é uma medição do quanto o processo atingiu o seu propósito. É esperado o seguinte resultado:
 - i. RAP 1 – O processo atinge seus resultados definidos. Espera-se que o processo transforme produtos de trabalho de entrada em produtos de trabalho de saída, permitindo, assim, que o propósito do processo seja atingido.

- b) AP 2.1 – O processo é gerenciado. Este atributo é responsável pela medição do gerenciamento do processo. São esperados os seguintes resultados:
 - i. RAP 2 - Existe uma política organizacional estabelecida e mantida para o processo. É visada à definição de políticas que conterão as diretrizes de como se dá o planejamento e a implementação dos processos da organização, de modo que as organizacionais alcance a execução dos processos e a indicação de como devem ser aderidos os principais aspectos de cada processo, conforme sua expectativa. As políticas de publicação são de produção exclusiva da organização.
 - ii. RAP 3 – A execução do processo é planejada. Neste resultado, o plano de execução do processo é definido, planejando-se os recursos, a responsabilidade e o tempo bem como as atividades de controle e monitoramento da execução do processo. Para isso, é criada a documentação do plano de execução do processo.

- iii. RAP 4 – (Para o nível G) A Execução do processo é monitorada e ajustes são realizados. Trata-se do monitoramento da execução dos processos de acordo com o que foi definido no planejamento e da asseguarção das ações corretivas, observando-se se elas são tomadas sempre que houver um desnível do planejamento. No decorrer do monitoramento dos processos, falhas podem ser identificadas e assim as ações corretivas deverão ser tomadas;
- iv. RAP 5 - (Até o nível F) As informações e os recursos necessários para a execução do processo são identificados e disponibilizados. É visada à garantia de que as informações e os recursos necessários para a execução do processo serão identificados e estarão disponíveis quando houver a necessidade. Incluem-se também, na descrição do processo, os recursos financeiros, as condições física adequadas, o pessoal e as ferramentas apropriadas. .
- v. RAP 6 – (Até o nível F) As responsabilidades e a autoridade para a execução do processo são definidas, atribuídas e comunicadas. Este resultado tem como objetivo assegurar as responsabilidades e a autoridade para a execução dos processos sejam claramente definidas e bem compreendidas.
- vi. RAP 7 – (Até o nível F) As pessoas que executam o processo são competentes em termos de formação, treinamento e experiência. É assegurado se as pessoas envolvidas na execução do projeto possuem a habilidade, o conhecimento e as experiências necessárias para exercer suas funções. O conhecimento e a habilidade não se restringem apenas

aos documentos de processo, mas aos trabalhos em grupo, à liderança, à análise e solução de problemas.

- vii. RAP 8 – A comunicação entre as partes interessadas no processo é gerenciada de forma a garantir o seu envolvimento. O foco deste resultado é identificar se as partes interessadas no processo e planejamento estão realmente envolvidas nas atividades como planejar, coordenar, revisar e definir requisitos em execução do processo. Comprovando-se, assim, a efetiva comunicação entre os envolvidos na execução.
- viii. RAP 9 – (Até o nível F) Os resultados do processo são revistos com a gerência de alto nível para fornecer visibilidade sobre a sua situação na organização. Este resultado objetiva fornecer viabilidade para a gerência acompanhar o estado da execução dos processos, levando em consideração as adequações, as operações e os recursos apropriados para o alcance dos resultados esperados. Este item não deve ser confundido com o monitoramento exposto na RAP 4.
- ix. RAP 10 – (Para o nível G) O processo planejado para o projeto é executado. Este resultado tem por intuito garantir que o projeto é conduzido a partir da execução do que fora planejado no processo. Comprovando-se, dessa forma, que os registros de execução das atividades do processo são, de fato, baseados no seu planejamento. Os quais devem ser mantidos e revisados periodicamente para garantir que os objetivos do projeto sejam sempre sendo levados em conta no processo.

6 TRABALHOS CORRELATOS

6.1 APLICAÇÃO DE MÉTRICAS DE SOFTWARE NO MÉTODO SCRUM DA METODOLOGIA ÁGIL

Este trabalho provém de um estudo de aplicação de métricas de software, realizado a fim de que fossem obtidos os prazos estimativos para o desenvolvimento de uma interação do método Scrum, utilizando-se a metodologia ágil. No desenvolvimento de um projeto, a metodologia ágil propõe-se a contribuir na implementação rápida e flexível de qualquer mudança. É considerada uma das metodologias que melhor atende às necessidades básicas para a obtenção de um resultado significativo no processo de desenvolvimento de software. Esta metodologia também favorece no gerenciamento de projeto, pois oferece flexibilidade e é adaptável a constantes mudanças.

6.2 ENGENHARIA DE SOFTWARE ORIENTADA AOS PROCESSOS

O presente estudo focou-se no processo de desenvolvimento do software orientado pelo cliente: utilizando-se de uma tecnologia de ponta, de desenvolvimento em tempo útil e dentro de determinado orçamento disponível e fazendo uso de uma modelagem obtida por meio da ferramenta Unified Modeling Language (UML). A qual é um instrumento de visualização, especificação, construção e documentação de um componente do projeto. Quando o processo é apresentado ao cliente, ele é construído na ferramenta Rational Unified

Process (RUP), sendo esta a ferramenta de condução dos engenheiros de software e dos clientes.

Foi proposto, neste trabalho, um modelo genérico para que a organização pudesse orientar seus processos. Os quais servem como base para que ela modele o seu alvo, e assim, o modelo de processo aderente a organizações que desenvolvem software, no foco de gestão e com isso os fluxos do processo de trabalho do RUP seria muito utilizado. Para o desenvolvimento desse trabalho foram utilizados os casos de UML e do RUP para o desenvolvimento de software para organizações oriundas de processos.

6.3 DEFINIÇÃO DE PROCESSOS EM UM AMBIENTE DE DESENVOLVIMENTO DE SOFTWARE

Foi reconhecido, no presente estudo, o potencial da qualidade dos produtos de software e dos processos de software utilizados no desenvolvimento e na manutenção.

As organizações estão em constante busca pela qualidade dos seus processos. As pesquisas na área dos processos de software têm-se tornado mais frequentes, explorando-se a abordagem para a modelagem, à análise e melhoria dos processos, e à tecnologia de apoio do processo de softwares.

O objetivo principal foi apresentar uma infra-estrutura de processos de software que define o processo construído e integrado ao ambiente da odontologia.

6.4 ENSINO DA ENGENHARIA DE REQUISITOS POR MEIO DE UM AMBIENTE COLABORATIVO

Este trabalho visa à análise do processo de engenharia de software desde o processo de definição de requisitos até a entrega do software ao cliente. Dentre todos os pontos importantes da engenharia de software, foi enfocada engenharia de requisitos. Dispondo o estudo dela de mecanismos apropriados ao entendimento, à análise e à avaliação da viabilidade do produto que o cliente almeja. É neste processo que surge o maior número de variáveis problemáticas tais como preservar o sucesso do desenvolvimento de software.

Os problemas encontrados na engenharia de requisitos podem ser minimizados com uma adequada qualificação profissional dos responsáveis por essa área. Qualificação esta que pode ser obtida por meio de aprimoramentos de técnicas, oferecido por instituições de ensino.

7 DESENVOLVIMENTO DE UMA METODOLOGIA BASEADA NO NÍVEL G DO MPS.BR PARA A ÁREA DE GRE

O presente trabalho tem como objetivo modelar um processo para o Gerenciamento e Desenvolvimento de Requisitos baseado no modelo Brasileiro de Qualidade de Processo, o MPS.BR.

7.1 METODOLOGIA DESENVOLVIDA

A formação metodológica deste trabalho está de acordo com os atributos definidos no nível G do MSP.BR e, conforme mencionado anteriormente, foi dado destaque ao processo de Desenvolvimento e Gerenciamento de Requisitos. Conforme pode ser visto na Figura 3, o desenvolvimento deste trabalho consiste em três etapas.



Figura 3. Composição do Trabalho Desenvolvido

Na primeira delas, foi realizado um estudo da ferramenta Eclipse Process Framework – EPF Composer, utilizada para desenvolver o modelo de processo. O EPF é uma

ferramenta que proporciona uma estrutura para a gestão de processo de software por meio do desenvolvimento iterativo, ágil e incremental, utilizado para pequenos projetos.

O Epf Composer apóia a produção de processos com características próprias, fornecendo ferramentas e conteúdos que podem ser utilizados como base para uma grande variedade de processos de TI. Este *framework* auxilia ainda na autoria de métodos, ferramentas, processos, gestão de configuração de bibliotecas de métodos e também publicação de processos.

Na segunda etapa, foi realizado o desenvolvimento do processo, consistindo na modelagem do processo de Gerenciamento e Desenvolvimento de Requisitos. Na terceira etapa, como resultado, foi realizada a publicação e apresentação do processo definido proporcionando sua utilização pelas empresas desenvolvedoras de software que pretendem aderir a modelagem de processos de desenvolvimento e melhoria da qualidade de software em suas linhas de produção. Nos Apêndices A e B constam algumas ilustrações do Modelo desenvolvido e visualização na Ferramenta utilizada para o desenvolvimento.

O processo foi desenvolvido na ferramenta EPF Composer, e dividida conforme os quatro resultados definidos pelo Nível G do Modelo de Referência para a Melhoria do Processo de Software Brasileiro (MR-MPS.BR). Basicamente, Basicamente, o processo desenvolvido é composto por quatro fases, conforme ilustra a figura 4. Abaixo serão descritas todas as fases junto de seus respectivos conceitos, atributos, tarefas e artefatos.

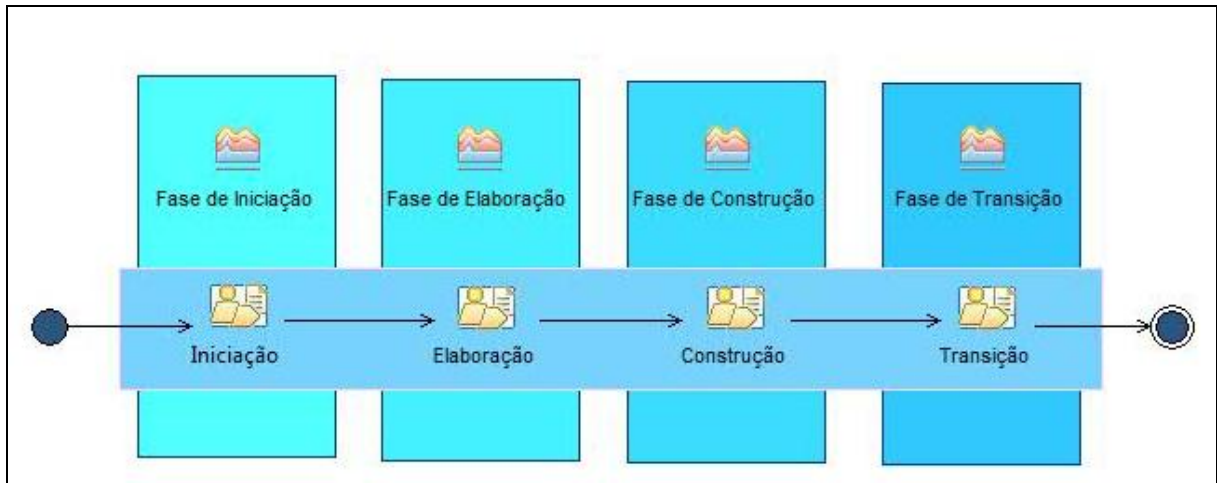


Figura 4. Processo Desenvolvido

Fase de Iniciação: A fase de iniciação é o *start* do processo. Fazendo-se necessário que haja um processo de elicitação dos requisitos de forma a não comprometer o processo de desenvolvimento. Conforme visualizado na figura 5, a fase é composta pelas tarefas de elicitação de requisitos, análise dos requisitos, Validação de requisitos com o cliente e a definição do modelo de requisitos.

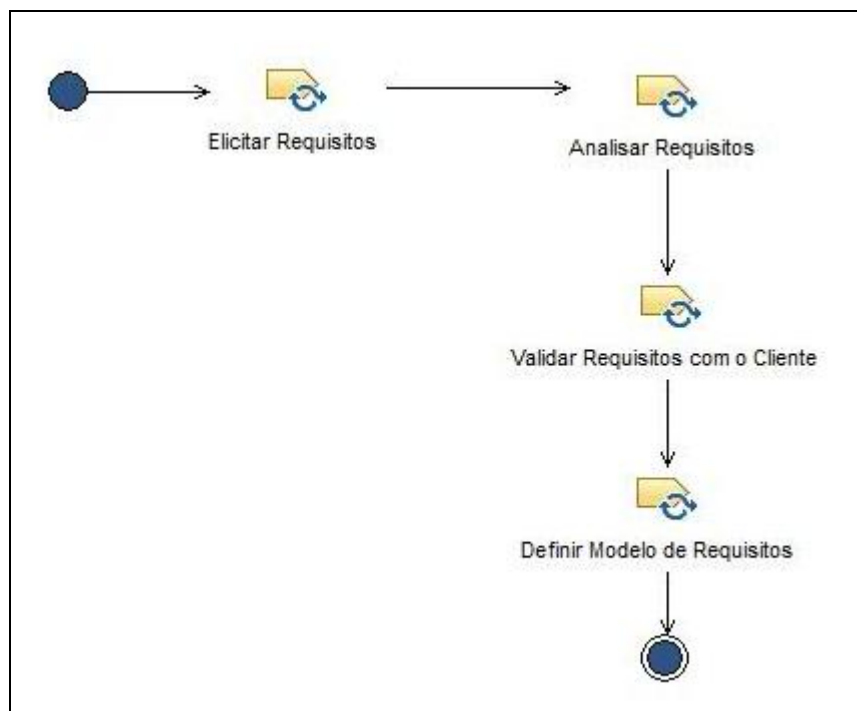


Figura 5. Fase de Iniciação

A tarefa de elicitação de requisitos é responsável pelo levantamento dos requisitos necessários para o desenvolvimento da aplicação, levando em conta as necessidades do cliente. O elicitor precisa estar presente na reunião realizada junto ao cliente, a fim de que sejam levantados todos os requisitos de usuário e de sistema, observando os fatores político-sociais da organização, pois estes, em determinado momento, poderão influenciar no projeto do sistema proposto. A tarefa de elicitar compete aos analistas de requisitos, clientes e fornecedores de requisitos.

Na análise de requisitos, são verificados todos os requisitos elicitados, os quais serão listados e encaminhados ao analista a fim de evitar a incompatibilidade de informações e, também, o desperdício de tempo e mão de obra. Sempre que os requisitos são bem analisados e corrigidos no decorrer da análise, a probabilidade de se obter sistemas que apresentem falhas torna-se cada vez menor; assim, o cliente consegue desempenhar eficientemente o seu trabalho e seus controles. O responsável por executar esta tarefa é o analista de requisitos.

A validação dos requisitos junto ao cliente é uma tarefa importante, pois é por dela que se obtém o aval do cliente para a continuidade do desenvolvimento do que foi solicitado. Nela o cliente é contatado e lhes são fornecidos todos os requisitos elicitados e analisados, os quais provêm do primeiro contato com o cliente. Após verificação, o cliente informa se os requisitos estão de acordo com o que ele havia solicitado. Caso haja alguma alteração ou algum requisito que não tenha sido bem especificado no processo de elicitação, reajustes são feitos. Assim que tudo estiver de acordo com a necessidade do cliente, o analista de requisitos encaminhará os requisitos para o processo de especificação. A validação dos requisitos compete ao analista de requisitos e ao cliente.

Já, na tarefa de definir modelos de requisitos, o analista de requisitos determinará qual o modelo de requisito será utilizado para representar os que foram elicitados e validados com o cliente. Esta tarefa precisa ser realizada, para que toda a equipe esteja apta a trabalhar com os novos requisitos a serem implementados no novo sistema ou nas alterações de módulos ou nos sistemas já existentes, sendo que, a representação pode ser realizada por meio de diagramas que apresentam todas as fases do software, facilitando a visualização e o processo de desenvolvimento do software e identificando os recursos que o software fornecerá ao cliente final.

Fase de elaboração: Esta fase é composta pela tarefa de especificar os requisitos a serem implementados no software, como alteração ou criação de novos processos. Depois da especificação, os requisitos são encaminhados ao processo de validação junto ao cliente. Conforme mostra a figura 6.

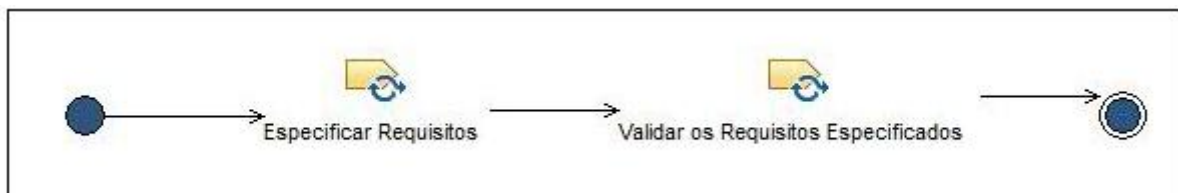


Figura 6. Fase de Elaboração

A especificação dos requisitos é responsável por descrever clara e objetivamente cada funcionalidade dos requisitos a serem implementados no software. Sendo que o especificador de requisitos pode descrevê-los e representá-los por meio do modelo - adotado pelo analista de requisitos, tratando-se este da constatação de todos os requisitos funcionais e não funcionais. Devem desenvolver esta tarefa os especificadores de requisitos e os analistas de sistemas.

Após a etapa de especificação de requisitos, entra-se em contato com o cliente a fim de se realizar a validação dos requisitos especificados. Caso haja algum requisito que não tenha sido especificado, ele será incluso na lista dos requisitos a serem desenvolvidos. Esta inclusão se dará juntamente com o analista de requisitos para que não haja nenhum problema no momento do desenvolvimento da funcionalidade do software. A responsabilidade pela execução desta tarefa compete aos especificadores de requisitos, analistas de sistemas, contando com a participação do cliente.

Fase de Construção: esta fase é responsável por liberar, implementar e testar os requisitos elicitados e definidos de acordo com a necessidade do cliente, tendo por tarefas a liberação, a implementação e o teste dos requisitos, conforme ilustra a Figura 7.

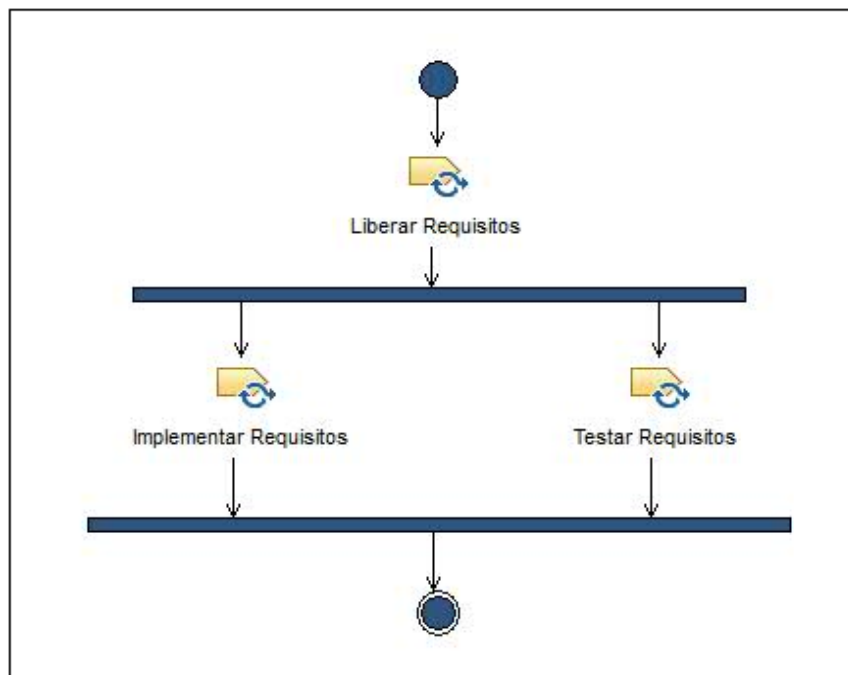


Figura 7. Fase de Construção

Na liberação dos requisitos, os requisitos já especificados na fase de elaboração são verificados e liberados aos desenvolvedores. A liberação somente será realizada se os requisitos estiverem totalmente elaborados e verificados pelo analista de requisitos.

A implementação de requisitos é responsável por codificar os requisitos liberados, mantendo-os ou criando os casos de uso. Efetuando-se, dessa forma, a rastreabilidade entre eles e também entre os componentes relacionados a eles a fim de que não haja nenhuma falha no momento dos testes e da liberação ao cliente.

No teste dos requisitos, todas as funcionalidades desenvolvidas são verificadas a fim de comprovar a eficácia dos requisitos implementados pelo desenvolvedor.

Fase de Entrega Requisitos: nesta fase, todos os requisitos elicitados, elaborados, implementados e testados são entregues ao cliente. Conforme, apresentado na Figura 8, nesta fase, há apenas a tarefa de entregar os requisitos.

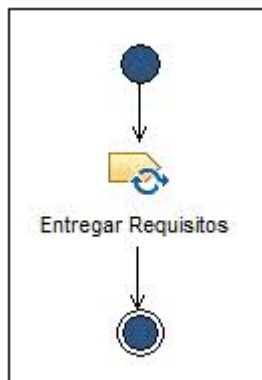


Figura 8. Fase de Entrega de Requisitos

A entrega dos requisitos consiste na disponibilização, ao cliente, dos elicitados e documentados de acordo as necessidades de na fase de iniciação.

A descrição completa das tarefas apresentadas neste trabalho pode ser observada no Processo Unificado da Ariela - PUA.

CONCLUSÃO

Atualmente, um dos grandes fatores de insatisfação por parte dos clientes que adquirem serviços ou produtos das empresas de desenvolvimento do software é a falta de qualidade fazendo com que as atualizações e/ou instalações dos softwares sejam liberados ao cliente com inconsistência dificultando seu trabalho e causando atrasos e prejuízos financeiros. Outro fator importante referente a insatisfação dos clientes na área de TI, é a baixa usabilidade dos sistemas proveniente de um levantamento de requisitos inexistente ou falho.

A elicitação de requisitos não é uma tarefa fácil de ser executada, pois exige um conhecimento aprofundado sobre técnicas de engenharia e manipulação de informações, onde estas, muitas vezes são mal interpretadas ou ignoradas pelo analista, gerando uma falha de qualidade no sistema, seja ela funcional ou não funcional como é o caso da usabilidade.

Uma das maneiras utilizadas na área de TI para minimizar as falhas ou inconsistências nas liberações de sistemas para o cliente final é a adequação dos processos de desenvolvimento de acordo com metodologias ou normas qualificadoras, como a ISO, o CMMI, o MPS.BR entre outras, as quais estabelecem regras que direcionam as etapas do processo de desenvolvimento.

Muitas empresas brasileiras de desenvolvimento de software estão adotando o MPS.BR para apoiar as equipes de trabalho no que se refere à garantia de qualidade nos produtos desenvolvidos, sendo essa uma norma que fomenta a qualidade na base, apoiando diretamente as micro, pequenas e médias empresas.

O modelo MPS.BR apóia as empresas para a melhoria nos processo de software e busca atender a necessidade de implantar os princípios de Engenharia de Software de forma

adequada, estando de acordo com as principais abordagens internacionais para definição, avaliação e melhoria de processos de software.

O modelo MPS.BR possui vários níveis de maturidade, sendo que a cada nível de maturidade a uma área de atuação. Um dos níveis de maturidade do modelo MPS.BR é o nível G que é baseado nas áreas de Gerência de Projeto e Gerência de Requisitos, sendo uma responsável por controlar todo o processo de desenvolvimento do software, desde as políticas organizacionais até o processo de final, e a outra, é responsável por controlar o processo de gerenciamento dos requisitos, desde a elicitação até o processo de liberação.

Neste trabalho desenvolveu-se uma metodologia que visa a definição de um modelo de processos para a área de processos gerenciamento de requisitos (GRE) do MPS.BR. Além disso, buscou-se o desenvolvimento de um processo que atendesse os resultados esperados do modelo, focando na aderência à padrões de qualidade consagrados e utilizados por diversas organizações brasileiras.

Apesar da utilização do modelo MPS.Br pelas organizações desenvolvedoras de software ser recente, os resultados apresentados são muito satisfatórios, sendo este um fator motivador do desenvolvimento deste trabalho. Com os estudos realizados na área de engenharia de software e requisitos, constatamos que a sua utilização minimiza consideravelmente os erros do software e a insatisfação dos clientes, uma vez que este processo aborda uma das áreas mais críticas da engenharia de software, a elicitação dos requisitos, pois é a partir dele que o software tem origem. Entendemos também, que a utilização de modelos de qualidade é uma processo de melhoria contínua, não podendo permanecer em uma única área de processo, e depende totalmente da maturidade da equipe executora.

A publicação do processo desenvolvido neste trabalho, poderá ser utilizada como base para as organizações de pretendem iniciar a organização dos seus processos de

levantamento e desenvolvimento de requisitos, pois o modelo apresentado foi baseado nos resultados exigidos para a área de processo de gerenciamento de requisitos do nível G do MPS.BR.

O modelo desenvolvido como produto deste trabalho deve ser otimizado pelas organizações interessadas antes de sua utilização, para atender suas necessidades, pois atualmente está apresentando características genéricas em sua definição, o que pode ser melhorado quando de sua implementação em uma linha de desenvolvimento. Sendo assim, como trabalhos futuros propõem-se:

- A implementação o processo dentro de uma organização, validando a aderência ao modelo MPS.BR.
- A implementação da área de processo GPR agrupando com as tarefas definidas no PUA para a área de processo GRE, alinhando com o nível G do MPS.BR.
- Desenvolver uma aplicação móvel do processo para ser acessado via *smartfone*, possibilitando a portabilidade na consulta do processo.

REFERÊNCIAS

- BERTOLLO, Gleidson. **Definição de Processos em um Ambiente de Software**. 2006. 108 f. Dissertação (Mestrado) - Universidade Federal do Espírito Santo, Vitória, 2006.
- BÔAS, Gisele Villas; ROCHA, Ana Regina Cavalcanti. Riscos e Fatores de Influência na definição de Estratégias para Projetos de Implementação de Melhoria de Processos de Software em Grupos de Empresas. **WAMPS 2009 – V Workshop Anual do MPS**. Campinas, 2009. 5, p. 176 - 184 , 10/2009
- BUENO, Isabella Cristina. **Ensino de engenharia de requisitos por meio de um ambiente Colaborativo**. 2009. 101 f. Monografia (Bacharel) - Universidade de Brasília, Brasília, 2009.
- COLENCI NETO, Alfredo. **Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS.BR**. 2008. 182 f. Tese (Doutor em Engenharia de Produtos) - Universidade de São Paulo, São Carlos, 2008.
- COHN, Mike. **User Stories Applied: For Agile Software Development**, Canada: Addison-Wesley, 2004
- DORFMANN, M. e THAYER, R. *Standards, Guidelines, and Examples of System and Software Requirements Engineering*. Los Alamitos, CA: IEEE Computer Society Press, 1990.
- DUARTE, Francisco José Monteiro. **Engenharia de Software Orientada aos Processos**. 2002. 104 f. Dissertação (Departamento de Informática) - Universidade do Minho Escola Engenharia, Braga, 2002.
- PFLEEGER, Shari L. **Engenharia de Software: Teoria na Prática**. 2. ed. São Paulo: Prentice Hall, 2004.
- PRESSMAN, Roger S. **Engenharia de Software**. 6. ed. São Paulo: McGraw-Hill, 2006.
- _____, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995. 1056 p. ISBN 8534602379
- REES, Michael. **A Feasible User Story Tool for Agile Software Development?** In: IEEE Ninth Asia- Pacific Software Engineering Conference(APSEC'02), 2002.
- SOFTEX, MPS. BR - Melhoria de Processo do Software Brasileiro: Guia Geral, Maio de 2009. Disponível em <http://www.softex.br/mpsbr/_guias/>. Acesso em: 01 Nov 2009.
- SOFTEX, MPS-BR - Melhoria de Processo do Software Brasileiro, Guia de Implementação, Parte 1 – Nível G. Disponível em: <http://www.softex.br/mpsbr/_guias/>. Acesso: 09 Nov de 2009.
- SOMMERVILLE, Ian. **Engenharia de Software**. 8ª ed. São Paulo: Pearson Addison-wesley, 2007.

_____, Ian. **Engenharia de software**. São Paulo: Addison-Wesley, 2003.

TRAVASSOS, Guilherme Horta; KALINOWSKI, Marcos. Variação de Desempenho nas Empresas que Adotaram o Modelo MPS: resultados iniciais iMPS 2009. **WAMPS 2009 – V Workshop Anual do MPS**. Campinas, 2009. 5, p. 20-29, 10/2009

ROUILLER, Ana Cristina; VASCONCELLOS, Alexandre Marcos Lins; BOAS, André Luis de Castro Villas. **ProQualiti - Qualidade na Produção de Software**: Núcleo de Estudos em Engenharia e Qualidade do software. 2. ed. Boa Viagem: Ana Cristina Rouiller, 2006. 118 p.

REFERENCIAS COMPLEMENTARES

COSTA, Marcelo Nascimento; KALINOWSKI, Marcos; SPINOLA, Rodrigo Oliveria. Soluções Concretas para Problemas Práticas da Engenharia de Requisitos. **Engenharia de Software**, Rio de Janeiro, n. , p.8-15

DINIZ, Alfraino de Souza. **PROREQ - Um guia facilitador para a implantação dos processos de gestão de requisitos**. 2007. 92 p. Dissertação (Mestrado em Ciências de Computação e Matemática) – Universidade de São Paulo, São Carlos. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-31032008-161043/>>. Acesso em: 20 set. 2009

ESPINHA, Rafael. **REVISTA Engenharia de Software: Qualidade de Software**. DevMedia Brasil, ano 2007, n.1. Disponível em: <<http://www.devmedia.com.br/articles/viewcomp.asp?comp=8029>>. Acesso em: 15 Out. 2008

FURTADO, Julio Cezar Costa; OLIVEIRA, Sandro Ronaldo Bezerra. Uma proposta de Mapeamento dos Processos Existentes no Guia de Aquisição do MPS.BR e no CMMI- ACQ. **WAMPS 2009 – V Workshop Anual do MPS**. Campinas, 2009. 5, p. 164 - 175 , 10/2009

GILLIS, Eric Robert; ANDRADE, Edméia Leonor Pereira de. Comparação entre a Instrução Normativa SLTI/MP nº04 e o Guia de Aquisição do MPS.BR. **WAMPS 2009 – V Workshop Anual do MPS**. Campinas, 2009. 5, p. 92 - 101 , 10/2009

GUERRA, Ana Cervigni; COLOMBO, Regina Maria Thienne. . **Tecnologia da informação: qualidade de produto de software**. Brasília: Ministério da Ciência e Tecnologia, 2009. 429 p.

GUSTAFSON, David A. **Teoria e problemas de engenharia de software**. Porto Alegre: Bookman, 2003. 207 p.

LIEBMAN, Alessandro. **Melhoria no processo de software: implantação do MPS. BR nível G em uma empresa de pequeno porte**. 2006. 43 f. Monografia (Bacharel em Ciência da Computação) - Universidade Federal de Lavras, Lavras, 2006. Disponível em: <http://www.bcc.ufla.br/monografias/2005/Melhoria_no_processo_de_software_implantacao>

[_do_MPS_BR_nivel_G_em_uma_empresa_de_pequeno_porte.pdf.>](#). Acesso em: 30 out. 2009.

LIMA, Geovane Nogueira; GOMES, Marcos André. Fatores críticos de sucesso em programas de melhoria de processo de forma cooperada. **WAMPS 2009 – V Workshop Anual do MPS**. Campinas, 2009. 5, p. 120- 127 , 10/2009

LOPES, Arthur Vargas. **Estruturas de dados: para a construção de software**. Canoas, RS: ULBRA - Universidade Luterana do Brasil, 1999. 2.v. ISBN 8585692650

LUCENA, Carlos J. P. **Inteligência artificial e engenharia de software**. Rio de Janeiro: J. Zahar, 1987. 305 p. ISBN 8571100136

MELLO, Marcelo Santos de. Gestão Integrada da Melhoria de Processos em Organizações de Software. **WAMPS 2009 – V Workshop Anual do MPS**. Campinas, 2009. 5, p. 34-41, 10/2009

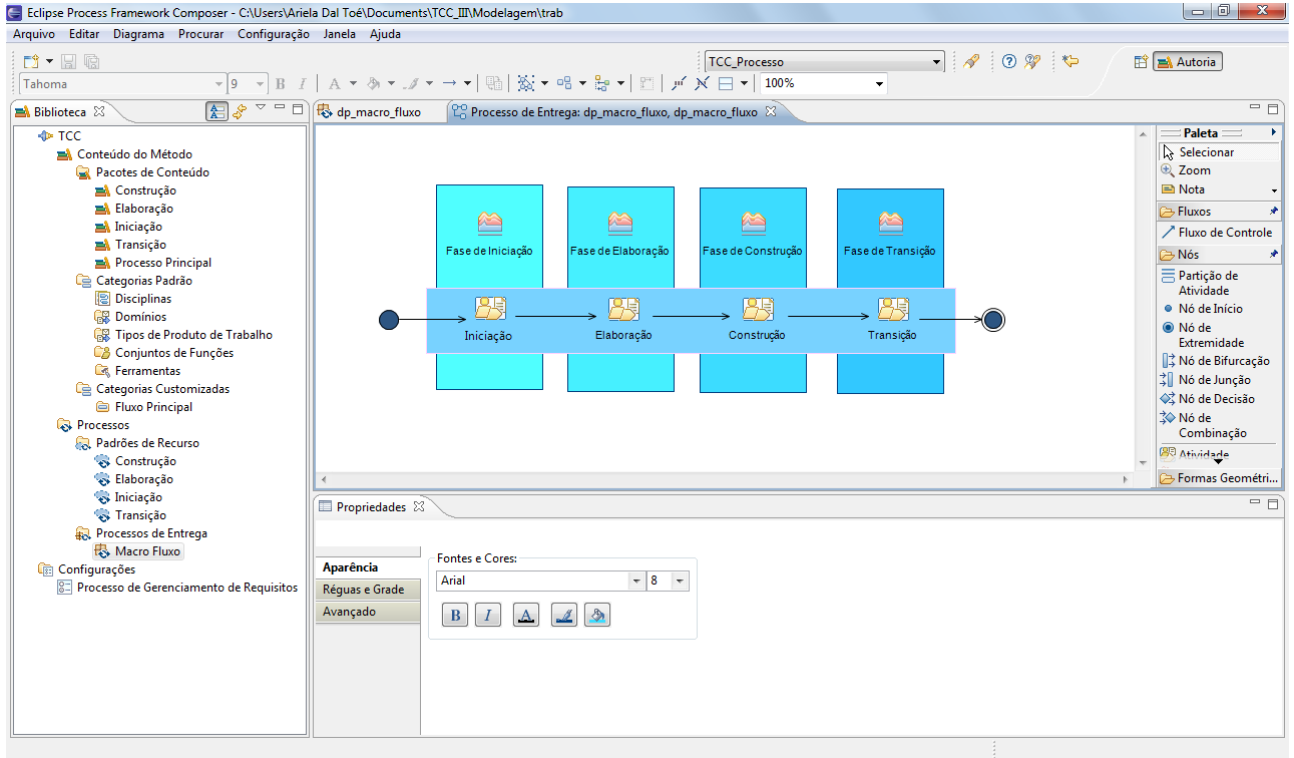
NASCIMENTO, Gustavo Vaz; LORENCIN, Wander Marcelo; NASSIF, Felício Fadlalla. Um Relato dos Desafios Encontrados e dos Benefícios conseguidos com a Implantação das Práticas Propostas pelo Nível F do Modelo De Referência de Processo MPS.BR . **WAMPS 2009 – V Workshop Anual do MPS**. Campinas, 2009. 5, p. 72 - 81, 10/2009

NASCIMENTO, Thercio Rodrigues do; RAMOS, Cristiane Soares; RIBEIRO, Luiz Carlos Miyadaira. Aplicação de Controle Estatístico de Processo (CEP) no contexto do MR-MPS em uma Fábrica de Software. **WAMPS 2009 – V Workshop Anual do MPS**. Campinas, 2009. 5, p. 62 - 71, 10/2009

ROCHA, Ana Regina Cavalcanti da; MALDONADO, José Carlos; WEBER, Kival Chaves. . **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, 2001. 303 p. ISBN 8587918540

TELES, Vinícius M. Um estudo de caso da adoção das práticas e valores do extreme programming - Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro - UFRJ, 2005.

APÊNDICE A – VISUALIZAÇÃO DO PROCESSO PRINCIPAL NO FRAMEWORK EPF COMPOSER



APÊNDICE C - ARTIGO

MODELAGEM DE UM PROCESSO PARA O DESENVOLVIMENTO E GERENCIAMENTO DE REQUISITOS BASEADA NO MODELO BRASILEIRO DE QUALIDADE DE PROCESSO MPS.BR

Ariela Dal Toe ¹, Gustavo Bisognin ²

1. Acadêmicas do Curso Ciência da Computação - Universidade do Extremo Sul Catarinense (UNESC) - Criciúma – BRAZIL

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

arieladaltoe@hotmail.com, gustavo@unesc.net

Resumo. *Este artigo descreve um pouco sobre a qualidade que se ter no momento de desenvolver um software, devido a muitos problemas encontrados por empresas que desenvolvem software foi criado um modelo qualidade de Processo Brasileiro baseado na norma ISO e no CMMI, sendo que será descrito um pouco sobre a composição desse modelo, é também será apresentado um modelo aderente ao Nível base do Modelo de Qualidade de Processo Brasileiro que é o Nível G do Modelo MPS.BR. Este Nível é composto pela Gerência de Projeto e Gerência de Requisitos, ambos descritos um pouco mais detalhado no decorrer do artigo.*

Palavras-chave: *Qualidade de Software, Melhoria de Processo de Software, MSP-BR.*

1 INTRODUÇÃO

Neste artigo será apresentada uma definição do modelo de Processo de Software Brasileiro o MPS.BR, onde neste conterà os níveis de maturidade que compõe esse modelo de Processo, dando em foque em um dos níveis de maturidade, que é o nível G. Este nível é o nível base do processo, pois para que uma empresa possua certificação MPS.BR é preciso que os seus processos estejam aderente ao nível G do modelo, onde No nível G as áreas de processo deste modelo é a Gerenciamento de Projeto e Gerenciamento de Requisitos, sendo que o foco deste artigo será o Gerenciamento de Requisitos bem como sua modelagem.

2 MELHORIA DE PROCESSO DO SOFTWARE BRASILEIRO - MPS-BR

SOFTEX (2009), afirma que para a definição dos processos foram utilizados os requisitos apresentados pela ISO/IEC 15504-2, declarando o propósito e os resultados esperados de sua execução. Assim permitindo avaliação e atribuição dos graus de efetividade na execução dos processos. A capacidade do processo é a caracterização na habilidade do processo para alcançar os objetivos de negócio, atuais e futuros; estando relacionada com o atendimento aos atributos de processo associados a cada nível de maturidade.

De acordo com a SOFTEX (2009), os níveis de maturidade estabelecem patamares na evolução dos processos, caracterizando assim os estágios de melhoria da implementação de processo nas organizações, onde de acordo com o seu nível de maturidade está permitirá prever o seu desempenho futuro ao executar um ou mais processos. Sendo que o MR-MPS define sete níveis de maturidade que são: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). Sendo que de acordo com escala de maturidade inicia-se no nível G e progride-se até o nível A.

2.1 NÍVEL G

O Nível de maturidade G possui dois pontos importantes que acabam tornando-o sua implementação um tanto desafiador que são: a mudança de cultura organizacional, onde está orientará na definição e na melhoria dos processos de desenvolvimento de software e a definição do conceito acerca do que é “projeto” para a organização.

De acordo com a SOFTEX (2009), o projeto usará os próprios padrões e procedimentos, onde nestes não são necessários os padrões em nível organizacional, sendo que se na organização existir processos que deverão ser adaptados aos processos do projeto, este fatores serão declarados durante o planejamento do projeto, onde essas adaptações pode ser inclusão de alteração em processos, atividades, ferramentas, técnicas, procedimentos, padrões, medidas, dentre outras.

2.2 Áreas de Processo do Nível G

Sendo que o nível G é constituído pelas áreas de Gerência de Projetos (GPR) e pela Gerência de Requisitos (GRE) de Software, onde:

a) Gerência de Projetos: Segundo a SOFTEX (2009), o processo de Gerência de Projetos tem como propósito determinar e manter os planos que definem as atividades, recursos e responsabilidades do projeto, também tem o a responsabilidade de promover as informações sobre o andamento do projeto, onde se torna possível a realização de correções quando no decorrer do processo ocorre algum desvio significativo no desempenho do projeto. O nível de maturidade cresce de acordo com a evolução dos processos.

b) Gerência de Requisitos: A Gerência de Requisitos tem como principal objetivo controlar a evolução dos requisitos, sendo responsável pelo gerenciamento de todos os requisitos recebidos ou gerados pelo projeto, incluindo ainda os requisitos funcionais e não-funcionais, bem como os requisitos impostos no projeto pela a organização. Para que haja um controle dos requisitos acordados a organização deverá executar um conjunto de passos definidos e apropriados. Quando houver a criação de um requisito novo no projeto este deverá ser revidado para não haja mau entendimento, antes deste ser acrescido no escopo do projeto.

Alguns resultados são esperados, dentro os quais destacam-se:

a) GRE1 – Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando-se critérios objetivos. Este resultado tem como principal objetivo garantir que os requisitos estejam claramente definidos de acordo com o entendimento - realizado junto ao fornecedor de requisito, . As informações dos fornecedores podem ser identificadas no plano do projeto e a forma de comunicação entre eles também poderá constar nele. , Registrando-se sempre por meio de atas, e-mail, ferramentas de comunicação, entre outros meios. Em relação à comprovação do entendimento dos requisitos, estes podem ser documentados de diferentes formas, tais como a lista de requisitos, as

especificações de caso de uso ou o detalhamento conforme a metodologia da organização, etc.. Depois de realizada a documentação dos requisitos, deve-se identificar se eles, bem como os componentes do produto do projeto, estão de acordo com a expectativa do cliente e dos usuários. Para isso, os requisitos são avaliados de acordo com a base de critérios e os objetivos previamente estabelecidos. Para que haja essa verificação, pode ser usado um checklist, ou seja, um documento com todos os tópicos tratados no decorrer da elaboração do projeto.

b) GRE2 - O comprometimento da equipe técnica com os requisitos aprovados é obtido. Ao aprovarem-se os requisitos formaliza-se o comprometimento da equipe técnica para com os requisitos aprovados. O registro formal pode ser feito por meio de ata de reunião, e-mail ou outro mecanismo. Quando há mudanças de requisitos aprovados pelos fornecedores, tais como alterações ou criações, o compromisso já estabelecido pela equipe técnica pode ser afetado. Assim, quando isso ocorrer, a equipe técnica deverá obter e registrar os requisitos após terem sido novamente aprovados a partir dos critérios estabelecidos na GRE1.

c) GRE3 – A Rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida. Este resultado indicará se há necessidade de se estabelecer um mecanismo que permita a rastreabilidade e a dependência dos requisitos e os produtos de trabalho. Quando já definida a rastreabilidade, está facilitará o processo de avaliação de impacto das mudanças que os requisitos podem ocasionar. Caso não haja um sistema de rastreamento, deve ser criado um mecanismo (sistema) que possibilite a realização da rastreabilidade bidirecional entre os requisitos e os produtos de trabalho.

d) GRE4 – Revisões em planos e produtos de trabalho do projeto são realizadas visando a identificar e corrigir inconsistências em relação aos requisitos. Este resultado tem por intuito a realização de revisões, por meio de algum mecanismo equivalente, para identificar inconsistências entre os requisitos e os demais elementos do projeto, tais como, planos, atividades e produtos de trabalho. Quando inconsistências forem identificadas, elas devem ser registradas, executando-se ações corretivas a fim de resolvê-las. Quando houver mudanças nos requisitos, é importante examinar se os demais artefatos estão consistentes com as alterações realizadas. As ações criadas para a correção das inconsistências devem ser acompanhadas até que sejam totalmente resolvidas.

e) GRE5 – Mudanças nos requisitos são gerenciadas ao longo do projeto. Durante a execução do projeto, os requisitos podem ser modificados por uma série de motivos. Assim, requisitos adicionais podem ser incorporados ao projeto, enquanto outros podem ser retirados ou sofrerem alterações. . Quando ocorrerem mudanças, elas devem ser registradas e os requisitos podem ser revistos conforme definido na GRE. Um histórico de decisões acerca dos requisitos deverá ser disponibilizado. As decisões deverão ser tomadas após a análise de impacto que inclui aspectos como a influência em outros requisitos, a expectativa dos interessados, o esforço, o cronograma, os riscos e o custo.

3 MODELO DE PROCESSO

O modelo desenvolvido está baseado nos resultados esperados pelo modelo de referência do MPS.BR. Onde neste modelo propõe o gerenciamento e desenvolvimento de requisitos, conforme a figura 1.

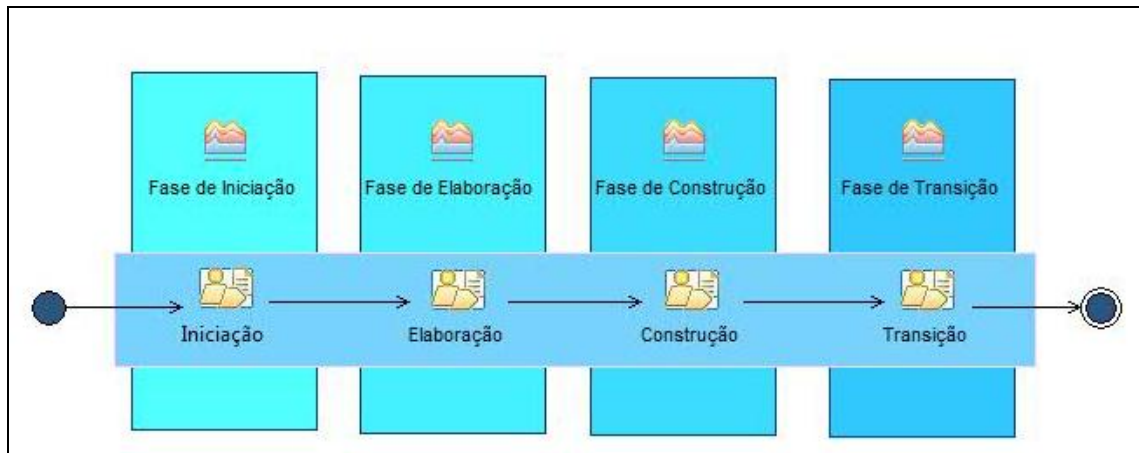


Figura 1. Processo Desenvolvido

No modelo acima foi definidos o papéis, responsabilidades, ferramentas, passos para execução das atividades, fluxo de atividades, mapeamento da atividade com o MPS-BR (prática que aquela atividade atende), artefatos e tarefas, que auxiliarão todos os profissionais envolvidos no processo. Segue abaixo descrição e composição de cada fase do processo.

Fase de Iniciação: nesta fase será dado o start do processo, ou seja, nesta fase são executadas as atividades de Levantar Requisitos e de Gerenciar Requisitos, onde a atividade de Levantar Requisitos está em processo de estudo pela melhor e mais adequada forma de relacionar os requisitos e a atividade de Gerenciar Requisitos está em processo de especificação o modo pelo qual os requisitos serão Gerenciados.

Fase de Elaboração: após a execução da atividade de levantamento de requisitos da primeira fase, os mesmos serão repassados para a fase de Elaboração, onde está é composta pelas atividades de Modelar e Validar os requisitos. No processo de Modelar os requisitos serão adotados as principais modelagem e o processo de validação de requisitos consiste na verificação junto ao cliente se os requisitos estão de acordo com a sua necessidade.

Fase de Construção: está fase é iniciada após a validação dos requisitos, ou seja, após a verificação junto ao cliente, caso os requisitos estejam de forma correta os mesmo serão encaminhados a está fase, onde nesta consta o artefato de Desenvolver os Requisitos, que é neste momento que os requisitos passam a ser desenvolvido na ferramenta EA (Enterprise Architect).

Fase de Transição: Após todos os requisitos levantados, modelados, validados e de construídos estão passam para de transição, ou seja, fase de documentação para posterior entregar a equipe responsável pelo desenvolvimento do projeto.

4 CONCLUSÃO

Concluí-se que o MPS.BR tem como principal propósito melhorar os processos do software, de forma que a mobilização vise à criação e aprimoramento do modelo em conformidade com as Normas Internacionais ISO/IEC 12207 [ISO/IEC, 2008a] e na ISO/IEC 15504 [ISO/IEC, 2003], com isso seja compatível com o as melhores práticas da engenharia de software e que seja adequado á realidade das empresas brasileiras. A adoção deste modelo pra as empresas é de custo razoável, tem todas as regiões Brasileiras.

5 REFERÊNCIAS BIBLIOGRÁFICAS

SOFTEX, MPS-BR - **Melhoria de Processo do Software Brasileiro**: Guia Geral. Maio de 2009. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS-BR_Guia_Geral_2009.pdf>. Acesso em: 03 nov. 2009.

SOFTEX, MPS-BR - **Melhoria de Processo do Software Brasileiro**, Guia de Implementação, Parte 1 - Fundamentação para implementação do Nível G do MR – MPS. Maio de 2009. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS-BR_Guia_de_Implementacao_Parte_1_2009.pdf>. Acesso em: 03 nov. de 2009.

LIEBMAN, Alessandro. **Melhoria no processo de software: implantação do MPS-BR nível G em uma empresa de pequeno porte**. 2006. 43 f. Monografia (Bacharel em Ciência da Computação) - Universidade Federal de Lavras, Lavras, 2006. Disponível em: <http://www.bcc.ufla.br/monografias/2005/Melhoria_no_processo_de_software_implantacao_do_MPS_BR_nivel_G_em_uma_empresa_de_pequeno_porte.pdf>. Acesso em: 28 out. 2009.

NETO, Alfredo, C.. **Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS-BR**. 2008. 182 p. Tese (Doutorado em Engenharia de Produção) – Programa de Pós – Graduação e Área de Concentração em Engenharia de Produção, Escola de Engenharia de São Carlos Universidade de São Paulo, 2008.