

RICARDO EMERSON JULIO

**ELABORAÇÃO E MODELAGEM DE UM PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE ADERENTE AO MPS.BR NÍVEL G**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

**LAVRAS
MINAS GERAIS – BRASIL
2007**

RICARDO EMERSON JULIO

**ELABORAÇÃO E MODELAGEM DE UM PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE ADERENTE AO MPS.BR NÍVEL G**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:
Engenharia de Software

Orientador:
Prof. Antônio Maria Pereira de Resende

**LAVRAS
MINAS GERAIS – BRASIL**

2007

**Ficha Catalográfica preparada pela Divisão de Processos Técnico
da Biblioteca Central da UFLA**

Julio, Ricardo Emerson

Elaboração e modelagem de um processo de desenvolvimento de software de acordo com o MPS.BR nível G / Ricardo Emerson Julio. Lavras – Minas Gerais, 2007.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Computação. 2. Engenharia de Software. 3. Qualidade de Processos.
I. JULIO, R. E. II. Universidade Federal de Lavras. III. Título.

RICARDO EMERSON JULIO

**ELABORAÇÃO E MODELAGEM DE UM PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE ADERENTE AO MPS.BR NÍVEL G**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do Curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 11 de fevereiro de 2008

Prof. Dr. Antônio Maria Pereira de Resende
(Orientador)

Prof. Dr. Heitor Augustus Xavier Costa

Prof. M. Eng. Paulo Henrique de Souza Bermejo

**LAVRAS
MINAS GERAIS – BRASIL**

Dedico esse trabalho aos meus pais, Dona Tina e João, às minhas irmãs, Rosângela, Adriana e Nayara, e à princesinha Paulinha pelo apoio e por sempre acreditarem que a realização do sonho era possível. Amo muito vocês! Obrigado por tudo...

Agradeço, primeiramente a Deus por estar sempre presente, me dando forças para continuar. Graças a Ele tenho o que agradecer e para quem agradecer.

Aos meus pais, pelo total apoio durante todo esse tempo. As vezes me pego dizendo que tudo o que eu SOU devo à minha mãe, Dona Tina, e tudo o que TENHO, devo ao meu pai, João. Realmente hoje vejo que isso é verdade. Amo muito vocês!

Às minhas irmãs, Rosângela, Adriana e Nayara, pelo orgulho e confiança depositados em mim. Sempre estiveram ao meu lado e foram, talvez sem que saibam, parte fundamental nessa conquista.

A todos os parentes que sempre estiveram do meu lado. Principal atenção à minha avó Dona Nazaré e ao meu avô Sr. Antônio. Aos tios e tias, primos e aos meus sobrinhos: Netinho, Pedro Henrique, Raissa e Paulinha. Amo vocês!

À minha princesinha Paula. Sem palavras para descrever...

Aos meus irmãos por consideração, André, Cidão, Juninho, Fábio, Thiago, Diego e p.S.

À família que Deus me permitiu escolher: meus amigos! Sempre muito importantes em tudo na minha vida. Sem vocês a vida não teria tanta graça. São tantos que seria impossível citar todos...

Àquelas que se tornaram “paixões”...

À galera do Apartamento 305, que se tornou minha casa e segunda família durante esses anos. Obrigado por tudo!

Ao meu orientador, professor Antônio Maria Pereira de Resende e à todos professores do Departamento de Ciência da Computação da UFLA e à Ângela e Deivson

Por fim, à todos colegas de faculdade, à todos que contribuíram de alguma forma e à UFLA, que foi o palco dessa batalha.

ELABORAÇÃO E MODELAGEM DE UM PROCESSO DE DESENVOLVIMENTO DE SOFTWARE ADERENTE AO MPS.BR NÍVEL G

RESUMO

Este trabalho apresenta a elaboração e a modelagem de um processo de desenvolvimento de software, para uma fábrica de software, aderente ao MPS.BR nível G. O processo foi desenvolvido de acordo com as atividades essenciais para a construção de produtos de software denominadas Análise e Definição de Requisitos, Projeto Arquitetural e Detalhado, Codificação, Testes e Entrega e engloba desde o estabelecimento do contrato inicial até a fase de entrega, homologação do produto e manutenção do software. Após a elaboração, o processo foi modelado através de uma representação com notação gráfica, indicando os papéis, artefatos de entrada e de saída, atividades e fluxo das atividades, facilitando a compreensão e o acompanhamento do processo. Finalmente, uma adequação do processo desenvolvido ao conjunto de resultados esperados pelo MPS.BR nível G foi demonstrado através de artefatos e atividades, presentes no processo, que implementam esses resultados.

Palavras-chave: Processo de Desenvolvimento de Software, Melhoria de Processo de Software, Qualidade de Software.

ELABORATION AND MODELLING OF A SOFTWARE DEVELOPMENT PROCESS ADHERENT TO MPS.BR LEVEL G

ABSTRACT

This work presents the elaboration and modelling of a software development process, for a software factory, adherent at the MPS.BR level G. The process was developed in agreement with the essential activities for the construction of software products denominated Analysis and Definition of Requirements, Project, Code, Tests and Delivery and it includes from the establishment of the initial contract to the delivery phase, approval of the product and maintenance.. After the elaboration, the process was modeled through a representation with graphic notation, indicating the roles, artifacts, activities, and activities flow, facilitating the understanding and accompaniment of the process. Finally, an adaptation of the process developed to the group of expected results by the MPS.BR level G it was demonstrated through artifacts and activities, presents in the process, that implement those results.

Key-words: Software Development Process, Software Process Improvement, Quality of Software.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1. INTRODUÇÃO	1
1.1. Motivação	1
1.2. Objetivos.....	2
1.3. Estrutura do Trabalho	3
2. PROCESSOS DE SOFTWARE	4
2.1. Considerações Iniciais	4
2.2. Definição	4
2.3. Atividades Essenciais	5
2.4. Atividades Auxiliares	7
2.5. Garantia de Qualidade de Software	8
2.6. Modelos de Processo de Software	11
2.7. Considerações Finais	14
3. FÁBRICA DE SOFTWARE	16
3.1. Considerações Iniciais	16
3.2. Conceitos	16
3.3. Tipos de Fábrica de Software	18
3.4. Processos de Desenvolvimento em Fábricas de Software	20
3.5. Considerações Finais	22
4. O MODELO MPS.BR	23
4.1. Considerações Iniciais	23
4.2. Introdução	23
4.3. Objetivos	24
4.4. MPS.BR	25
4.5. O MPS.BR nível G	31
4.6. Considerações Finais	38
5. METODOLOGIA	39
3.1. Tipo de Pesquisa	39
3.2. Procedimentos Metodológicos	39
6. RESULTADOS E DISCUSSÃO	41
6.1. Considerações Iniciais	41
6.2. O Processo de Desenvolvimento de Software	41
6.2.1. Análise de Negócio	43
6.2.2. Análise de Requisitos e Planejamento	46
6.2.3. Modelagem e Implementação	51
6.2.4. Entrega e Manutenção	56

6.2.5. Considerações finais sobre o processo	56
6.3. Aderência ao MPS.BR nível G	60
6.3.1. Gerência de Projetos (GPR)	60
6.3.2. Gerência de Requisitos (GRE)	66
6.4. Considerações Finais	67
7. CONSIDERAÇÕES FINAIS	69
7.1. Conclusões	69
7.2. Contribuições	70
7.3. Trabalhos Futuros	70
REFERÊNCIAS BIBLIOGRÁFICAS	71

LISTA DE FIGURAS

Figura 2.1 – Um diagrama do ciclo de vida cascata (<i>waterfall</i>)	13
Figura 2.2 – Exemplo da representação de um processo	14
Figura 4.1 – Componentes do MPS.BR (Fonte: MPS.BR – Guia Geral (2006))	25
Figura 4.2 – Processos do MPS.BR (Fonte: MPS.BR – Guia Geral (2006)).....	28
Figura 4.3 – Níveis de Maturidade (Fonte: MPS.BR – Guia Geral (2006))	31
Figura 6.1 – Ciclo de vida do processo	41
Figura 6.2(a) – Análise de Negócio	44
Figura 6.2(b) – Análise de Negócio	45
Figura 6.3(a) – Análise de Requisitos e Planejamento	48
Figura 6.3(b) – Análise de Requisitos e Planejamento	59
Figura 6.4(a) – Modelagem e Implementação	51
Figura 6.4(b) – Modelagem e Implementação	52
Figura 6.5 – Testes	53
Figura 6.6 – Entrega	55
Figura 6.7(a) – Manutenção	56
Figura 6.7(b) – Manutenção	57
Figura 6.8(a) – Gerência de Projetos	62
Figura 6.8(b) – Gerência de Projetos	63
Figura 6.9 – Processo disponibilizado em um ambiente Web	66

LISTA DE TABELAS

Tabela 2.1 – Características do processo (Fonte: Sommerville, 2003)	10
Tabela 6.1 – Relação das atividades essenciais com as fases do processo	42
Tabela 6.2 – Documentação de atividades de uma fase	43
Tabela 6.3 – Principais artefatos do processo	54
Tabela 6.4 – Implementação dos resultados esperados (GPR)	61
Tabela 6.5 – Implementação dos resultados esperados (GRE)	65

1. INTRODUÇÃO

O desafio fundamental que a Engenharia de Software enfrenta é o mesmo há décadas: Como construir software com qualidade maior, custos menores e dentro do prazo? Entre processos, ferramentas e organizações alternativas, a literatura consolida experiência e inovação resultantes do trabalho de milhares de pesquisadores, motivados em descrever (e prescrever) formas mais eficazes de lidar com os problemas inerentes à construção de software (PFLEEGER, 2004). Resulta destes esforços um conjunto de modelos e normas: CMM, SPICE, ISO/IEC 12.207 e MPS.BR (ISO, 1995).

A exigência do mercado pela qualidade de produtos e/ou de serviços vem-se intensificando. Conseqüentemente, a preocupação com a melhoria do processo de desenvolvimento é impulsionada nas empresas de software. Estas empresas têm investido fortemente na correção de seus processos e em sua capacitação para se tornarem mais competitivas.

1.1 Motivação

A qualidade dos processos das empresas desenvolvedoras afeta diretamente o produto ou o serviço gerado por estes processos, tornando-se uma vantagem competitiva entre elas. Mas, para atingir esta qualidade, as empresas sofrem inúmeros desafios, alguns apresentados desde o início deste século, a partir da globalização da economia de software e maior competitividade no mercado (CÂNDIDO, 2004).

Ter um processo de software desordenado e confuso resulta em ineficiência, desorganização, falta de qualidade e perda na competitividade, o que desperta interesses por parte das empresas, em investir na melhoria de seus processos, através da implantação de processos para garantia da qualidade. Para empresas de desenvolvimento de produtos de software, desenvolvê-los em tempo hábil, com custo mínimo e com alta qualidade são fatores essenciais para elas terem preferência no mercado. Segundo Oliveira (2007), observa-se um aumento da conscientização das empresas brasileiras de desenvolvimento de produtos de software em relação à qualidade de seus produtos e serviços. Mas, a qualidade de um produto de software deve ser observada durante a execução do processo e não após a sua finalização.

Ainda de acordo com Oliveira (2007), a garantia da qualidade do produto ou do

serviço implica em lucros para a empresa, redução de gastos com re-trabalho, aumento da produtividade e uso racional dos recursos dentro da empresa. Ao aplicar um processo de qualidade em uma organização, além destas vantagens, pode-se observar a redução do risco de falhas de um projeto. Ter garantia e controle da qualidade de produtos e de serviços em empresas de software resulta em alta disponibilidade, precisão e performance para as mais complexas atividades do ciclo de desenvolvimento e, conseqüentemente, o aperfeiçoamento da qualidade da atividade durante a execução do processo.

A qualidade do produto ou do serviço de software, obtida por meio da melhoria da qualidade do processo, pode ser alcançada mais facilmente através da aplicação de modelos de melhoria de processo considerados repositórios de melhores práticas desenvolvidas ao longo de vários anos com sucesso. Além disso, esses modelos definem os requisitos que os processos devem atender, apresentando flexibilidade em relação à aderência e permitem avaliações dos processos de forma objetiva e a detecção de pontos fortes e fracos.

Modelos e normas como CMMI, ISO 15504, ISO 12207 e MPS.BR definem um caminho evolucionário para melhoria de processo. O MPS.BR foi escolhido para o desenvolvimento deste trabalho entre os modelos existentes para melhoria de processo, pois, além de ter uma representação estagiada, proporciona melhoria de processo a baixo custo, o que o torna acessível a grande massa de micro, pequenas e médias empresas brasileiras de software. A maioria destas empresas possui poucos recursos e necessitam de grande melhora no seu processo em um curto espaço de tempo para se tornarem competitivas no mercado.

1.2. Objetivos

O objetivo geral deste trabalho consiste em elaborar um processo de desenvolvimento de software para fábricas de software de pequeno porte respeitando o primeiro nível de maturidade do MPS.BR (nível G) que possa ser controlado, utilizado para o desenvolvimento de produtos de software de qualidade e que atenda às necessidades da empresa.

Para o cumprimento do objetivo geral proposto, os seguintes objetivos específicos foram definidos:

- Estabelecer um processo de desenvolvimento de software, adequado ao MPS.BR

nível G, que envolva desde o estabelecimento do contrato inicial com um cliente até as fases de entrega, de homologação do produto e de manutenção do software e que seja aplicável em fábricas de software de pequeno porte;

- Fazer uma modelagem do processo através de uma representação com notação gráfica, indicando os papéis, os artefatos de entrada e de saída, as atividades e os fluxos das atividades, devidamente separado nas principais fases da Engenharia de Software e de Gerência de Projetos, a saber: Iniciação, Elicitação de Requisitos, Análise, Projeto, Codificação, Testes, Manutenção e Finalização.

1.3 Estrutura do Trabalho

Os capítulos subsequentes deste trabalho de pesquisa estão organizados em seis capítulos como descrito a seguir.

No Capítulo 2, é apresentado o conceito de processo e qualidade de software. O Capítulo 3 descreve Fábricas de Software e apresenta uma classificação segundo os tipos e os processos. O Capítulo 4 apresenta os conceitos e uma explicação sobre o MPS.BR e as normas e modelos em que foi baseado.

A metodologia e o tipo de pesquisa utilizado para a execução deste trabalho, bem como as ferramentas e as técnicas são descritas no Capítulo 5.

O Capítulo 6 apresenta o processo desenvolvido e a sua aderência ao modelo MPS.BR nível G. O Capítulo 7 apresenta as conclusões, as contribuições e sugestões de trabalhos futuros.

2. PROCESSOS DE SOFTWARE

2.1 Considerações Iniciais

Neste Capítulo, são abordados os conceitos referentes a processos de software. Primeiramente, são mostradas as principais definições de processo de software encontradas na literatura. Em seguida, é apresentada uma explicação sobre as atividades consideradas essenciais e auxiliares na construção de software. Por fim, apresenta-se os principais conceitos de Garantia de Qualidade de Software e os modelos de processo de software que descrevem o ciclo de vida de um projeto de construção de software.

2.2 Definição

Um processo pode ser entendido como um conjunto de tarefas ordenadas ou uma série de etapas. Geralmente, ele envolve ferramentas e técnicas que prescrevem as suas atividades, utiliza recursos e está sujeito a um conjunto de restrições, gerando produtos intermediários e finais. Cada atividade do processo tem critérios de entrada e saída, de modo que possa ser possível saber quando o processo começa e quando termina (PFLEEGER, 2004).

Com base nessas afirmações, uma definição útil de processo de software para esse trabalho pode ser sintetizada como um conjunto de atividades realizadas para construir software, levando em consideração os produtos sendo construídos, as pessoas envolvidas, as ferramentas com as quais trabalham e a matéria prima utilizada (PFLEEGER, 2004).

Quando o processo envolve a elaboração de um produto, algumas vezes refere-se a ele como ciclo de vida. Assim, o processo de desenvolvimento de software pode ser chamado de ciclo de vida do software, pois ele descreve a vida do produto de software desde a concepção até a implementação, entrega, utilização, implantação e manutenção.

Os processos são importantes porque imprimem consistência e estrutura a um conjunto de atividades orientando as ações, permitindo examinar, entender, controlar e aprimorar as atividades que o compõe. Segundo Lonchamp (1993), processo de desenvolvimento de produtos de software é composto por passos ordenados com o propósito de produzir e manter produtos de software finais requeridos.

Pessoa (2003) define que um bom funcionamento do processo de desenvolvimento de produtos de software apóia-se na integração perfeita de suas atividades promovendo o

desenvolvimento eficaz.

Ainda de acordo com Pessoa (2003), é importante ressaltar que um processo de software particular, da forma como implementado por uma organização, é algo individualizado: cada equipe de desenvolvimento instancia sua versão. Uma descrição de um processo de software específico oferece uma forma de entender e avaliar seus objetivos, forças e fraquezas.

2.3 Atividades Essenciais

Segundo Pfleeger (2004), as atividades consideradas essenciais para a construção de software são Análise e Definição de Requisitos, Projeto Arquitetural e Detalhado, Codificação, Teste de Unidade, de Integração e de Sistema e Entrega. Estas atividades podem ser resumidas como:

- **Análise e Definição de Requisitos:** é o processo de descobrir e de detalhar as funções e as limitações funcionais do software; um requisito é a descrição de algo que o sistema deve ser capaz de realizar. Em projetos convencionais, é freqüentemente realizado um processo de engenharia de requisitos com a participação do usuário final do sistema, pois normalmente ele é bom conhecedor do domínio do problema a ser solucionado. Freqüentemente, as atividades do processo de software são associadas a um ou mais artefatos: são seus produtos concretos. O artefato convencionalmente associado à atividade de análise de requisitos é um documento de especificação de requisitos;
- **Projeto Arquitetural e Detalhado:** esta atividade é um detalhamento iterativo de uma solução computacional para atender aos requisitos elicitados. O projeto é, normalmente, feito em níveis; nos níveis mais abstratos, são definidas as entradas e as saídas do software e as entidades com quais ele se relaciona. Progressivamente, refina-se este projeto em partes menores: sub-sistemas, módulos, componentes, classes, funções e variáveis. Aos níveis mais abstratos, é dado o nome de projeto arquitetural; e aos níveis mais refinados, projeto detalhado. Existem diversas técnicas para realizar este refinamento e descrevê-lo; em geral, os artefatos desta atividade são descrições textuais e diagramas que descrevem os componentes e seus relacionamentos. Existem diversas notações diferentes para projeto, incluindo a Linguagem de Modelagem Unificada (*Unified Modelling Language – UML*);

- **Codificação:** consiste na criação do código-fonte cujo objetivo é implementar a funcionalidade especificada nos requisitos, de acordo com o modelo definido no projeto. Esta atividade envolve escrever código, depurá-lo e integrá-lo progressivamente. Para a realização desta atividade, desenvolvedores utilizam artefatos de requisitos e de projeto produzidos e interagem com as equipes responsáveis por estes;
- **Teste de Unidade, de Integração e de Sistema:** embora seja comum testes *ad hoc*, isto é, testes não devidamente projetados, durante o processo de codificação, a atividade de teste envolve uma abordagem mais sistemática. Assim como no projeto, o teste pode ser descrito como uma atividade progressiva, indo do elemento menor para o maior. Por exemplo, o teste de unidade cobre partes individuais do sistema (funções, módulos e interfaces); o teste de integração verifica o funcionamento destas partes uma vez combinadas. O teste de sistema valida os aspectos funcionais do sistema, incluindo desempenho, adequação e consistência da funcionalidade implementada;
- **Entrega:** uma vez pronto e validado, o sistema pode ser entregue e pode-se consolidar a documentação e ajuda *online*, o empacotamento do software e a sua distribuição, entrega ou instalação. Considera-se como uma fase crítica pelo fato de ser o primeiro contato do usuário com o software. Em processos iterativos ou cíclicos, torna-se comum ocorrerem múltiplas mini-entregas; neste caso, a atividade de entrega tem impacto minimizado pelo fato de reduzir os riscos associados à entrada do sistema no ambiente do usuário; esta é feita gradualmente, com cada mini-entrega individual.

De acordo com Davis *et al* (1994), estas atividades não são necessariamente seqüenciais, freqüentemente ocorrendo superposição entre elas por períodos de tempo. Em outras palavras, não se pode determinar uma relação temporal estrita entre elas; existe, na realidade, uma relação mais sutil de interdependência que tende a ordenar sua realização.

Por exemplo, a atividade de implementação requer que se saiba o que está sendo construído, de forma que depende de resultados obtidos durante a análise de requisitos. No entanto, não é realista dizer que para todo projeto a implementação só se inicia quando os requisitos estiverem completamente levantados e documentados.

2.4 Atividades Auxiliares

Segundo Reis (2003), além das atividades essenciais da seção anterior, um processo de desenvolvimento de software eficaz inclui algumas atividades diferenciadas. Estas atividades existem para viabilizar e apoiar a execução das essenciais, sendo voltadas para otimizar recursos e minimizar os riscos. De maneira resumida, as atividades auxiliares fundamentais incluem:

- **Gerência de Alocação:** A tarefa de gerência de alocação consiste em atribuir e em administrar recursos (espaço, equipamento) e pessoas entre as tarefas do projeto, com o objetivo de maximizar a produtividade da equipe;
- **Gerência de Cronograma:** normalmente, software é desenvolvido com base em um prazo esperado para sua entrega; em projetos iterativos ou de maior escala, é freqüente alguns marcos intermediários que permitem verificar o progresso do trabalho. Gerência de cronograma envolve administrar um equilíbrio entre a funcionalidade esperada, o desempenho da equipe e as datas chaves do projeto;
- **Gerência de Configuração:** este termo cobre um conjunto de políticas e de atividades que têm como objetivo controlar os artefatos do projeto (incluindo código-fonte), possibilitando revisar, armazenar, auditar e relatar as alterações produzidas ao longo do tempo. Uma parte importante da gerência de configuração é o controle de versões do software, normalmente assistido por uma ferramenta de software como ClearCase, SourceSafe e CVS. Outras atividades incluem revisão e inspeção do código, auditoria pós-integração e relatórios de alterações;
- **Documentação:** ao longo do processo de construção do software, os artefatos, o código e a funcionalidade são descritos textual ou graficamente na forma de documentação. Existem diversos produtos diferentes que podem ser considerados documentação. A criação, o gerenciamento e a atualização da documentação são considerados essenciais para um processo saudável.

Em organizações mais complexas e em projetos de grande porte, podem ocorrer subdivisões importantes destas atividades auxiliares, assim como a inclusão de outras atividades específicas.

2.5 Garantia de Qualidade de Software

Ao conjunto de esforços feitos para manter qualidade ao longo do processo é dado o nome de Garantia de Qualidade de Software (ou QA, de *[Software] Quality Assurance*). QA não é apenas uma atividade, mas uma série de micro-processos e políticas estabelecidas para supervisionar a qualidade dos artefatos e do próprio processo (REIS, 2003).

Ainda de acordo com Reis (2003), entre estas atividades, pode-se destacar como importantes do ponto de vista de QA:

- revisão e auditoria de artefatos, incluindo código-fonte;
- levantamento e análise de métricas;
- teste, incluindo teste público (*alfa e beta*);
- padronização interna (políticas e regras internas à organização);
- padronização e certificação externa (em relação aos padrões estabelecidos).

Esta lista não é exaustiva, mas oferece uma idéia do tipo de garantia que QA busca oferecer.

A qualidade de produto e a qualidade de processo são complementares e interdependentes. Espera-se que a qualidade do processo de fabricação tenha impacto positivo sobre o produto de software obtido. Entretanto, tal objetivo será atingido se houver compreensão clara que os processos devem fornecer mecanismos necessários para especificar o produto e controlar a sua fabricação.

A qualidade do processo de software é determinada pelo grau de flexibilidade para incorporar características implícitas de qualidade de produto e novos métodos, técnicas e ferramentas ao processo de desenvolvimento de produtos de software. Este processo, de acordo com Pessoa (2003), é a seqüência de passos para construção de um produto de software e abrange as relações com o fornecedor e com o cliente, gerenciamentos (de projeto, de qualidade, de configuração, de requisitos, de custo, de tempo e de risco) e a engenharia do produto. A qualidade de um produto de software é conseguida de forma consistente, em longo prazo, a partir da qualidade do processo.

Segundo Vasconcelos *et al* (2003), os avanços tecnológicos, a crescente preocupação na eliminação de defeitos, o aumento na produtividade e a redução de custos motivaram o surgimento de modelos de qualidade para o processo de manufatura. A partir da década de 60, começaram a surgir critérios, modelos e técnicas para a garantia da

qualidade no processo de produção.

Ainda de acordo com Vasconcelos *et al* (2003), diversos modelos de qualidade de software vêm sendo propostos ao longo dos últimos anos. Esses modelos têm sido fortemente adotados por organizações em todo o mundo.

Em 1947, a criação da ISO (*International Standard Organization*) formalizou a necessidade da definição de padrões internacionais no setor da indústria e muito contribuiu para a evolução do setor, definindo normas para a garantia da qualidade direcionada para produção, serviços, gerenciamento entre outros contextos. Esta organização é não governamental, elabora normas internacionais e tem a missão de promover o desenvolvimento da normalização, com objetivo de facilitar a troca internacional de bens e serviços e a cooperação no desenvolvimento intelectual de atividades científicas, tecnológicas e econômicas (Moreira, 2004).

As normas ISO têm como objetivo estabelecer um conjunto de padrões para as organizações do mundo. No Brasil, a ISO é representada pela ABNT (Associação Brasileira de Normas Técnicas).

Para alcançar a qualidade, a Engenharia de Software utiliza melhoria de processos e mecanismos para planejar e gerenciar o processo de desenvolvimento. Processos de software são complexos e abrangem alto número de atividades inter-relacionadas. Com isso, segundo Sommerville (2003), melhorar o processo de software de uma empresa não significa simplesmente admitir métodos ou ferramentas específicas ou algum modelo de processo que tenha sido utilizado em outro projeto, pois os processos de software são inerentemente complexos e envolvem um número muito grande de atividades.

Ainda segundo Sommerville (2003), durante os últimos anos, ampliou-se o interesse por parte das organizações que desenvolvem software pela melhoria de seus processos. A melhoria do processo significa compreender os processos existentes e modificá-los, a fim de melhorar a qualidade do produto e/ou reduzir os custos e o tempo de desenvolvimento. A maior parte da literatura relacionada a esse assunto tem se concentrado em aprimorar os processos para melhorar a qualidade do produto e, em particular, para reduzir o número de defeitos nos software fornecidos. Uma vez que esse objetivo é alcançado, a redução dos custos e do tempo pode se tornar a principal meta da melhoria.

Assim como os produtos, os processos também têm atributos ou características (Tabela 2.1).

Tabela 2.1 – Características do processo (Fonte: Sommerville, 2003)

Característica do processo	Descrição
Facilidade de compreensão	Até que ponto o processo está explicitamente definido e com que facilidade se pode compreender a definição do processo?
Visibilidade	As atividades de processo culminam em resultados nítidos, de modo que o progresso do processo seja externamente visível?
Facilidade de suporte	Até que ponto as atividades do processo podem ser apoiadas por ferramentas CASE?
Aceitabilidade	O processo definido é aceitável e utilizável pelos engenheiros responsáveis pela produção do produto de software?
Confiabilidade	O processo está projetado de tal maneira que seus erros sejam evitados ou identificados antes que resultem em erros no produto?
Robustez	O processo pode continuar, mesmo que surjam problemas inesperados?
Facilidade de manutenção	O processo pode evoluir para refletir os requisitos mutáveis da organização ou melhorias de processo identificadas?
Rapidez	Com que rapidez pode ser concluído o processo de entrega de um sistema, a partir de uma determinada especificação?

Executando as atividades de um processo corretamente, ocorrerá a melhoria do processo. Estas atividades dentre outras, podem ser (SOMMERVILLE, 2003):

- Análise de processo: estuda os processos existentes e cria um modelo específico

para documentar e entender o processo;

- Identificação de melhoria: a melhoria do processo deve visar à eliminação dos problemas que se opõem à qualidade do produto, encontrados no estágio anterior de análise de processo. A melhoria também deve expor novos procedimentos, métodos e ferramentas para a resolução destes problemas;
- Introdução de mudanças de processo: estabelecem novos procedimentos, métodos, ferramentas e os junta com outras atividades de processo. No entanto, é preciso ter tempo para implantar estas alterações e garantir que sejam conciliáveis às outras atividades do processo, métodos e padrões da empresa;
- Treinamento em mudanças de processo: melhorias impostas sem um treinamento adequado têm resultado contrário à qualidade do produto. Esta fase é considerada essencial ao processo de melhoria;
- Ajuste de mudanças: as alterações serão eficientes após os problemas encontrados com a mudança no processo sejam eliminados. A fase de ajuste propõe e aplica novas modificações com os erros gerados na mudança.

A implementação de programas de melhoria de processo tem se mostrado como um ponto principal para o sucesso das empresas desenvolvedoras de produtos de software. Mas, isto depende do cumprimento com as metas determinadas, da disponibilidade de recursos e do apoio e participação dos colaboradores da empresa.

2.6 Modelos de Processo de Software

Para auxiliar a compreensão do processo de software, alguns autores, como Pfleeger (2004), Pressman (2002) e Sommerville (2003), oferecem modelos que descrevem de forma mais concisa o ciclo de vida de um projeto de construção de software. Normalmente, são representados graficamente, através de um diagrama. Para Pfleeger (2004), alguns modelos são prescrições para o desenvolvimento de software demonstrando seu progresso e outros são descrições desse desenvolvimento na atualidade. Na teoria, os dois grupos de modelos podem ser similares ou iguais, mas na prática eles não são.

Pfleeger (2004) apresenta uma classificação dos modelos pré-existentes de acordo com a sua seqüencialidade:

- **Modelos Seqüenciais:** tem como característica uma seqüência linear de atividades; dos requisitos seguem projeto arquitetural e detalhado, codificação, e assim por

diante. Estes modelos são prescritivos uma vez que raramente são construídos software obedecendo uma seqüência estritamente linear. Existem diversas variantes dos modelos seqüenciais que buscam reduzir a sua estrita linearidade. Uma técnica utilizada se chama prototipação, que envolve a construção de um software que ofereça uma primeira experiência ao usuário final, mas que não implementa a funcionalidade completa. Após construído o protótipo, existem algumas alternativas: pode ser descartado em favor de implementar sistema completo ou pode ser evoluído gradativamente até atingir a funcionalidade requisitada;

- **Modelos Evolucionários ou Iterativos:** ao invés de uma seqüência linear, o software é gradualmente produzido e lançado; para cada ciclo de lançamento, as atividades básicas são realizadas de acordo com o ritmo de execução. Uma variante possível é um ciclo, normalmente aplicado a software que não contempla uma entrega final; em outras palavras, software que permanece em desenvolvimento contínuo, como é o caso de grande parte dos produtos de software modernos de consumo geral.

A Figura 2.1 apresenta um diagrama que descreve um dos primeiros modelos a ser propostos, o modelo cascata (*waterfall*). Embora seja um modelo prescritivo e diversas críticas à sua natureza tenham sido publicadas (PFLEEGER, 2004), o modelo cascata é pertinente para este trabalho por apresentar claramente uma distinção entre as atividades de desenvolvimento do projeto e a sua simplicidade e maior didática.

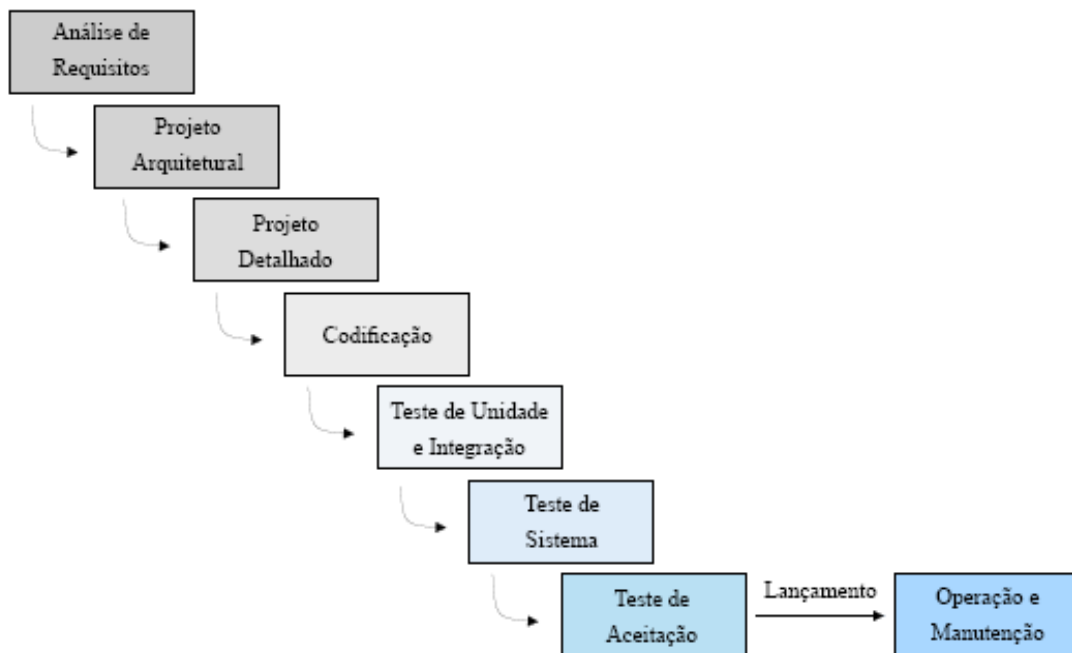


Figura 2.1 – Um diagrama do ciclo de vida cascata (*waterfall*)

Segundo Araújo *et al* (2001), um processo pode ser descrito por meio de atividades, artefatos de entrada e saída, papéis, rota ou fluxo das atividades e regras.

As atividades correspondem a uma etapa a ser executada dentro de um processo. Elas são definidas através de um nome, seus objetivos, instruções a serem seguidas durante a sua execução e os dados ou documentos necessários para a sua realização.

Um artefato é um produto de trabalho do processo. Os artefatos de entrada são utilizados durante a execução de atividades. As atividades ao serem executadas produzem artefatos, que são chamados de artefatos de saída.

A definição de um fluxo de atividades compreende a explicitação do encadeamento de atividades do processo. Esse encadeamento ocorre, de um modo geral, segundo um grafo orientado, onde atividades (representadas pelos nós do grafo) podem ser executadas em paralelo, segundo uma ordenação parcial, e/ou sequencialmente, quando o grafo orientado de execução de atividades particularizar-se em uma seqüência única de atividades. Expressões de guarda podem ainda ser aplicadas às arestas dos grafos para adicionar-se condições de execução de parte das atividades.

A execução de qualquer atividade em um ambiente organizado pressupõe a aceitação e obediência a regras pré-estabelecidas. Quando a execução de atividades deve ser feita de forma coordenada, envolvendo mais de um executor, as regras desempenham uma função vital para a consecução do objetivo final de um processo. Regras dizem

respeito a restrições e diretrizes impostas por um negócio e/ou pela cultura de uma organização. Regras definem quais informações irão transitar pelo fluxo e sob quais condições, ou seja, elas são atributos que definem de que forma os dados que trafegam no fluxo de trabalho devem ser processados, roteados e controlados.

A Figura 2.2 mostra uma representação gráfica de um processo de análise de currículos. Nesta representação, podem ser observados as atividades, o fluxo das atividades, os artefatos de entrada e saída e os papéis responsáveis por executar a atividade.

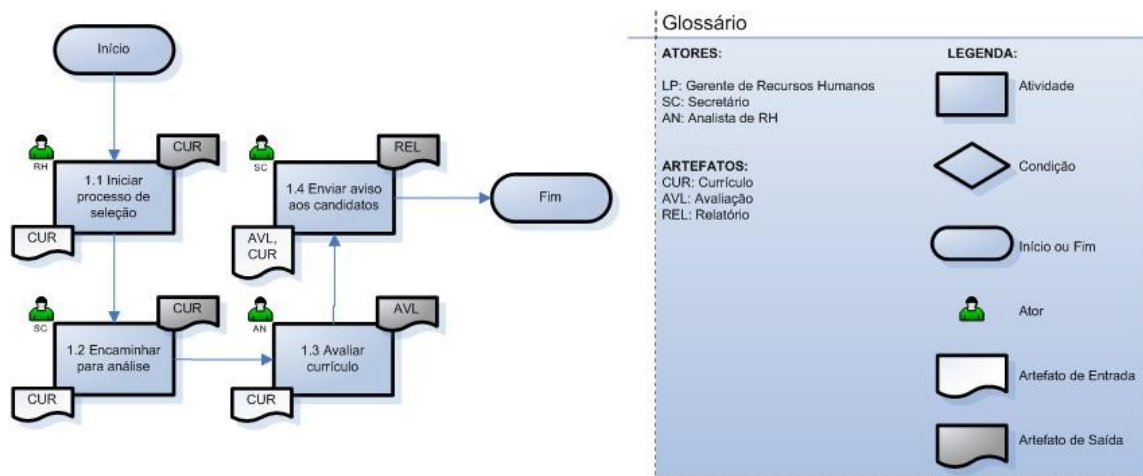


Figura 2.2 – Exemplo da representação de um processo

2.7 Considerações Finais

Os processos de software podem ser entendidos como um conjunto de etapas que descrevem o ciclo de vida do projeto de construção de software. Leva-se em consideração os produtos construídos durante o desenvolvimento, as pessoas envolvidas, as ferramentas e a matéria prima utilizada.

Algumas atividades são consideradas essenciais durante a criação de software. Além dessas atividades essenciais, um processo de desenvolvimento de software eficaz inclui algumas atividades auxiliares com o objetivo de otimizar recursos e minimizar os riscos.

Uma série de atividades são estabelecidas com o intuito de manter a qualidade ao longo do processo. Ao conjunto desses esforços, é dado o nome de Garantia de Qualidade de Software, que torna-se importante para a concepção de um produto de software de qualidade.

Para finalizar, foram apresentados alguns modelos de processo de software utilizados para auxiliar na compreensão do processo e descrever o ciclo de vida de produção de software.

3. FÁBRICA DE SOFTWARE

3.1 Considerações Iniciais

Este Capítulo descreve os principais conceitos de fábricas de software e os requisitos que devem ser atendidos para que uma organização seja classificada como fábrica de software.

Além disso, apresenta uma classificação segundo os tipos e os processos, descrevendo os tipos conforme o foco dentro do ciclo de desenvolvimento de software.

Para finalizar, são apresentados os principais processos de desenvolvimento encontrados em fábricas de software.

3.2 Conceitos

O conceito de fábrica de software vem sendo utilizado desde os anos 60, porém as pesquisas se intensificaram a partir da década de 80 devido ao crescimento deste modelo no cenário mundial. Países como o Brasil buscam atingir um modelo semelhante ao das fábricas de software da Índia, referência de qualidade e sucesso no desenvolvimento de produtos de software, para conseguir resultados tão positivos quanto. No entanto, para chegar neste padrão, deve-se aumentar o interesse e o investimento com processos e padrões de qualidade, pois eles interferem na qualidade final do produto.

Segundo Bemer *apud* Cusumano (1991), o termo fábrica de software é discutido desde sua primeira citação, final dos anos 60, e está evoluindo e refinando-se até os dias atuais. De acordo com Aaen *et al.* (1997), apesar das primeiras fábricas de software terem surgido no final da década de 60, o termo fábrica diverge quando o desenvolvimento de software é comparado à produção em massa.

Embora esta produção incluía operações centralizadas de larga escala, a associação do termo fábrica ao desenvolvimento de software propõe o uso de técnicas para produção em larga escala, de forma coordenada e de alta qualidade (CUSUMANO, 1991).

No entanto, mesmo fábrica de software sendo um modelo antigo, não tem sido adotado intensivamente. Os principais esforços partiam de associações entre institutos de pesquisa, projetos governamentais e grandes corporações de bens e de serviços de informática. Durante os últimos anos, em diferentes partes do mundo, surgiram iniciativas para criação de novas fábricas de software que adotaram estratégias diferentes

para organização e execução de seus processos, por exemplo, foco na adoção de processos ou priorização da adoção de ferramentas de automação. Com isso, o termo voltou a ganhar espaço dentro de grandes corporações como *Hewlett-Packard Development Company* (HP), *International Business Machines Corporation* (IBM) e *Unisys* (CASTOR, 2004).

Ainda de acordo com Castor (2004), a característica principal do modelo fábrica de software é a adoção de técnicas usadas na engenharia de produção em série com o objetivo de criar um ambiente de desenvolvimento produtivo de produtos de software com qualidade e baixo custo, visto que dois dos principais problemas encontrados neste ambiente são baixa produtividade e alto custo na produção de sistemas.

Outras características do modelo são: i) reusabilidade; ii) utilização de ferramentas para suportar o desenvolvimento; iii) sistemas de controle e gerenciamento; iv) modularização; e v) produção de famílias de produtos como básicas para uma organização (ROCHA *et al.*, 2004).

Em uma definição coerente e completa de Fernandes *et al.* (2004), é apresentada fábrica de software, caracterizando-a como um processo estruturado, controlado e melhorado de forma contínua, considerando abordagens de engenharia industrial orientado para o atendimento a múltiplas demandas de natureza e escopo distintos e visando à geração de produtos de software, conforme os requisitos documentados dos usuários e/ou clientes, da forma mais produtiva e econômica.

Ainda de acordo com Fernandes *et al.* (2004), para que uma instituição seja qualificada como fábrica de software, ela deve atender aos seguintes requisitos:

- processo definido e padrão (desenvolvimento, controle e planejamento);
- interação controlada com o cliente (entradas e saídas da fábrica);
- solicitações de serviço à fábrica de software devem ser padronizadas;
- estimativas de custos e prazos baseadas no conhecimento real da capacidade produtiva com métodos de obtenção baseados em dados históricos;
- controle rigoroso dos recursos envolvidos em cada demanda da fábrica de software;
- controle e armazenamento em bibliotecas de itens de software (documentos, código, métodos, etc);
- controle dos *status* e execução das demandas;
- produtos gerados de acordo com os padrões estabelecidos pela organização;
- equipe treinada e capacitada nos processos organizacionais e produtivos;
- controle da qualidade do produto;

- processos de atendimento ao cliente;
- métricas definidas e controle dos acordos de nível de serviço definidos com o cliente.

Neste trabalho, adota-se a definição de Siy *et al.* (2001) que propõe uma fábrica de software como sendo uma instituição que fornece serviços e produtos em um curto prazo de tempo, com alta qualidade e custos baixos, uma vez que o desenvolvimento utiliza processo bem definido e tecnologia avançada.

Embora algumas visões em relação ao termo diverjam, há pontos comuns como a produção de produtos de software de qualidade com baixo custo e alta produtividade. As diferenças estão relacionadas aos conceitos, aos processos utilizados no desenvolvimento e às diferentes formas de estruturas de uma fábrica de software (CASTOR, 2004).

3.3 Tipos de Fábrica de Software

Além das definições dos requisitos que devem ser atendidos para que uma organização seja classificada como fábrica de software, ainda podem ser reconhecidos quatro tipos de fábricas de software, conforme o foco dentro do ciclo de desenvolvimento do software (FERNANDES *et al.*, 2004):

- Fábrica de Programas: é a unidade menor e menos complexa que tem o escopo na codificação e nos testes das implementações. O processo de desenvolvimento engloba fases de construção e testes unitários;
- Fábrica de Projetos: tem foco mais abrangente no desenvolvimento, englobando as fases referentes à fábrica de programas e outras fases como projeto conceitual, especificação lógica, projeto detalhado e testes de integração e aceitação. Dependendo da interface com o cliente, Fábrica de Projetos pode se caracterizar por Fábrica de Projetos de Software – deve haver conhecimento por parte da Fábrica do negócio do cliente – ou Fábrica de Projetos Físicos; no entanto, estas duas caracterizações têm princípios básicos semelhantes;
- Fábrica de Projetos Ampliada: além de englobar os outros tipos de fábricas, possui soluções mais abrangentes de Tecnologia da Informação, atuando também na concepção da solução e no mapeamento dos requisitos junto ao usuário/cliente, responsabilizando-se pela arquitetura da solução. Fernandes *et al.* (2004) define que este tipo pode ser apresentado no mesmo nível de fábrica de projetos de software;

- Fábrica de Componentes: onde o processo de desenvolvimento reutiliza artefatos vantajosos de projetos anteriores.

Aaen *et al.* (1997) indicam quatro estratégias de fábrica de software e definem os lugares do mundo onde prevalecem. Estas abordagens foram apresentadas a partir de um estudo de quatro visões de fábrica de software bem conhecidas, para uma padronização de operações de desenvolvimento de software:

- Fábrica Industrializada: Abordagem japonesa baseada no conceito da fábrica de software da Toshiba. Esta abordagem visa a melhorar a qualidade do produto de software e aumentar a produtividade, além de criar um ambiente em que o projeto, a programação, o teste, a instalação e a manutenção podem ser executados de uma maneira unificada. A estratégia desta abordagem tem elementos como: projetar estruturas que suportem o processo de desenvolvimento de produtos de software, construir ferramentas de suporte para as atividades deste processo de desenvolvimento e estabelecer uma organização que controle e monitore este processo. Além disso, a organização desta fábrica de software é determinada por: ferramentas de suporte, gerência de projeto, reusabilidade, medição de qualidade e produtividade;
- Fábrica Genérica: Abordagem européia financiada pelo programa Eureka, a qual tem como finalidade a criação de uma arquitetura e um *framework* para sistemas integrados de desenvolvimento de produtos de software, componentes gerais e aplicações para áreas de negócio. A estratégia desta abordagem é desenvolver componentes e ambientes de produção que são partes da fábrica de software juntamente com diretrizes e padrões para componentes de produtos de software. Na organização dessa fábrica, as pessoas que são colocadas nesse contexto de trabalho fazem parte do modelo de produção de produtos de software, onde é necessário dar suporte ao trabalho individual, aumentando a previsibilidade do processo, e criar melhor interação entre as tarefas humanas e as computadorizadas;
- Fábrica de Componentes baseada em Experiência: Abordagem norte-americana que foi desenvolvida no *Software Engineering Institute* (SEI) que tem o objetivo de melhorar a eficácia de processos, a redução de re-trabalho e o reuso do ciclo de vida de produtos. A estratégia dessa abordagem consiste em três elementos chave: i) melhoria no paradigma; ii) organização experiente e dedicada; e iii) planos de contingência. A organização dessa fábrica é em primeiro lugar comprometida com

o aprendizado e a transferência de tecnologia, focando no entendimento de soluções e agrupando experiências para o reuso;

- Amadurecimento Organizacional: Esta abordagem é a definição de fábrica de software de acordo com o modelo *Capability Maturity Model* (CMM). Seu objetivo inicial foi a construção de um *framework* para melhoria de processos de software visando a aumentar previsibilidade, confiabilidade e auto-melhoramento do processo de software com alta qualidade. A estratégia dessa abordagem é o melhoramento passo a passo da organização de software. Para tal, o CMM definiu a ordem de evolução da organização da empresa baseada na adoção de processos. Tal abordagem defende que a organização ideal tem um desenvolvimento disciplinado de planejamento e de acompanhamento dos projetos de software, onde casos de sucesso são reaproveitados.

3.4 Processos de Desenvolvimento em Fábricas de Software

De acordo com Souza (2004), processo é um conjunto de operações realizadas no desenvolvimento de um produto. É uma série de passos realizados para um determinado objetivo. Para a Associação Brasileira de Normas e Técnicas (ABNT, 1994), processo é um conjunto de atividades inter-relacionadas que converte entradas em saídas.

Segundo Lonchamp (1993), processo de desenvolvimento de produtos de software é composto por passos ordenados com o propósito de produzir e manter produtos de software finais requeridos. Pessoa (2003) define que, para um bom funcionamento do processo de desenvolvimento de produtos de software, deve-se ter suas atividades integradas perfeitamente, para que o desenvolvimento ocorra de forma eficaz.

Rocha *et al.* (2004) identifica dois processos de desenvolvimento de produtos de software: tradicional e ágil.

O processo tradicional é considerado um processo rigoroso onde requisitos são previsíveis e comandados e controlados por um planejamento prévio ao desenvolvimento do produto de software. Este processo tem o foco principal na especificação de requisitos, o qual é base para a elaboração de vários outros documentos do processo, dificultando o processo, se alterações forem realizadas (PRESSMAN, 2002).

O processo ágil, um processo leve, tem o foco na codificação, não se importando

tanto com o planejamento, o qual é realizado de acordo com as mudanças ocorridas nos requisitos (BECK, 2000). Mas, para os dois tipos de processos, existem vantagens e desvantagens (ROCHA *et al.* 2004):

- Processo Tradicional:
 - Desvantagens: i) processo burocrata, devido à divisão das atividades e que devem ser realizadas hierarquicamente; e ii) não apropriado, pois o planejado não acontece na realidade;
 - Vantagens: i) fundamentado em fluxos de trabalho de processo; ii) guiado de acordo com o planejamento; iii) garantia alta de que o processo sairá do papel; iv) esboçado para admitir requisições durante o desenvolvimento; v) inclui suporte a ferramentas; e vi) permite ser modificado para se adaptar ao domínio da aplicação ou da organização;

- Processo Ágil:
 - Desvantagens: i) a escalabilidade para equipes grandes e/ou dispersas; e ii) mudança de cultura de paradigma;
 - Vantagens: i) revisões do código durante o desenvolvimento; ii) teste realizado pela equipe ou pelo cliente; iii) iterações curtas, orientadas de acordo com as pessoas do desenvolvimento; e iv) modularidade no nível de desenvolvimento.

O termo fábrica de software está baseado na idéia de fornecer uma linha de produção de soluções que atendam às necessidades específicas de cada cliente. Isto é possível através da formalização das atividades e de seus produtos, trabalhando com etapas e tarefas bem definidas para cada tipo de profissional e partindo das tarefas básicas da linha de produção até rotinas de controle de qualidade.

Logo, com a alta especialização dos profissionais, estes garantem a produtividade da etapa de produção em que estão engajados e a qualidade do artefato produzido para a etapa seguinte.

Além disso, considera-se que o desenvolvimento de produtos de software é um processo especializado para ser assimilado e subsidiado por empresas não pertencentes ao ramo, implicando na crescente terceirização na área de tecnologia de informação, principalmente na área de desenvolvimento de produtos de software.

Atualmente, em paralelo à crescente terceirização, aumenta a necessidade de melhoria em termos de qualidade e custo do produto de software para atender ao mercado. Com isso, empresas de software estão investindo no seu processo de desenvolvimento, o que, juntamente com a evolução da engenharia de software e das tecnologias envolvidas, tornam as fábricas de software uma realidade mais presente no mercado.

3.5 Considerações Finais

O conceito de fábrica de software vem evoluindo desde sua criação até os dias atuais. Uma fábrica de software pode ser apresentada como uma organização que adota técnicas utilizadas na engenharia de produção em série, criando um ambiente produtivo que desenvolva software de qualidade e baixo custo.

Isto é possível através da formalização das atividades e de seus produtos, trabalhando como convém a linha de produção, com etapas e tarefas bem definidas para cada tipo de profissional, partindo das tarefas básicas da linha de produção até rotinas de controle de qualidade.

Atualmente, as empresas têm investido na qualidade de seus processos com o objetivo de garantir produtos de software de qualidade e que atendam às expectativas de seus clientes, tornando as fábricas de software uma realidade cada vez mais presente no mercado.

4. O MODELO MPS.BR

4.1 Considerações Iniciais

Neste Capítulo, são abordados os conceitos referentes ao MPS.BR. Primeiramente, são mostrados a importância, a motivação e os órgãos responsáveis pela sua criação. Em seguida, é apresentado os principais objetivos do MPS.BR, a base técnica para a construção e o aprimoramento do modelo, os componentes, os processos e os níveis de maturidade. Finalmente, apresenta-se o nível G no MPS.BR, juntamente com seus processos e resultados esperados.

4.2 Importância

Várias mudanças estão ocorrendo nos ambiente de negócio, motivando as empresas a modificar suas estruturas organizacionais e adquirir uma visão mais centrada no cliente. Alcançar competitividade pela qualidade, para as empresas de software, implica tanto na melhoria da qualidade dos seus produtos, como na melhoria da qualidade dos processos de produção e distribuição de software.

Deste modo, a qualidade se torna um fator crítico de sucesso para a indústria de software. Para que o Brasil tenha um setor de software competitivo, nacional e internacionalmente, é essencial que os empreendedores do setor coloquem a eficiência e a eficácia dos seus processos em foco nas empresas, visando a oferta de produtos e serviços de software conforme padrões internacionais de qualidade.

Segundo o MPS.BR – Guia Geral (2006), em 2003, no início da concepção do MPS.BR, dados da Secretaria de Política de Informática e Tecnologia do Ministério da Ciência e Tecnologia (MCT/SEITEC), mostravam que apenas 30 empresas no Brasil possuíam avaliação SW-CMM® (*Capability Maturity Model*): 24 no nível 2; 5 no nível 3; 1 no nível 4; e nenhuma no nível 5. Observando-se esta pirâmide pôde-se concluir que a qualidade do processo de software no Brasil podia ser dividida em dois tipos de empresas. No topo da pirâmide, normalmente, estavam as empresas exportadoras de software e outras grandes empresas que desejavam atingir níveis mais altos de maturidade (4 ou 5) do CMMI-SE/SWSM por estágio e serem formalmente avaliadas pelo SEI (*Software Engineering Institute*), em um esforço que pode levar de 4 a 10 anos. Na base da pirâmide, em geral, encontrava-se a grande massa de micro, pequenas e médias empresas de software

brasileiras, com poucos recursos e que necessitam obter melhorias significativas nos seus processos de software em 1 ou 2 anos.

Ainda segundo o MPS.BR – Guia Geral (2006), o foco principal do MPS.BR, está neste segundo grupo de empresas, embora não exclua os outros grupos. Busca-se que ele seja adequado ao perfil de empresas com diferentes tamanhos e características, públicas e privadas, embora com especial atenção às micro, pequenas e médias empresas. Também, espera-se que o MPS.BR seja compatível com os padrões de qualidade aceitos internacionalmente e que tenha como pressuposto o aproveitamento da competência existente nos padrões e modelos de melhoria de processo disponíveis. Dessa forma, ele tem como base os requisitos de processos definidos nos modelos de melhoria de processo e atende a necessidade de implantar os princípios de Engenharia de Software de forma adequada ao contexto das empresas brasileiras, estando em consonância com as principais abordagens internacionais para definição, avaliação e melhoria de processos de software.

O MPS.BR baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de produtos e serviços de software. Está em desenvolvimento desde dezembro de 2003 e é coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), contando com apoio do Ministério da Ciência e Tecnologia (MCT), da Financiadora de Estudos e Projetos (FINEP) e do Banco Interamericano de Desenvolvimento (BID).

A coordenação do Programa MPS.BR conta com duas estruturas de apoio para o desenvolvimento de suas atividades, o Fórum de Credenciamento e Controle (FCC) e a Equipe Técnica do Modelo (ETM). Através destas estruturas, o MPS.BR obtém a participação de representantes de Universidades, Instituições Governamentais, Centros de Pesquisa e organizações privadas, os quais contribuem com suas visões complementares que agregam qualidade ao empreendimento.

4.3 Objetivos

O modelo MPS.BR visa, preferencialmente, as micro, pequenas e médias empresas. Um dos objetivos do MPS.BR é definir e aprimorar um modelo de melhoria e avaliação de processos de software de forma a atender as necessidades de negócio das empresas e ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software. O MPS.BR também estabelece um processo e um método de avaliação, o qual dá

sustentação e garante que o MPS.BR seja empregado de forma coerente com as suas definições.

Segundo o MPS.BR – Guia Geral (2006), a base técnica para a construção e o aprimoramento deste modelo de melhoria e avaliação de processo de software é composta pelas normas NBR ISO/IEC 12207 – Processo de Ciclo de Vida de Software, pelas emendas 1 e 2 da norma internacional ISO/IEC 12207 e pela ISO/IEC 15504 – Avaliação de Processo (também conhecida por SPICE: *Software Process Improvement and Capability dEtermination*). Este modelo também cobre o conteúdo do CMMI-SE/SWSM, através da inclusão de processos e resultados esperados além dos estabelecidos na Norma ISO/IEC 12207.

4.4 MPS.BR

O MPS.BR possui três componentes, mostrados na Figura 4.1: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS). Cada componente é descrito por meio de Guias e/ou de Documentos do MPS.BR (MPS.BR – Guia Geral, 2006).

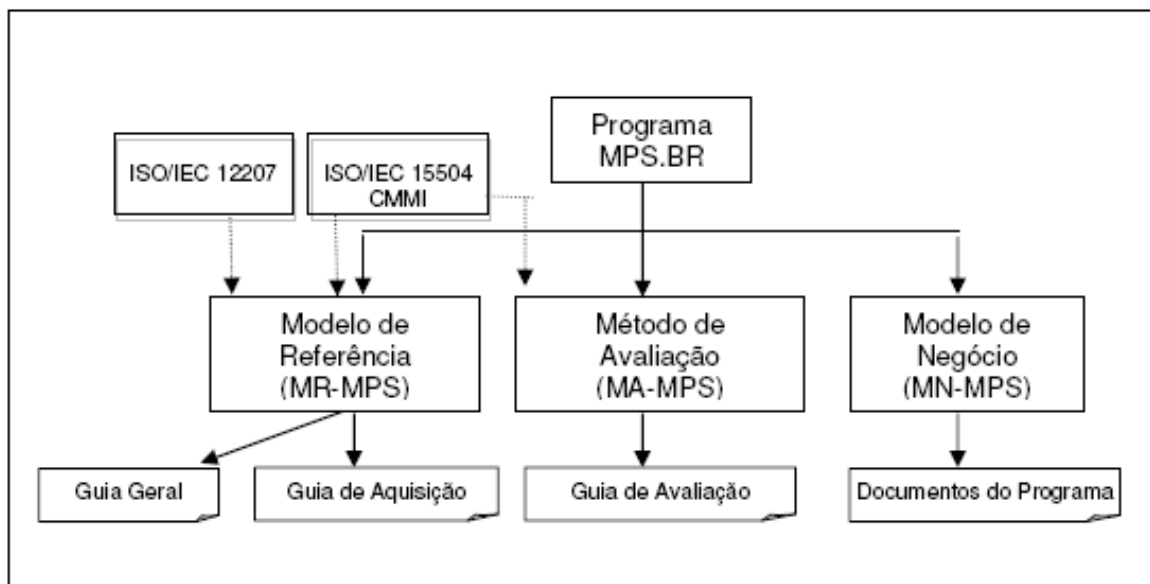


Figura 4.1 – Componentes do MPS.BR (Fonte: MPS.BR – Guia Geral (2006))

O Modelo de Referência MR-MPS define os níveis de maturidade, que são uma combinação entre processos e sua capacidade. A definição dos processos segue a forma apresentada na Emenda 1 da ISO/IEC 12207, declarando o propósito e os resultados

esperados de sua execução. Isso permite avaliar e atribuir graus de efetividade na execução dos processos. As atividades e tarefas necessárias para atender ao propósito e aos resultados esperados não são definidas no guia, devendo ficar a cargo dos usuários do MR-MPS.

O Guia Geral apresenta os processos, os níveis de maturidade e os principais conceitos do MPS.BR. Nível de maturidade é um grau de melhoria de processo para um pré-determinado conjunto de processos no qual os propósitos dentro do conjunto são atendidos (ISO/IEC 15504-1, 2004). O Guia de Aquisição contém boas práticas para a aquisição de software e serviços correlatos. Ele é um documento complementar destinado a organizações que pretendam adquirir produtos e serviços de software.

O Guia de Avaliação contém o processo e o método de avaliação MA-MPS, os requisitos para os avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA). O processo e o método de avaliação MA-MPS está em conformidade com a norma ISO/IEC 15504-2 (ISO/IEC 15504-2, 2003).

O Modelo de Negócio MN-MPS descreve regras de negócio para implementação do MR-MPS pelas Instituições Implementadoras (II), avaliação seguindo o MA-MPS pelas IAs, organização de grupos de empresas para implementação do MR-MPS e avaliação MA-MPS pelas Instituições Organizadoras de Grupos de Empresas (IOGE), certificação de consultores de aquisição e programas anuais de treinamento por meio de cursos, provas e workshops MPS.BR. Um resumo executivo dessas regras de negócio está disponível no Portal SOFTEX (<http://www.softex.br/>).

Conforme abordado anteriormente, no Modelo de Referência, há níveis de maturidade que estabelecem patamares de evolução de processos, descrevendo os estágios de melhoria da implementação de processos na organização. O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos.

De acordo com o MPS.BR – Guia Geral (2006), o MR-MPS define sete níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). A escala de maturidade se inicia no nível G e progride até o nível A.

A mudança do nível de maturidade é alcançada quando são atendidos os resultados esperados e o propósito do processo e dos atributos de processo relacionados aos níveis anteriores.

Para possibilitar a implementação e a avaliação mais adequadas às micro, pequenas e médias empresas, a divisão em estágios, embora baseada nos níveis de maturidade do CMMISE/SWSM, tem graduação diferente. A possibilidade de realizar avaliações considerando mais níveis também permite uma visibilidade dos resultados de melhoria de processos em prazos mais curtos.

Os processos no MR-MPS são descritos em termos de propósito, resultados e informações adicionais. O propósito descreve o objetivo geral a ser atingido durante a execução do processo. Os resultados esperados do processo estabelecem os resultados a serem obtidos com a efetiva implementação do processo. Estes resultados podem ser evidenciados por um artefato produzido ou uma mudança significativa de estado ao executar o processo (MPS.BR – Guia Geral, 2006).

Segundo o MPS.BR – Guia Geral (2006), os processos são agrupados, por uma questão de organização, de acordo com a sua natureza, ou seja, o seu objetivo principal no ciclo de vida de software. Esse agrupamento resultou em três classes de processos (Figura 4.2):

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

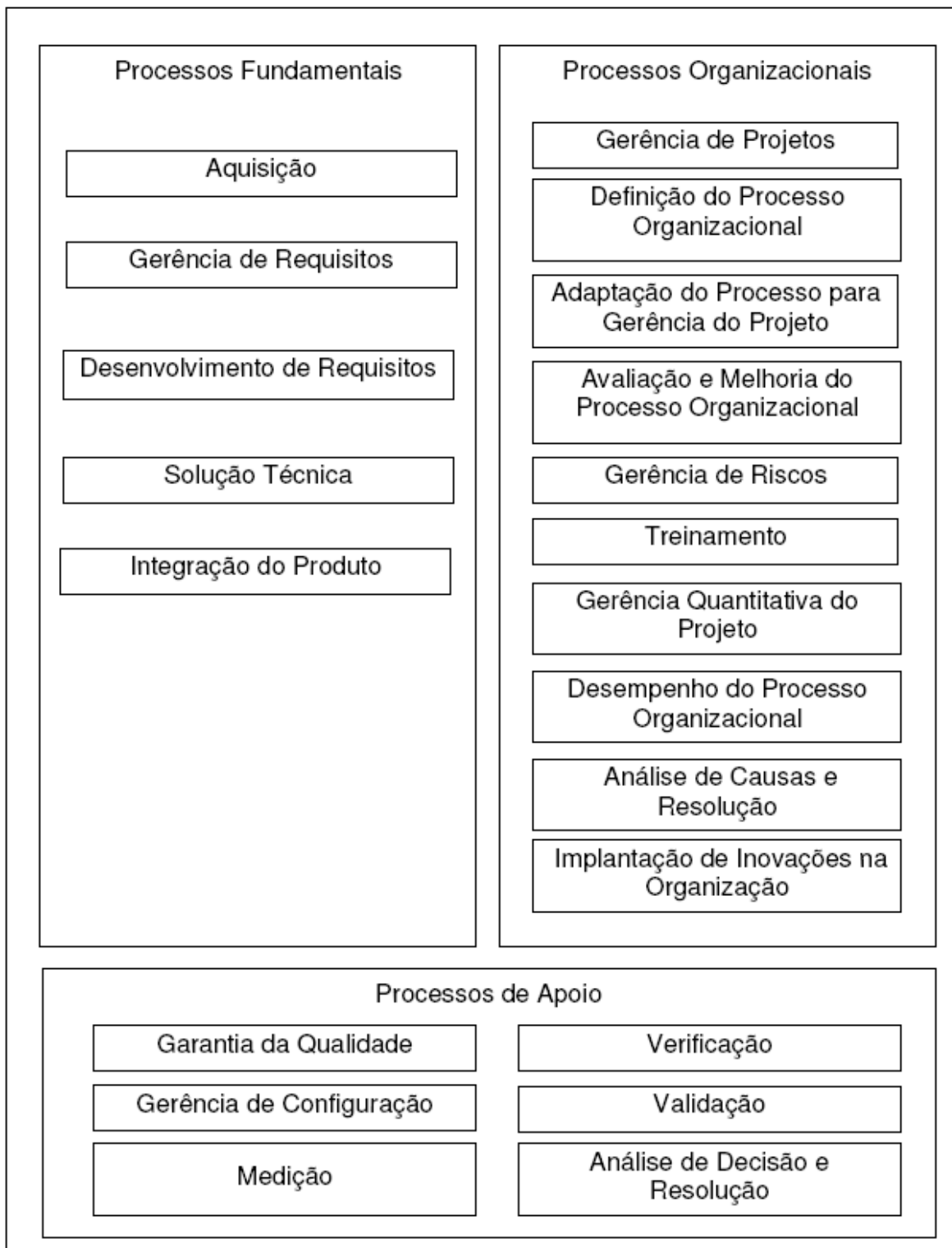


Figura 4.2 – Processos do MPS.BR (Fonte: MPS.BR – Guia Geral (2006))

De acordo com o MPS.BR – Guia Geral (2006), a capacidade do processo é representada por um conjunto de atributos descrito em termos de resultados esperados, expressando o grau de refinamento e institucionalização com que o processo é executado

na organização. No MPS.BR, à medida que a organização evolui nos níveis de maturidade, maior nível de capacidade para desempenhar o processo deve ser atingido pela organização.

O atendimento aos atributos do processo (AP), através do atendimento aos resultados esperados dos atributos do processo (RAP) é exigido para todos os processos no nível correspondente ao nível de maturidade, sendo a sua execução acumulativa, o que significa que ao passar para o nível F, os processos do nível G também devem ser executados.

A capacidade do processo no MPS.BR possui cinco APs: AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2. Cada AP está detalhado em termos do RAP, conforme descrito no MPS.BR – Guia Geral (2006):

- **AP 1.1 O processo é executado**

Este atributo é uma medida da extensão na qual o processo atinge o seu propósito.

Resultado esperado:

RAP1. O processo atinge seus resultados definidos.

- **AP 2.1 O processo é gerenciado**

Este atributo é uma medida da extensão na qual a execução do processo é gerenciada.

Resultados esperados:

RAP 2. Existe uma política organizacional estabelecida e mantida para o processo;

RAP 3. A execução do processo é planejada;

RAP 4 (para o Nível G). A execução do processo é monitorada e ajustes são realizados para atender aos planos;

RAP 4 (a partir do Nível F). Medidas são planejadas e coletadas para monitoração da execução do processo;

RAP 5. Os recursos necessários para a execução do processo são identificados e disponibilizados;

RAP 6. As pessoas que executam o processo são competentes em termos de formação, treinamento e experiência;

RAP 7. A comunicação entre as partes interessadas no processo é gerenciada de forma a garantir o seu envolvimento no projeto;

RAP 8. O estado, as atividades e os resultados do processo são revistos com os

níveis adequados de gerência (incluindo a gerência de alto nível) e problemas pertinentes são tratados.

- **AP 2.2 Os produtos de trabalho do processo são gerenciados**

Este atributo é uma medida da extensão na qual os produtos de trabalho produzidos pelo processo são gerenciados apropriadamente.

Resultado esperado:

RAP 9. Os produtos de trabalho são documentados, revistos e controlados em níveis apropriados de gerência de configuração.

- **AP 3.1. O processo é definido**

Este atributo é uma medida da extensão na qual um processo-padrão é mantido para apoiar a implementação do processo definido.

Resultados esperados:

RAP 10. Um processo padrão é definido, incluindo diretrizes para sua adaptação para o processo definido;

RAP 11. A seqüência e a interação do processo-padrão com outros processos são determinadas;

- **AP 3.2 O processo está implementado**

Este atributo é uma medida da extensão na qual o processo-padrão é efetivamente implementado como um processo definido para atingir seus resultados.

Resultado esperado:

RAP 12. Dados apropriados são coletados e analisados, constituindo uma base para o entendimento do comportamento do processo, para demonstrar a adequação e a eficácia do processo e avaliar onde pode ser feita a melhoria contínua do processo.

A Figura 4.3 apresenta os níveis de maturidade do MR-MPS, os processos e os atributos de processo correspondentes a cada nível, segundo o MPS.BR – Guia Geral (2006).

Nível	Processos	Atributos de Processo
A (mais alto)	Implantação de Inovações na Organização	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP3.2
	Análise de Causas e Resolução	
B	Desempenho do Processo Organizacional	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP3.2
	Gerência Quantitativa do Projeto	
C	Análise de Decisão e Resolução	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP3.2
	Gerência de Riscos	
D	Desenvolvimento de Requisitos	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP3.2
	Solução Técnica	
	Integração do Produto	
	Verificação	
	Validação	
E	Treinamento	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP3.2
	Definição do Processo Organizacional	
	Avaliação e Melhoria do Processo Organizacional	
	Adaptação do Processo para Gerência do Projeto	
F	Medição	AP 1.1, AP 2.1 e AP 2.2
	Gerência de Configuração	
	Aquisição	
	Garantia da Qualidade	
G	Gerência de Requisitos	AP 1.1 e AP 2.1
	Gerência do Projeto	

Figura 4.3 – Níveis de Maturidade (Fonte: MPS.BR – Guia Geral (2006))

4.5 O MPS.BR nível G

O nível G é o primeiro nível de maturidade do MR-MPS. Ele estabelece o início dos trabalhos em implantação de melhoria dos processos de software, portanto deve ser observado com cautela. Ao final da implantação deste nível, a empresa deve ser capaz de gerenciar parcialmente seus projetos de desenvolvimento de software.

Durante a implantação do nível G, pode-se observar uma mudança na cultura

organizacional e uma definição acerca do que é projeto para a organização. Diversas empresas de software precisam adequar sua forma de trabalhar para se tornarem empresas orientadas a projetos. Ser orientada a projetos significa: redefinir algumas operações (atividades de rotina) em andamento, como projeto, estabelecendo objetivos, prazos e escopo para sua execução.

Segundo o MPS.BR – Guia de Implementação – Parte 1 (2007), o nível G descreve dois processos: Gerência de Projetos e Gerência de Requisitos:

- **Gerência de Projetos (GPR)**

O propósito do processo Gerência de Projetos é identificar, estabelecer, coordenar e monitorar as atividades, as tarefas e os recursos que um projeto necessita para produzir produto e/ou serviço.

Várias atividades estão incluídas no processo de Gerência de Projetos (GPR). Através dele, pode-se: i) desenvolver um plano geral de controle do projeto; ii) obter o comprometimento e mantê-lo ao longo da execução do projeto; e iii) conhecer o progresso do projeto, de modo que ele possa ser corrigido ao desviar do planejado.

O desenvolvimento do plano do projeto inclui: i) identificar e estimar o escopo, os produtos de trabalho e as tarefas do projeto; ii) estabelecer recursos necessários; iii) identificar e analisar riscos do projeto; iv) estabelecer compromissos; e v) definir cronograma de execução baseado no ciclo de vida definido para o projeto.

Os resultados esperados do GPR podem ser resumidos de acordo com o MPS.BR - Guia de Implementação – Parte 1 (2007):

GPR1 – O escopo do trabalho para o projeto é definido

O escopo do projeto estabelece o que está incluído no projeto. Ele é o ponto de partida para o planejamento do projeto e define o trabalho necessário para ele possa ser terminado com sucesso.

O escopo pode ser representado através de uma Estrutura Analítica do Projeto (EAP), também conhecida como WBS (*Work Breakdown Structure*), que fornece um esquema para identificar e organizar as unidades lógicas de trabalho a serem gerenciadas. Este resultado também pode ser implementado através de um Documento de Visão ou outro documento que defina, claramente, o escopo do trabalho.

GPR2 – As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados

O escopo do projeto, representado através da EAP, deve ser decomposto em

componentes menores, mais facilmente gerenciáveis e possíveis de serem estimados. O tamanho é a principal entrada de muitos modelos utilizados para estimar o esforço, custo e cronograma.

O tamanho é a dimensão do que se tem a fazer, por exemplo, número de tabelas, relatórios, telas, acessos a banco de dados, cálculos, programas, módulos, linhas de código, casos de uso, classes, número de requisitos e objetos.

No nível G, a estimativa de escopo, de produtos e de tarefas pode ser feita baseada no número de requisitos e no uso da EAP, utilizando dados históricos e a experiência em projetos anteriores.

GPR3 – O modelo e as fases do ciclo de vida do projeto são definidas

O ciclo de vida de projeto define um conjunto de fases, em que cada fase gera produtos de trabalho necessários para o desenvolvimento de fases posteriores. Essa organização em fases permite planejar o projeto incluindo marcos importantes para o controle e revisões.

A determinação das fases do ciclo de vida do projeto permite que a tomada de decisões e a avaliação do andamento do projeto sejam planejadas em períodos pré-estabelecidos. Tais pontos fornecem os eventos planejados nos quais as correções de curso, as definições de futuro, a reavaliação de escopo e o custo do projeto podem ser realizados.

GPR4 – O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas

Os dados históricos incluem os dados de custo, de esforço e de tempo de projetos anteriores, além de dados apropriados de escala para equilibrar as diferenças de tamanho e complexidade. Através da análise dos dados históricos são feitas as estimativas, considerando escopo, tarefas, produtos de trabalho, riscos, mudanças previstas, viagens, ciclo de vida e grau de competência da equipe do projeto.

GPR5 – O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos

Com base na EAP e nas estimativas de esforço e custo, é definido o cronograma do projeto. É importante manter a coerência entre o ciclo de vida, a EAP, as estimativas e o cronograma.

Logo em seguida, é estabelecido o orçamento do projeto, com base nas informações presentes no cronograma e na estimativa de custos. Este resultado é importante porque o cronograma e o orçamento são ferramentas importante no acompanhamento do projeto.

GPR6 – Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados

Projetos normalmente têm riscos e estes precisam ser identificados, analisados e priorizados. Uma planilha de riscos, contendo dados como identificador, descrição, probabilidade, impacto e prioridades no seu tratamento, pode ser utilizada. É importante demonstrar que esta planilha está sendo monitorada e atualizada.

No nível G, os riscos são acompanhados para verificar como afetam o projeto e para tomar medidas, mesmo que ainda sem um gerenciamento completo.

GPR7 – Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessário para executá-lo

As funções, as responsabilidades e as relações hierárquicas do projeto são determinadas durante o planejamento de recursos humanos. O planejamento de recursos humanos inclui informações de como e quando o recurso será envolvido no projeto, critérios para sua liberação, competência necessária para a execução das atividades, mapa de competências da equipe e identificação de necessidades de treinamento.

GPR8 – As tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados

Alguns recursos precisam ser adquiridos ou alocados para o projeto. Eles precisam ser planejados, pois necessitam de orçamento, o que pode tornar crítico em alguns projetos.

As tarefas, os recursos e o ambiente necessário, como equipamentos, ferramentas, serviços, componentes e viagens, podem ser especificados com base na EAP ou estrutura equivalente. Os recursos humanos, incluindo treinamento, são abordados no GPR7.

GPR9 – Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, de armazenamento e de distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança

Os dados do projeto (relatórios, dados informais, estudos e análises, atas de reuniões, documentação, lições aprendidas e artefatos gerados) devem ser identificados, coletados, armazenados e distribuídos de acordo com um planejamento prévio. Se pertinente, regras de segurança e confidencialidade devem ser incluídas.

Diversas informações como o motivo de coletar cada dado, os dados relevantes do projeto e a inexistência de dados confidenciais devem ser explicitadas no planejamento.

GPR10 – Planos para a execução do projeto são estabelecidos e reunidos no Plano do Projeto

Para monitoração efetiva do projeto, é importante que as informações definidas e coletadas para o projeto, como escopo, ciclo de vida, tarefas, orçamento, cronograma, riscos, recursos e planejamento dos dados, sejam organizadas em um Plano de Projeto.

A utilização do Plano de Projeto melhora a visibilidade do projeto, facilitando o seu gerenciamento e a formação de uma base histórica.

GPR11 – A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados

Muitas vezes, é melhor não iniciar ou parar um projeto em andamento do que continuar com um projeto inviável. Através do estudo de viabilidade, pode-se analisar se o prosseguimento de um projeto não acarretará perdas para os fornecedores e para o cliente.

O estudo de viabilidade considera o escopo do projeto e examina aspectos técnicos (requisitos e recursos), financeiros (capacidade da organização) e humanos (disponibilidade de pessoas com a capacitação necessária). Deve-se considerar também os objetivos de negócio da organização.

No início do projeto, uma avaliação preliminar deve ser conduzida. À medida que o projeto evolui, a viabilidade de sucesso pode ser reavaliada com mais precisão. As mudanças de requisitos são eventos que impõem a realização de uma reavaliação da viabilidade do projeto.

GPR12 – O Plano do Projeto é revisado com os interessados e o compromisso com ele é obtido

Para que o projeto atinja as metas estabelecidas, é importante que haja compromisso das partes interessadas. Para isso, uma revisão do planejamento deve ser realizada com o objetivo de solucionar os conflitos entre as variáveis do projeto, como requisitos, custos e prazos.

Uma reunião de início de projeto (*kick off*) pode ser utilizada para obter o comprometimento e resolver as diferenças existentes entre os recursos estimados e disponíveis.

GPR13 – O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados

Acompanhar o que foi planejado é uma atividade importante do gerenciamento. O

Plano de Projeto deve ser avaliado continuamente durante o ciclo de vida do projeto.

O uso dos recursos, a aderência ao cronograma e outras variáveis do projeto devem ser monitorados.

GPR14 – O envolvimento das partes interessadas no projeto é gerenciado

O cliente, o usuário, a direção da organização e os membros da equipe do projeto devem ser identificados, juntamente com as fases em que eles são importantes e como eles serão envolvidos.

Planejar a comunicação é importante porque o distanciamento da gerência do projeto em relação aos interessados pode provocar desvios do planejamento ou das necessidades do projeto.

GPR15 – Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento

Os marcos do projeto precisam ser definidos no planejamento. Os marcos podem ser, por exemplo, o início ou o final de cada fase do projeto. Essas revisões tem o objetivo de verificar o andamento do projeto de uma forma ampla e se as condições para o início de uma fase estão atendidas.

GPR16 – Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas

Vários problemas podem ser identificados no projeto através do monitoramento e das revisões em marcos. Estes problemas devem ser analisados, registrados e tratados com as partes interessadas do projeto.

GPR17 – Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão

Após os problemas serem identificados, analisados e registrados, ações corretivas devem ser estabelecidas. Estas ações devem ser gerenciadas até a sua conclusão, verificando, com certa frequência, se o problema foi resolvido.

- **Gerência de Requisitos (GRE)**

Segundo o MPS.BR – Guia de Implementação – Parte 1 (2007), o propósito do GRE é gerenciar os requisitos dos produtos e dos componentes do produto do projeto e identificar inconsistências entre esses requisitos e os planos e produtos de trabalho do projeto.

Ainda de acordo com o MPS.BR – Guia de Implementação – Parte 1 (2007), o principal objetivo da Gerência de Requisitos é controlar a evolução dos requisitos. O GRE gerencia os requisitos recebidos ou gerados pelo projeto, incluindo requisitos funcionais e não-funcionais, documenta as mudanças nos requisitos e mantém a rastreabilidade bidirecional entre os requisitos, os planos de projeto e os produtos de trabalho.

O comprometimento dos envolvidos no projeto sobre os requisitos é adquirido através de uma revisão, com o objetivo de resolver questões e prevenir o mau entendimento.

Os resultados esperados do GRE podem ser resumidos de acordo com o MPS.BR - Guia de Implementação – Parte 1 (2007):

GRE1 – O entendimento dos requisitos é obtido junto aos fornecedores de requisitos

O objetivo deste resultado é garantir que os requisitos estejam claramente definidos. É necessário rever com o cliente se as necessidades e as expectativas são atendidas com os requisitos propostos. O entendimento deve ser obtido junto aos fornecedores de requisitos e o resultado das comunicações deve ser registrado formalmente. Este resultado pode ser implementado através de um documento de requisitos.

GRE2 – Os requisitos de software são aprovados utilizando critérios objetivos

Os requisitos de software devem ser aprovados pelas partes interessadas após uma avaliação com base em um conjunto de critérios objetivos. Estes critérios podem ser: i) possuir identificação única; ii) estar claro e apropriadamente declarado; iii) ser relevante; iv) ser completo; v) estar consistente com os demais requisitos; e vi) ser implementável, testável e rastreável. Esta avaliação pode ser feita através de um *checklist* realizado em uma reunião de *kick off*.

GRE3 – A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida

A rastreabilidade bidirecional pode ser dividida em rastreabilidade vertical e rastreabilidade horizontal. A rastreabilidade vertical determina se os requisitos fonte foram completamente tratados e se os requisitos de mais baixo nível ou códigos de unidade podem ser rastreados para um requisito fonte válido. A rastreabilidade horizontal estabelece a dependência entre os requisitos ou produtos de trabalho em um mesmo nível, por exemplo, rastreabilidade dos requisitos entre si ou rastreabilidade entre códigos de unidades dependentes.

Esse resultado pode ser implementado através de um mecanismo que permita

rastrear a dependência entre os requisitos e os produtos de trabalho, com o objetivo de avaliar o impacto das mudanças de requisitos que possam ocorrer no projeto.

GRE4 – Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos

Uma avaliação para identificar se os requisitos estão consistentes com os produtos de trabalho do projeto deve ser executada a fim de encontrar problemas. Após a identificação das inconsistências, elas devem ser registradas e corrigidas. As ações corretivas devem ser acompanhadas até que os problemas sejam resolvidos.

GRE5 – Mudanças nos requisitos são gerenciadas ao longo do tempo

As mudanças devem ser registradas e um histórico das decisões relacionadas aos requisitos deve estar disponível aos envolvidos. Uma análise do impacto das mudanças no projeto deve ser realizada.

4.6 Considerações Finais

O MPS.BR é um modelo de melhoria do processo de software baseado em níveis de maturidade. Ele foi desenvolvido de acordo com diversas normas de qualidade internacionais e visando principalmente às micro, pequenas e médias empresas de software brasileiras.

Ele é apresentado através de diversos componentes que são descritos através de guias e documentos. Um processo e um método de avaliação é definido garantindo sustentação ao modelo.

O nível G do MPS.BR é o primeiro nível de maturidade do modelo e é onde são iniciados os trabalhos de melhoria do processo. O objetivo dos processos descritos para o nível G do MPS.BR (GPR e GRE) é alcançado através de resultados esperados que são descritos pelo modelo.

5. METODOLOGIA

5.1 Tipo de Pesquisa

Observando o método científico, tem-se que a presente pesquisa classifica-se, quanto a sua natureza, como uma pesquisa aplicada ou tecnológica, com objetivos de caráter exploratório e com base em referências bibliográficas e documentais.

Define-se como uma pesquisa aplicada ou tecnológica pela aplicação de conhecimentos básicos com o intuito de elaborar um processo.

O trabalho possui objetivos de caráter exploratório por chamar a atenção das fábricas de software de pequeno porte para a importância de processos de desenvolvimento de software atendendo a normas de qualidade.

5.2 Procedimentos Metodológicos

Para atingir os objetivos descritos, foi realizada uma busca de informações com o intuito de adquirir maior familiaridade com o tema e o aperfeiçoamento de idéias através da pesquisa bibliográfica. Pesquisou-se temas referentes a processos de desenvolvimento de software, fábricas de software e modelos de qualidade e melhoria do processo de software, enfatizando o MPS.BR.

Segundo Jung (2004), a pesquisa bibliográfica tem como principal objetivo constituir uma base conceitual consistente a partir de informações existentes, oferecendo uma vasta aprendizagem para o entendimento fundamental do assunto e possibilitando ao pesquisador ousar ao sugerir novos argumentos que expliquem as descobertas.

A pesquisa sobre o MPS.BR baseou-se nos diversos guias e documentos do modelo que podem ser encontrados no *site* da SOFTEX (<http://www.softex.br>).

O processo de desenvolvimento de software elaborado foi desenvolvido baseado no *Rational Unified Process* (RUP). Como o processo foi inicialmente elaborado para ser implantado na Fábrica de Software do DCC – UFLA, ele foi modelado através de uma representação gráfica própria usada pela empresa. A utilização de uma outra representação gráfica poderia causar gastos com treinamento e maior impacto ambiental na organização; por isso, não foi avaliada.

Os fluxos de trabalho (*workflows*) do processo foram feitos utilizando a ferramenta

Microsoft Visio 2003.

Após a elaboração e a modelagem, foi feita uma comparação do processo desenvolvido com os resultados esperados nos dois processos do MPS.BR nível G. Essa comparação foi executada através da verificação da presença, no processo elaborado, de artefatos ou atividades que implementassem os resultados esperados do GPR e do GRE.

6. RESULTADOS E DISCUSSÃO

6.1 Considerações Iniciais

Neste capítulo, são apresentados os resultados da elaboração e da modelagem do processo de desenvolvimento de software e sua proposta de adequação ao nível G do modelo MPS.BR com o objetivo de fornecer uma descrição das fases do processo, as suas principais atividades, os artefatos e os papéis envolvidos.

6.2 O Processo de Desenvolvimento de Software

Com base nos estudos realizados, percebe-se a importância de um processo de desenvolvimento de software de qualidade. Dessa forma, esta seção tem como finalidade descrever o ciclo de vida do processo proposto.

O processo elaborado é focado nas atividades consideradas essenciais para a construção de software, denominadas análise e definição de requisitos, projeto arquitetural e detalhado, codificação, testes (de unidade, integração e sistema) e entrega. De acordo com essas atividades foi definido o ciclo de vida do processo de acordo com quatro fases, conforme a Figura 6.1.

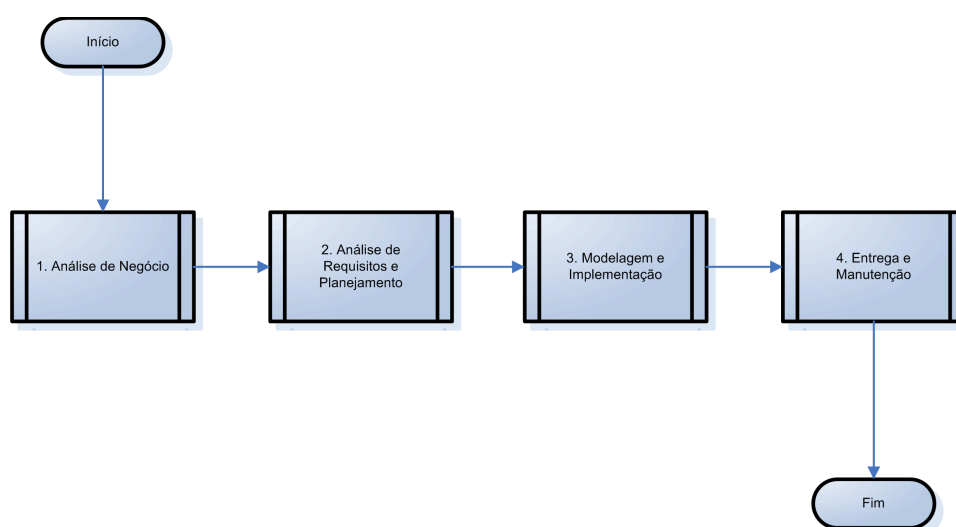


Figura 6.1 – Ciclo de vida do processo

A Tabela 6.1 ilustra a relação entre as quatro fases do ciclo de vida do processo com as atividades consideradas mínimas necessárias para a construção efetiva de software, segundo Pfleeger (2004).

Tabela 6.1 – Relação das atividades essenciais com as fases do processo

Atividade essencial	Fases do ciclo de vida do processo
Análise e Definição de Requisitos	Análise de Requisitos e Planejamento
Projeto Arquitetural e Detalhado, Codificação e Testes	Modelagem e Implementação
Entrega	Entrega e Manutenção

Para cada fase, foram definidas as atividades para que fossem gerados os produtos de trabalho que, nesse contexto, são os artefatos que devem ser produzidos ao longo do processo de software. Para cada atividade, foi atribuído um ou mais responsáveis pela sua execução, denominado papel responsável.

Uma atividade possui um propósito, um papel responsável e artefatos de entrada e de saída. As atividades são executadas com a finalidade de atingir o objetivo do processo. Os artefatos de entrada podem ser *templates* de documentos, documentos preenchidos, padrões do processo ou elementos informativos. Os artefatos de saída podem ser um ou mais artefatos de entrada atualizados ou preenchidos ou um novo produto de trabalho.

Para auxiliar as informações do fluxo do processo, foi proposto um modelo de documento de referência para as atividades das fases, que deve ser preenchido de acordo com a experiência e os fatores ambientais da instituição no qual o processo será implantado (Tabela 6.2). Neste documento, para cada atividade, são apresentados:

- Nome e identificador da atividade;
- Descrição;
- Artefatos de Entrada e Saída;
- Responsável pela execução da atividade;
- Ferramentas utilizadas;
- Passos.

Tabela 6.2 – Documentação de Atividade de uma Fase

Atividade 2.2 – Desenvolver Visão
<p>Descrição: Atividade responsável por:</p> <ul style="list-style-type: none"> • Estabelecer um acordo sobre quais problemas precisam ser resolvidos. • Identificar os envolvidos para o sistema. • Definir as fronteiras do sistema. • Descrever os principais recursos do sistema.
<p>Artefatos de Entrada:</p> <ul style="list-style-type: none"> • <i>Template</i> do Documento de Visão (xVIS) <p>Artefatos de Saída:</p> <ul style="list-style-type: none"> • Documento de Visão
<p>Papel:</p> <ul style="list-style-type: none"> • Engenheiro de Requisitos
<p>Ferramentas:</p> <ul style="list-style-type: none"> • Editor de texto
<p>Passos:</p> <ul style="list-style-type: none"> • Estabelecer um acordo sobre quais problemas precisam ser resolvidos. • Identificar os envolvidos para o sistema. • Definir as fronteiras do sistema. • Identificar as restrições a serem impostas ao sistema. • Formular a descrição do problema. • Definir recursos do sistema. • Avaliar os resultados.

As próximas subseções descrevem as quatro fases do ciclo de vida do processo de desenvolvimento de software elaborado.

6.2.1 Análise de Negócio

A fase Análise de Negócio é a primeira fase do projeto, onde são estabelecidos os objetivos do produto de software, bem como delimitado o seu escopo. Neste momento, o foco do trabalho está voltado para o levantamento dos requisitos preliminares e as funções gerais do produto de software. Nessa fase, são estabelecidos e acordados custos e cronograma de desenvolvimento.

A fase Análise de Negócio engloba as atividades de atendimento ao cliente, análise, proposta e negociação como apresentado na Figura 6.2(a) e Figura 6.2(b).

Glossário

PAPÉIS ENVOLVIDOS:
 GN: Gerente de Negócios
 GP: Gerente de Projetos

ARTEFATOS:
 - DRP: Documento de Requisitos Preliminar
 - EST: Documento de Estimativa
 - FSS: Formulário de Solicitação de Serviços
 - HIS: Histórico de Estimativas
 - PCO: Proposta Comercial
 - PTE: Proposta Técnica
 - VIS: Documento de Visão
 OBS: Os artefatos precedidos de "x" são templates
 " * " significa elaboração parcial do documento

LEGENDA:

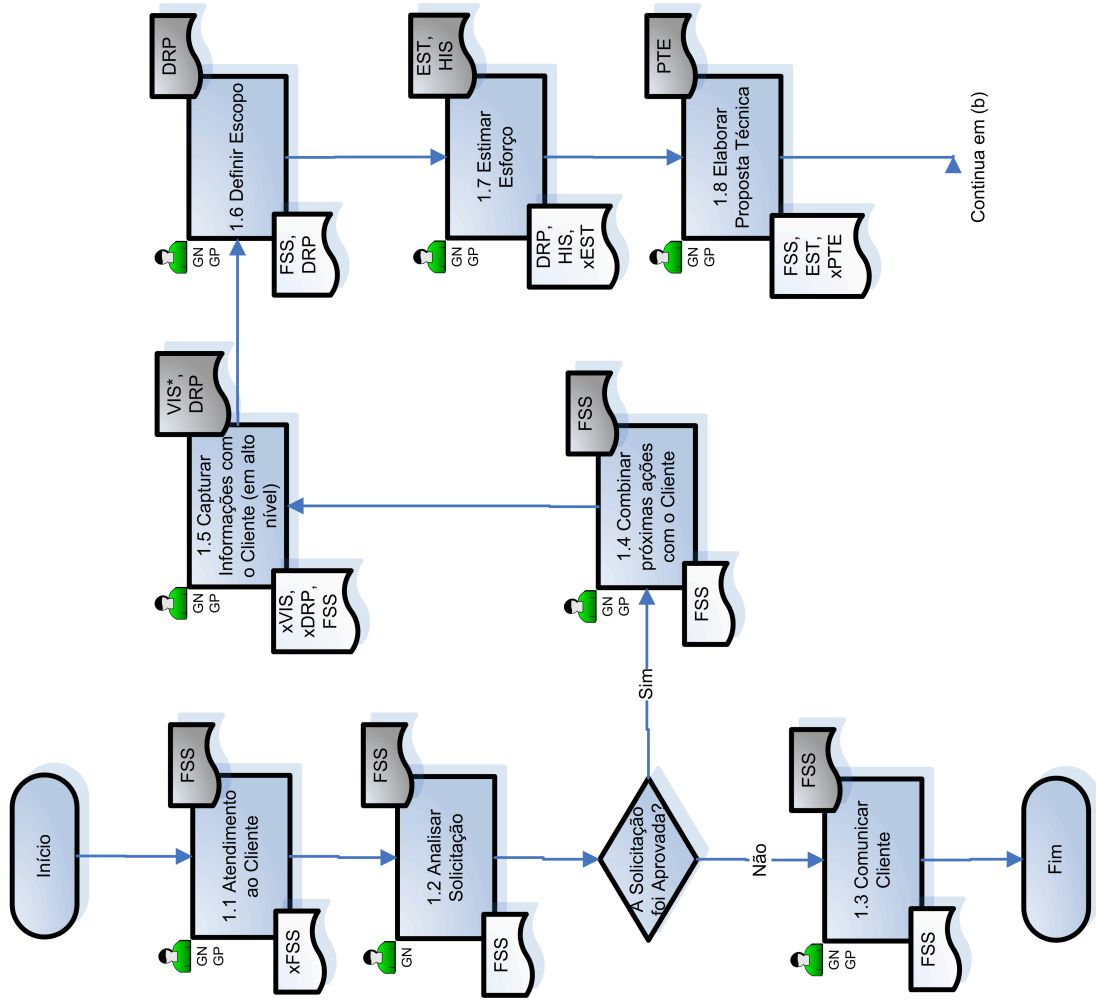
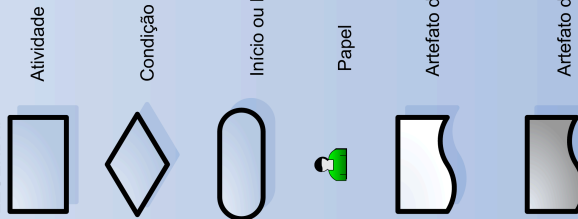


Figura 6.2(a) – Fluxo de Trabalho do processo de Análise de Negócio

Glossário

PAPEIS ENVOLVIDOS:
 GN: Gerente de Negócios
 GP: Gerente de Projetos

ARTEFATOS:

- DRP: Documento de Requisitos Preliminar
 - EST: Documento de Estimativa
 - FSS: Formulário de Solicitação de Serviços
 - PCO: Proposta Comercial
 - PTE: Proposta Técnica
- OBS: Os artefatos precedidos de "x" são templates

LEGENDA:

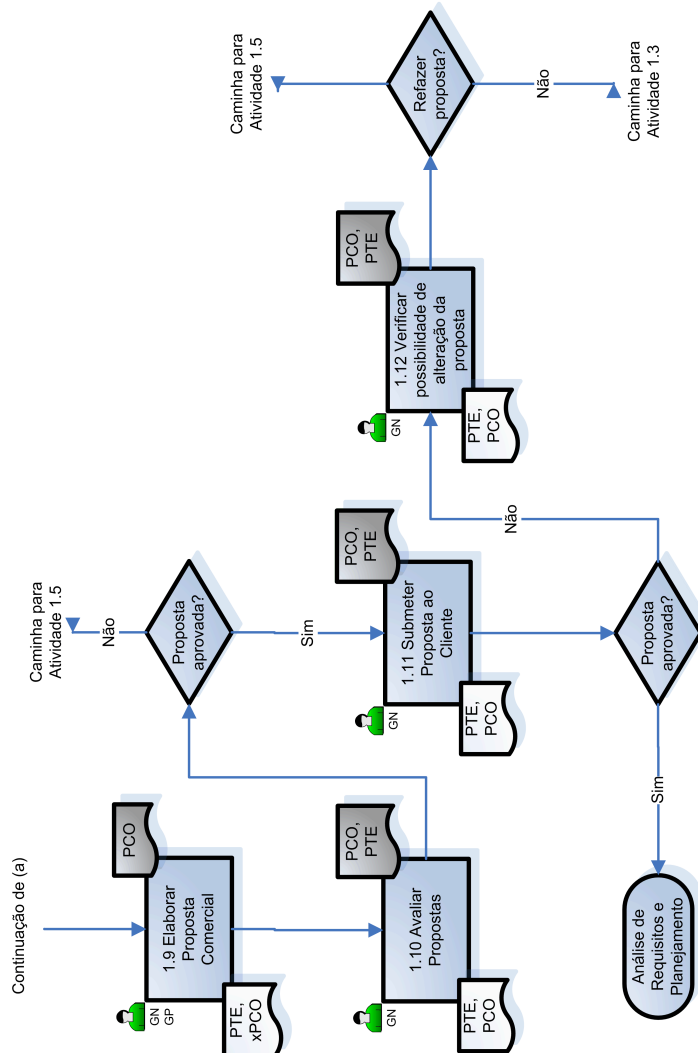
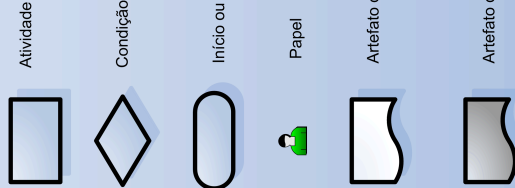


Figura 6.2(b) – Fluxo de Trabalho do processo de Análise de Negócio

Na Atividade 1.1 – Atendimento ao Cliente, é realizado o primeiro contato com o cliente e registrada a solicitação de um serviço ou produto. Nesta etapa, pode ocorrer a comunicação ao cliente que a empresa não prestará seus serviços se não achar viável o projeto (Atividade 1.3 – Comunicar Cliente).

Após o atendimento inicial, verifica-se a viabilidade do projeto, sob o ponto de vista organizacional, durante a Atividade 1.2 – Analisar Solicitação. São analisados aspectos, como: tipo do cliente, viabilidade técnica, compromissos assumidos, viabilidade financeira e mercado. Se o projeto for viável, a empresa e o cliente combinam reuniões, forma de interação e interfaces para definição do escopo do projeto (Atividade 1.4 – Combinar próximas ações com o cliente).

Após definido o escopo e estimados os esforços, é elaborada uma proposta técnica, na Atividade 1.8 – Elaborar Proposta Técnica, onde são especificados: o escopo do projeto, a estratégia de execução e os produtos de trabalho que serão entregues. Na Atividade 1.9 – Elaborar Proposta Comercial, é realizada uma análise do ambiente de execução interno da empresa para a elaboração da proposta comercial definindo o valor do projeto, a forma de pagamento, as propriedades de licença, os critérios de aceitação e ressalvas. As propostas são analisadas antes de serem enviadas para o cliente.

Durante a negociação, executada na Atividade 1.11 – Submeter Proposta ao Cliente, as propostas são levadas ao cliente onde este pode aceitá-la ou recusá-la. Caso a proposta seja aceita, o projeto passa para a fase de Análise de Requisitos e Planejamento, é registrado como aceito e cria-se um ambiente para o projeto na empresa; caso a proposta seja recusada e dependendo do motivo alegado pelo cliente, é analisada a possibilidade de alteração das propostas iniciais.

Os principais artefatos produzidos durante a fase de Análise de Negócio são a Proposta Técnica e a Proposta Comercial. Eles contêm as informações necessárias para o início da fase seguinte.

6.2.2 Análise de Requisitos e Planejamento

A fase Análise de Requisitos e Planejamento estabelece e mantém uma concordância com os clientes e outros envolvidos sobre o que o sistema deve fazer, oferece aos desenvolvedores do sistema melhor compreensão dos requisitos, define as fronteiras do sistema (ou delimita o sistema) e fornece uma base para estimar o custo e o tempo de desenvolvimento do sistema.

O planejamento consiste em uma análise mais refinada do produto de software a ser construído juntamente com o plano detalhado do trabalho a ser realizado. O foco do trabalho está voltado para o levantamento de requisitos e análise, buscando uma compreensão clara do problema.

Durante essa fase, são elaborados o Documento de Visão e o Glossário para definir a visão que os envolvidos têm do produto a ser desenvolvido, em termos das necessidades e características mais importantes, e melhorar o entendimento entre as partes.

Um dos principais artefatos resultantes dessa fase é o Documento de Requisitos, com a descrição da funcionalidade de uma ou mais partes do sistema, baseado no escopo identificado na Proposta Técnica. Após sua elaboração, o documento de requisitos é analisado pela empresa, para identificar erros e inconsistências que causem alguma dificuldade de entendimento para o desenvolvimento e/ou homologação do produto de software, e avaliado em conjunto com o cliente, com a finalidade de obter um bom entendimento dos requisitos. Além disso, os requisitos são apresentados à equipe do projeto para obter o comprometimento e o seu entendimento.

Durante o planejamento, são estimados os esforços das atividades, definidas as fases e o ciclo de vida do projeto, identificados e classificados os riscos, identificados os recursos necessários e os materiais de consumo internos e externos, especificadas e documentadas as revisões, as verificações e as validações realizadas nos produtos de trabalho e estimados os custos das atividades, dos materiais de consumo, da locomoção e da hospedagem necessários para execução do projeto. Caso haja discordância entre as estimativas feitas anteriormente deve ser feita uma negociação com o cliente.

Estas atividades são executadas durante a Atividade 2.13 – Desenvolver Plano de Projeto, que se trata de uma tarefa muito mais complexa. A Atividade 2.3 engloba a maioria das atividades do processo de Gerência de Projetos.

A partir destes dados, é verificado se o planejamento está coerente com o projeto e se será possível alcançar os objetivos do projeto. Caso o planejamento esteja coerente, o projeto passa para a fase de Modelagem e Implementação; caso contrário, avalia-se a possibilidade de realizar um novo planejamento. Se for possível, o projeto volta para o início da etapa, senão é formalizado o cancelamento do projeto.

O fluxo de atividades da fase Análise de Requisitos e Planejamento pode ser visto na Figura 6.3(a) e Figura 6.3(b).

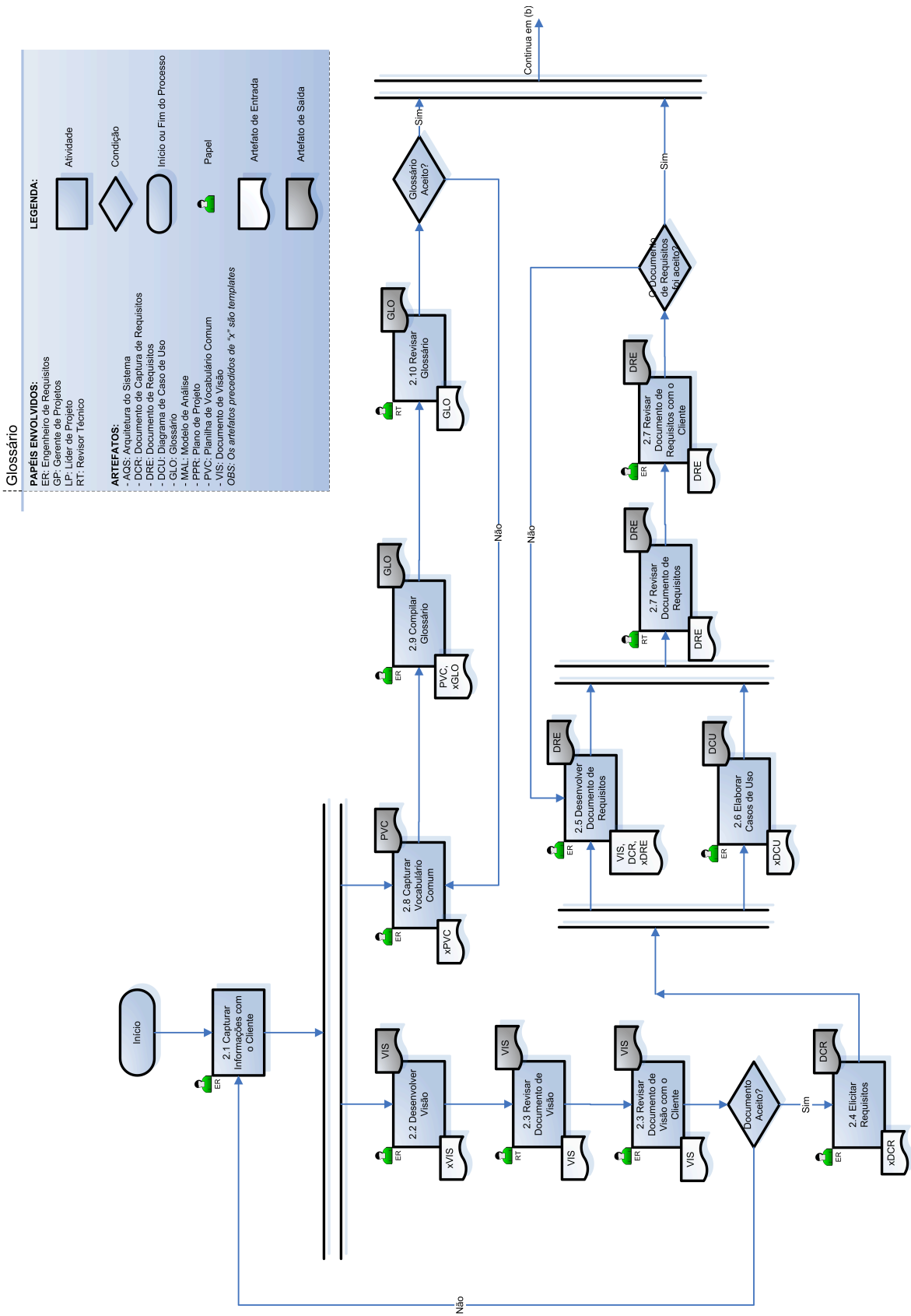


Figura 6.3(a) – Fluxo de Trabalho do processo de Análise de Requisitos e Planejamento

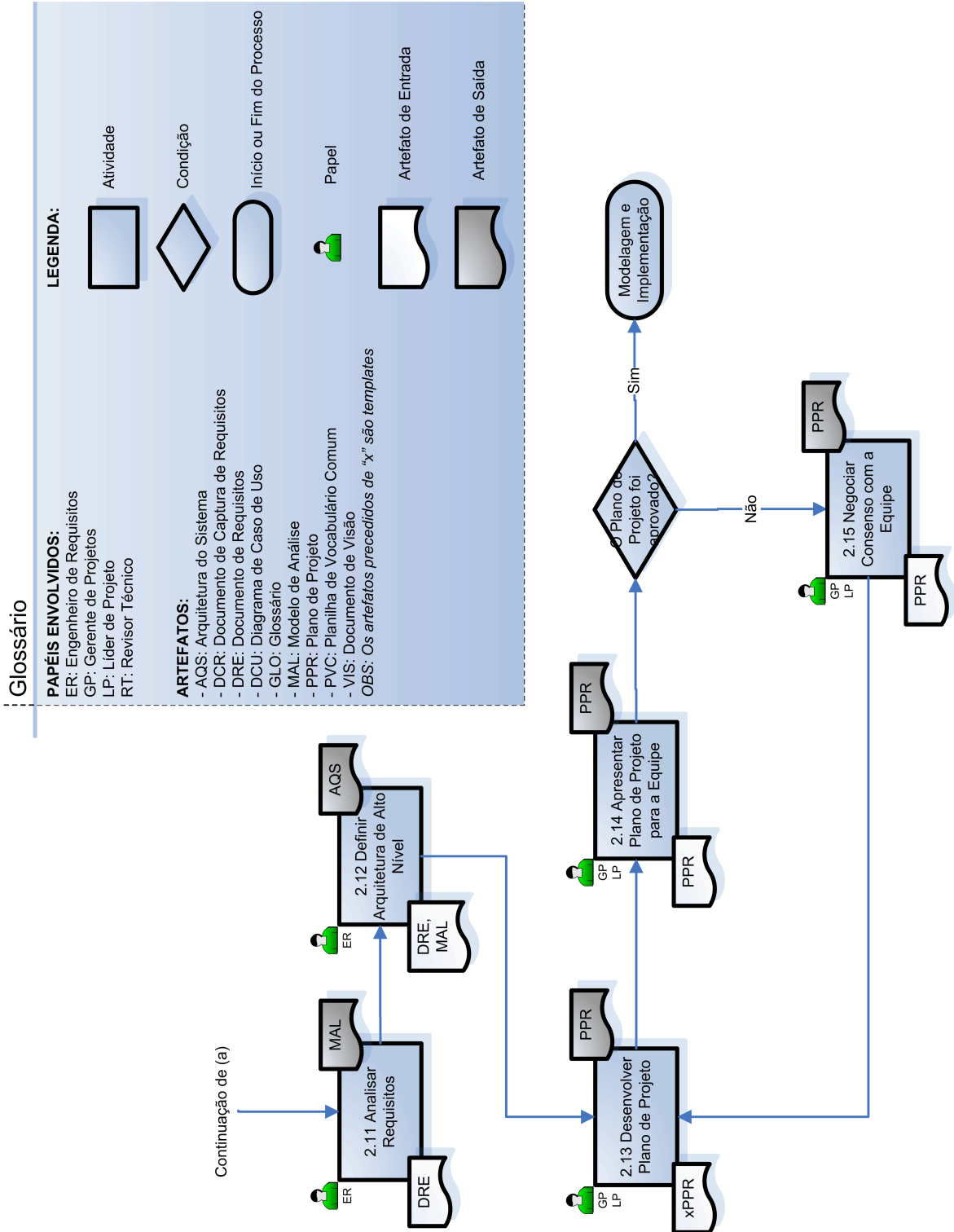


Figura 6.3(b) – Fluxo de Trabalho do processo de Análise de Requisitos e Planejamento

6.2.3 Modelagem e Implementação

Durante a Modelagem, transforma-se os requisitos em um modelo do sistema a ser criado, desenvolve uma arquitetura para o sistema e adapta o modelo para que corresponda ao ambiente de implementação.

A Implementação define a organização do código em termos de subsistemas de implementação organizados em camadas, implementa classes e objetos em termos de componentes (arquivos-fonte, binários, executáveis e outros), testa os componentes desenvolvidos como unidades e integra os resultados produzidos por implementadores individuais (ou equipes) ao sistema executável.

Na fase Modelagem e Implementação, um produto completo é desenvolvido com base no conhecimento adquirido das fases anteriores. Os componentes que demandam maior atenção nesse momento são: a modelagem, a implementação e o teste como mostrado na Figura 6.4(a), na Figura 6.4(b) e na Figura 6.5. O produto final é obtido ao final desta fase.

É elaborada uma modelagem do produto de software, indicando como a funcionalidade será implementada. Esta modelagem é revisada para refletir corretamente a especificação dos requisitos.

Durante a Atividade 3.7 – Elaborar Plano de Testes, são definidos, para cada caso de uso, os testes específicos que devem ser realizados durante a atividade de teste. Após esta atividade, são implementados os requisitos gerando o código-fonte, que deve ser verificado pelo programador se está de acordo com os testes parciais definidos anteriormente.

Com os artefatos gerados até o momento, o Engenheiro de Testes busca inconsistências e erros na implementação dos requisitos presentes na iteração atual, o que ajudará a identificar as atividades que restam para finalizar a iteração e a situação do projeto. É realizado um teste global buscando inconsistências e erros na implementação dos requisitos presentes nas iterações. A partir deste teste, é avaliada se a versão do produto de software deve ou não ser liberada para implantação final. Se a versão for liberada, o projeto passa para a atividade de construção dos Manuais Técnicos e de Usuário (Atividade 3.21 – Criar Manuais). Com a elaboração do pacote de instalação, o projeto encaminha para a fase de Entrega e Manutenção. Caso contrário, são corrigidos os erros encontrados.

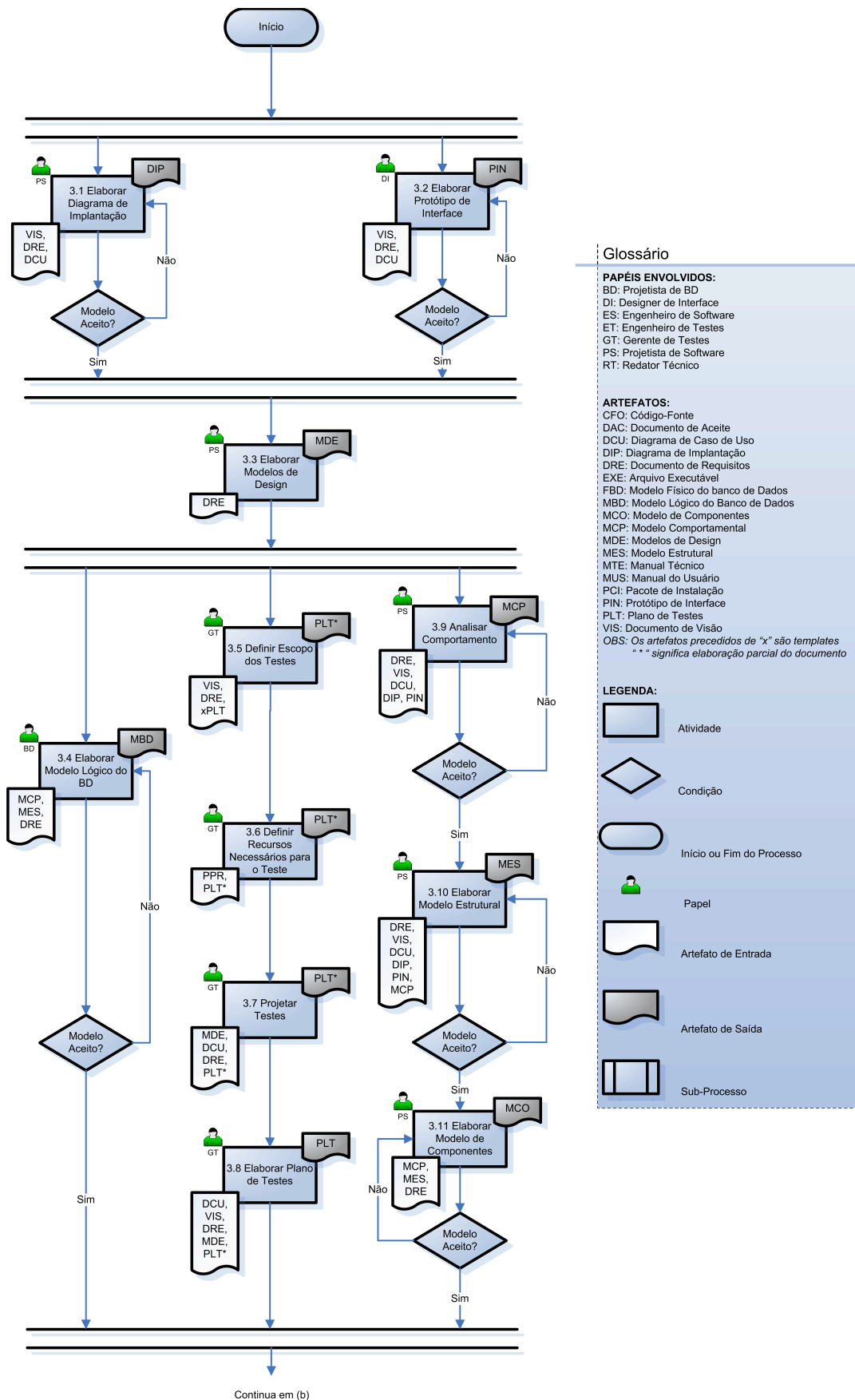
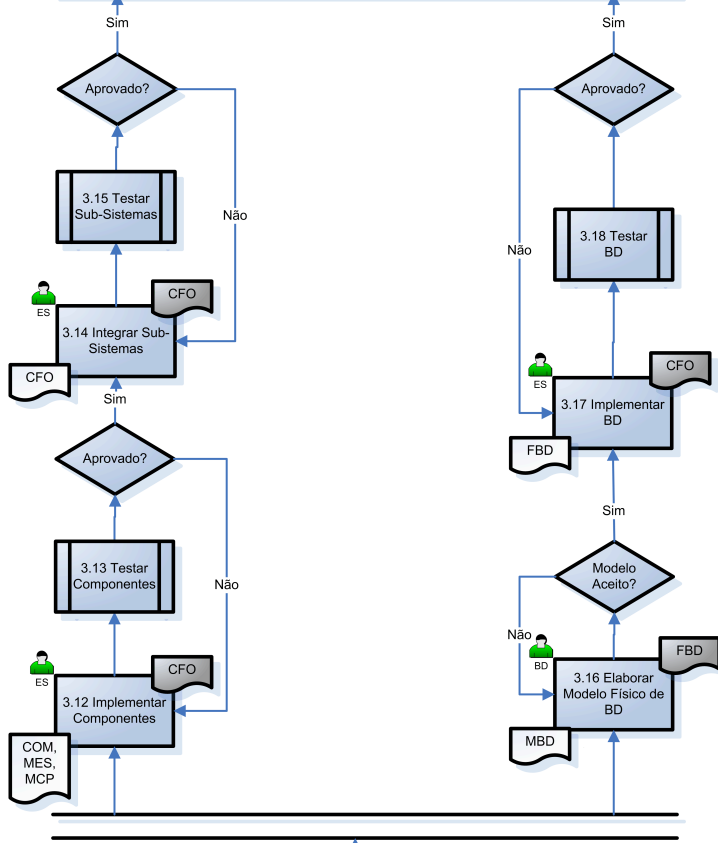
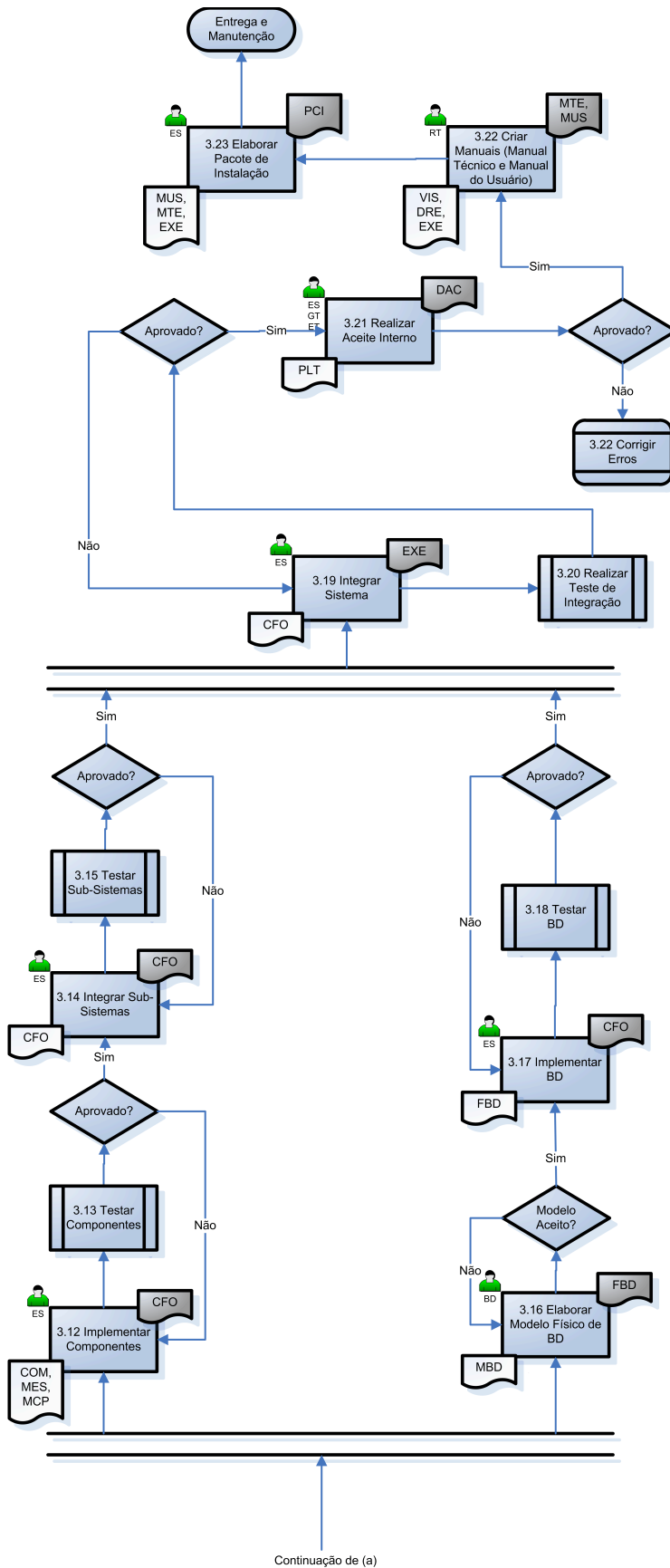


Figura 6.4(a) – Fluxo de Trabalho do processo de Modelagem e Implementação



Continuação de (a)

Figura 6.4(b) – Fluxo de Trabalho do processo de Modelagem e Implementação

Glossário

PAPÉIS ENVOLVIDOS:

BD: Projetista de BD
 ES: Engenheiro de Software
 ET: Engenheiro de Testes
 GT: Gerente de Testes
 PS: Projetista de Software
 RT: Redator Técnico

ARTEFATOS:

CFO: Código-Fonte
 DAC: Documento de Aceite Interno
 DCU: Diagrama de Caso de Uso
 DIP: Diagrama de Implantação
 DRE: Documento de Requisitos
 EXE: Arquivo Executável
 FBD: Modelo Físico do banco de Dados
 MBD: Modelo Lógico do Banco de Dados
 MCP: Modelo de Componentes
 MDE: Modelo Comportamental
 MDE: Modelos de Design
 MES: Modelo Estrutural
 MTE: Manual Técnico
 MUS: Manual do Usuário
 PCI: Pacote de Instalação
 PIN: Protótipo de Interface
 PLT: Plano de Testes
 VIS: Documento de Visão
 OBS: Os artefatos precedidos de "x" são templates
 * * * significa elaboração parcial do documento

LEGENDA:

- Atividade
- Condição
- Início ou Fim do Processo
- Papel
- Artefato de Entrada
- Artefato de Saída
- Sub-Processo
- Sub-Processo que deverá ser tratado em trabalhos futuros

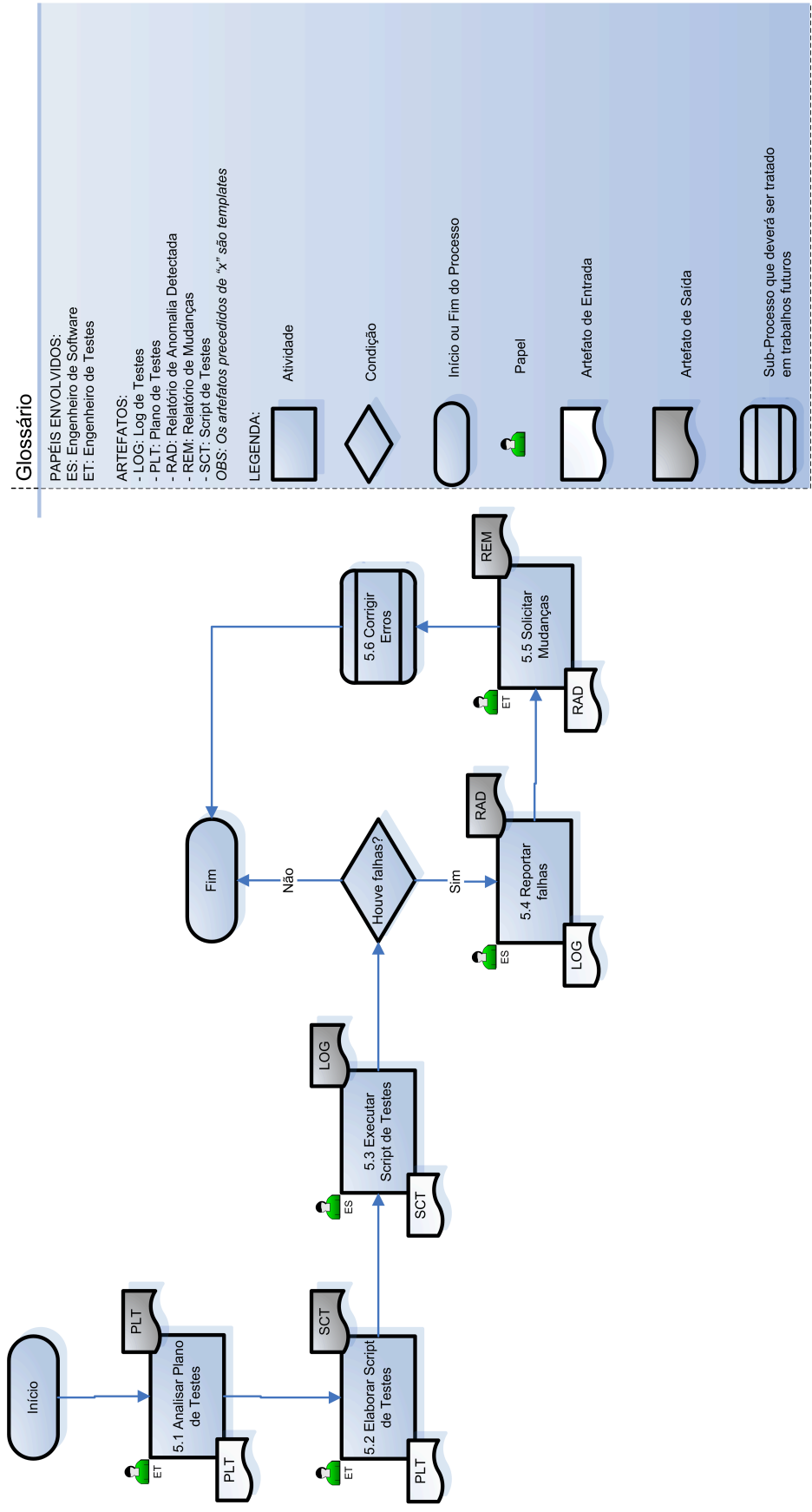


Figura 6.5 – Fluxo de Trabalho do processo de Testes

6.2.4 Entrega e Manutenção

Nessa fase, são definidas as atividades que garantem que o produto de software será disponibilizado a seus usuários finais. Ela engloba a preparação do ambiente para disponibilizar o produto de software ao usuário e a sua homologação (validação da funcionalidade pelo usuário final), como pode ser observado na Figura 6.6. Testa-se o produto no local de desenvolvimento antes dele ser finalmente oferecido ao cliente.

Durante a Entrega, é estudada a possibilidade de realizar treinamento, dependendo do que foi acordado, e são avaliados os pontos positivos e negativos do projeto. Caso não se tenha o aceite, o cliente reporta as não-conformidades que devem ser corrigidas e testadas.

Na Manutenção, é recebido um pedido, feito pelo cliente, de manutenção do produto. O pedido é analisado pela equipe responsável e, caso seja aceito, são elaboradas as alterações a serem feitas no sistema para ser adequado ao pedido de manutenção e descritas na Proposta de Alteração (Figura 6.7(a) e Figura 6.7(b)).

6.2.5 Considerações Finais sobre o Processo

O objetivo de cada uma das fases do ciclo de vida do processo, mostradas nas subseções anteriores, é a criação de produtos necessários para a construção de software de acordo com a metodologia utilizada pelo processo elaborado no presente trabalho.

Alguns artefatos gerados são de extrema importância para um produto de software de qualidade e para um entendimento entre a equipe desenvolvedora e os demais envolvidos no projeto. Para melhor acompanhamento do processo, a Tabela 6.3 mostra os principais artefatos produzidos em cada uma das fases do ciclo de vida do projeto.

Tabela 6.3 – Principais artefatos do processo

Fases do ciclo de vida	Artefatos
1. Análise de Negócio	Proposta Técnica, Proposta Comercial
2. Análise de Requisitos e Planejamento	Documento de Visão, Documento de Requisitos, Plano de Projeto
3. Modelagem e Implementação	Modelos e Diagramas, Código-Fonte, Pacote de Instalação
4. Entrega e Manutenção	Documento de Homologação, Documento de Aceite do Usuário

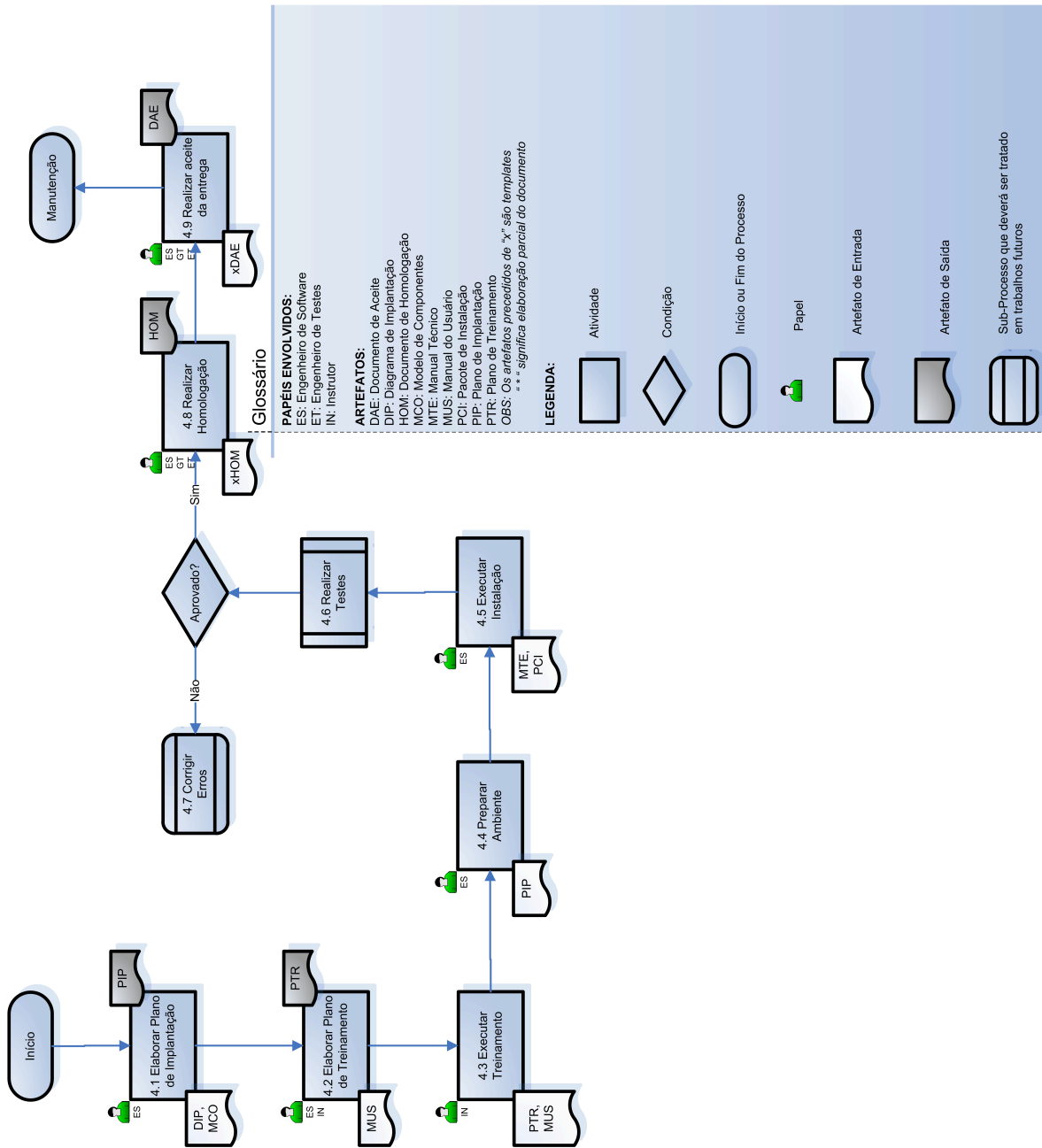
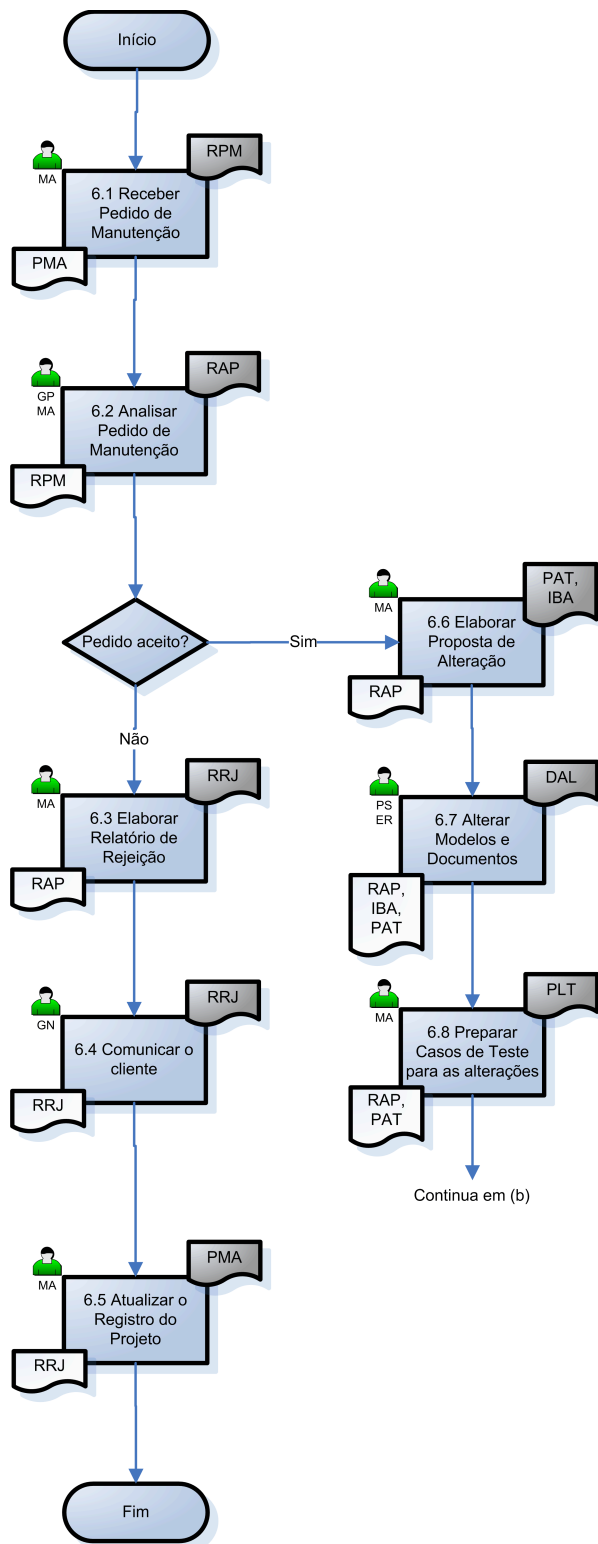


Figura 6.6 – Fluxo de Trabalho do processo de Entrega



Glossário

PAPÉIS ENVOLVIDOS:

ES: Engenheiro de Software
 ET: Engenheiro de Testes
 GN: Gerente de Negócios
 GP: Gerente de Projetos
 GT: Gerente de Testes
 MA: Mantenedor
 PS: Projetista de Software
 RT: Redator Técnico

ARTEFATOS:

CFO: Código-Fonte
 DAL: Documentos e Modelos Alterados
 DRE: Documento de Requisitos
 EXE: Arquivo Executável
 HOM: Documento de Homologação
 IBA: Itens Básicos a serem Alterados (Documentos e Modelos)**
 MTE: Manual Técnico
 MUS: Manual do Usuário
 PAT: Proposta de Alteração
 PCI: Pacote de Instalação
 PLT: Plano de Testes
 PMA: Pedido de Manutenção
 RAP: Relatório de Avaliação do Pedido
 RPM: Relatório de Pedido de Manutenção
 RRJ: Relatório de Rejeição
 VIS: Documento de Visão

OBS: Os artefatos precedidos de "x" são templates
 "x" significa elaboração parcial do documento

** devem ser definidos de acordo com particularidades de cada projeto e/ou tipo de alteração

LEGENDA:

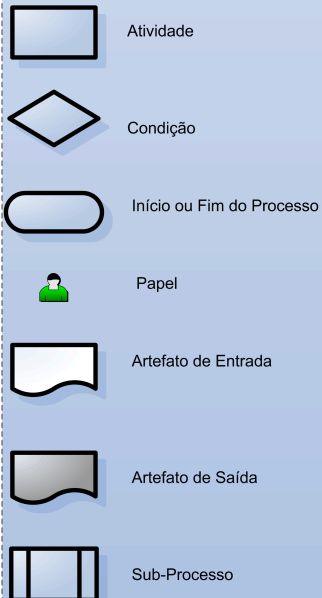


Figura 6.7(a) – Fluxo de Trabalho do processo de Manutenção

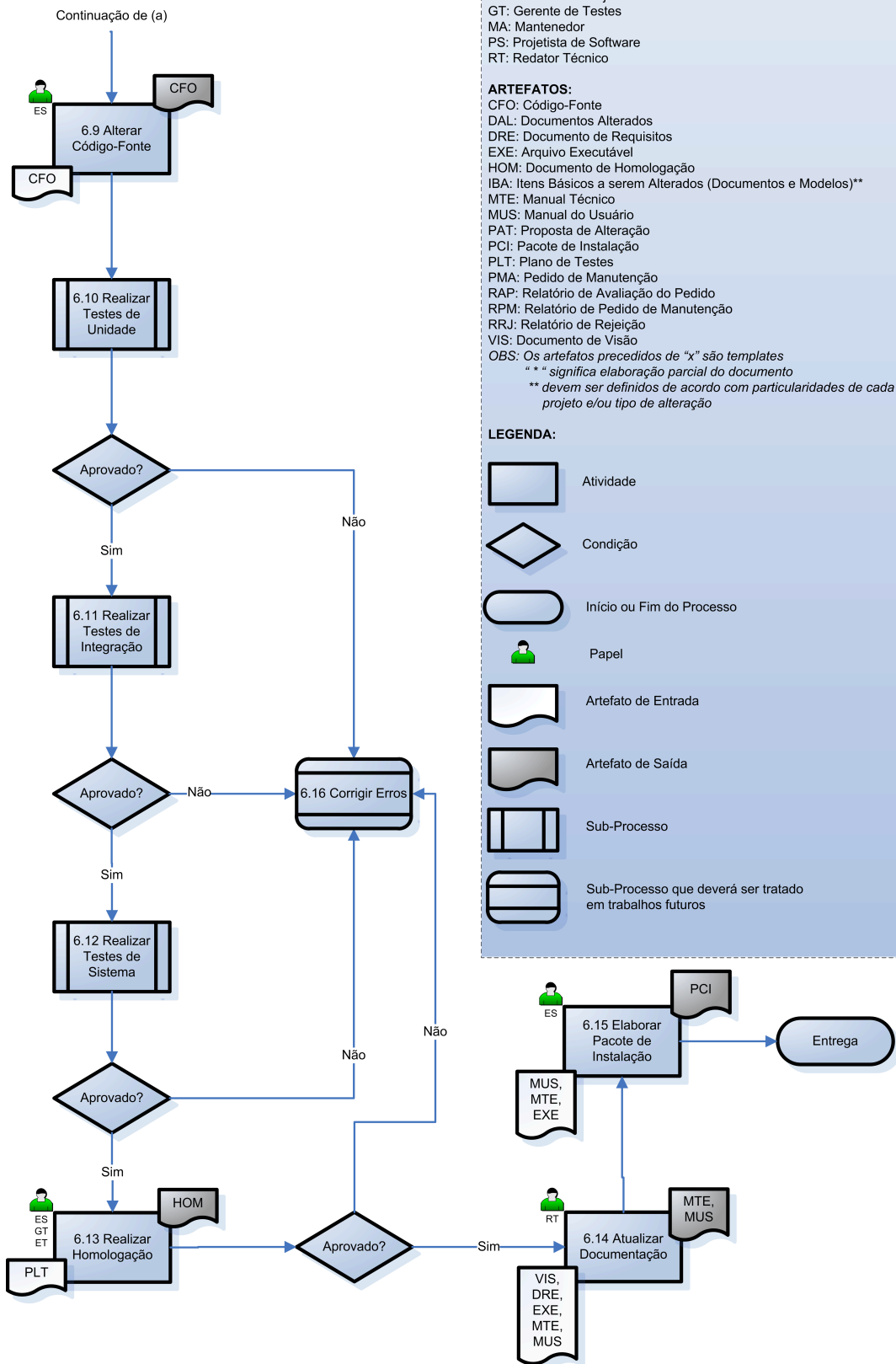


Figura 6.7(b) – Fluxo de Trabalho do processo de Manutenção

6.3 Aderência ao MPS.BR nível G

Após a elaboração e a modelagem, foi proposta uma aderência do processo desenvolvido à uma série de resultados esperados para a sua adequação ao primeiro nível de maturidade do MPS.BR (nível G).

Os resultados esperados são resultados observáveis que demonstram que os objetivos gerais da execução do processo foram atingidos com sucesso. Um resultado pode ser um artefato produzido, uma mudança significativa de estado e o atendimento das especificações, por exemplo, metas e requisitos.

Segundo o MPS.BR – Guia de Implementação – Parte 1 (2007), o nível G descreve dois processos: Gerência de Projetos (GPR) e Gerência de Requisitos (GRE), necessários para que se atinja o primeiro nível de maturidade.

Nas próximas subseções, é apresentado como o processo desenvolvido implementa os resultados esperados do GPR e do GRE.

6.3.1 Gerência de Projetos (GPR)

Para o processo de Gerência de Projetos, os seguintes resultados esperados são listados. O processo desenvolvido que descreve as atividades do gerenciamento de projetos pode ser observado na Figura 6.8(a) e na Figura 6.8(b).

- GPR1 – O escopo do trabalho para o projeto é definido;
- GPR2 – As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
- GPR3 – O modelo e as fases do ciclo de vida do projeto são definidas;
- GPR4 – O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;
- GPR5 – O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;
- GPR6 – Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
- GPR7 – Os recursos humanos para o projeto são planejados considerando o perfil e

o conhecimento necessário para a sua executá-lo;

- GPR8 – As tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;
- GPR9 – Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, de armazenamento e de distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente questões de privacidade e segurança;
- GPR10 – Planos para a execução do projeto são estabelecidos e reunidos no Plano de Projeto;
- GPR11 – A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados;
- GPR12 – O Plano de Projeto é revisado com os interessados e o compromisso com ele é obtido;
- GPR13 – O progresso do projeto é monitorado com relação ao estabelecido no Plano de Projeto e os resultados são documentados;
- GPR14 – O envolvimento das partes interessadas no projeto é gerenciado;
- GPR15 – Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;
- GPR16 – Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;
- GPR17 – Ações para corrigir desvios em relação ao planejado e para prevenir a repetição de problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão.

A definição do escopo deve estabelecer o que está e o que não está incluído no projeto. Para isso, deve-se definir o objetivo, a motivação, os limites, as restrições, os produtos que serão entregues e os outros produtos gerados pelo projeto. O processo proposto representa o escopo no Documento de Visão, construído durante a fase Análise de Requisitos e Planejamento, que define, claramente, o escopo do trabalho.

Após a definição do escopo do projeto e através da construção de uma Estrutura

Análítica do Projeto (EAP), as tarefas e os produtos de trabalho do projeto são dimensionados. A estimativa é feita baseada na complexidade, no número de requisitos ou no uso da EAP juntamente com dados históricos e a experiência em projetos anteriores, atingindo o GPR2.

O ciclo de vida do projeto consiste de fases e de atividades que devem ser definidas de acordo com a sua natureza, o escopo dos requisitos e as estimativas para os recursos com o objetivo de oferecer maior controle gerencial do projeto.

O ciclo de vida do projeto define um conjunto de fases, em que cada fase gera produtos de trabalho necessários para o desenvolvimento de fases posteriores. Essa organização em fases permite planejar o projeto incluindo marcos importantes para o controle e as revisões. Para implementar o GPR3, o processo de Gerência de Projetos apresenta a Atividade 7.5 – Definir fases do ciclo de vida do projeto. Nesta atividade, as fases do ciclo de vida do projeto são definidas e documentadas na Lista de Marcos, Artefato de Saída da atividade.

O artefato Lista de Atividade contém a definição e o sequenciamento das atividades. Através desse artefato, o Líder de Projeto estima os recursos, a duração e os custos necessários para as atividades listadas. As estimativas de esforço e de custo são, normalmente, baseadas nos resultados de análises utilizando modelos e/ou dados históricos aplicados ao tamanho, atividades e outros parâmetros de planejamento.

Com base na EAP e nas estimativas de esforço e de custo (GPR4), define-se o cronograma, considerando as dependências entre as atividades. Através da elaboração do cronograma e de acordo com a estimativa de custos, é estabelecido o orçamento do projeto.

Vários documentos são estabelecidos durante o andamento do processo, como Plano de Riscos, Plano de Gerenciamento de Recursos Humanos, Plano de Gerenciamento de Recursos e Ambiente e o Plano de Gerenciamento da Comunicação. Esses artefatos são reunidos no Plano de Projeto.

O Plano de Projeto é revisado com a equipe e os envolvidos no projeto. Após o aceite, é executado um monitoramento contínuo do andamento do projeto. São identificados, analisados, registrados, corrigidos e documentados os problemas para prevenir sua repetição.

A Tabela 6.4 mostra os resultados esperados do GPR, os artefatos construídos e as atividades do processo proposto que implementam esses resultados.

Tabela 6.4 – Implementação dos resultados esperados (GPR)

Resultado Esperado	Artefato	Atividade
GPR1	Documento de Visão (VIS)	2.2 Desenvolver Visão
GPR2	Estrutura Analítica do Projeto (EAP)	7.4 Criar EAP
GPR3	Lista de Marcos (LMA)	7.5 Definir fases do ciclo de vida do projeto
GPR4	Lista de Atividades (LAT)	7.6 Definir atividades 7.7 Seqüenciar atividades 7.8 Estimar recursos das atividades 7.9 Estimar duração das atividades 7.10 Estimar custos das atividades
GPR5	Cronograma (CNG), Orçamento (ORM)	7.12 Desenvolver cronograma 7.18 Desenvolver orçamento
GPR6	Plano de Riscos do Projeto (PRP)	7.13 Identificar riscos do projeto
GPR7	Plano de Gerenciamento de Recursos Humanos (PRH)	7.14 Planejar recursos humanos
GPR8	Plano de Gerenciamento de Recursos e Ambiente (PRA)	7.15 Planejar os recursos e o ambiente
GPR9	Plano de Gerenciamento da Comunicação (PGC)	7.19 Distribuir as informações do projeto
GPR10	Plano do Projeto (PPR)	7.20 Elaborar Plano do Projeto
GPR11	Proposta Técnica (PTE)	1.10 Avaliar propostas
GPR12	---	7.21 Revisar Plano de Projeto
GPR13	Plano do Projeto (PPR)	7.22 Monitorar andamento do projeto
GPR14	Plano de Gerenciamento da Comunicação (PGC)	7.16 Planejar a comunicação
GPR15	---	7.22 Monitorar andamento do projeto
GPR16	Plano do Projeto (PPR)	7.23 Identificar, analisar e registrar os problemas
GPR17	Plano do Projeto (PPR)	7.24 Corrigir erros 7.25 Documentar resultados

Glossário

PAPEIS ENVOLVIDOS:

CL: Cliente
 EQ: Equipe
 LP: Líder do Projeto

ARTEFATOS:

DRE: Documento de Requisitos
 DRP: Documento de Requisitos Preliminar
 EAP: Estrutura Analítica do Projeto
 CNG: Cronograma
 HIS: Histórico de Estimativas
 LAT: Lista de Atividades
 LMA: Lista de Marcos
 ORM: Orçamento
 PGC: Plano de Gerenciamento da Comunicação
 PGE: Plano de Gerenciamento de Escopo
 PRG: Plano de Gerenciamento de Recursos e Ambiente
 PRH: Plano de Gerenciamento de Riscos e Ambientes
 PRP: Plano de Gerenciamento de Recursos Humanos
 PRP: Plano de Riscos do Projeto
 PRP: Plano de Projeto
 OBS: Os artefatos precedidos de "x" são templates
 "x" significa elaboração parcial do documento

LEGENDA:

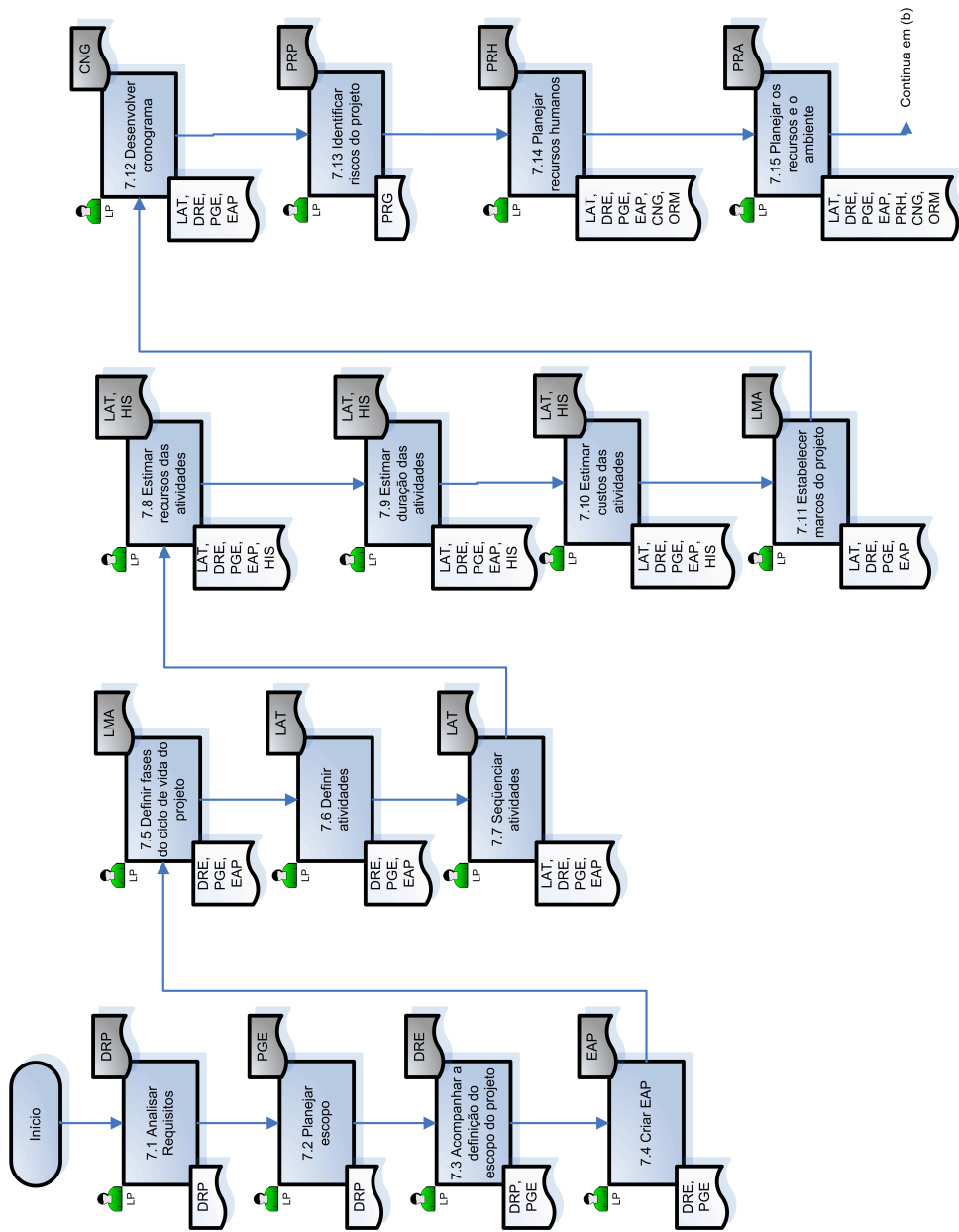
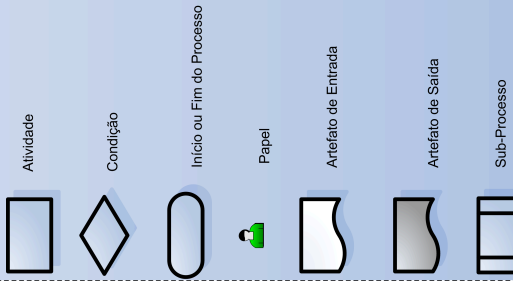


Figura 6.8(a) – Fluxo de Trabalho do processo de Gerência de Projetos

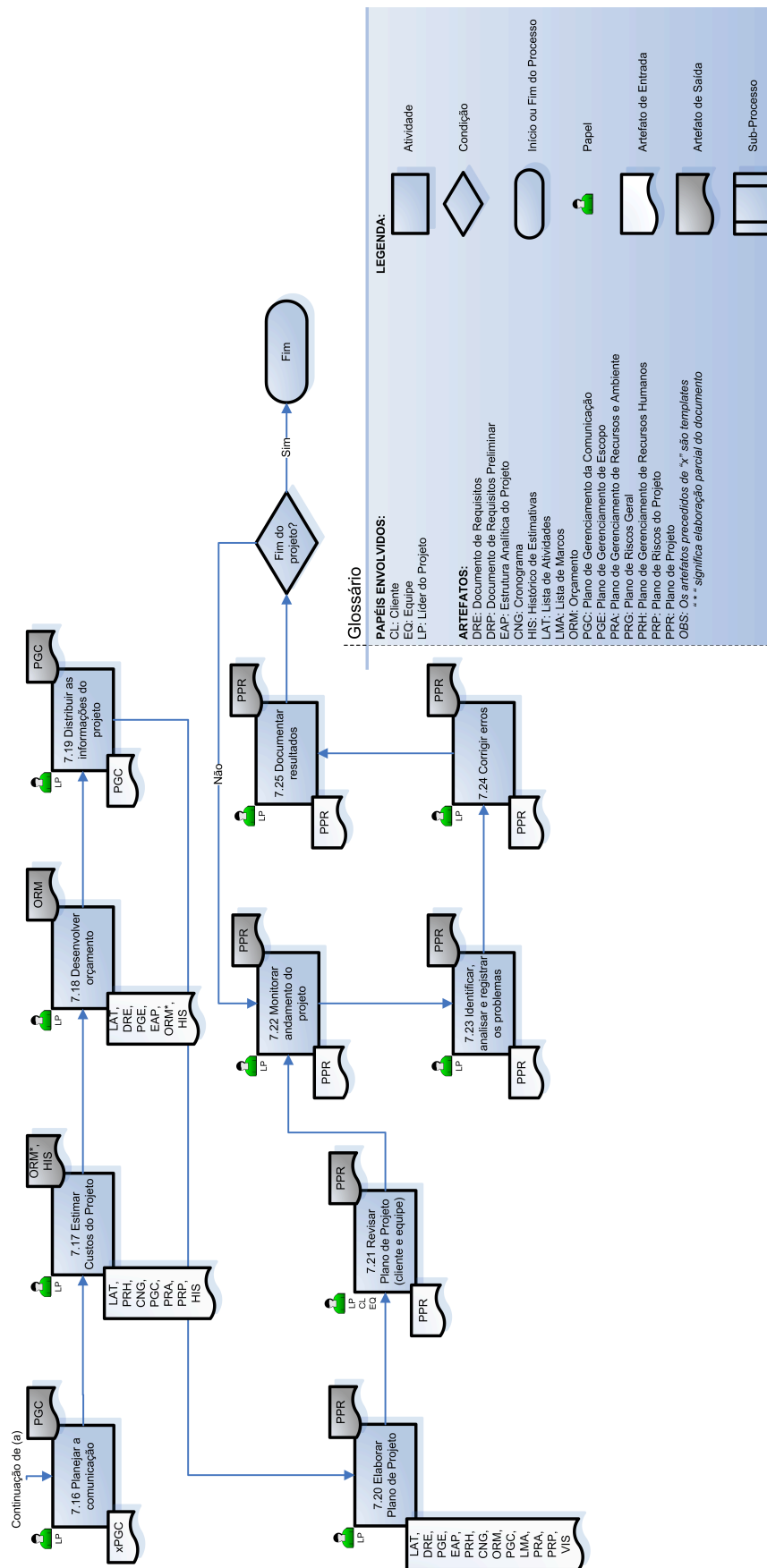


Figura 6.8(b) – Fluxo de Trabalho do processo de Gerência de Projetos

6.3.2 Gerência de Requisitos (GRE)

De acordo com o Guia de Implementação – Parte 1: Nível G do MPS.BR, os seguintes resultados esperados são listados para o processo de Gerência de Requisitos:

- GRE1 – O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
- GRE2 – Os requisitos de software são aprovados utilizando critérios objetivos;
- GRE3 – A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- GRE4 – Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
- GRE5 – Mudanças nos requisitos são gerenciadas ao longo do tempo;

O bom entendimento dos requisitos do projeto é um dos fatores determinantes para o seu sucesso. O objetivo do GRE1 é garantir que os requisitos estejam claramente definidos e revistos com o cliente para garantir que as necessidades e as expectativas sejam atendidas com os requisitos propostos.

Esse objetivo é implementado através da elicitação dos requisitos, elaboração de um Documento de Requisitos e da sua revisão junto ao cliente. Os fornecedores de requisitos e a definição de como as mudanças poderão ser solicitadas podem ser registrados no Documento de Requisitos.

O GRE2 define que os requisitos de software estabelecidos devem ser aprovados pelas partes interessadas, após uma avaliação desses requisitos com base em um conjunto de critérios objetivos, previamente estabelecidos.

Para atingir esse resultado, são definidas as atividades para revisar e aprovar os requisitos: Atividade 2.7 – Revisar Documento de Requisitos e Atividade 2.8 – Revisar Documento de Requisitos com o cliente. No final dessas atividades, os requisitos são aprovados entre os interessados e o Documento de Requisitos é assinado por ambas as partes (contratada e contratante).

Durante a Atividade 7.1 – Analisar Requisitos, deve-se estabelecer a criação de um sistema de rastreamento, permitindo rastrear a dependência entre os requisitos e os produtos de trabalho a fim de facilitar a avaliação do impacto das mudanças de requisitos que possam ocorrer, por exemplo, nas estimativas do escopo, nos produtos de trabalho ou nas tarefas do projeto.

Após analisados os requisitos e durante a Atividade 7.22 – Monitorar o andamento

do projeto, são feitas revisões nos planos e nos produtos de trabalho do projeto visando identificar e corrigir inconsistências em relação aos requisitos. Quando há mudança nos requisitos, uma vez identificados os impactos da mudança com a utilização do instrumento de rastreabilidade dos requisitos, é importante verificar se os demais artefatos estão consistentes com as alterações realizadas.

A Tabela 6.5 mostra os resultados esperados do GRE, os artefatos construídos e as atividades do processo proposto que implementam esses resultados.

Tabela 6.5 – Implementação dos resultados esperados (GRE)

Resultado Esperado	Artefato	Atividade
GRE1	Documento de Requisitos (DRE)	2.4 Elicitar requisitos 2.5 Desenvolver Documento de Requisitos 2.7 Revisar Documento de Requisitos 2.8 Revisar Documento de Requisitos com o cliente
GRE2	Documento de Requisitos (DRE)	2.7 Revisar Documento de Requisitos 2.8 Revisar Documento de Requisitos com o cliente
GRE3	---	7.1 Analisar Requisitos
GRE4	Plano do Projeto (PPR)	7.22 Monitorar andamento do projeto
GRE5	Plano do Projeto (PPR)	7.22 Monitorar andamento do projeto

6.4 Considerações Finais

Durante a execução desse trabalho, foi elaborado um processo de desenvolvimento de software de acordo com os resultados esperados dos processos necessários para a adequação ao primeiro nível do MPS.BR.

De acordo com as definições feitas por Pressman (2002) e Rocha *et al* (2004), apresentadas na Seção 3.4, o processo desenvolvido pode ser classificado como tradicional por ser fundamentado em fluxos de trabalho e ter foco principal na especificação dos requisitos, o qual é base para vários outros documentos do processo e, principalmente, construído para uma Fábrica de Projetos Ampliada, como definido na Seção 3.3 por Fernandes *et al* (2004).

Como o processo não foi implantado, não foi possível capturar dados de projetos reais para uma análise dos pontos fracos e pontos fortes do processo. Com a implantação do processo em uma fábrica de software, espera-se aumentar a qualidade dos produtos, a

produtividade e a confiabilidade dos prazos estabelecidos e diminuir os custos do processo de desenvolvimento.

Os fluxos de trabalho modelados foram exportados para HTML, facilitando a sua disponibilização para a equipe em um ambiente Web, como pode ser visto na Figura 6.9.

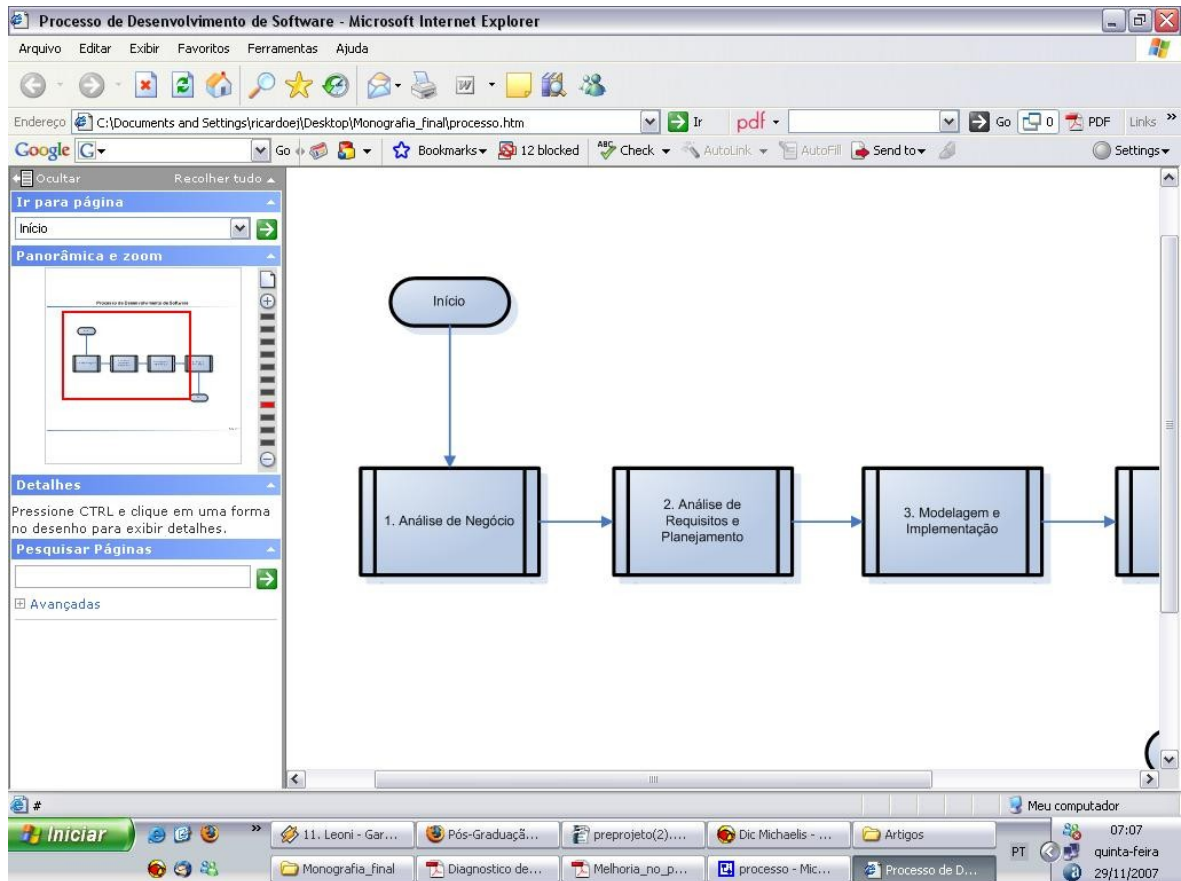


Figura 6.9 – Processo disponibilizado em um ambiente Web

Conceitualmente, o processo foi concebido e adaptado para ser aderente ao nível G do MPS.BR. Atualmente, o processo está sendo preparado para ser colocado em prática na Fábrica de Software do Departamento de Ciência da Computação (DCC) da Universidade Federal de Lavras (UFLA) em conjunto com a CompJr.

7. CONSIDERAÇÕES FINAIS

7.1 Conclusões

Este trabalho buscou elaborar e modelar um processo de desenvolvimento de software, aplicado a Fábricas de Software, seguindo algumas normas que proporcionassem algumas melhorias no processo.

Para tanto, utilizou-se o modelo MPS.BR – Melhoria de Processo de Software Brasileiro, como modelo de referência para aplicar a melhoria do processo de software atendendo ao nível G. Esse modelo se mostrou como uma alternativa viável para a implantação de melhorias de uma maneira gradual e, segundo o MPS.BR – Guia Geral (2006), voltado à realidade das empresas brasileiras.

A abordagem teórica relacionada à Engenharia, Qualidade e Modelo de Qualidade do Processo de Software foi imprescindível para a concepção deste trabalho, em função dos conceitos, práticas e metodologias apreendidos. O processo desenvolvido foi baseado em atividades consideradas essenciais pela literatura para a construção efetiva de produtos de software.

A qualidade dos processos das empresas desenvolvedoras afeta diretamente o produto ou o serviço gerado por estes processos, tornando-se uma vantagem competitiva entre elas.

Com a implantação do processo elaborado no presente trabalho em uma fábrica de software, pretende-se construir software com mais qualidade, menos custos e dentro dos prazos estipulados.

As fases do ciclo de vida do processo, seus principais artefatos e atividades foram apresentados mostrando sua importância durante o desenvolvimento do produto.

Finalmente, uma adequação do processo desenvolvido ao conjunto de resultados esperados pelo MPS.BR nível G foi demonstrado. Teoricamente, observou-se que o processo atende significativamente aos objetivos gerais da execução do processo descritos no nível G do modelo MPS.BR. Foi analisando a presença de artefatos ou atividades no processo desenvolvido que implementam os resultados esperados do GPR e do GRE.

7.2 Contribuições

Os seguintes pontos se destacam como contribuições deste trabalho:

- elaboração de um processo de desenvolvimento de software, aplicável em Fábricas de Software, de acordo com as atividades essenciais para a construção efetiva de software denominadas Análise e Definição de Requisitos, Projeto Arquitetural e Detalhado, Codificação, Testes e Entrega e que engloba desde o estabelecimento do contrato inicial até a fase de entrega, homologação do produto e manutenção do software;
- modelagem do processo através de uma representação com notação gráfica, indicando os papéis, artefatos de entrada e de saída, atividades e fluxo das atividades, facilitando assim, a compreensão e acompanhamento do processo;
- a realização sistemática das atividades do processo com qualidade diminui os defeitos do produto ou serviço final, uma vez que suas atividades serão realizadas com maior monitoração e controle;
- destacou-se a importância de modelos de melhoria de processos de software, como o MPS.BR, demonstrando como implementar essas melhorias em um processo.

7.3 Trabalhos Futuros

Alguns desdobramentos de trabalhos futuros que podem ser vislumbrados, a saber:

- implantação do processo desenvolvido em Fábricas de Software para um acompanhamento da efetividade do processo em projetos reais;
- detalhamento do sub-processo de Corrigir Erros;
- adequação do processo a níveis subseqüentes de maturidade do modelo MPS.BR;
- automação do processo de desenvolvimento através de uma ferramenta de gerenciamento de workflows (Sistema de Gerenciamento de Workflows).

8. REFERÊNCIAS BIBLIOGRÁFICAS

AAEN, I; BOTTCHEER, P.; MATHIASSEN, L. **The Software Factory: Contributions and Illusions**, In the Proceedings of the Twentieth Information Systems Research Seminar, Scandinavia, Oslo, 1997.

Associação Brasileira de Normas Técnicas. NBR ISO 8402/1994 – Gestão da qualidade e garantia da qualidade - Terminologia. Rio de Janeiro: ABNT, 1994.

Associação para Promoção da Excelência do Software Brasileiro. Guia Geral do MPS.BR. Abril, 2005.

Associação para Promoção da Excelência do Software Brasileiro. Guia de Implementação – Parte 1: Nível G. Dezembro, 2006.

Associação para Promoção da Excelência do Software Brasileiro. Guia de Avaliação do MPS.BR. Dezembro, 2006.

ARAÚJO, R. M.; BORGES, M. R. S. **Sistemas de Workflow**. XX Jornada de Atualização em Informática – Congresso da Sociedade Brasileira de Computação. Fortaleza, CE, 2001.

BECK, K. **Extreme Programming Explained**, Addison-Wesley, 2000.

CÂNDIDO, E. J. D. **Uma Simplificação da Técnica Análise de Pontos de Função para Estimar Tamanho de Aplicativos Web**. 2004. 120p. Dissertação (Mestrado em Computação) – Universidade de São Paulo, São Paulo.

CASTOR, E. M. **Fábrica de Software: Passado, Presente e Futuro**. Pós-Graduação Lato Sensu em Tecnologia da Informação - UNIRRATEC - União dos Institutos Brasileiros de Tecnologia, 2004.

CUSUMANO, M. A. **Japan's Software Factories**. Oxford University Press, 1991.

DAVIS, A. M.; SITARAM, P. A. **Concurrent Process Model of Software Development**. *ACM SIGSOFT Software Engineering Notes*, v. 9, n. 2, p. 38–51, abril 1994.

FERNANDES, A. A.; TEIXEIRA, S. D. **Fábrica de Software: Implantação e Gestão de Operações**. Instituto de Tecnologia de Software. São Paulo: Atlas, 2004.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **Information technology – Software life cycle processes: ISO/IEC 12207:1995**. Genebra, Suíça, 1995.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **ISO/IEC 15504-1: Information Technology - Process Assessment – Part 1 - Concepts and Vocabulary**. Geneve: ISO, 2004.

JUNG, C. F. **Metodologia Para Pesquisa & Desenvolvimento – Aplicada a Novas Tecnologias, Produtos e Processos**. Rio de Janeiro: Axcel Books, 2004. 328p.

LONCHAMP, J. **A Structured Conceptual and Terminological Framework for the Software Process Engineering**, In: International Conference On The Software Process, 1993, Berlin. Proceedings... Berlin: IEEE Computer Society

PESSOA, M. S. P. **Introdução ao CMM - Modelo de Maturidade da Capacidade de Processo de Software**. Lavras: UFLA/FAEPE, 2003. Curso de Pós-graduação “Lato Sensu” à Distância: Melhoria de Processo de Software. 77 p.

PFLEEGER, Shari Lawrence. **Engenharia de Software –Teoria e Prática**. 2ª edição. São Paulo: Pearson Prentice Hall, 2004.

PMBOK, 2004 - PROJECT MANAGEMENT INSTITUTE – PMI. **A Guide to the Project Management Body of Knowledge - PMBOK™**, Syba: PMI Publishing Division, 2004. Disponível em: <www.pmi.org>.

PRESSMAN, ROGER S. **Engenharia de Software**. 5. ed., Rio de Janeiro: McGraw-Hill, 2002. 843p.

REIS, C. R. **Caracterização de um processo de software para projetos de software livre**. Dissertação de Mestrado, Universidade de São Paulo, São Carlos – SP, 2003.

ROCHA, T. A. da; OLIVEIRA, S. R. B.; VASCONCELOS, A. M. L. de (2004) **Adequação de Processos para Fábricas de Software**, In: VI Simpósio Internacional de Melhoria de Processos de Software (SIMPROS 2004), São Paulo, SP, novembro de 2004.

SIY, H, P., MOCKUS, A., HERBSLEB, J, D., KRISHNAN, M., TUCKER, G, T., **Making The Software Factory Work:Lessons from a decade of experience**. In the Seventh International Symposium on Software Metrics, London, England, April 2001.

SOMMERVILLE, I. **Engenharia de Software**. 6^a ed. São Paulo, Addison Wesley, 2003.

SOUZA, A. D. **Estudo e avaliação da área de processo de planejamento de projeto de acordo com o modelo CMMI-SW Nível 2 na empresa SWQuality situada em Lavras-MG**. 2004. Monografia (Graduação em Ciência da Computação) – Universidade Federal de Lavras, Lavras. Disponível em, <<http://www.comp.ufla.br/monografias/ano2004/ano2004.htm>>. Acessado no dia 06 de julho de 2007.

VASCONCELOS, A. M. L de; MACIEL, T. M. de M. **Introdução à engenharia de software e aos princípios de qualidade**. Lavras: UFLA/FAEPE, 2003. Curso de Pós-graduação “Lato Sensu” à Distância: Melhoria de Processo de Software. 104 p.