

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARIDIANE CORRÊA PLÁCIDO

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

CRICIÚMA, NOVEMBRO DE 2010

MARIDIANE CORRÊA PLÁCIDO

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

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

Orientador: Prof. MSc. Gustavo Bisognin

CRICIÚMA, NOVEMBRO DE 2010

MARIDIANE CORRÊA PLÁCIDO

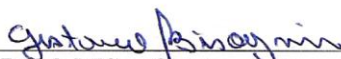
**DESENVOLVIMENTO DE UMA METODOLOGIA PARA O GERENCIAMENTO
DE PROJETOS BASEADO NA ÁREA DE PROCESSO: GERÊNCIA DE
PROJETOS (GPR) DO MODELO DE MATURIDADE MPS- BR**

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



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

Banca Examinadora:



Prof. MSc. Gustavo Bisognin (UNESC)
Orientador



Prof.^a MSc. Ana Claudia Garcia Barbosa (UNESC)



Prof. Esp. Cristiano Damiani Tomasi (FASC)

Dedico este trabalho aos meus familiares e amigos por todo apoio recebido ao longo de minha formação e pela confiança e compreensão.

AGRADECIMENTOS

Quero agradecer a Deus e aos anjos que iluminaram o meu caminho e guiaram os meus passos.

A minha família por todo apoio e incentivo nos momentos de dificuldades.

Aos meus pais que nunca mediram esforços na minha educação e acreditaram nos meus sonhos, incentivando-me nos bons e maus momentos ao longo desse período de estudo até a realização deste trabalho.

Aos meus futuros sogros, pela confiança depositada em mim e por sempre estarem ao meu lado, apoiando-me e me incentivando.

Ao meu noivo, Diego, pelo carinho e compreensão demonstrados nessa longa caminhada.

Agradeço, em especial, o meu orientador, Prof. MSc Gustavo Bisognin, o qual admiro e por quem tenho profundo respeito, por orientar este trabalho e pela dedicação e auxílio prestado na sua elaboração.

Por fim, a todos os colegas e professores do curso de Ciência da Computação, especialmente a Margarete, a secretária do departamento que com todo carinho sempre me apoiou nos momentos difíceis desses anos de estudo.

Finalmente, agradeço a todos que de alguma forma contribuíram e me ajudaram indiretamente na conclusão deste trabalho.

*"Na vida, não vale tanto o que temos, nem tanto importa o que somos.
Vale o que realizamos com aquilo que possuímos e, acima de tudo, importa o que fazemos de nós!"*

Chico Xavier

RESUMO

Com o constante avanço das tecnologias disponíveis e técnicas de desenvolvimento, a metodologia de trabalho destaca-se como um diferencial das organizações e, conseqüentemente, a qualidade dos produtos de software. Para padronizar as boas práticas de desenvolvimento de sistemas, surgiram os modelos de processos de desenvolvimento de software, como, por exemplo, a ISO, O CMMI, o MPS.BR, entre outros . Além da existência de modelos, as empresas precisam estruturar-se para customizá-los a fim de obter resultados. Porém, isso é um grande desafio, pois, para definir e manter um processo em constante melhoria, é necessário que haja uma equipe qualificada para o desenvolvimento da qualidade dos processos, organizada para garantir a evolução contínua do modelo. Nesse contexto, o presente trabalho apresenta a definição de uma metodologia para o processo de desenvolvimento baseado no modelo de referência MPS.BR e mostra ainda as melhores práticas de gerenciamento de projetos para profissionais da área de tecnologia da informação. Espera-se que a utilização deste trabalho possa contribuir e auxiliar as empresas que buscam adequar seus processos dentro de um contexto de qualidade. Além disso, a utilização de um processo aderente e baseado em um modelo de maturidade fortalece a saúde financeira da organização, agregando qualidade aos seus produtos e serviços, além de fortalecer a sua marca.

Palavra-chave: Engenharia de Software; Melhoria de Processo de Software; MPS.BR.

ABSTRACT

With the great advancement of available technologies and development techniques, the methodology of work stands out as a differential of organizations and, consequently, the quality of software products. To standardize the best practices of systems development, process models of software development have emerged, such as ISO, CMMI, the MPS.BR, among others. Besides the existence of models, companies must be structured to customize them to achieve results. However, this is a big challenge, therefore, to define and maintain a continuous improvement process, there must be a qualified team for the development of quality processes, organized to ensure the continued evolution of the model. In this context, this paper presents the definition of a methodology for the development process based on the reference model MPS.BR and brings the best practices for project management professionals in Information Technology. It is expected that the use of this work can contribute and assist companies that seeks to adapt its processes inside a context of quality. Furthermore, the use of a favorable process and based on a maturity model strengthens the organization's financial issues, adding quality to their products and services, and make the brand stronger.

Keywords: Software Engineering; Software Process Improvement; MPS.BR

LISTA DE ILUSTRAÇÕES

Figura 1. Modelo Cascata.....	26
Figura 2. Modelo Protótipo	28
Figura 3. Modelo Espiral	29
Figura 4. Modelo Iterativo e Incremental	31
Figura 5. Níveis de Maturidade do CMMI.....	37
Figura 6. Construção do modelo MPS.BR	39
Figura 7. Estrutura do MPS.BR.....	40
Figura 8. Estrutura de Níveis de Maturidade.....	42
Figura 9. Processos do gerenciamento de projetos do PMBOK	53
Figura 10. Integração entre os processos de gerenciamento de projetos através de um ciclo do tipo PDCA.....	54
Figura 11. As áreas de conhecimento de gerenciamento do PMBOK	56
Figura 12. Conceitos principais do EPF que são baseados no UMA	62
Figura 13. Os elementos do Conteúdo de Método	64
Figura 14. Área de processo do MPS.BR configurada como disciplina no EPF	65
Figura 15. Ciclo de Vida do Processo	66
Figura 16. Padrão de Recurso para reutilização das Fases	66
Figura 17. Conteúdo de Método do Processo.....	67
Figura 18. Tarefas disponíveis para serem utilizadas no Processo	68
Figura 19. Visualização da Tarefa: Alocar Gerente de Projetos	69
Figura 20. Funções disponíveis para definição do Processo	69
Figura 21. Modelagem da Atividade: Iniciação	72
Figura 22. Modelagem da Atividade: Planejamento	74
Figura 23. Modelagem da Atividade: Execução	77

Figura 24. Modelagem das Atividades: Acompanhamento e Controle.....	79
Figura 25. Modelagem da Atividade: Encerramento	80

LISTA DE QUADROS

Quadro 1. Comparação entre os níveis de Capacidade e de Maturidade	36
Quadro 2. Níveis de Maturidade de Processos	43

LISTAS DE ABREVIACÕES E SIGLAS

AP	Atributos de Processo
CMMI	Capability Maturity Model Integration
EPF	Eclipse Process Framework Composer
EUA	Estados Unidos da América
GPR	Gerência de Projetos
GRE	Gerência de Requisitos
IBM	International Business Machines
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
MA – MPS	Método de Avaliação de Melhoria de Processos do Software
MN – MPS	Modelo de Negócios de Melhoria de Processos do Software
MPS.BR	Melhoria de Processo do Software Brasileiro
MR – MPS	Modelo de Referência de Melhoria de Processos do Software
OMG	Object Management Group
PDCA	Plan, Do, Control, Action (Planejar, Executar, Controlar, Agir)
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PMP	Project Management Professional
RAP	Resultado do Atributo de Processo
RUP	Rational Unified Process
SEI	Software Engineering Institute
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro
SPEM	Software Process Engineering Metamodel

TI	Tecnologia da Informação
UMA	Unified Method Architecture
UML	Unified Modeling Language

SUMÁRIO

1	INTRODUÇÃO	16
1.1	OBJETIVO GERAL	18
1.2	OBJETIVO ESPECÍFICOS	18
1.3	JUSTIFICATIVA.....	18
1.4	ESTRUTURA DO TRABALHO.....	21
2	ENGENHARIA DE SOFTWARE	22
2.1	A CRISE DE SOFTWARE.....	22
2.2	MODELOS DE CICLOS DE VIDA DE SOFTWARE.....	24
2.2.1	Modelo Cascata	25
2.2.2	Modelo Protótipo	27
2.2.3	Modelo Espiral	29
2.2.4	Modelo Iterativo e Incremental	31
3	MELHORIA DO PROCESSO DE SOFTWARE	32
3.1	QUALIDADE DE SOFTWARE	33
3.1.1	Qualidade do Produto x Qualidade do Processo	33
3.2	MODELOS DE QUALIDADE DE PROCESSOS DE SOFTWARE.....	35
3.2.1	CMMI	35
3.2.2	MPS.BR	38
3.2.2.1	Níveis de Maturidade de Processos	41
3.2.2.2	Nível G - Parcialmente Gerenciado	44
3.2.2.3	Capacidade do Processo	46
4	GERÊNCIA DE PROJETOS	49
4.1	GERENCIAMENTO DE PROJETOS SEGUNDO O PMBOK	49

4.1.1	Definição de Projeto	50
4.1.2	O Gerente de Projetos	51
4.1.3	Processos de Gerenciamento de Projetos de um Projeto	52
4.1.4	As áreas de Conhecimento de Gerenciamento de Projetos	56
5	TRABALHOS CORRELATOS	59
5.1	MELHORIA NO PROCESSO DE SOFTWARE: IMPLANTAÇÃO DO MPS.BR NÍVEL G EM UMA EMPRESA DE PEQUENO PORTE	59
5.2	ELABORAÇÃO E MODELAGEM DE UM PROCESSO DE DESENVOLVIMENTO DE SOFTWARE ADERENTE AO MPS- BR NÍVEL G	59
5.3	PROPOSTA DE UM MODELO DE REFERÊNCIA PARA DESENVOLVIMENTO DE SOFTWARE COM FOCO NA CERTIFICAÇÃO DO MPS.BR	60
5.4	PROCESSO DE GARANTIA DA QUALIDADE BASEADO NO MODELO MPS.BR	60
6	DESENVOLVIMENTO DO PROCESSO	61
6.1	METODOLOGIA	61
6.2	DESENVOLVIMENTO DE UMA METODOLOGIA PARA O MODELO DE PROCESSO DE SOFTWARE	65
6.2.1	Modelagem de Processo de Software	65
6.2.2	Definição dos Fluxos de Atividades do Processo	70
6.2.2.1	Iniciação	71
6.2.2.2	Planejamento	73
6.2.2.3	Execução	76
6.2.2.4	Acompanhamento e Controle	79
6.2.2.5	Encerramento	80

CONCLUSÃO	82
REFERÊNCIAS	85
APÊNDICE A – PUBLICAÇÃO DO MODELO DE PROCESSO UNIFICADO	88
APÊNDICE B - PUBLICAÇÃO DO CICLO DE VIDA DO PROJETO.....	89
APÊNDICE C – ARTIGO CIENTÍFICO	90

1 INTRODUÇÃO

Segundo o Guia de Conhecimentos em Gerenciamento de Projetos (PMBOK) um dos principais fatores para a obtenção de sucesso no desenvolvimento de projetos de software está relacionado diretamente ao nível do seu planejamento. Um dado estatístico muito interessante publicado nesse manual afirma que mais de 80% dos projetos de desenvolvimento de software atrasam. Esses atrasos são causados pelo mau planejamento ou pela inexistência de artefatos para o gerenciamento e controle dos projetos na linha de produção de software (PMBOK, 2008).

Espinha (2007) observa que um dos fatores importantes para a construção de um software de qualidade é a metodologia de desenvolvimento utilizada e a forma como este é implantado na organização.

A inexistência de métodos bem definidos e de boas práticas de desenvolvimento, mesmo que simples, fazem com que o desenvolvimento de software seja realizado de forma *ad-hoc*, ou seja, sem a formalização de qualquer tipo de planejamento ou controle das atividades referentes ao desenvolvimento do produto de software. Nesses modelos, as equipes tendem a permanecer altamente dependentes da experiência e do conhecimento das pessoas envolvidas no negócio.

Em virtude da necessidade de se ter qualidade nos produtos de software, a aderência por modelos de maturidade, normas de qualidade e guias de boas práticas na definição de processos tem se tornado cada vez mais frequente.

Uma das maiores dificuldades das organizações que iniciam num programa de melhoria de processos é a maneira de adaptar este conjunto de boas práticas à sua realidade, identificando quais áreas devem ser abordadas de acordo com a relevância e prioridade.

Cada organização possui políticas, crenças e uma cultura específica que deve ser levada em conta a fim de que as melhorias sejam bem aceitas e realmente contribuam para um desenvolvimento mais eficiente (ESPINHA, 2007).

Na busca da qualidade e do sucesso no desenvolvimento de software, é de fundamental importância a utilização de métodos, técnicas, ferramentas e de um especialista capacitado na gerência de projetos. Essa relevância tem sido adotada por organizações na área de software.

As causas do insucesso nos projetos de software podem estar relacionadas com a má Gerência de Projetos e com a completa falta de gerenciamento. Como forma de minimizar esses problemas, propõe-se o desenvolvimento de uma metodologia para o gerenciamento e controle dos projetos de software na linha de produção.

Esta proposta tem como objetivo de pesquisa realizar um estudo detalhado da área de gerenciamento de software usando características do conhecimento especializado e organizado de PMBOK nas áreas de negócio, tais como: práticas de gerenciamento, técnicas e ferramentas. E utilizando, como referência, o Programa de Melhoria de Processo de Software Brasileiro (MPS.BR), o qual é um modelo de maturidade nacional e agrega padrões de qualidade internacionais na qual as empresas se certificam em diferentes níveis de acordo com seu grau de maturidade.

Dentro desse contexto, percebeu-se a oportunidade de realizar um trabalho que identificasse as práticas mais utilizadas por implementadores MPS a fim de demonstrar as exigências do modelo de maturidade brasileiro, visando a auxiliar, a orientar e a minimizar as dificuldades enfrentadas na gerência de projetos de software. Dessa forma, é possível gerar um melhor entendimento por parte das empresas e dos implementadores sobre como demonstrar os resultados exigidos para a certificação do nível G do modelo MPS.

1.1 OBJETIVO GERAL

Desenvolver uma metodologia para o gerenciamento de projetos baseada na área de processo Gerência de Projetos (GPR) do modelo de maturidade MPS.BR.

1.2 OBJETIVO ESPECÍFICOS

Os objetivos específicos do projeto são baseados nos seguintes itens:

- a) realizar um estudo dos conhecimentos da área de Gerenciamento de Projetos;
- b) desenvolver o aprendizado da Área de Processo GPR do nível G, do modelo de maturidade MPS.BR;
- c) elaborar um processo formal para a disciplina Gerência de Projetos;
- d) desenvolver e publicar os artefatos relacionados à área de processo GPR.

1.3 JUSTIFICATIVA

Nos últimos anos, observou-se que muitas organizações produzem software desprovidas de metodologia nos seus processos de desenvolvimento.

A maior parte dos problemas se concentra nas dificuldades encontradas durante o desenvolvimento de um software. Isso ocorre em razão da ausência de definição de um processo padrão de trabalho a ser seguido por todos os envolvidos no projeto de desenvolvimento.

Conforme descreve Vargas (2005) projeto é um conjunto de ações realizadas de maneira organizada, caracterizadas por sequências claras e lógicas de eventos com o início,

meio e fim, vinculadas ao propósito de atingir um ou mais objetivos definidos e planejados por especialistas da área.

A Engenharia de Software contém fases rotuladas, tais como foco de qualidade, processos, métodos e ferramentas, etc. Especificamente a camada de processos é o alvo principal na vinculação de definições e padronização da qualidade para uma organização (PRESSMAN, 2002).

Pode-se afirmar que o projeto de software é uma sequência de etapas necessárias para o desenvolvimento e manutenção de se um produto seguindo um ciclo de vida definido e gerenciado.

A imaturidade do processo ou a ausência dele, juntamente com a execução de diferentes rotinas de trabalho por membros da equipe, causa muitos problemas no decorrer das etapas de produção dos softwares. Entre os mais comuns, pode-se citar: atrasos nos cronogramas dos projetos, custos maiores que o estimado, um produto final de baixa qualidade com a ocorrência de muitas falhas, gerando conseqüentemente a insatisfação e a falta de confiança do usuário em relação ao software que lhe foi entregue.

A qualidade de software visa garantir a confiabilidade das propriedades do produto final e, para isso, utiliza uma combinação de modelos, ferramentas e também um especialista (PRESSMAN, 2002).

Para adquirir qualidade de processos, produtos e serviços, as empresas devem adaptar-se a padrões de desenvolvimento de software, por meio de processos definidos e modelados pode-se alcançar um nível de padronização e amadurecimento gradativo dos processos e conseqüentemente da qualidade dos seus produtos.

A crise do software, que tem persistido nas últimas décadas apesar do impressionante progresso tecnológico, ocorre principalmente em razão da falta de métodos e técnicas bem fundamentados, contrastando com outras áreas da engenharia (PRESSMAN, 2006).

Por meio dos anos de evolução da engenharia de software, obteve-se como lição aprendida que o aumento gradativo da qualidade e maturidade das equipes de desenvolvimento é obtido somente com a padronização e o controle efetivo dos processos de engenharia nas empresas.

Vários padrões já são mundialmente conhecidos e adotados em parte ou integralmente, como, por exemplo, *International Organization for Standardization (ISO)*, *Capability Maturity Model Integration (CMMI)*, PMBOK, entre outros, os quais estabelecem um consenso para alcançar soluções que vão de encontro das necessidades dos negócios e da sociedade. Outro modelo que tem obtido grande sucesso entre as empresas Brasileiras, é o MPS.BR, o qual introduz conceitos que unem práticas das normas ISO e CMMI, tornando-se um modelo ideal a ser largamente implementado nas mais diversas linhas de desenvolvimento.

O MPS.BR, é um programa coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), é baseado nos conceitos de maturidade e capacidade de processo.

Garantia de qualidade é o processo por meio do qual é definida a forma como a qualidade de software pode ser atingida e de que modo a organização de desenvolvimento sabe se o software possui o nível de qualidade necessário (SOMMERVILLE, 2007).

Assim como todas as áreas do MPS.BR, a Gerência de Projetos possui uma lista de resultados esperados que devem ser atendidos, pois são eles que guiam uma avaliação e apontam o nível em que a empresa se encontra.

Com base nos problemas apresentados no decorrer desta proposta e na ampla quantidade de técnicas, padrões e ferramentas presentes atualmente na área da engenharia de software, propõe-se como trabalho final de graduação, o desenvolvimento de uma metodologia aderente à área de Processo de Gerência de Projetos (GPR) do modelo MPS.BR a ser publicada como modelo aplicável em equipes de diferentes tamanhos e linhas de negócios.

1.4 ESTRUTURA DO TRABALHO

Quanto à estrutura formal, este trabalho divide-se em seis capítulos.

No primeiro deles, é contextualizada a introdução referente ao trabalho, são expostas as definições dos objetivos geral e específicos, sendo concluído com a apresentação da justificativa do trabalho realizado.

O segundo capítulo trata da Engenharia de Software e discorre sobre os principais padrões de qualidade do processo de software.

No capítulo três, procura-se apresentar um estudo detalhado dos modelos de padrões de qualidade, enfatizando a área de Processo de Gerência de Projetos (GPR) referentes ao nível G do modelo de qualidade MPS.BR.

O quarto capítulo apresenta o entendimento da área de Gerência de Projetos e suas principais características, objetivando apresentar algumas definições para o tema proposto, utilizando com base o guia PMBOK no gerenciamento de desenvolvimento de software.

A apresentação dos trabalhos correlatos está descrita no quinto capítulo e, por fim, o capítulo seis exhibe a metodologia e a definição do processo proposto.

2 ENGENHARIA DE SOFTWARE

O objetivo deste capítulo é apresentar uma base de entendimento acerca da Engenharia de Software bem como sua importância nos dias atuais. Além disso, será apresentada uma breve introdução dos conceitos básicos que consolidam o conhecimento dentro do tema abordado neste trabalho.

2.1 A CRISE DE SOFTWARE

A Engenharia de Software é uma ciência relativamente nova, definida como tentativa de solucionar os vários problemas apresentados no desenvolvimento de sistemas computacionais, tais como: estimativas de custos, prazos vencidos, insatisfação do cliente, dificuldade de manutenção, ausência de técnicas bem estabelecidas e a necessidade de softwares mais complexos.

Em meados de 1968, ocorreu, na Alemanha, a Conferência da Organização do Tratado do Atlântico Norte (OTAN) sobre Engenharia de Software, organizada para se discutir a respeito da chamada *Crise de Software*.

Infelizmente a comunidade de software ainda não estava preparada para essas mudanças e, por isso, muitos projetos de software fracassaram durante esse período.

Sommerville (2007) afirma que os softwares produzidos naquela época não eram confiáveis, o prazo e o custo eram maiores que os estimados e, muitas vezes, eram entregues com atraso ou apresentavam um péssimo desempenho.

Na tentativa de contornar as causas dos problemas encontrados, o objetivo da OTAN, era encontrar algo que a ajudasse a definir, padronizar e amadurecer os processos de desenvolvimento de software. Por essa razão, o encontro foi considerado um marco do

surgimento do termo *Engenharia de Software*, sendo esta uma disciplina criada exclusivamente para estudar e aprimorar o desenvolvimento de software (SOMMERVILLE, 2007).

À medida que a disciplina de Engenharia de Software evoluiu, suas técnicas e modelos de processos foram sendo amplamente utilizados e adaptados até os dias atuais. No entanto, o desenvolvimento de software tem se tornado mais complexo ao longo dos anos, as exigências por parte dos clientes são cada vez maiores em termos de produtividade e qualidade, e os prazos têm diminuído. Fatos esses, que fomentam a procura por gerenciamento, modelagem e melhoria contínua de processos.

Um processo de software pode ser visto como um conjunto de atividades, métodos, ferramentas e práticas utilizadas para construir um produto de software (PFLEEGER, 2004).

Baseada nessa definição, a composição de um processo de software deve considerar as seguintes informações:

- a) atividades a serem realizadas;
- b) recursos necessários;
- c) artefatos requeridos e produzidos;
- d) procedimentos adotados;
- e) modelo de ciclo de vida a ser utilizado.

Resulta, do processo um produto que reflete a forma como os projetos de desenvolvimento serão conduzidos. Embora existam vários processos para o desenvolvimento de softwares, as atividades de especificação, projeto, validação e evolução são comuns a todos eles. Diante disso, Sommerville (2007) relata que são quatro as atividades a serem executadas no desenvolvimento:

- a) **especificação de software:** trata-se da definição das funcionalidades (requisitos) e das restrições do software. Geralmente é a fase em que o desenvolvedor conversa com o cliente para definir as características do sistema;
- b) **projeto e implementação de software:** o software é produzido de acordo com as especificações. Nesta fase, são propostos modelos por meio de diagramas, sendo eles implementados em alguma linguagem de programação;
- c) **validação de software:** nesta etapa, o sistema é validado a fim de garantir que todas as funcionalidades especificadas tenham sido implementadas;
- d) **evolução de software:** o software precisa evoluir para continuar sendo útil ao cliente.

Segundo Sommerville (2007) As fases são executadas em todos os modelos, no entanto, as organizações podem executá-las de forma particular forma particular e de acordo com sua organização.

2.2 MODELOS DE CICLOS DE VIDA DE SOFTWARE

Os modelos de ciclos de vida de desenvolvimento de sistemas computacionais são propostos para descrever a forma com que as fases interagem no processo de desenvolvimento. Eles podem ser utilizados para explicar as diferentes abordagens adotadas durante o processo de desenvolvimento ou manutenção de sistemas computacionais, definindo, dessa forma, o ciclo de vida de sua concepção.

O ciclo de vida pode ser considerado como um conjunto de fases do projeto, pois descreve a vida do produto de software desde a concepção, passando pela implementação, entrega, utilização e manutenção.

A escolha acertada do modelo de ciclo de vida é uma das mais importantes decisões para se obter o sucesso do projeto, pois colabora para que o processo de desenvolvimento do software seja eficiente, alcançado um produto com mais qualidade, minimizando os riscos inerentes ao processo de desenvolvimento e o ônus financeiro associado a sua concepção.

A seguir, serão abordados alguns dos modelos de ciclo de vida existentes, sendo eles os processos tradicionais de desenvolvimento, criados pela Engenharia de Software a partir da década de 70.

2.2.1 Modelo Cascata

O modelo Cascata foi o primeiro paradigma de desenvolvimento criado pela Engenharia de Software. É uma abordagem linear e sequencial de todas as atividades envolvidas no desenvolvimento de software (PRESSMAN, 2006) e, por se tratar do encadeamento de uma fase a outra, esse modelo também ficou conhecido como “ciclo de vida” do software (SOMMERVILLE, 2007).

Os estágios do modelo em Cascata *Waterflow* são executadas em sequência, como uma cascata, sendo que o desenvolvimento de uma fase não deve começar antes da conclusão da fase anterior, conforme ilustrado na Figura 1:

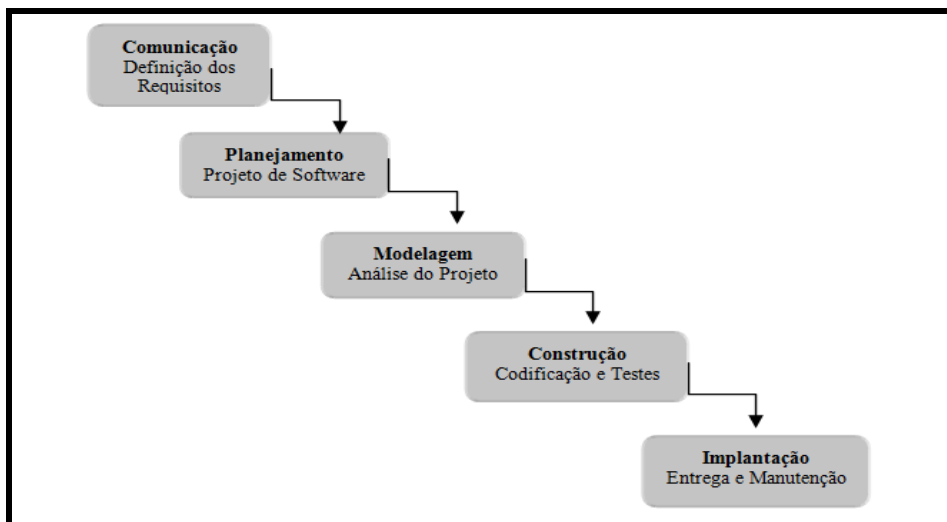


Figura 1. Modelo Cascata

Fonte: Adaptado de PRESSMAN, R. (2006);
SOMMERLIVE, I (2007)

Segundo Sommerville (2007) os principais estágios do modelo retratam as atividades fundamentais de desenvolvimento:

- a) **análise e definição de requisitos:** as funções, as restrições e os objetivos do sistema são estabelecidos por meio da consulta aos usuários do sistema. Em seguida, tais dados são detalhadamente definidos, servindo como uma especificação do sistema;
- b) **projeto de sistemas e de software:** o processo de projeto de sistemas agrupa os requisitos em sistemas de hardware ou de software. Ele estabelece uma arquitetura do sistema geral. O projeto de software envolve a identificação e a descrição das abstrações fundamentais do sistema de software e de suas relações;
- c) **implementação e teste de unidades:** durante este estágio, o projeto de software é compreendido como um conjunto de programas ou de unidades do programa. O teste de unidades envolve a análise de cada unidade a fim de se verificar se elas atendem a suas especificações;
- d) **integração e teste de sistemas:** as unidades de programa, ou programas individuais, são integradas e testadas como um sistema completo com o intuito de

garantir que os requisitos de software tenham sido atendidos. Depois da realização dos testes, o sistema de software é entregue ao cliente;

e) **operação e manutenção:** normalmente, embora não necessariamente, será a fase mais longa do ciclo de vida, pois se refere à usabilidade do sistema durante a operação dele após a instalação. A manutenção envolve a correção de erros não descobertos em estágios anteriores do ciclo de vida, melhorando a implementação das unidades do sistema e aumentando as funções dele à medida que novos requisitos são descobertos.

Desse modo, o modelo apresenta uma visão de alto nível do que realmente acontece durante o desenvolvimento, além de sugerir, aos desenvolvedores, uma sequência de atividades previstas. (PFLEEGER, 2004).

Os resultados de cada atividade fornecem um *feedback* aos desenvolvedores responsáveis pelas atividades anteriores, por meio do qual são identificadas e descritas as atividades, estimulando a melhoria e a evolução do software (PETERS, ; PEDRYCZ, 2001).

Portanto, o modelo em Cascata é utilizado em processos de software quando se tem os requisitos fixos, quando não há muitas modificações.

2.2.2 Modelo Protótipo

O modelo em Protótipo pode ser usado como um modelo de processo independente. Ele é utilizado como técnica, juntamente com qualquer outro modelo, o desenvolvedor interage diretamente com o usuário, escutando seus pedidos e desenvolvendo. Imediatamente, é criado um protótipo do produto desejado. O usuário, então, utiliza esse protótipo e fornece ao desenvolvedor novas informações que o levam às atualizações do

sistema, adaptações e implementações no software, em tempo de projeto e desenvolvimento (PRESSMAN, 2006).

Conforme ilustrado na Figura 2, o modelo começa com a definição de requisitos, feita pelo desenvolvedor e pelo usuário. São definidos os objetivos gerais e criado um projeto inicial, o qual leva à construção de um protótipo a ser avaliado pelo usuário a fim de que os requisitos do software a ser desenvolvido sejam refinados. Logo, o processo de refinamento dos requisitos traz benefícios para ambas as partes envolvidas no processo de desenvolvimento do software.

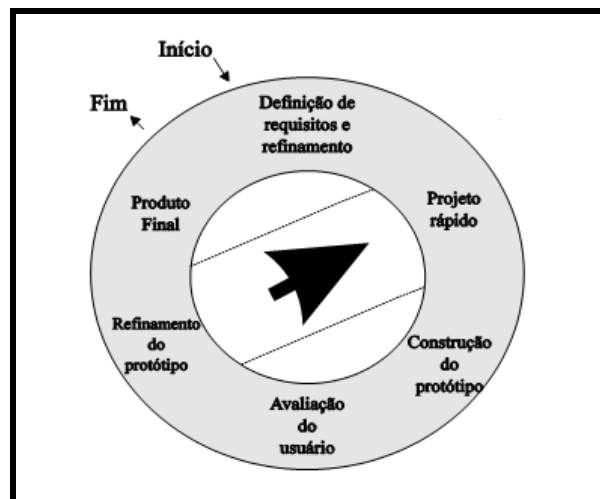


Figura 2. Modelo Protótipo
Fonte: Adaptado de PRESSMAN, R (2006).

O usuário, ao compartilhar no processo de desenvolvimento, tem condições de contribuir para um maior refinamento dos requisitos, aumentando a confiança no software em desenvolvimento.

Outro benefício a ser apresentado se refere à compreensão do desenvolvedor quanto ao software em que está trabalhando, pois, a partir dos pontos abordados pelo usuário bem como suas sugestões, poderá ser desenvolvido um produto de melhor qualidade e mais próximo ao que foi requisitado.

A rapidez na construção de um protótipo pode ser obtida com a reutilização de softwares e ferramentas, contribuindo para o aumento da velocidade do desenvolvimento do produto de software. Assim sendo, a reutilização utiliza códigos já testados anteriormente em projetos validados (PRESSMAN, 2006).

2.2.3 Modelo Espiral

Durante muitos anos, o modelo Cascata foi a base da maior parte do desenvolvimento de projetos de software, mas em 1988, Boehm sugeriu o modelo Espiral, desenvolvido para abranger as melhores características tanto do ciclo de vida Cascata quanto da Prototipação, acrescentando, ao mesmo tempo, um novo elemento, a análise de riscos, com o intuito de minimizá-los e controlá-los (PRESSMAN, 2006).

O modelo usa ciclos da espiral para repetir todas as fases que forem necessárias para o desenvolvimento do software, sendo assim, cada volta na espiral representa uma fase no processo.

O modelo define cinco fases conforme representado na Figura 3:

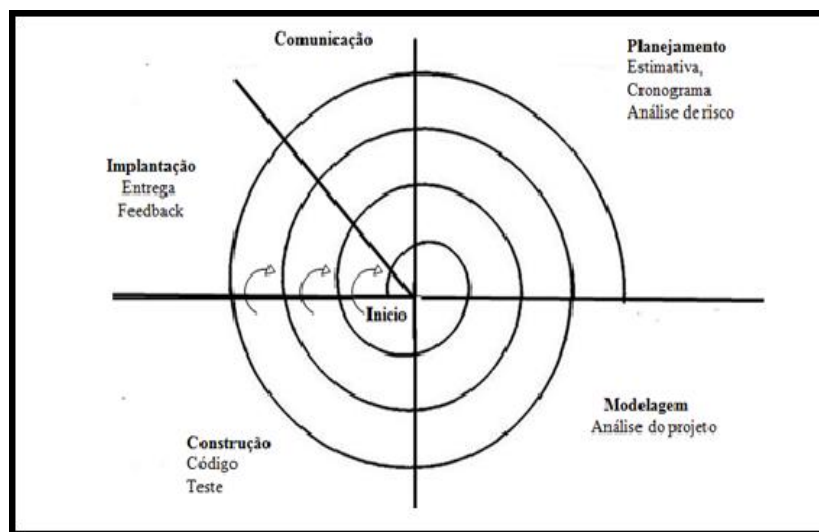


Figura 3. Modelo Espiral
Fonte: Adaptado de PRESSMAN, R. (2006).

Em cada ciclo da espiral, algumas atividades são realizadas em ordem sequencial, sendo elas:

- a) **comunicação**: entra-se em contato com o cliente a fim de se adquirir informações úteis para a definição dos objetivos, das restrições e alternativas para a fase inicial do projeto;
- b) **planejamento**: avalia-se as alternativas com relação aos objetivos e restrições e identifica-se as principais fontes de riscos;
- c) **modelagem**: consiste na análise do projeto, na qual são identificados os possíveis riscos no desenvolvimento do software mesmo antes de ele ser iniciado. O software passa por fases interativas durante todo o seu desenvolvimento;
- d) **construção**: referente à implementação, aos testes, à instalação e ao suporte da aplicação;
- e) **implantação**: momento em que o software é entregue ao cliente e é dado o *feedback* quanto ao incremento, ao desenvolvimento e a possíveis problemas.

O modelo Espiral permite que, ao longo de cada interação, sejam obtidas versões do sistema cada vez mais completas, recorrendo-se à prototipagem para reduzirem-se os riscos.

Durante o primeiro giro em torno da espiral, os objetivos, as alternativas e as restrições são definidas; e os riscos são identificados e analisados.

Se a análise do risco identificar que há problemas ou incertezas nos requisitos, passa-se para o planejamento a fim de reduzirem-se os riscos antes mesmo de se tornarem problemáticos. O sistema é avaliado e são apresentadas sugestões para modificações. Baseadas nessas informações, ocorrem as seguintes fases: modelagem, seguida da construção e, por fim, a implantação (PRESSMAN, 2006).

2.2.4 Modelo Iterativo e Incremental

A idéia principal deste modelo de desenvolvimento é a de que um sistema deve ser desenvolvido de forma interativa, assim, cada incremento vai adicionando, ao sistema, novas habilidades funcionais até a obtenção do sistema final, sendo que, a cada passo realizado, novas modificações podem ser introduzidas.

O modelo de ciclo de vida Incremental é a abordagem intermediária dos modelos Evolucionário e Cascata, conforme ilustrado na Figura 4. Porém, o trabalho é agrupado em pequenos ciclos sequenciais e cada ciclo segue o modelo cascata (SOMMERVILLE, 2007).

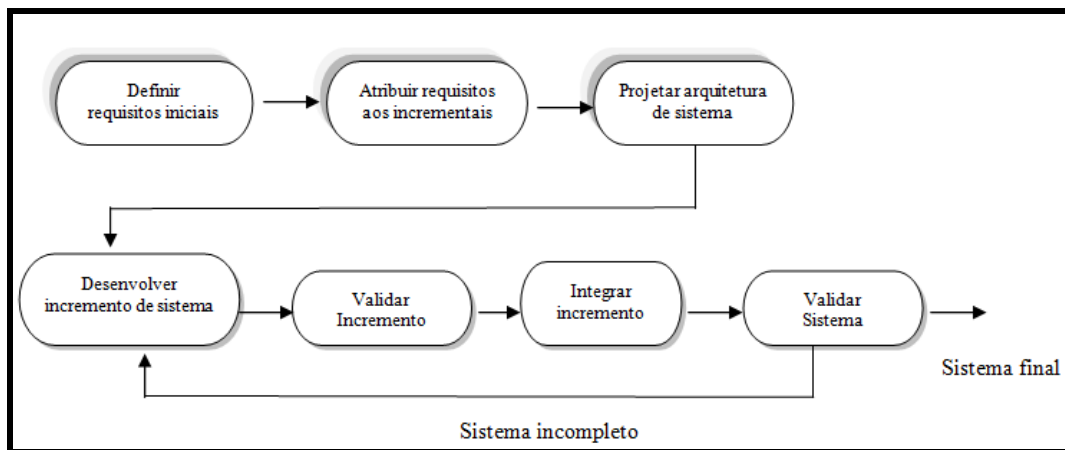


Figura 4. Modelo Iterativo e Incremental

Fonte: Adaptado de SOMMERLLIVE, I. (2007).

Este modelo é eficiente quando o conjunto de requisitos do projeto apresenta alto risco de sofrer mudanças. Assim, se os requisitos tiverem de ser modificados, o orçamento será menos comprometido do que se tivesse sido utilizado um modelo sequencial (SOMMERVILLE, 2007).

Uma das vantagens desta abordagem é a facilidade em testar o sistema, uma vez que a realização de testes, em cada nível de desenvolvimento, é, sem dúvida, mais fácil do que testar o sistema integralmente ao final.

3 MELHORIA DO PROCESSO DE SOFTWARE

Inicialmente, este capítulo introduzirá os conceitos de melhoria de processo de software. Em seguida, explanará sobre a qualidade de software, processos e produtos bem como acerca dos principais modelos de processos de software.

Nos últimos anos, muitas organizações de software buscam pela melhoria da qualidade de seus processos no desenvolvimento e manutenção do software. São comprovadamente gratificantes os benefícios conferidos pela aderência a modelos de maturidade, normas de qualidade e guias de boas práticas, no intuito de aprimorar as técnicas e habilidades no desenvolvimento de softwares bem direcionados..

Começar, talvez, seja a mais importante e difícil fase de qualquer trabalho. No caso da melhoria do processo, o levantamento de como o software está sendo desenvolvido na organização é a primeira atividade a ser executada. O sucesso de um programa de melhoria do processo depende da confiabilidade dos dados adquiridos na fase de avaliação, visto que uma avaliação adequada permite a elaboração de um plano de melhoria mais eficiente e compatível com a realidade da organização (CAROSIA, 2003).

O objetivo de um programa de melhoria de processo não é simplesmente impor regras pré-estabelecidas de como fazer as coisas, mas levar em consideração as características e as necessidades da organização (CAROSIA, 2003). Na próxima seção serão apresentados alguns conceitos de qualidade do produto e de processo.

3.1 QUALIDADE DE SOFTWARE

Obter qualidade nos processos e produtos de engenharia de software não é uma tarefa fácil, são vários os fatores que dificultam o atingimento dos objetivos de qualidade (ROCHA; MALDONADO; WEBER, 2001).

A qualidade do software está diretamente relacionada à forma como são implementados os requisitos para ele elicitados.

O gerenciamento de requisitos de software se refere à forma como o sistema será construído bem como a sua organização, ditando-se funcionalidades e prioridades de implementação. A organização destes requisitos em tarefas depende basicamente da disciplina Gerenciamento de Projetos, a qual apóia a execução das atividades de desenvolvimento do software, inferindo qualidade no seu processo e consequentemente no seu produto.

A apresentação dos conceitos de qualidade de produto e de processo, e suas respectivas definições serão abordadas a seguir.

3.1.1 Qualidade do Produto x Qualidade do Processo

Raramente obtém-se qualidade logo após a conclusão do desenvolvimento do processo. Assim, a qualidade do produto de software é um dos objetivos do processo de desenvolvimento. Portanto, ao se desenvolver um produto, é importante que já estejam definidas as finalidades e os objetivos de qualidade que se deseja atingir (CAROSIA, 2004).

Se o processo de desenvolvimento de software considerar as características de qualidade, há uma grande probabilidade de que o produto final apresente tais características.

A qualidade do produto pode ser observada como um conjunto de objetivos a serem alcançados num determinado grau a fim de que o produto sane as necessidades de seus

usuários. Cada objetivo pode ser detalhado a um amplo conjunto de atributos que apresentam a qualidade do produto de software. Portanto, é necessária a escolha de um modelo que estabeleça os atributos e permita avaliar-se a qualidade do software (ROCHA; MALDONADO; WEBER, 2001).

A qualidade de software está diretamente relacionada à qualidade do processo que o produz. Dessa forma, a definição de um processo de software apropriado à aplicação a ser desenvolvida e também alinhado à cultura dos profissionais envolvidos no projeto de software é um fator crítico de sucesso para qualquer organização de software.

O processo de software é um conjunto de ferramentas, métodos e práticas usadas para produzir software e é representado por um conjunto sequencial de atividades, objetivos, transformações e eventos que integram estratégias para cumprimento da evolução de software (PRESSMAN, 2006).

Sommerville (2007) afirma que a melhoria do processo se baseia na suposição de que o principal fator que influencia a qualidade do produto é o processo de desenvolvimento dele. O processo é aperfeiçoado com o objetivo de reduzir-se o número de defeitos do produto até que ele possa ser repetido. Dessa maneira, os resultados do processo são previsíveis e o número de defeitos é reduzido. Obviamente isso resultará um produto de melhor qualidade e a garantia da qualidade de software fornece métodos que garantem que os produtos de trabalho e processos estejam sendo executados em conformidade com os padrões, procedimentos e métodos estabelecidos.

A importância da qualidade no setor de software é indiscutível. Assim, muitas organizações se preocuparam em criar modelos para auxiliar na busca pela tão desejada qualidade, tanto do produto quanto do processo de software.

3.2 MODELOS DE QUALIDADE DE PROCESSOS DE SOFTWARE

Para que as ações de melhoria possam ser desencadeadas, as organizações que desejam melhorar seus processos precisam utilizar-se de normas e modelos capazes de expor possíveis padronizações a fim de obter o aperfeiçoamento que conduz à melhoria contínua da qualidade no processo de desenvolvimento de software.

Vários modelos e normas de qualidade de software vêm ganhando destaque, seja por sua qualidade ou aplicação.

3.2.1 CMMI

O CMMI é um modelo de qualidade desenvolvido pelo *Software Engineering Institute* (SEI). (SEI, 2006), baseia-se num conjunto de capacidades da Engenharia de Software, as quais devem estar presentes à medida que a organização alcança diferentes níveis de maturidade de processo. Para obtê-las, a o SEI alega que as empresas devem desenvolver um modelo que siga às normas do CMMI (PRESSMAN, 2006).

Desde 1991, o SEI tem desenvolvido modelos de qualidade à engenharia de sistemas, à engenharia de software, à aquisição de software e desenvolvimento integrado de produto e processo (SAMARANI, 2005).

O CMMI é constituído pelas melhores práticas que abrangem o desenvolvimento e a manutenção de produtos/serviços, englobando o ciclo de vida do produto desde a sua iniciação até a entrega e manutenção (SAMARANI, 2005).

Segundo Pressman (2006) os modelos de desenvolvimento contidos na constelação CMMI contêm práticas que possibilitam uma abordagem de melhoria de processo e de avaliação utilizando duas diferentes representações, conforme apresentado no Quadro 1.

Níveis	Representação contínua	Representação por estágio
	Níveis de capacidade	Níveis de maturidade
0	Incompleto	
1	Executado	Inicial
2	Gerenciado	Gerenciado
3	Definido	Definido
4	Quantitativamente gerenciado	Quantitativamente gerenciado
5	Em otimização	Em otimização

Quadro 1. Comparação entre os níveis de Capacidade e de Maturidade
 Fonte: Adaptado FERRARI, S. (2007)

a) **contínua**: baseia-se na seleção de processos de uma ou mais áreas de processo a serem executados pela organização a fim de alinhar as atividades de melhoria de processos aos objetivos de negócio da organização;

b) **estagiada**: utiliza conjuntos pré-definidos de áreas de processo para definir um roteiro de melhoria para a organização. Este roteiro é caracterizado por níveis de maturidade.

Quadro 1 compara os níveis de capacidade aos níveis de maturidade. Cabe ressaltar que os nomes de quatro dos níveis são os mesmos em ambas as representações, apresentando como diferença a inexistência de nível 0 (zero) de maturidade para a representação por estágio e também o fato de, no nível um (1), o nível de capacidade ser “Executado” enquanto o nível de maturidade é “Inicial”. Portanto, o ponto de início difere-se para as duas representações (FERRARI, 2007).

Cada nível de maturidade é constituído por um conjunto de áreas de processo que caracterizam diferentes comportamentos organizacionais (PRESSMAN, 2006).

O CMMI propõe cinco níveis de maturidade. Quanto maior o nível de maturidade de uma empresa, maior a produtividade e a qualidade e conseqüentemente menor o risco.

Fernandes e Correa (2008) destacam que cada nível de maturidade possui características bem distintas, conforme demonstrada na Figura 5.

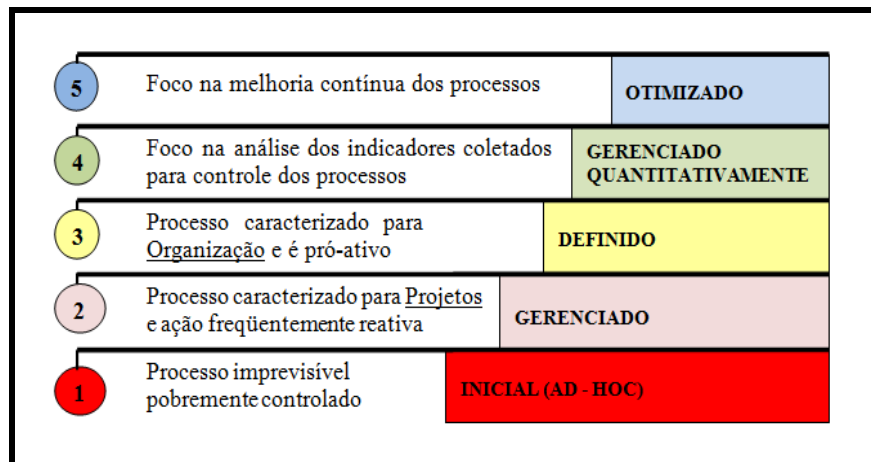


Figura 5. Níveis de Maturidade do CMMI
Fonte: FERNANDES, M.; CORREA, S. (2008).

Ainda de acordo com os mesmos autores:

- a) Nível 1 - inicial: processo de software caracterizado como “ad hoc”. Poucos processos de desenvolvimento definidos e a obtenção de sucesso neles depende de esforço individual;
- b) Nível 2 - repetível: as políticas de gerência de desenvolvimento de software são definidas e seguidas. É o nível mais difícil de ser alcançado por se tratar de uma quebra de paradigma;
- c) Nível 3 - definido: o processo básico de software para as atividades de gestão e engenharia é documentado, padronizado e integrado num processo de software padrão para a organização;
- d) Nível 4 - gerenciado: medidas detalhadas do processo de software e da qualidade do produto são realizadas. O processo e os produtos de software e da qualidade do produto são quantitativamente compreendidos e controlados;
- e) Nível 5 - otimização: a melhoria contínua do processo é proporcionada pelo *feedback* quantitativo do processo e pelas idéias e tecnologias inovadoras.

O CMMI é um modelo internacional muito valorizado por empresas do mundo inteiro, sendo, para algumas delas, pré-requisito na contratação de serviços.

Entretanto, o custo para a certificação ainda é muito alto, fator que motivou a criação do MPS.BR.

3.2.2 MPS.BR

A qualidade de software tem sido muito aprimorada nos últimos anos e isso explica a razão pela qual muitas empresas adotaram novas técnicas e tecnologias, modificando sua estrutura organizacional bem como a de seus processos, conferindo maior importância ao gerenciamento de qualidade de software.

A alta concorrência e as exigências dos clientes, a globalização dos mercados e o desenvolvimento de novas tecnologias são também motivadores dessa mudança (SOFTEX, 2009).

Para nível nacional, é de extrema importância que empreendedores dêem ênfase à eficiência e à eficácia dos seus processos, visando à oferta de produtos conforme os padrões internacionais de qualidade.

O objetivo do MPS.BR é estar adequado ao perfil das mais variadas organizações, mas, em especial, das micro, pequenas e médias empresas. Também se espera que ele seja compatível aos padrões de qualidade aceitos internacionalmente e utilize todo o conhecimento existente nos modelos de melhoria de processo já disponibilizados (SOFTEX, 2009).

O MPS.BR é um programa coordenado pela SOFTEX e uma de suas metas é definir e aprimorar um modelo de melhoria e de avaliação de processo de software a ser aplicado nas empresas desenvolvedoras de software brasileiro para garantir qualidade nos seus produtos. Dentro das definições desse modelo, é composto pelas normas *International*

Organization for Standardization e o *International Electrotechnical Commission* ISO/IEC 12207, e ISO/IEC 15504-2, o modelo também foi definido em conformidade ao CMMI.

A SOFTEX, com o apoio do governo, de universidades, de centros de pesquisa e de organizações privadas, confirmou a realidade das empresas brasileiras. Esta estrutura é apresentada na ilustração Figura 6.

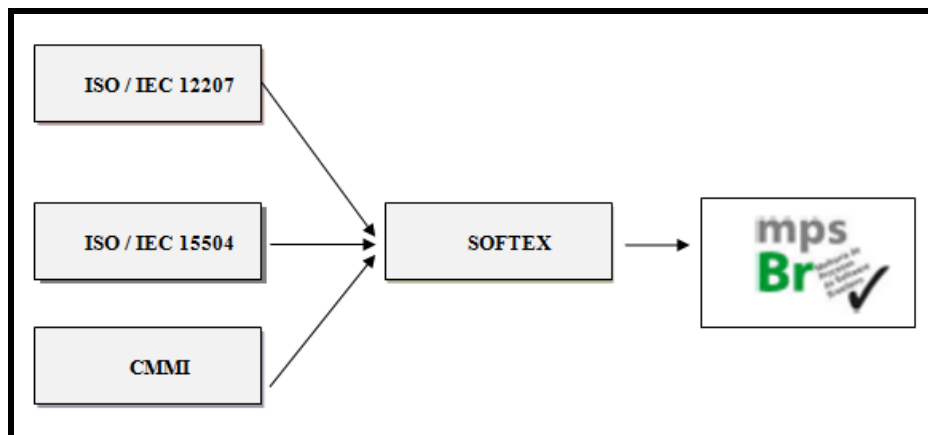


Figura 6. Construção do modelo MPS.BR
 Fonte: (KOSCIANSKI; SOARES, 2006), A. (2006).

A norma internacional ISO/IEC 12207 foi criada pela ISO tendo como principal objetivo estabelecer definições dos processos de ciclo de vida do desenvolvimento de software.

A norma ISO/IEC 12207 tem como objetivo ajudar as organizações na concepção dos artefatos presentes na aquisição, no fornecimento, no desenvolvimento, na operação e na manutenção de software (SOFTEX, 2009).

A norma ISO/IEC 15504 foi criada após o estudo das necessidades e exigências de uma avaliação de processos de software, no qual foi concluído que a nova norma deveria ser aplicável à melhoria de processos e à determinação da capacidade. A norma em questão tem por objetivo prover uma infra estrutura para a avaliação de processos a fim de que seja obtida a melhoria de processos e de determinar a capacidade de processos de uma unidade organizacional (SOFTEX, 2009).

Estruturalmente o MPS.BR é dividido em três componentes: O Modelo de Referência (MR-MPS), o Método de Avaliação (MA-MPS) e o Modelo de Negócio (MN-MPS) (SOFTEX, 2009).

Conforme na ilustração da Figura 7, todos os componentes do modelo foram descrito por meio de documentos específicos, como tais respectivos guias.

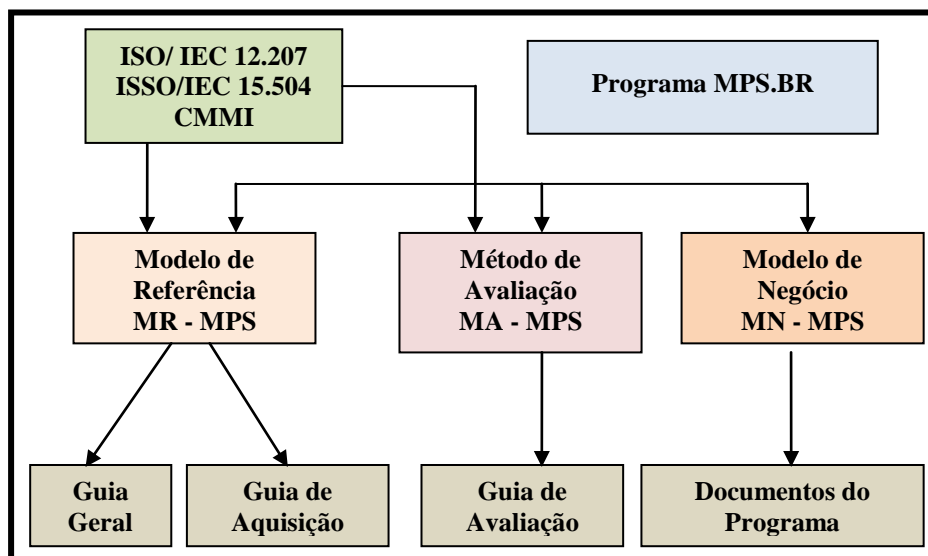


Figura 7. Estrutura do MPS.BR
Fonte: (KOSCIANSKI; SOARES, 2006).

O Modelo de Referência (MR-MPS) define os processos de software que contêm os requisitos a serem realizados pelas empresas que pretendem estar em conformidade com ele (KOSCIANSKI; SOARES, 2006).

O Método de Avaliação (MA-MPS) está descrito no Guia de Avaliação, no qual se encontra o processo de avaliação, os requisitos para os avaliadores e os requisitos para se verificar a conformidade que auxilia na confirmação de que o MPS.BR está sendo utilizado de acordo com suas definições (KOSCIANSKI; SOARES, 2006).

Já o Modelo de Negócio (MN-MPS) descreve as regras para que empresas brasileiras desenvolvedoras de software implementem o MR-MPS (KOSCIANSKI; SOARES, 2006).

Os níveis de maturidade representam um grau de melhoria na implementação de processos em uma organização. Observando o nível de maturidade em que uma empresa se encontra, pode-se prever o seu desempenho ao executar um processo (SOFTEX, , 2009, p 16).

Segundo SOFTEX (2009) o MPS.BR – Guia Geral (2009) e o MR-MPS definem sete níveis de maturidade, sendo eles:

- a) A - Em otimização;
- b) B - Gerenciado Quantitativamente;
- c) C - Definido;
- d) D - Largamente Definido;
- e) E - Parcialmente Definido;
- f) F - Gerenciado;
- g) G - Parcialmente Gerenciado.

Os níveis de maturidade do modelo MPS.BR serão vistos a seguir, identificando-se quais propósitos e atributos serão estabelecidos.

3.2.2.1 Níveis de Maturidade de Processos

Os níveis de maturidade são uma combinação entre os processos e sua capacidade, conforme estrutura apresentada na Figura 8. Para cada processo, é declarado um propósito bem como os resultados esperados com a sua execução. Assim, é possível avaliar se uma empresa atingiu os objetivos de cada processo.

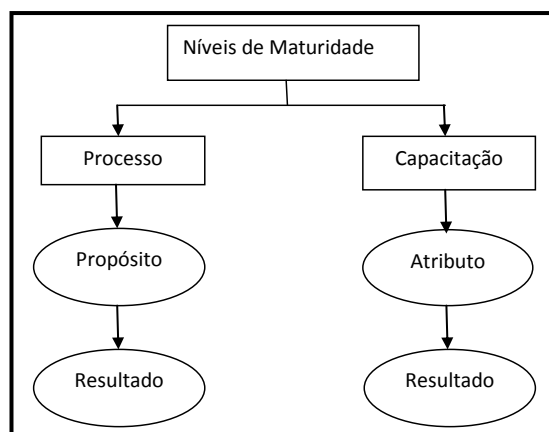


Figura 8. Estrutura de Níveis de Maturidade
 Fonte: KOSCIANSKI; SOARES (2006, p.145).

A melhoria e o alcance de um determinado nível de maturidade do MR-MPS obtêm-se, quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e os resultados esperados dos atributos de processo estabelecidos para aquele nível.

Conforme uma organização evolui nos níveis de maturidade, os processos devem ser desempenhados com um maior nível de capacidade. Para cada Atributo de Processo (AP) são descritos os Resultados Esperados do Processo (RAP), que devem ser atendidos por todos os processos no nível correspondente ao nível de maturidade.

Começa no nível G e termina no nível A, sendo que a organização evolui nesta escala quando os propósitos e resultados esperados para os processos e atributos de processo daquele nível são atingidos (SOFTEX, 2009).

Os níveis de maturidade, os processos e atributos de processo correspondentes a cada nível podem ser visualizados no Quadro 2.

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP, 4.2
C	Gerência de Riscos – GRI	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Desenvolvimento para Reutilização – DRU	
	Gerência de Decisões – GDE	
D	Verificação – VER	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Validação – VAL	
	Projeto e Construção do Produto – PCP	
	Integração do Produto – ITP	
	Desenvolvimento de Requisitos – DRE	
E	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de Reutilização – GRU	
	Gerência de Recursos Humanos – GRH	
	Definição do Processo Organizacional – DFP	
	Avaliação e Melhoria do Processo Organizacional - AMP	
F	Medição – MED	AP 1.1, AP 2.1 e AP 2.2
	Garantia da Qualidade – GQA	
	Gerência de Portfólio de Projetos – GPP	
	Gerência de Configuração – GCO	
	Aquisição – AQU	
G	Gerência de Requisitos (GRE):	AP 1.1 e AP 2.1
	Gerência de Projetos (GPR):	

Quadro 2. Níveis de Maturidade de Processos
Fonte: SOFTEX (2009)

Pode-se observar, na tabela acima, que o modelo MPS.BR é dividido em níveis de maturidade. Cada nível é composto por um conjunto de áreas de processo que são implementadas de acordo com alguns resultados estabelecidos para cada atributo de processo definido pelo modelo.

3.2.2.2 Nível G - Parcialmente Gerenciado

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 e, 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 (SOFTEX, 2009).

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, ou seja, poder redefinir algumas operações (atividades de rotina) em andamento, estabelecendo objetivos, prazos e escopo para sua execução (SOFTEX, 2009).

Segundo o SOFTEX (2009) o nível G é composto pelos processos de Gerência de Projetos e Gerência de Requisitos, satisfazendo os atributos de processos AP 1.1 e AP 1.2, conforme observado no Quadro 2.

A SOFTEX (2009) afirma que, neste nível, devem ser implementados os atributos de processo de acordo com a descrição de propósitos e resultados esperados para cada nível de maturidade.

a) **Gerência de Projetos (GPR):** tem o propósito de identificar, estabelecer, coordenar e monitorar as atividades e recursos que o projeto precisa para apresentar o resultado esperado, atendendo aos seus requisitos e restrições.

Apresenta como resultados esperados:

GPR 1: o escopo do trabalho para o projeto é definido.

GPR 2: o escopo, os produtos de trabalho e as tarefas do projeto são estimados, por meio de métodos apropriados.

GPR 3: as fases do ciclo de vida do projeto são definidas.

GPR 4: o esforço e o custo para os produtos de trabalho e às tarefas são estimados baseados em dados históricos ou referências técnicas.

GPR 5: o cronograma e o orçamento do projeto são estabelecidos e mantidos.

GPR 6: os riscos do projeto são identificados e os impactos, probabilidades de ocorrência e prioridades de tratamento são determinadas e documentadas.

GPR 7: os recursos humanos para o projeto são planejados considerando-se o perfil e o conhecimento necessários para executá-lo.

GPR 8: as tarefas, a infraestrutura e os recursos necessários para completar o trabalho são planejados.

GPR 9: os dados relevantes do projeto são identificados, coletados, armazenados e distribuídos. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança.

GPR 10: o planejamento do projeto é monitorado no que se refere ao cronograma, aos custos, recursos, riscos, dados e ao envolvimento dos interessados.

GPR 11: a avaliação da viabilidade de se atingir as metas do projeto, considerando as restrições e os recursos disponíveis, realizando-se ajustes se necessário.

GPR 12: o planejamento do projeto é revisado com todos os interessados e o compromisso formal deles é obtido.

GPR 13: o projeto é gerenciado utilizando-se o Plano do Projeto e outros planos que afetam o projeto e os resultados são documentados.

GPR 14: o envolvimento dos interessados no projeto é planejado.

GPR 15: são estabelecidos registros e análises dos problemas identificados nas monitorações.

GPR 16. ações corretivas são estabelecidas quando necessário e gerenciadas até a conclusão.

GPR 17: revisões são realizadas em marcos do projeto, conforme estabelecido no planejamento.

b) Gerência de Requisitos (GRE): tem como propósito gerenciar os requisitos do projeto bem como investigar possíveis inconsistências entre os requisitos e os planos e produtos de trabalho. Apresenta como resultados esperados:

GRE 1: os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando-se critérios objetivos.

GRE 2: o comprometimento da equipe técnica em relação aos requisitos aprovados é obtido.

GRE 3: a rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida.

GRE 4: revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos.

GRE 5: mudanças nos requisitos são gerenciadas ao longo do projeto.

3.2.2.3 Capacidade do Processo

Segundo SOFTEX (2009), a capacidade do processo é um conjunto de atributos de processo descrito em termos de resultados esperados, os quais proporcionam o atendimento dos atributos de processo. A capacidade estabelece o grau de refinamento e de institucionalização com que o processo é executado na organização. À medida que a empresa evolui nos níveis de maturidade, um maior ganho de capacidade para desempenhar o processo é atingido pela organização.

Conforme SOFTEX (2009) a Capacidade do Processo possui cinco AP:

a) AP 1.1: o processo é executado;

- b) AP 2.1: o processo é gerenciado;
- c) AP 2.2: os produtos de trabalho do processo são gerenciados;
- d) AP 3.1: o processo é definido;
- AP 3.2: o processo está implementado.

Os diferentes níveis de capacidade dos processos são descritos por AP. O alcance de cada atributo de processo é avaliado utilizando-se os respectivos RAP do nível G, conforme definido a seguir:

a) O Processo é Executado (AP 1.1): este atributo é a medida do quanto o processo atinge o seu propósito. Apresenta como resultado esperado:

RAP 1: o processo atinge seus resultados definidos;

b) O processo é Gerenciado (AP 2.1): este atributo é a medida do quanto a execução do processo é gerenciada. Apresenta como resultado esperado:

RAP 2: existe uma política organizacional estabelecida e mantida para o processo.

RAP 3: a execução do processo é planejada.

RAP 4: a execução do processo é monitorada e ajustes são realizados (Para o nível G).

RAP 5: as informações e os recursos necessários para a execução do processo são identificados e disponibilizados (até o nível F).

RAP 6: as responsabilidades e a autoridade para executar o processo são definidas, atribuídas e comunicadas (até o nível F).

RAP 7: as pessoas que executam o processo são competentes em termos de formação, treinamento e experiência (até o nível F).

RAP 8: a comunicação entre as partes interessadas no processo é gerenciada de forma a garantir o seu envolvimento.

RAP 9: os resultados do processo são revistos com a gerência de alto nível para fornecer visibilidade sobre a sua situação na organização (até o nível F).

RAP 10: o processo planejado para o projeto é executado (para o nível G).

Para se compreender os conceitos adotados pelo modelo MR-MPS, observou-se a necessidade do entendimento da área de Gerência de Projetos, sobre a qual será explanado no próximo capítulo.

4 GERÊNCIA DE PROJETOS

Este capítulo trata de assuntos relacionados à disciplina de Gerenciamento de Projetos com o objetivo de apresentar algumas definições, características e a importância de sua utilização no desenvolvimento de produtos de software.

Também serão apresentados os conceitos de projetos, gerenciamento de projetos bem como o papel do gerente de projetos.

4.1 GERENCIAMENTO DE PROJETOS SEGUNDO O PMBOK

Um dos principais difusores do gerenciamento de projetos e da profissionalização do gerente de projetos é o *Project Management Institute* (PMI), fundado nos Estados Unidos da América (EUA) e atualmente estendido a mais de cem países. Na difusão do conhecimento em gerenciamento de projetos, duas das principais iniciativas do PMI são:

- a) a certificação profissional em Gerência de Projetos *Project Management Professional* (PMP);
- b) e a publicação de um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos - PMBOK.

O objetivo do PMBOK é de auxiliar na execução de um projeto, servindo como um modelo de referência que descreve o gerenciamento das atividades essenciais no andamento do projeto, os processos envolvidos e as áreas de conhecimento do gerenciamento de projeto (PMBOK, 2008).

Segundo PMBOK (2008) o gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas das atividades do projeto, a fim de atender aos requisitos do projeto.

Gerenciar um projeto significa, resumidamente, planejar a sua execução antes de iniciá-lo e posteriormente acompanhar a sua execução e o seu controle. Algumas práticas gerenciais, quando bem aplicadas, contribuem para a melhoria da qualidade da gerência de um projeto.

Com a utilização do gerenciamento de projetos, alguns benefícios podem ser alcançados, dentre eles: evitar a ocorrência de surpresas durante a execução dos trabalhos; antecipar situações desfavoráveis desde que ações preventivas e corretivas sejam tomadas antes que estas situações se tornem um problema para o projeto; disponibilizar os orçamentos antes do início dos trabalhos; e gerar documentação no intuito de facilitar estimativas para futuros projetos.

Procedimentos e padrões devem ser definidos pelos gerentes de projeto e utilizados por todos os envolvidos no processo de desenvolvimento para poder atingir-se um alto nível de qualidade (SOMMERVILLE, 2007).

4.1.1 Definição de Projeto

Projeto pode ser definido como um esforço temporário empregado para criar um produto, serviço ou resultado exclusivo. Os projetos caracterizam-se por serem: temporais, tendo início e fim bem definidos; exclusivos, pois produzem resultados únicos; e também progressivos, porque podem ser realizados em passos de forma incremental (PMBOK, 2008).

Para Pfleeger (2004) projeto é o processo criativo de transformar o problema em uma solução. Segundo a definição de Sommerville (2007) descreve que projeto é um processo criativo, pois estabelece requisitos funcionais e não funcionais que satisfazem o sistema.

Um projeto é caracterizado por eventos com início e fim, conduzido por pessoas e destinados a atingir um objetivo claro, Tais pessoas são chamadas de *stakeholders*.

O principal profissional do projeto, responsável por acompanhar e controlar todos os objetivos do projeto, é o Gerente de Projetos, o qual identificará as necessidades do cliente, estabelecendo objetivos claros e alcançáveis, equilibrando as demandas conflitantes de escopo, custo e prazo.

Para gerenciar a área de gerenciamentos de projetos, é necessário possuir grande conhecimento em gerenciamento de equipes bem como deter competências, habilidades e atitudes para assumir tal função. Na próxima seção, será apresentado, juntamente com as definições de gerenciamento de projetos, a função do Gerente de Projetos e também o quanto este papel é importante para a realização do controle e da qualidade de um projeto.

4.1.2 O Gerente de Projetos

Segundo PMBOK (2008) o gerente de projetos é um profissional designado pela organização na área de gerenciamento de projetos, que tem a responsabilidade de atingir os seguintes objetivos dos projetos: planejar, controlar, executar e gerenciar.

O gerente de projetos possui várias atividades e responsabilidades, tais como: definir e controlar os objetivos do projeto, os requisitos do produto e os riscos do projeto; definir e avaliar os fatores críticos de sucesso do projeto e os pontos fortes e fracos do projeto; definir e controlar o cronograma; alocar e gerenciar recursos; definir prioridades; coordenar interações entre os envolvidos no projeto; assegurar que prazos e custos estão sendo mantidos conforme o planejado e que os produtos do projeto atendam aos critérios de qualidade, estando de acordo com os padrões estabelecidos; e participar de reuniões de acompanhamento e de revisão do projeto (PMBOK, 2008).

A responsabilidade do gerente é criar uma metodologia de gerenciamento de projetos que abranja ferramentas e técnicas de gerenciamentos de projetos, reconhecidas

como boas práticas e necessárias para compreender e aplicar o conhecimento. No entanto, isso não é suficiente para um gerenciamento eficaz, pois, além das habilidades relacionadas aos processos, o gerente de projetos deverá ser capaz de se comunicar de forma verbal e escrita, uma vez que interagem diretamente com pessoas. São responsáveis pelos planos, criação e alocação de equipes, cronogramas e relatórios do projeto, apresentados aos clientes e à organização.

O gerenciamento de projetos é um trabalho que envolve vários desafios, pois desempenha um papel de grande responsabilidade durante todo ciclo de vida do projeto. Envolve a supervisão e assecuração de que tudo está sendo realizado dentro dos padrões exigidos, além de verificar se o desenvolvimento está dentro do prazo e dentro do orçamento planejado (SOMMERVILLE, 2006).

4.1.3 Processos de Gerenciamento de Projetos de um Projeto

“Um processo é um conjunto de ações e atividades inter-relacionadas realizadas para obter um conjunto pré-especificado de produtos, resultados ou serviços” (PMBOK, 2008 p. 34).

O PMBOK (2008) considera que um projeto, ao longo de seu ciclo de vida, é composto por processos, um conjunto de ações realizadas por pessoas e que se destina à obtenção de resultados.

Segundo Viera (2003) cada projeto é composto por vários processos executados por pessoas e podem ser divididos em Gerência de Projeto e processos orientados ao produto.

Os processos podem ser também executados pelas equipes de projetos, podendo ser classificados em duas categorias de processos num projeto, conforme descreve PMBOK (2008):

a) **processos de gerência de projetos:** tem por objetivo a garantia do uso de ferramentas e das técnicas ao longo da existência do projeto;

b) **processos orientados ao produto:** especificam e elaboram o produto do projeto que são aqueles relacionados aos requisitos, às especificações e à geração do projeto, que são definidos pelo ciclo de vida projeto.

O gerenciamento de projetos é realizado por meio da aplicação e da integração entre os processos, das interações deles e dos objetivos a que atendem. Estes processos são agregados em cinco grupos definidos como grupos de processos de gerenciamento de projetos (PMBOK, 2008), conforme ilustrado na Figura 9:

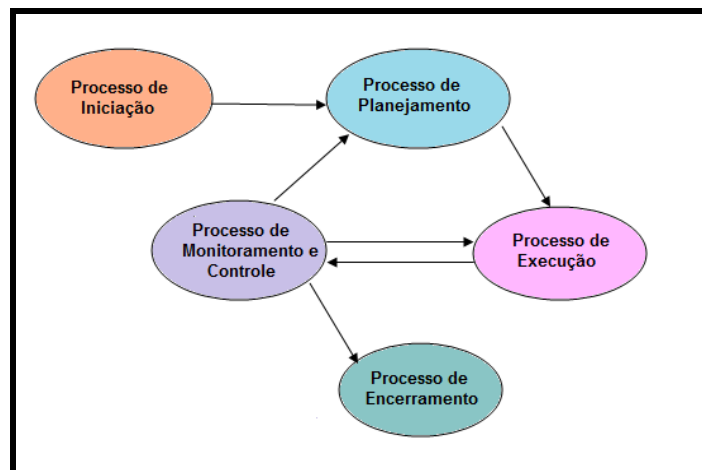


Figura 9. Processos do gerenciamento de projetos do PMBOK
Fonte: PMBOK(2008).

Segundo PMBOK (2008) os processos de gerenciamento de projetos têm suas iterações entre os processos e os objetivos que são:

a) **grupo de processos de iniciação:** define e autoriza o projeto ou uma fase do projeto;

b) **grupo de processos de planejamento:** define e refina os objetivos e planeja a ação necessária para alcançar os objetivos bem como o escopo para o qual o projeto foi planejado;

c) **grupo de processos de execução:** é a execução da definição do plano de gerenciamento do projeto a fim de atender as especificações já desempenhadas anteriormente;

d) **grupo de processos de monitoramento e controle:** é necessário que o desenvolvimento seja regularmente acompanhado para identificar desvios do plano de gerenciamento do projeto. Podendo, dessa forma, serem tomadas ações corretivas quando necessário a fim de se atender aos objetivos do projeto.

e) **grupo de processos de encerramento:** é formalizada a finalização das atividades de todos os grupos de processo.

Segundo PMBOK (2008) a integração entre os processos de gerenciamento de projetos por meio de um ciclo do tipo PDCA (do inglês: *plan-do-check-act*. Traduzido para o português: planejar, fazer, verificar e agir). Este ciclo é ligado por resultados, tornando-se uma parte do ciclo a entrada para outra fase, conforme ilustra a Figura 10.

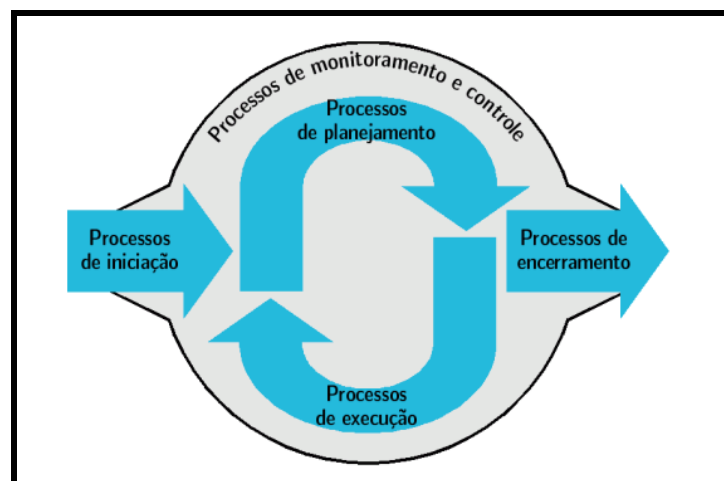


Figura 10. Integração entre os processos de gerenciamento de projetos através de um ciclo do tipo PDCA
Fonte: PMBOK (2008)

O grupo de processos de planejamento corresponde ao componente “planejar” do ciclo PDCA. O grupo de processos de execução corresponde ao “fazer” e o grupo de processos de monitoramento e controle corresponde aos componentes “verificar” e “agir”. Além disso, como o gerenciamento de um projeto é um esforço finito, o grupo de processos

de iniciação começa os ciclos e o grupo de processos de encerramento os finaliza (PMBOK, 2008).

A natureza integradora do gerenciamento de projetos exige a interação do grupo de processos de monitoramento e controle com todos os aspectos dos outros grupos de processos (PMBOK, 2008).

Conforme detalhado por Vargas (2007) e Vieira (2003) o emprego de boas práticas de gerenciamento de projetos proporciona vários benefícios, tais como:

- a) evita surpresas durante a execução dos trabalhos;
- b) antecipa as situações desfavoráveis que poderão ser encontradas, realizando, assim, antes que aconteça o problema, ações preventivas e corretivas;
- c) adapta os trabalhos ao mercado e ao cliente;
- d) agiliza as decisões;
- e) aumenta o controle gerencial de todas as fases;
- f) documenta e facilita as estimativas para futuros projetos;
- g) melhora a coordenação da equipe;
- h) melhora o relacionamento com o cliente;
- i) diminui o tempo de implementação.

Conforme Vargas (2007) ainda ressalta, não basta apenas realizar o gerenciamento do projeto, é preciso ser eficaz, pois o contrário disso pode levar o projeto ao fracasso. Os principais aspectos que indicam o insucesso de um projeto são:

- a) pouca compreensão da complexidade do projeto;
- b) o projeto inclui muitas atividades e muito pouco tempo para realizá-las;
- c) o projeto não teve um gerente de projetos;
- d) não se conheciam as necessidades de pessoal, equipamentos e materiais;
- e) as pessoas não estavam envolvidas nos mesmos padrões estabelecidos.

As empresas que adotam as práticas de gerenciamento de projetos se beneficiam e se tornam cada vez mais competitivas, se destacam no mercado e principalmente, demonstram para os seus clientes que estão organizadas de acordo com as práticas e metodologias reconhecidas internacionalmente para realizar projetos com qualidade, cumprindo o que foi prometido, dentro do que foi orçado e de acordo com o tempo previsto. (VIEIRA, 2003, p. 18).

4.1.4 As áreas de Conhecimento de Gerenciamento de Projetos

Conforme dito anteriormente, em cada processo são executadas ações gerenciais que, segundo o PMBOK (2008), podem abranger até nove áreas de conhecimento, conforme ilustrado na Figura 11.

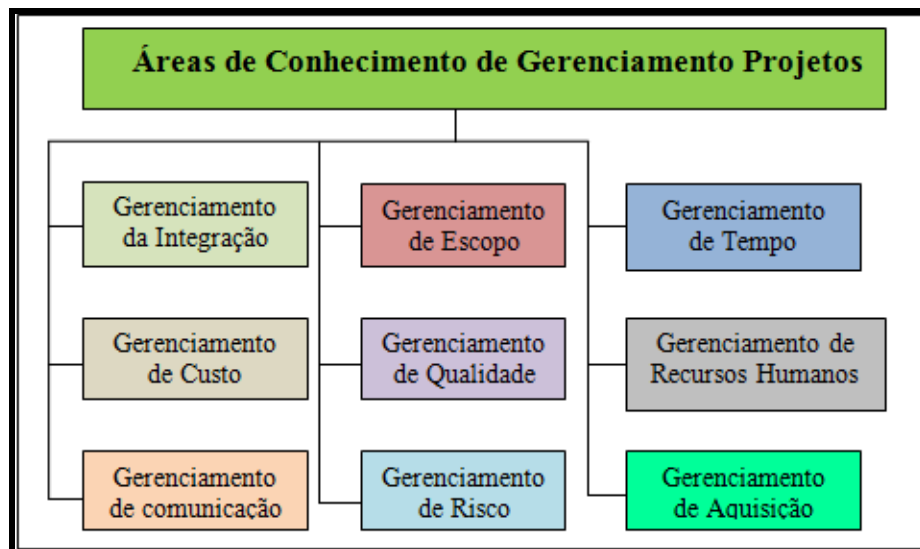


Figura 11. As áreas de conhecimento de gerenciamento do PMBOK
Fonte: PMBOK (2008).

- a) **gerenciamento de integração:** são incluídas as atividades necessárias para os requisitos requeridos a fim de ser assegurado que todos os elementos do projeto sejam adequadamente coordenados e identificados, garantindo o sucesso do projeto;
- b) **gerenciamento de escopo:** assegurar que, no projeto, esteja incluído todo o trabalho requerido e somente o trabalho necessário para concluí-lo de maneira bem sucedida;

- c) **gerenciamento de tempo:** determina o prazo previsto para a conclusão do projeto;
- d) **gerenciamento de custos:** elabora uma estimativa e controla os custos, além de assegurar que o projeto seja concluído de acordo com seu orçamento previsto;
- e) **gerenciamento de qualidade:** garante que os produtos ou serviços do projeto estarão em conformidade com o solicitado pelo cliente;
- f) **gerenciamento de recursos humanos:** consiste na organização da equipe, ou seja, de todos os envolvidos que possuem responsabilidades designadas para a conclusão do projeto.
- g) **gerenciamento de comunicações:** garantir que as informações do projeto sejam geradas e distribuídas com rapidez e eficiência para os envolvidos no projeto;
- h) **gerenciamento de riscos:** visa planejar, identificar e monitorar os riscos de modo que diminuam os impactos negativos no projeto;
- i) **gerenciamento de aquisições:** é a forma de adquirir bens e serviços ou resultados externos à equipe envolvida.

A não execução de processos de uma área afetará negativamente o projeto, pois ele é um esforço integrado. Pode ser usada, como exemplo, uma mudança de escopo, a qual quase sempre afeta o custo do projeto, entretanto, ela poderá, ou não, afetar a qualidade do produto.

O processo de Gerência do Projeto fornece uma base a todas as outras ações da organização em relação ao projeto em andamento à medida que envolve atividades fundamentais na organização e no controle do que irá ocorrer durante todo o desenvolvimento do projeto.

A abordagem do PMBOK, como guia de boas práticas para o gerenciamento de projetos, fornece uma base clara e concisa de como definir um processo para esta disciplina. Assim como os modelos de qualidade de software CMMI e ISO, o MPS.BR também utiliza, em seu modelo de referência, os conceitos abordados pelo PMBOK, sendo que o estudo e o entendimento deste destacam-se como uma importante atividade para as equipes responsáveis pela definição dos processos nas organizações desenvolvedoras de soluções baseadas em Software.

5 TRABALHOS CORRELATOS

Durante os estudos desta pesquisa, foram analisados alguns trabalhos científicos cujos propósitos eram semelhantes a ela, porém com focos em outros objetivos específicos. Serão descritos, a seguir, alguns trabalhos selecionados que envolvem processos de software.

5.1 MELHORIA NO PROCESSO DE SOFTWARE: IMPLANTAÇÃO DO MPS.BR NÍVEL G EM UMA EMPRESA DE PEQUENO PORTE

Trabalho de Conclusão de Curso Alessandro Liebman para obtenção do grau de Bacharel em Ciência da Computação, em 2006, pelo Departamento de Ciência da Computação da Universidade Federal de Lavras, do Estado de Minas Gerais.

O trabalho tem como objetivo descrever e discutir a implantação do modelo de melhoria MPS.BR no processo de software numa empresa de pequeno porte (LIEBMAN, 2006).

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

Trabalho de Conclusão de Curso de Ricardo Emerson Julio para obtenção do grau de Bacharel em Ciência da Computação, em 2007, pelo Departamento de Ciência da Computação da Universidade Federal de Lavras, do Estado de Minas Gerais.

O trabalho tem por intuito a elaboração e a modelagem de um processo de desenvolvimento de software para uma fábrica de software aderente ao nível G do MPS.BR. É conferido destaque à importância dos modelos de melhoria de processos de software

MPS.BR, sendo demonstrado como implementar tais melhorias em um processo (JULIO, 2007).

5.3 PROPOSTA DE UM MODELO DE REFERÊNCIA PARA DESENVOLVIMENTO DE SOFTWARE COM FOCO NA CERTIFICAÇÃO DO MPS.BR

Tese de Alfredo Colenci Neto para obtenção do grau de Doutor em Engenharia de Produção, em 2008, pela Escola de Engenharia de Software de São Carlos, da Universidade de São Paulo, do Estado de São Paulo.

A tese tem por objetivo propor um modelo de referência aplicável às pequenas empresas produtoras de Software fim de dar o suporte ao seu processo de desenvolvimento de produtos (PDP). Sendo realizada uma avaliação contextualizada, como base método de avaliação MARES (COLENCI, 2008).

5.4 PROCESSO DE GARANTIA DA QUALIDADE BASEADO NO MODELO MPS.BR

Trabalho de Conclusão de Curso de Anildo Loos para obtenção do grau de Bacharel em Ciência da Computação, em 2009, pelo Curso de Ciência da Computação da Universidade Regional de Blumenau, do Estado de Santa Catarina.

O objetivo deste trabalho foi desenvolver uma ferramenta de apoio ao processo de garantia da qualidade baseado nos resultados previstos na versão 1.2 do modelo MPS.BR. A ferramenta atende à maioria dos resultados esperados do processo de garantia da qualidade do modelo MPS.BR (LOOS, 2009).

6 DESENVOLVIMENTO DO PROCESSO

A modelagem de um processo de software deve possibilitar o entendimento e a padronização do processo, para tanto, deve-se observar os principais objetivos para modelar o processo.

6.1 METODOLOGIA

A metodologia utilizada para o desenvolvimento deste trabalho iniciou-se pela etapa de levantamento bibliográfico, na qual foram estudados os conceitos relacionados à *Engenharia de Software* e às melhores práticas de gerenciamento de processos na área de Gerência de Projetos.

Com a pesquisa realizada em livros, dissertações, testes e sites na Web, procurou-se compreender os modelos existentes bem como alguns temas referentes a processos de desenvolvimento de software, gerenciamento de projetos, modelos de qualidade e melhoria de processos de software, enfatizando o MPS.BR .

A pesquisa sobre o MPS.BR foi baseada nos diversos guias e documentos do modelo, podendo serem encontrados no site da SOFTEX, disponíveis para downloads.

O processo elaborado para o desenvolvimento do software foi desenvolvido com base no *Rational Unified Process* (RUP), pois ele aborda o conceito de orientação a objetos, utilizando notações da *Unified Modeling Language* (UML) para a definição de fluxo de processos.

Os fluxos de trabalho do processo foram feitos utilizando-se do *Eclipse Process Framework Composer* (EPF), uma ferramenta gratuita, criada pela *International Business Machines* (IBM), que proporciona uma estrutura para a gestão de processo de software

(SERAFIM, 2008). O EPF contém uma estrutura de bibliotecas e *plugins* suportando processos com base em diferentes abordagens de desenvolvimento em vários modelos de ciclo de vida, incluindo: cascata, iterativos incrementais e ágeis.

O EPF foi projetado para ser um sistema de gerenciamento de conteúdo, fornecendo uma estrutura de gerenciamento comum a fim de solucionar problemas encontrados pelas equipes de desenvolvimento de projetos durante sua concepção (ECLIPSE, 2009).

O esquema do EPF Composer para a modelagem de processos é uma evolução do *Software Process Engineering Metamodel* (SPEM), referenciando o *Unified Method Architecture* (UMA), que é o meta modelo proposto pela *Object Management Group* (OMG). O SPEM propõe a unificação entre as diferentes metodologias propostas para a modelagem de processos, padronizando termos e significados (SERAFIM, 2008).

A grande particularidade do EPF é a categorização dos elementos padronizados do SPEM, pois ela define uma estrutura para representar o método e o processo. A SPEM separa, do Processo, o Conteúdo de Métodos, conforme ilustrado na Figura 12.

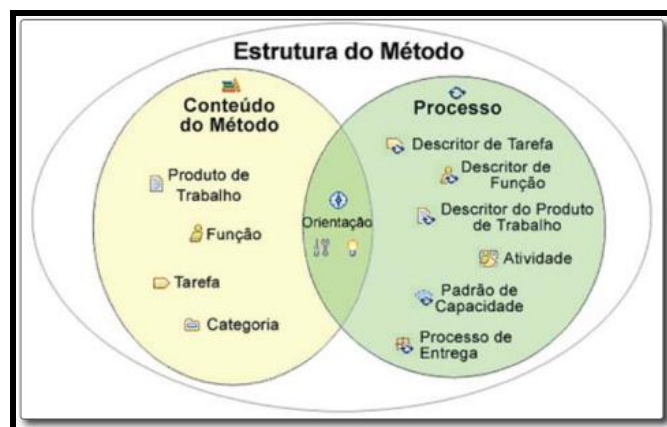


Figura 12. Conceitos principais do EPF que são baseados no UMA
Fonte: SERAFIM (2008)

O Método de Conteúdo descreve o que deve ser produzido como Produtos de Trabalho, as competências necessárias exigidas pela Função e as instruções passo a passo que explicam como os objetivos serão atingidos.

Os processos descrevem o desenvolvimento do ciclo de vida, relacionando os elementos dos Métodos de Conteúdo em sequências personalizadas para determinados tipos de projetos. As descrições dos Métodos de Conteúdo são independentes do desenvolvimento do ciclo de vida.

Como mostra a Figura 12, o Método de Conteúdo é expresso por meio de Produtos de Trabalho (artefatos), Função, Tarefa e Orientação (guia). Os guias estão posicionados na interseção entre Métodos de Conteúdo e Processo, pois podem conter informações para as duas categorias definidas no EPF (ECLIPSE, 2009).

O principal conceito que representa a categoria Processo é a Atividades, pois o alinhamento delas define os fluxos de trabalho. As Atividades contêm referências para os Métodos de Conteúdo por meio dos conceitos do Descritor (Descrição das Tarefas, Função, Produtos de Trabalho e das Atividades). As Atividades são usadas para definir processos, dentre os quais, o EPF sugere os seguintes: Padrão de Capacidade e Processo de Entrega (ECLIPSE, 2009).

O Processo de Entrega representa um processo completo e integrado e pode ser usado como referência para projetos semelhantes. O Padrão de Capacidade pode ser considerado como representações de subprocessos que expressam interesses específicos, tais como o processo de uma área ou disciplina específica que podem ser reusado dentro de processos diferentes (ECLIPSE, 2009).

Podem ser visualizados na Figura 13 os principais elementos do conteúdo de método.

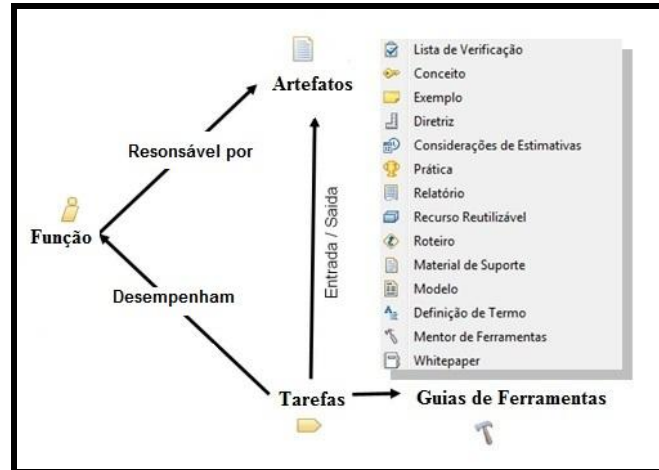


Figura 13. Os elementos do Conteúdo de Método

Os elementos básicos utilizados no processo desenvolvido são:

- a) **Artefatos**: o que é produzido;
- b) **Tarefa**: como realizar o trabalho;
- c) **Função**: quem executa o trabalho;
- d) **Guias de Ferramentas**: modelos, *checklists*, exemplos, relatórios, conceitos,

etc.

Para a definição do processo, optou-se por utilizar todos os conceitos sugeridos pelo EPF, pois o objetivo é proporcionar uma documentação organizada e clara. Dessa forma, ao usá-lo, pode-se criar o seu próprio processo de desenvolvimento de software e estruturá-lo em sentido específico. O processo define os métodos que podem ser reutilizados durante a definição de outros processos.

O *framework* possui uma funcionalidade de publicação em que o processo pode ser apresentado ou publicado em um ambiente digital no formato de uma web site, disponibilizando, assim, a todos os *stakeholders* da organização, o processo de desenvolvimento de software definido para a consulta: conforme é apresentado na ilustração do Apêndice A.

6.2 DESENVOLVIMENTO DE UMA METODOLOGIA PARA O MODELO DE PROCESSO DE SOFTWARE

Nesta seção, são apresentados os resultados da modelagem do processo de desenvolvimento de software, baseado nos resultados estabelecidos no nível G do modelo MPS.BR, com o objetivo de fornecer uma descrição das fases do processo: principais atividades, artefatos e papéis.

6.2.1 Modelagem de Processo de Software

Com base nos estudos realizados, percebe-se a importância de um processo de desenvolvimento de software de qualidade. Assim sendo, o primeiro passo para a criação do processo é a definição do modelo de ciclo de vida de projetos, por meio do qual são descritas as fases do processo.

Para a elaboração da metodologia, foi criada uma disciplina na área de processo do nível G do modelo MPS.BR: Gerência de Projetos, atuante como uma categoria padrão oferecida pela EPF e pode ser utilizada para categorizar as atividades e tarefas do processo, conforme ilustrado na Figura 14.

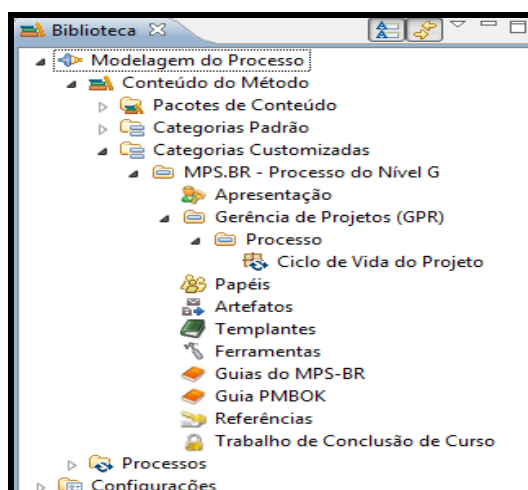


Figura 14. Área de processo do MPS.BR configurada como disciplina no EPF

De acordo com tais atividades, foi definido um ciclo de vida do processo contendo quatro fases, conforme ilustrado na Figura 15.

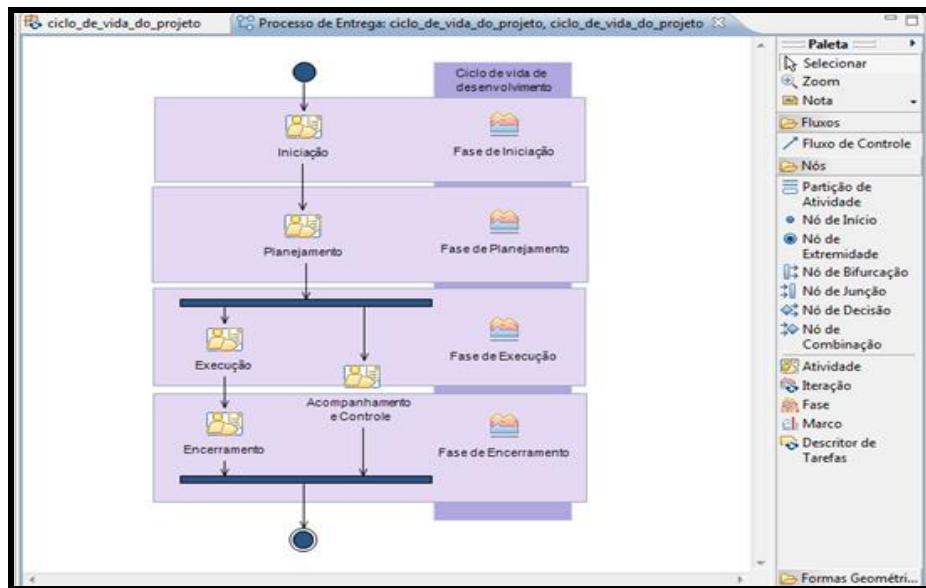


Figura 15. Ciclo de Vida do Processo

Estas fases têm objetivos bem definidos e fornecem uma base para a forma como o trabalho do projeto será estruturado.

Para criar a estrutura do processo e padronizar as atividades de desenvolvimento, optou-se por utilizar as denominações existentes no PMBOK: Iniciação, Planejamento, Acompanhamento/Controle e Encerramento; conforme apresentado na Figura 16.

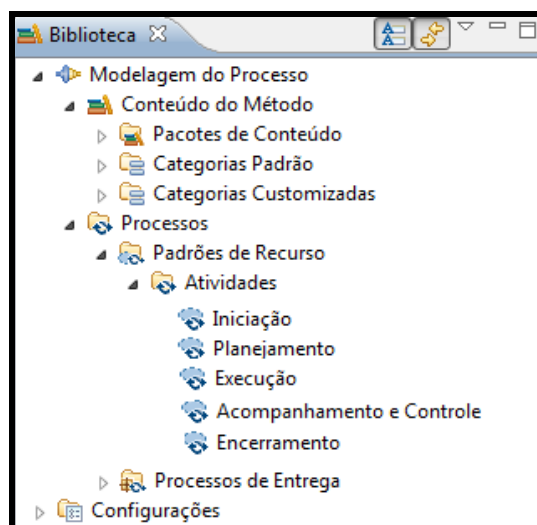


Figura 16. Padrão de Recurso para reutilização das Fases

Com o objetivo de padronizar estas fases e registrar o conhecimento para posterior reutilização, foi criado o Padrão de Recurso. Ele permite a criação de conjuntos específicos de atividades que podem ser agrupadas por área de processo ou por assunto, dependendo da melhor definição adotada pelo desenvolvedor do processo.

No presente trabalho, foi criado um conjunto de atividades dentro do Padrão de Recurso para agrupar o ciclo de vida principal do processo definido, conforme poder ser visto na Figura 16.

Na modelagem, optou-se em utilizar um *plugin* do Conteúdo de Método e foram criados os seguintes pacotes de métodos: Iniciação, Planejamento, Execução, Acompanhamento e Controle, e Encerramento, conforme apresentado na Figura 17.

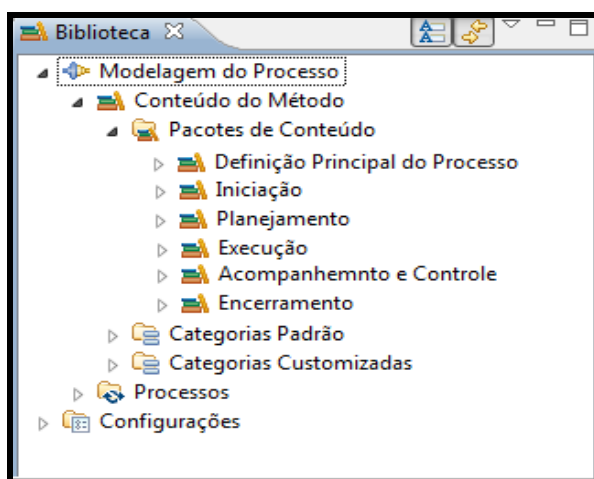


Figura 17. Conteúdo de Método do Processo

Esses métodos são compostos por tarefas, de acordo com o que esta sendo ilustrado na Figura 18, e servem para descrever o que é necessário para a conclusão de cada tarefa.

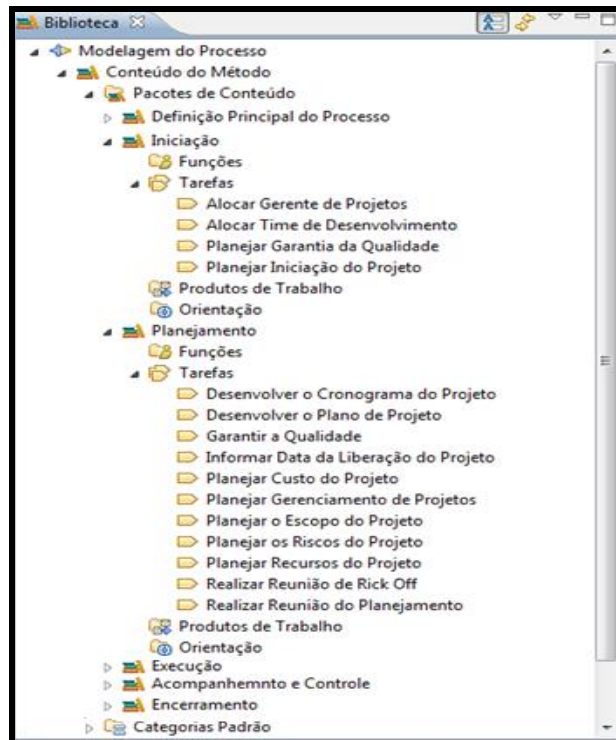


Figura 18. Tarefas disponíveis para serem utilizadas no Processo

Estas tarefas têm como finalidade a descrição dos objetivos desempenhados dentro do processo, podendo ser analisadas por todos os envolvidos do projeto. Vale destacar que as tarefas descritas definem o mínimo a ser realizado.

Caso o responsável considere necessário realizar outras atividades, poderá agregá-las ao seu processo.

As tarefas são muito importantes e, com a utilização do EPF Composer, torna-se simples descrevê-las, pois são definidas por uma série de etapas, funções e produtos de trabalho. Dessa forma, o processo fornece uma compreensão intuitiva de como as tarefas podem ser planejadas, quem será o responsável pela criação ou atualização dos artefatos, oferecendo ainda um guia prático sobre o significado da tarefa e como ela deve ser executada.

Na Figura 19, são apresentados como exemplo do processo definido, os artefatos de entrada e saída bem como o responsável pela tarefa “Alocar_Gerente_de_Projetos”, contendo também as descrições referentes à execução, finalidade e descrição principal.



Figura 19. Visualização da Tarefa: Alocar Gerente de Projetos

Outro fator importante é a criação das Funções. Conforme ilustrado na Figura 20, foi criado um pacote de método chamado Descrição Principal do Processo, local em que foram designadas todas as funções envolvidas no projeto.

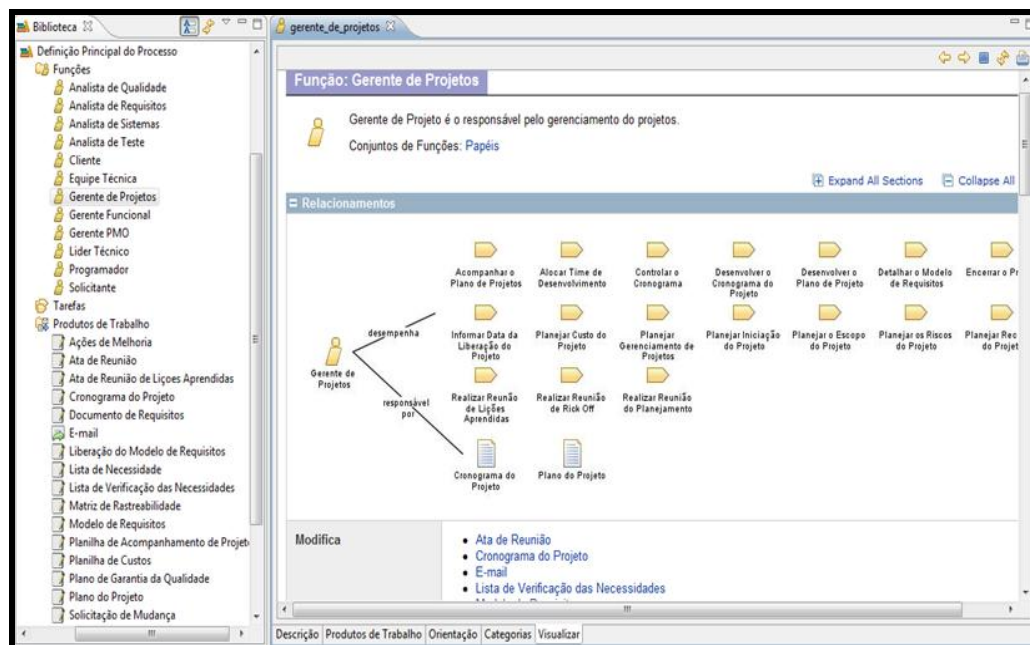


Figura 20. Funções disponíveis para definição do Processo

Dentro deste pacote, foram criados os Produtos de Trabalho, os artefatos, que possuem a descrição de sua funcionalidade no processo. O modelo proposto ainda agrupa um conjunto de *templates* padrão que podem ser utilizados na implementação do processo.

Todas as funções foram descritas, destacando-se os papéis e as responsabilidades delas no processo. A descrição conta com informações claras e organizadas a fim de que todos possam entender o objetivo do projeto.

Dentro de uma organização, um profissional pode assumir várias funções, desde que possua as competências necessárias pra exercê-las, conforme estabelecido na definição do processo.

Para cada fase do processo de desenvolvimento de software, há uma série de atividades a serem executadas. As quais foram organizadas em um conjunto lógico que, quando aplicado em uma linha de desenvolvimento, apóia a criação de um produto de software com qualidade.

Na modelagem, o ciclo de vida do processo possui atividades que apontam e determinam as fases responsáveis por agrupar as atividades que, por sua vez, possuem tarefas cuja função é descrever um fluxo de trabalho para que os objetivos do projeto sejam alcançados. As próximas subseções deste capítulo apresentam as atividades descritas no ciclo de vida do projeto.

6.2.2 Definição dos Fluxos de Atividades do Processo

O fluxo de atividades do processo de Gerência de Projetos foi definido com base nos resultados esperados pelo processo GPR do MPS.BR, além de ser influenciado pelo modelo de referência para o desenvolvimento da modelagem utilizada neste trabalho.

Por meio do fluxo de atividades, o processo Gerência de Projetos é sistematizado a fim de possibilitar que as atividades, os recursos e as responsabilidades do projeto sejam definidos. E também para fornecer informações, acerca do andamento do projeto, que permitam a realização de correções quando houver desvios significativos em seu desempenho.

Ao passo que é feita a identificação dos assuntos pertinentes ao projeto, são analisadas e planejadas as atividades e tarefas a serem executadas para se alcançar os objetivos. São definidos também os recursos que serão necessários para que o produto/serviço seja obtido. Ao se estabelecer as tarefas e as atividades, deve-se atribuir as responsabilidades pela execução delas bem como alocar os recursos necessários para que elas sejam feitas de forma correta. A ação de coordenar atividades, tarefas e recursos de um projeto envolve designar prioridades e integrar atividades e tarefas e, desse modo, distribuir recursos corretamente.

Ao concluir os primeiros passos, é preciso haver monitoramento constante no que está sendo realizado e utilizado a fim de que as possíveis inconformidades sejam identificadas e tratadas antes de causarem danos ao plano estabelecido para a execução do projeto.

Os propósitos do processo de GPR constam nas atividades a serem realizadas dentro do processo e serão vistas nas subseções que detalham cada fase e suas respectivas atividades. Tais fases podem ser visualizadas no modelo desenvolvido conforme apresentado na ilustração do Apêndice B.

6.2.2.1 Iniciação

Na fase de Iniciação, são concedidas várias atribuições ao projeto, conforme ilustrado na Figura 21. Esta fase é a integração de todos os envolvidos no processo, das definições e da autorização para dar-se início ao projeto.

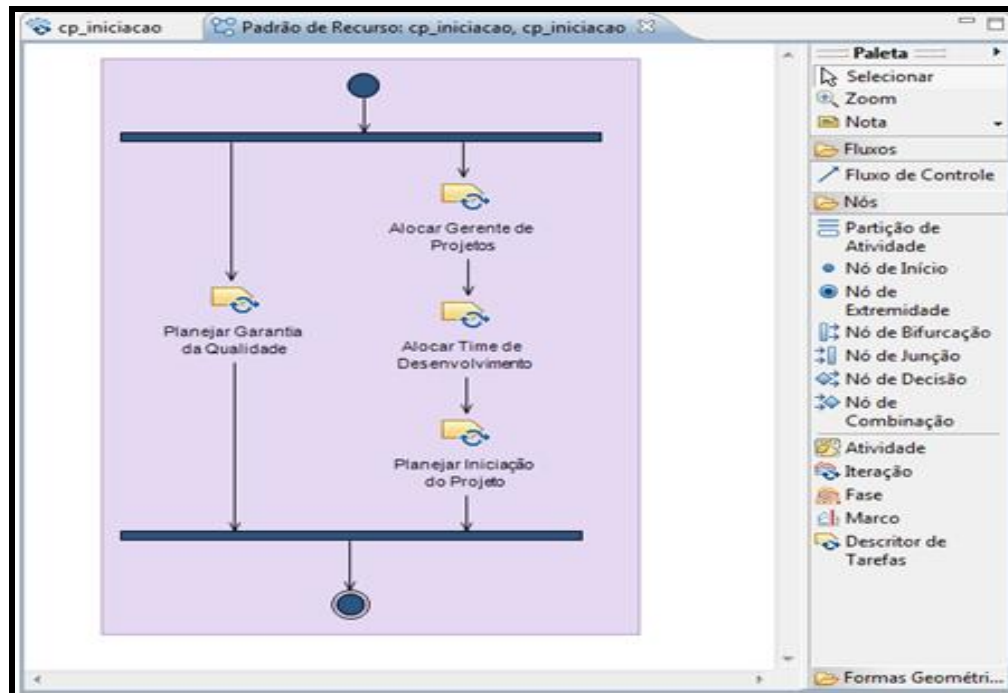


Figura 21. Modelagem da Atividade: Iniciação

O *Project Management Office* (PMO) é responsável por receber a Solicitação de Projetos e analisar os requisitos enviados pelo solicitante. Caso estejam de acordo, ele designará o Gerente de Projetos, pois o papel do PMO é liderar a equipe de gerenciamento de projetos e avaliar a viabilidade do planejamento executado pela sua equipe dentro dos projetos estratégicos da organização.

Estando designado, o Gerente de Projetos assume a responsabilidade de alocar o time responsável pela execução das tarefas definidas no escopo do projeto.

Na modelagem, cada profissional envolvido tem um departamento que lhe compete e cada departamento, por sua vez, possui um Líder Técnico que irá se reunir com Gerente de Projetos a fim de selecionar o profissional disponível e apto a exercer a função de cada projeto.

Com a iniciação do projeto, é necessário o planejamento da garantia da qualidade do projeto a ser desenvolvido. Esta tarefa tem por objetivo verificar se as atividades do projeto estão de acordo com o processo definido, obtendo, dessa forma, um gerenciamento

controlado capaz de garantir que a execução das tarefas está dentro de um padrão de qualidade.

Após a definição das atividades, é importante reunir a equipe técnica para planejar o cronograma e o plano de projeto. Os participantes deverão assinar a Ata da reunião e assumir o compromisso correspondente às responsabilidades que lhes cabem no desenvolvimento do projeto.

6.2.2.2 Planejamento

Na fase de planejamento, é estabelecida uma análise apurada das tarefas e atividades e planejada a ação necessária para atingir os objetivos do escopo do projeto.

Nesta fase, também são detalhadas as informações feitas na iniciação e é criada uma base para as decisões que serão tomadas ao longo do projeto, conforme ilustrado na Figura 22:

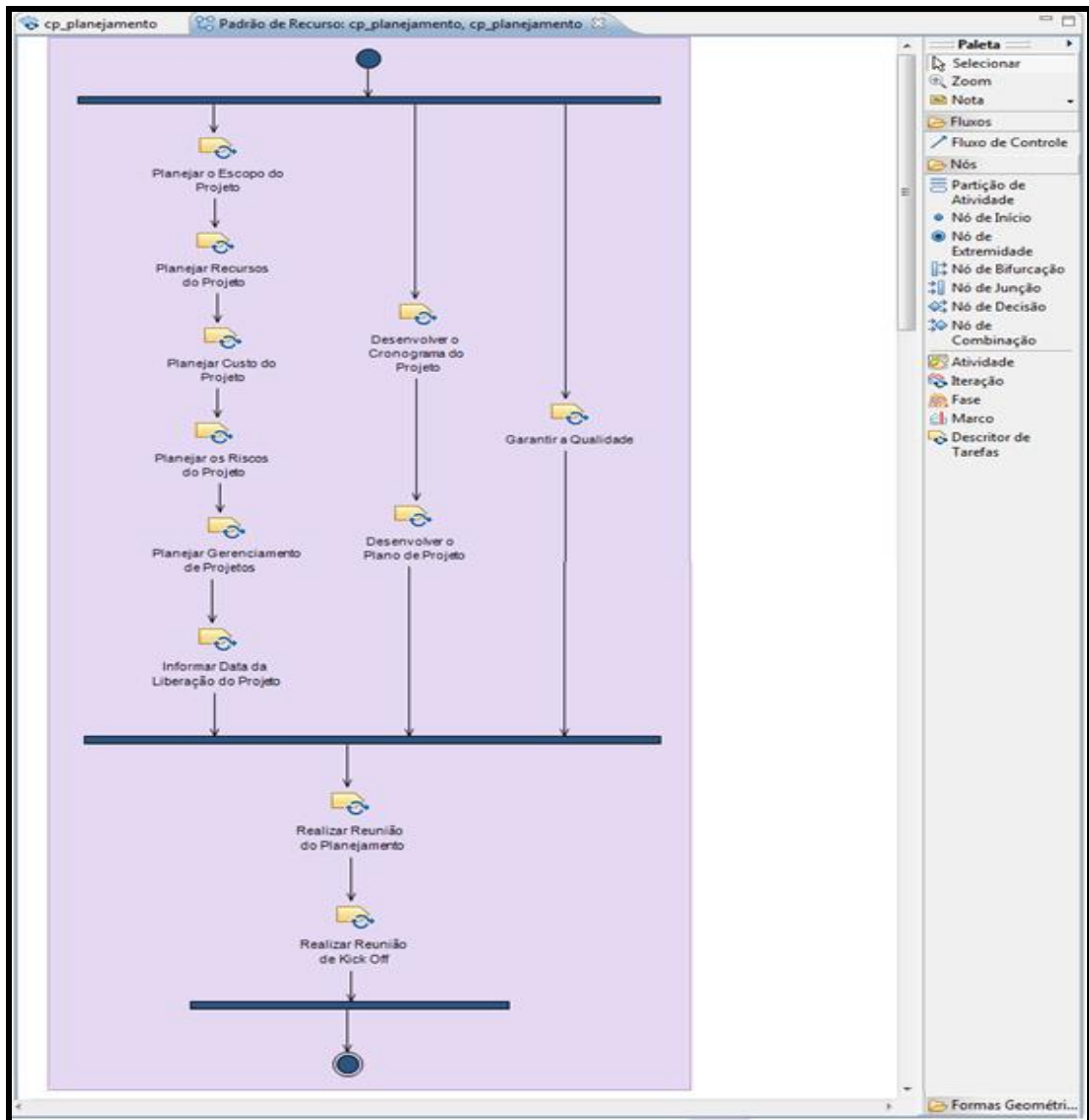


Figura 22. Modelagem da Atividade: Planejamento

As primeiras preocupações do Gerente de Projetos são: o detalhamento do escopo e dos requisitos do produto, e analisar os objetivos que devem ser alcançados no final do projeto.

Após o detalhamento do escopo, parte-se para as definições das atividades necessárias para o desenvolvimento do produto a ser entregue, além do planejamento dos recursos necessários, atividade coordenada pelo Gerente Funcional da empresa, responsável por analisar recursos materiais e recursos humanos para o desenvolvimento do projeto. Em seguida, é estimado o custo dos recursos necessários para completar as atividades do projeto

que contém informações dos custos previstos para cada projeto, sendo esse anexado ao plano de projeto.

Além de estimar custos, é necessário realizar um levantamento dos prováveis riscos do projeto, analisando-se os efeitos que podem impactar negativamente no projeto a fim de evitá-los.

O Gerente de Projetos é responsável por desempenhar o gerenciamento das atividades descritas no plano de projetos e verificar se elas estão sendo atendidas, garantindo, assim, um gerenciamento controlado das atividades. É necessário definir-se o tempo de execução de cada atividade e, com base nisso, estimar-se o custo.

O cronograma é elaborado nesta fase, pois é um modo de acompanhar as atividades. Para cada atividade, serão definidas as datas iniciais e finais de execução. O Gerente de Projetos pode decidir como e quando elas serão realizadas bem como analisar os recursos para atingir os objetivos de custo e tempo delas.

Cada atividade executada tem de ter assegurada a sua garantia de qualidade quanto aos desvios e pontos de melhoria. O objetivo da garantia da qualidade é dar suporte ao controle da qualidade no processo, para que atenda as definições da padronização feita na fase de iniciação e ainda para que a execução seja realizada conforme elicitado. O Analista de Qualidade tem a responsabilidade de apontar a melhoria contínua do processo e é encarregado pela padronização da qualidade do produto.

O desenvolvimento do plano de projeto consiste em relatar todas as informações referentes a ele, sendo elas documentadas para serem controladas e monitoradas. As modificações do escopo do projeto devem ser relatadas e anexadas ao plano de projeto.

Após o término das definições detalhadas no planejamento, é feita uma reunião para se revisar o que fora realizado. A revisão não é elaborada pela mesma equipe que está envolvida, é selecionado um profissional, geralmente outro Gerente de Projetos, para analisar

se as atividades planejadas estão de acordo com o escopo e, assim, efetuar a validação do projeto. O profissional incumbido desta análise deverá documentar e encaminhar, ao Gerente de Projetos, as possíveis não conformidades encontradas. Ele deverá analisar também a viabilidade do projeto.

Em seguida, é realizada uma reunião denominada *kickOff*, com todos os membros diretamente envolvidos no projeto, na qual o Gerente de Projetos irá informar o resultado das análises feitas e entregar o planejamento do projeto, certificando-se ainda de que todos tenham entendimento sobre suas funções e responsabilidades dentro do projeto. É formalizado o começo do projeto bem como os principais tópicos, sendo eles: o objetivo do projeto, os principais riscos, o prazo, o custo e a data de entrega.

Estas informações já devem constar no plano de projeto, assim sendo, o intuito principal da reunião é esclarecer certos pontos e entrosar a equipe responsável por executá-lo, motivando-os a cumprir os objetivos do projeto cientes de suas funções.

6.2.2.3 Execução

Esta fase é a etapa mais delicada do processo, pois as mudanças começam a surgir quando chega o momento de colocar em prática tudo que foi planejado. Como foi visto anteriormente, na fase de planejamento, são definidas quais atividades serão realizadas, definições estas que servirão de base para a condução da fase de execução.

O Gerente de Projetos e os envolvidos deverão obter uma comunicação objetiva e clara, em virtude dos desvios do projeto, pois, com uma comunicação eficaz, minimiza-se a dificuldade de solucionar os problemas que surgem ao longo do projeto.

A maior responsabilidade do Gerente de Projetos corresponde à integração das atividades com a equipe técnica, pois ela tem de ser mantida alinhada aos objetivos do projeto.

Na modelagem, optou-se em abordar algumas tarefas que expressassem a fase de execução, as quais podem ser visualizadas na Figura 23.

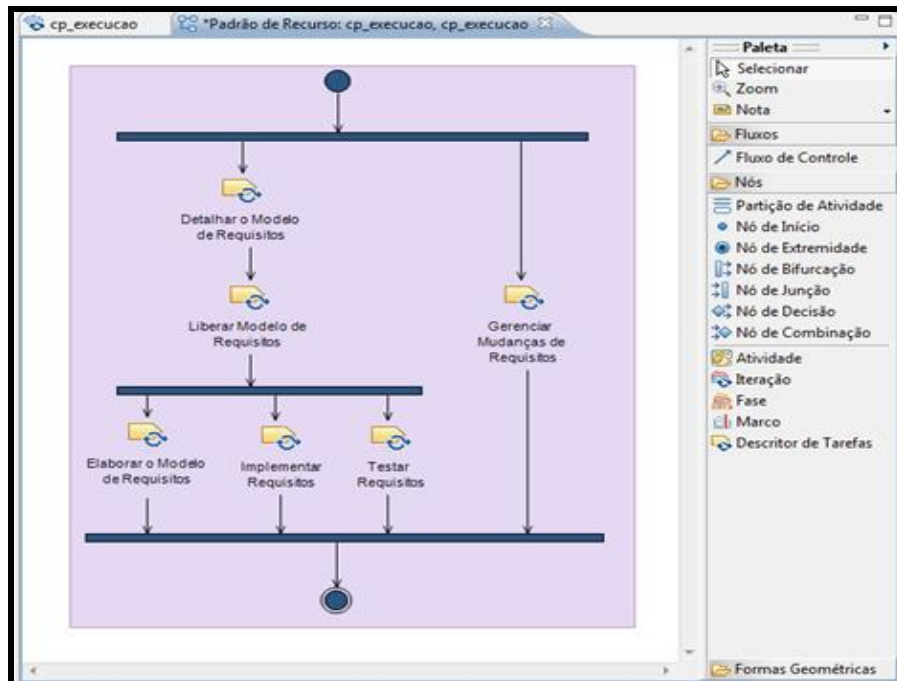


Figura 23. Modelagem da Atividade: Execução

Os requisitos dos softwares são estudados pelo Analista de Requisitos e passam por análises para que não haja problemas futuramente. Os requisitos são detalhados como objetivo de organizar um conjunto de artefatos que descrevam os requisitos a serem implementados. A finalidade é transformá-los em conjuntos técnicos desejados para o sistema que guiarão o projeto.

Em seguida ao detalhamento dos requisitos, é elaborado o modelo de requisitos, no qual as especificações são detalhadas de maneira mais técnica. Estes modelos são representações gráficas que descrevem processos e sistemas a serem desenvolvidos.

Os modelos de requisitos podem tratar de requisitos funcionais e não funcionais. Sendo considerados funcionais: os casos de uso do sistema que descrevem os fluxos de eventos, as prioridades, os atores, as entradas e saídas de cada caso de uso a ser implementado. Já os não funcionais são: a usabilidade, a confiabilidade, o desempenho, a segurança, a distribuição, a adequação a padrões e os requisitos de hardware e software.

Na tarefa de implementar os requisitos, o programador irá receber o modelo de requisitos e codificar os elementos especificados afim de garantir o que foi planejado. Logo depois da codificação dos requisitos, é de extrema importância realizar-se o teste dos requisitos cuja finalidade é obter a qualidade do produto desenvolvido. .

O Analista de Teste irá detectar previamente os erros mais comuns, sendo eles: a falta de clareza dos requisitos, a duplicidade entre eles e a falha na modelagem deles. O analista elabora um artefato com as solicitações de mudança que servirá para documentar e controlar os erros, podendo fornecer também um registro das decisões para garantir que sejam atendidas no projeto.

Todas as modificações realizadas dentro da fase de execução passam por um gerenciamento de requisitos que tem por objetivo gerenciar as mudanças do projeto. Estas solicitações de mudanças são analisadas e controladas, pois podem apresentar erros de inconsistência descobertos durante a validação de requisitos ou, mais tarde, durante o processo de desenvolvimento.

A clareza dos objetivos dos requisitos é fundamental, pois contribui para a qualidade do produto a ser desenvolvido.

6.2.2.4 Acompanhamento e Controle

Esta fase ocorre paralelamente à fase de execução do projeto. Ela tem como objetivo acompanhar e controlar o que está sendo realizado no projeto e propor ações corretivas e preventivas no menor tempo possível logo após a detecção dos problemas. A modelagem da fase de execução pode ser visualizada na Figura 24.

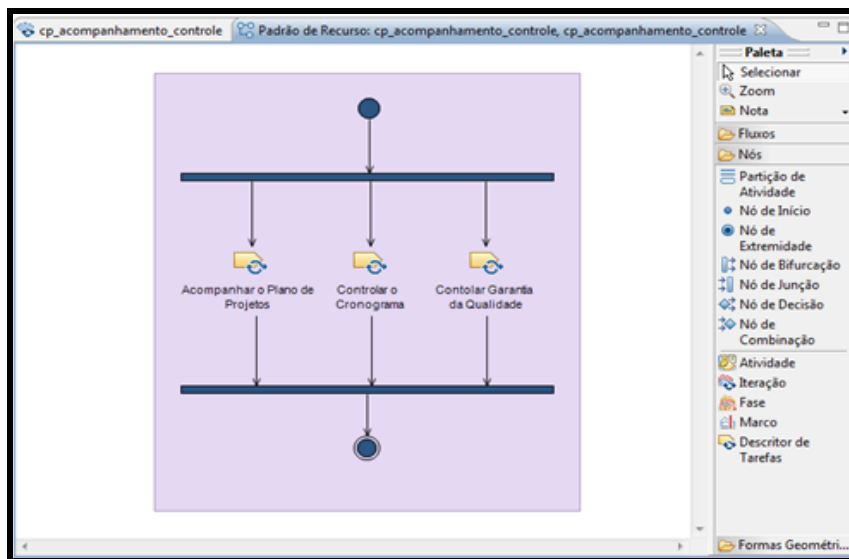


Figura 24. Modelagem das Atividades: Acompanhamento e Controle

A finalidade de acompanhar e controlar é comparar o plano de projeto e o cronograma que está sendo seguido. Verificando-se, assim, se está seguindo o planejado bem como os padrões de qualidade adotados. Caso ao contrário, será necessário aplicar ações corretivas nos desvios identificados.

O profissional responsável por acompanhar e controlar tais atividades é o Gerente de Projetos, que vai coordenar as tarefas e atividades e irá analisar e atualizar o Plano de Projeto e o cronograma.

A tarefa de acompanhar o plano de projeto é o monitoramento da execução das tarefas conforme descritas no Plano do Projeto. O controle do cronograma é garantia de que as tarefas definidas no início do projeto estão sendo feitas no prazo estimado caso haja

desvios do projeto. O Gerente de Projetos terá a visão global das tarefas e atividades, podendo controlar os desvios causados pelas mudanças e gerenciar de modo a minimizar os efeitos negativos no projeto.

Na tarefa de controlar a garantia da qualidade, o Gerente de Projetos, juntamente com Analista de Qualidade, tem por objetivo monitorar os resultados específicos do projeto para determinar se eles atingem os padrões de qualidade e identificar as ações responsáveis por eliminar as causas de desempenhos insatisfatórios.

O Analista de Qualidade é o responsável por criar o artefato “Ações de Melhoria”, uma lista de ações a serem implementadas para garantir a melhoria contínua do processo.

6.2.2.5 Encerramento

A fase de Encerramento é a finalização das atividades concluídas do processo. As atividades de encerramento são apresentadas na Figura 25.

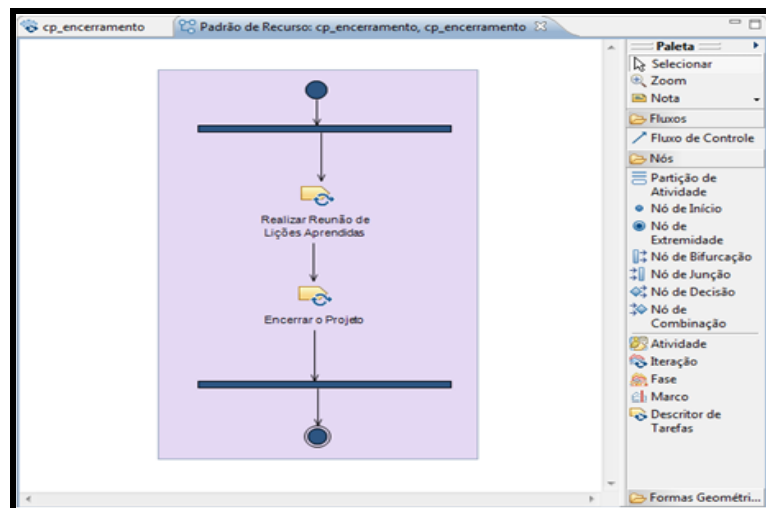


Figura 25. Modelagem da Atividade: Encerramento

É realizada uma reunião para apontar todas as falhas ocorridas durante o projeto, as quais são analisadas a fim de que erros similares não ocorram em novos projetos. É criado

um artefato “Lições Aprendidas”; nele são documentadas as principais lições aprendidas vivenciadas durante a execução do projeto. Isso é feito com o intuito de se aprimorar o conhecimento dos participantes do projeto, proporcionando maior probabilidade de se obter sucesso nos próximos projetos e a consequente satisfação do cliente.

O projeto é, então, formalmente encerrado para todos os envolvidos, especialmente ao cliente, informando-lhe que o projeto está concluído e que as novas solicitações serão atendidas em novo projeto.

O cliente assinará o termo de encerramento, documento no qual são registrados dados referentes ao atendimento das solicitações do projeto, podendo ter sido sanadas total ou parcialmente. Caso o projeto tenha sido cancelado, serão detalhadas as razões da suspensão. O documento em questão é emitido pelo Gerente de Projetos que se encarregará de enviá-lo ao PMO a fim de que ele seja armazenado digitalmente.

CONCLUSÃO

O estudo da Gerência de Projetos provou ser uma área de extrema complexidade e que necessita da utilização de ferramentas para a gestão de processos. Em virtude do elevado número de projetos com orçamentos e cronogramas acima do previstos, torna-se clara a necessidade de um efetivo gerenciamento de projetos, levando-se em conta os padrões de qualidade.

Optou-se em estudar o MPS.BR como modelo de referência para a aplicação das definições de melhoria de processo de software, atendo-se apenas à Gerência de Projetos (GPR), área de processo definida no nível G de maturidade do referido modelo.

Dessa forma, foi desenvolvida, neste trabalho, uma metodologia para auxiliar às definições de processos de software, tendo como base as boas práticas estabelecidas pelos resultados do modelo MPS.BR adequado às principais normas e modelos de qualidade de software existentes no mercado.

Para a modelagem do processo, foi estudado o framework EPF Composer, ferramenta desenvolvida para modelar e publicar processos, sendo de grande importância para o presente trabalho, pois sua estrutura é de fácil compreensão, contribuindo com a definição e a modelagem dos fluxos do processo.

A metodologia proposta apresentou um conjunto de fases e atividades que induzem à melhoria do processo de software de forma planejada. Além disso, foram propostos *templates* de documentos que auxiliam a execução da metodologia proposta.

Na elaboração da metodologia, comprovou-se que o conhecimento adquirido com essa experiência é de muita importância, pois o entendimento do objeto de estudo deste trabalho, proporciona a evolução acadêmico - científica e profissional na área de engenharia de software e planejamento de projetos, podendo ser utilizada com eficiência nas mais

diversas linhas de desenvolvimento presentes atualmente, na área de tecnologia da informação.

Como visto no estudo apresentado, considera-se viável o uso da metodologia de processos em organizações que visam estabelecer um caminho para melhorar seus produtos e serviços por meio do amadurecimento de processos e consequente adequação da sua estrutura organizacional. Fator que tende a auxiliar os profissionais da área de TI na execução de um gerenciamento controlado e com qualidade comprovada, minimizando atrasos e os pontos de insucesso, constantemente associados à realização de projetos de software, reduzindo, dessa forma, o custo e aumentando a confiabilidade da organização.

Desta forma, conclui-se que o MPS.BR é um modelo que apóia fortemente a prática de melhoria contínua em todas as etapas presentes no processo de desenvolvimento de software. Ele oferece recursos que identificam e minimizam riscos, proporcionando maior flexibilidade no atendimento dos requisitos a serem implementados, de forma sistemática, organiza todas as etapas bem como suas respectivas atividades presentes ao longo do processo de desenvolvimento, facilitando, assim, o planejamento, a execução, o controle e a avaliação contínua.

A principal contribuição deste trabalho foi a definição de uma metodologia de melhoria voltada para as organizações que desenvolvem software. O MPS.BR é visto como um modelo adequado às empresas brasileiras que buscam a evolução contínua da qualidade de seus produtos, projetos e da capacitação dos recursos empregados no desenvolvimento do software produzido.

A partir desta metodologia, as organizações que desejarem padronizar seus processos podem utilizar, como base, a metodologia desenvolvida neste trabalho e adequá-la para atender a realidade da organização conforme sua necessidade.

Durante o desenvolvimento da metodologia proposta, algumas possibilidades de trabalhos futuros foram identificadas:

- a) Desenvolver a implementação do modelo de maturidade do MPS.BR, utilizando-se a metodologia desenvolvida neste trabalho.
- b) Avaliar a metodologia desenvolvida por meio de uma análise comparativa de *gap*, usando os seguintes elementos comparativos: modelos referência e o modelo desenvolvido neste trabalho.
- c) Abordar novas áreas de processo para os níveis mais avançados do modelo MPS.BR.

REFERÊNCIAS

ABNT, **Associação Brasileira de Normas e Técnicas. NBR 6023: informação e documentação: referências: elaboração.** Rio de Janeiro, 2002.

CAROSIA, Jaciara S. **Levantamento da qualidade do Processo de Software com foco em pequenas organizações.** 2004, 158f. . Dissertação (Mestrado em Computação Aplicada) - Curso da Pós-Graduação em Computação Aplicada – Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos. 2004.

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

ECLIPSE, **Process Framework Project (EPF) Composer.** Version 1.5.0.4. 2009. Disponível em: <<http://www.eclipse.org/epf/>>. Acesso em: 05 ago. de 2010.

ESPINHA, Rafael; SOUZA, João. Melhorando Processos Através da Análise de Riscos e Conformidades. **Engenharia de Software Magazine: Qualidade de software: Entenda os principais conceitos sobre Testes e Inspeção de Software.** Brasil, n. 1, p.10-21, 2007. Disponível em: <<http://www.devmedia.com.br> >. Acesso em: 28 out. 2009.

FERNANDES, Marley; CORREA, Samuel da S. **Os modelos de maturidade de processos de software e o MPS.BR .** 2008. 48 p. Trabalho de Conclusão de Curso – Faculdade de Minas. Curso de Sistemas de Informação da FAMINAS-BH, Belo Horizonte 2008. Disponível em: <http://www.ricardoterra.com.br/publications/students/2008_fernandes_e_correa.pdf>. Acesso em: 15 out. de 2009.

FERRARI, Sandra. **Proposta de metodologia para controle de qualidade em uma fábrica de software.** 2007. 302 f. Tese (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <http://www.tede.ufsc.br/tedesimplificado//tde_busca/arquivo.php?codArquivo=77>. Acesso em: 15 out. 2009.

JULIO, Ricardo E.. **Elaboração e Modelagem de um processo de desenvolvimento de software aderente ao MPS.BR do nível G.** 2007. 91 p. Monografia (Bacharel em Ciência da Computação) - Universidade Federal de Lavras, Lavras, 2007. Disponível em:<http://www.bcc.ufla.br/monografias/2007/Elaboracao_e_modelagem_de_um_processo_

de_desenvolvimento_de_software_aderente_ao_mps_br_nivel_G.pdf>. Acesso em: 14 out. 2009.

KOSCIANSKI, André; SOARES Santos, Michel dos. **Qualidade de Software**. Novatec, 2006.

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

LOOS, Anildo. **Processo de Garantia da Qualidade baseados no Modelo MPS – BR**. 2009. 101 p.(Bacharel em Ciência da Computação). Trabalho de Conclusão de Curso de Ciência da Computação da Universidade Regional de Blumenau. Blumenau. 2009.

PETERS, James F.; PEDRYCZ, Wiltold. **Engenharia de Software: Teoria na Prática**. Tradução: Ana Patricia Gargia 3.ed. Rio de Janeiro: Campus, 2001.

PFLEEGER, Shari L. **Engenharia de Software: Teoria na Prática**. 2.ed. São Paulo: Prentice Hall, 2004.

PMBOK, **Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos** (Guia PMBOK ®). 2004. 3ª Edição. Disponível em: <<http://www.cos.ufrj.br/~mareas/livros/PMBOK.pdf>>. Acesso em: 19 out. 2009.

PRESSMAN, Roger S. **Engenharia de Software**. 5º ed. Rio de Janeiro. McGraw-Hill, 2002;

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

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

SAMARANI, Paulo R. de M.. **Um modelo de implementação do Capabilit Maturity Integration nível 2**. 2005. 135f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porta Alegre. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/5698>>. Acesso em: 24 mar. de 2010.

SERAFIM, Gustavo. **Eclipse Process Framework**: Um ambiente de Engenharia de Software livre para publicar e manter métodos e processos. Análise de Riscos e Conformidades. **Engenharia de Software Magazine**: Metodologias Ágeis. Brasil, n. 4, p.50-59, 2008. Disponível em: <<http://www.devmedia.com.br>>. Acesso em: 03 nov. 2010.

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

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

SOMMERVILLE, Ian. **Engenharia de Software**. 8.ed. São Paulo: Pearson Addison-wesley, 2007.

VARGAS, Ricardo Viana. **Manual prático do plano de projeto**: utilizando o PMBOK Guide. 3. ed. Rio de Janeiro: Brasport, 2007.

VIEIRA, Marconi Fábio. **Gerenciamento de projetos de tecnologia da informação**. Rio de Janeiro: Campus, 2003.

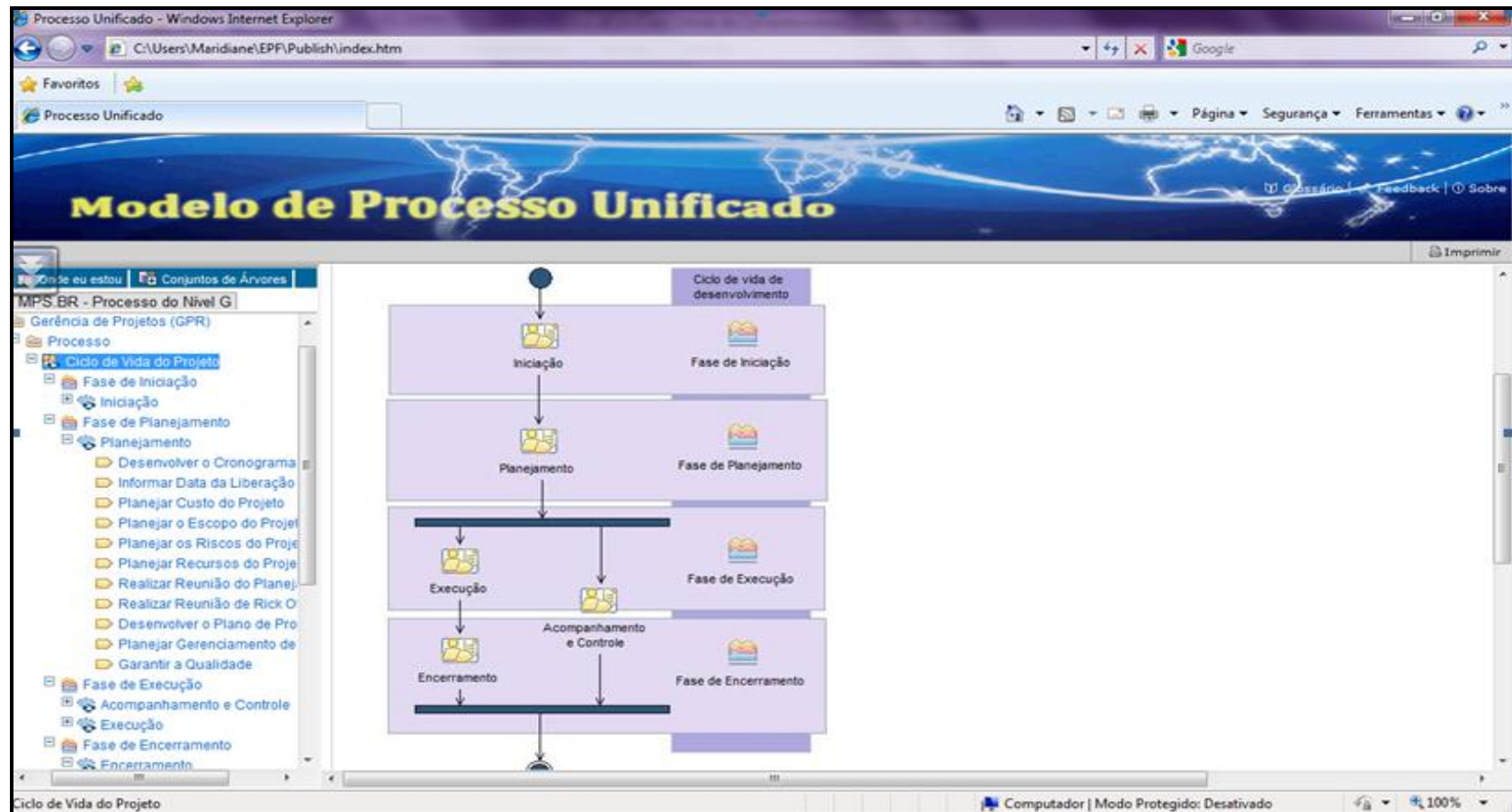
APÊNDICE A – PUBLICAÇÃO DO MODELO DE PROCESSO UNIFICADO

A publicação do modelo de processo desenvolvido pode ser visualizada conforme o apresentado a seguir:



APÊNDICE B - PUBLICAÇÃO DO CICLO DE VIDA DO PROJETO

As fases descritas no modelo de processo podem ser vistas na publicação em uma web site:



APÊNDICE C – ARTIGO CIENTÍFICO

DESENVOLVIMENTO DE UMA METODOLOGIA PARA GERENCIAMENTO DE PROJETOS BASEADA NA ÁREA DE PROCESSO: GERÊNCIA DE PROJETOS GPR DO MODELO DE REFERÊNCIA MPS.BR.

Maridiane Corrêa Plácido ¹, Gustavo Bisognin ²

¹ Acadêmica do Curso Ciência da Computação - Universidade do Extremo Sul Catarinense (UNESC) - Criciúma – Brasil

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

maridianecorrea@hotmail.com, gustavo@unesc.net

ABSTRACT. *With the great advancement of available technologies and development techniques, the methodology of work stands out as a differential of organizations and, consequently, the quality of software products. In this context, this paper presents the definition of a methodology for the development process based on the reference model MPS.BR. It is expected that the use of this methodology can contribute and assist companies that seek to adapt its processes inside a context of quality, aggregating it in their products and services in order to minimize the problems that happen in the management and project control*

RESUMO. *Com o constante avanço das tecnologias disponíveis e técnicas de desenvolvimento, a metodologia de trabalho destaca-se como um diferencial das organizações e, conseqüentemente, a qualidade dos produtos de software. Nesse contexto, o presente artigo apresenta a definição de uma metodologia para o processo de desenvolvimento de software, baseado no modelo de referência MPS.BR. Espera-se que a utilização desta metodologia possa contribuir e auxiliar as empresas que buscam adequar seus processos dentro de um conjunto de qualidade agregando em seus produtos e serviços, afim de minimizar os problemas ocorridos no gerenciamento e controle dos projetos.*

1. INTRODUÇÃO

Nos últimos anos, constatou-se que muitas organizações brasileiras desenvolvedoras de software o fabricam de forma indevida, desprovidas de planejamento ou controle sobre o projeto em razão da ausência de uma metodologia de trabalho a ser seguida por todos os envolvidos. Diante desse cenário, as organizações precisam se adequar a um programa de melhoria a fim de que seus processos sejam padronizados e, com isso, minimizar as falhas ocorridas no gerenciamento de projetos.

Os processos de desenvolvimento de software estão ligados a padrões de qualidade, tais como: *International Organization for Standardization (ISO)*, *Capability Maturity Model Integration (CMMI)* e *Melhoria de Processo do Software Brasileiro (MPS-*

BR). Tais padrões consistem em normas que auxiliam o desenvolvimento de processos. Além da existência de modelos, as empresas precisam estruturar-se para customizá-los a fim de obter resultados. Porém, isso é um grande desafio, pois, para definir e manter um processo em constante melhoria, é necessário que haja uma equipe qualificada apta para promover o desenvolvimento da qualidade dos processos, de maneira organizada, garantindo a evolução contínua do modelo.

Nesse contexto, o presente artigo apresenta a definição de uma metodologia para o gerenciamento de projetos, baseada na Área de Processo GPR no Nível G do modelo de referência MPS.BR. Mostra ainda as melhores práticas de gerenciamento de projetos a fim de auxiliar os profissionais da área de tecnologia da informação, minimizando os atrasos e os fatores de insucesso dos projetos.

2. Melhoria de Processos de Software

Nos últimos anos, muitas organizações de software têm buscado pela melhoria da qualidade de seus processos no desenvolvimento e manutenção do software. São comprovadamente gratificantes os benefícios conferidos pela aderência a modelos de maturidade, normas de qualidade e guias de boas práticas, no intuito de aprimorar as técnicas e habilidades bem direcionadas no desenvolvimento de softwares.

Começar, talvez, seja a mais importante e difícil fase de qualquer trabalho. No caso da melhoria do processo, o levantamento acerca de como o software está sendo desenvolvido na organização é a primeira atividade a ser executada. O sucesso de um programa de melhoria do processo depende da confiabilidade dos dados adquiridos na fase de avaliação, visto que uma avaliação adequada permite a elaboração de um plano de melhoria mais eficiente e compatível com a realidade da organização (CAROSIA, 2003).

O objetivo de um programa de melhoria de processo não é simplesmente impor regras pré-estabelecidas sobre como as coisas devem ser feitas, mas levar em consideração as características e as necessidades da organização (CAROSIA, 2003).

2.1. MPS.BR

O MPS.BR é um programa coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) e uma de suas metas é definir e aprimorar um modelo de melhoria e de avaliação de processo de software a ser aplicado nas empresas desenvolvedoras de software brasileiro, a fim de garantir qualidade nos seus produtos. Estruturalmente, o MPS.BR é dividido em três componentes: O Modelo de Referência (MR-MPS), o Método de Avaliação (MA-MPS) e o Modelo de Negócio (MN-MPS) (SOFTEX, 2009).

Ele se inicia no nível G e termina no nível A, sendo que a organização evolui nesta escala quando os propósitos e resultados esperados para os processos e Atributos de Processo (AP) daquele nível são atingidos (SOFTEX, 2009).

Os níveis de maturidade, os processos e atributos de processo correspondentes a cada nível podem ser visualizados conforme a ilustração do Quadro 1.

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP, 4.2
C	Gerência de Riscos – GRI	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Desenvolvimento para Reutilização – DRU	
	Gerência de Decisões – GDE	
D	Verificação – VER	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Validação – VAL	
	Projeto e Construção do Produto – PCP	
	Integração do Produto – ITP	
	Desenvolvimento de Requisitos – DRE	
E	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de Reutilização – GRU	
	Gerência de Recursos Humanos – GRH	
	Definição do Processo Organizacional – DFP	
	Avaliação e Melhoria do Processo Organizacional - AMP	
F	Medição – MED	AP 1.1, AP 2.1 e AP 2.2
	Garantia da Qualidade – GQA	
	Gerência de Portfólio de Projetos – GPP	
	Gerência de Configuração – GCO	
	Aquisição – AQU	
G	Gerência de Requisitos (GRE):	AP 1.1 e AP 2.1
	Gerência de Projetos (GPR):	

Quadro 1. Níveis de Maturidade de Processos

Pode-se observar, na tabela acima, que o modelo MPS.BR é dividido em níveis de maturidade. Cada nível é composto por um conjunto de áreas de processo que são implementadas de acordo com alguns resultados estabelecidos para cada atributo de processo definido pelo modelo.

2.1.1 Nível G

O nível de maturidade G é o primeiro do MPS.BR. Ele estabelece o início dos trabalhos em implantação de melhoria dos processos de software e, 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 (SOFTEX, 2009).

Durante a implantação do nível G, pode-se observar uma mudança na cultura organizacional e na 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, ou seja, poder redefinir algumas operações (atividades de rotina) em andamento, estabelecendo objetivos, prazos e escopo para sua execução (SOFTEX, 2009).

Segundo a SOFTEX (2009), o nível G é composto pelos processos de Gerência de Projetos e Gerência de Requisitos, satisfazendo os Atributos de Processos AP 1.1 e AP 1.2, conforme observado no Quadro 1.

a) Gerência de Projetos (GPR): tem o propósito de identificar, estabelecer, coordenar e monitorar as atividades e recursos que o projeto precisa para apresentar o resultado esperado, atendendo aos seus requisitos e restrições.

b) Gerência de Requisitos (GRE): tem como propósito gerenciar os requisitos do projeto bem como investigar possíveis inconsistências entre os requisitos e os planos e produtos de trabalho.

De acordo com SOFTEX (2009), a capacidade do processo é um conjunto de atributos de processo descrito em termos de resultados esperados, os quais proporcionam o atendimento dos atributos de processo.

À medida que a empresa evolui nos níveis de maturidade, um maior ganho de capacidade para desempenhar o processo é atingido por ela.

Os diferentes níveis de capacidade dos processos são descritos por AP. O alcance de cada atributo de processo é avaliado utilizando-se os respectivos Resultado do Atributo de Processo (RAP) do nível G, conforme definido a seguir:

a) O Processo é Executado (AP 1.1): este atributo refere-se à medida do quanto o processo atinge o seu propósito.

b) O processo é Gerenciado (AP 2.1): este atributo corresponde à medida do quanto a execução do processo é gerenciada.

O presente artigo aborda a Área de Processo GPR do nível G do modelo MPS.BR, utilizando as definições dos resultados para o desenvolvimento de uma metodologia para melhoria de processos de software.

3. Desenvolvimento do Processo

A modelagem de um processo de software deve possibilitar o seu entendimento e sua padronização, para tanto, deve-se observar os principais objetivos para modelá-lo.

3.1. Metodologia

A metodologia utilizada para o desenvolvimento deste trabalho iniciou-se pela etapa de levantamento bibliográfico, na qual foram estudados os conceitos relacionados à *Engenharia de Software* e às melhores práticas de gerenciamento de processos na área de Gerência de Projetos.

Com a pesquisa realizada em livros, dissertações, testes e sites na Web, procurou-se compreender os modelos existentes bem como alguns temas referentes a processos de desenvolvimento de software, gerenciamento de projetos, modelos de qualidade e melhoria de processos de software, enfatizando o MPS.BR.

A pesquisa sobre o MPS.BR foi baseada nos diversos guias e documentos do modelo, os quais podem ser encontrados no site da SOFTEX, disponíveis para downloads.

O processo elaborado para o desenvolvimento do software foi desenvolvido com base no *Rational Unified Process* (RUP), pois este aborda o conceito de orientação a objetos,

utilizando notações da *Unified Modeling Language* (UML) para a definição de fluxo de processos.

Os fluxos de trabalho do processo foram feitos utilizando-se do *Eclipse Process Framework Composer* (EPF), uma ferramenta gratuita, criada pela *International Business Machines* (IBM), que proporciona uma estrutura para a gestão de processo de software (SERAFIM, 2008). O EPF contém uma estrutura de bibliotecas e *plugins* suportando processos com base em diferentes abordagens de desenvolvimento em vários modelos de ciclo de vida, incluindo: cascata, iterativos incrementais e ágeis.

O EPF foi projetado para ser um sistema de gerenciamento de conteúdo, fornecendo uma estrutura de gerenciamento comum a fim de solucionar problemas encontrados pelas equipes de desenvolvimento de projetos durante a concepção dele (ECLIPSE, 2009).

O *framework* possui uma funcionalidade de publicação em que o processo pode ser apresentado ou publicado num ambiente digital no formato de uma web site, disponibilizando, assim, a todos os *stakeholders* da organização, o processo de desenvolvimento de software definido para a consulta.

3.2. Modelagem do Processo de Software

Com base nos estudos realizados, percebe-se a importância de um processo de desenvolvimento de software de qualidade. Assim sendo, o primeiro passo para a criação do processo é a definição do modelo de ciclo de vida de projetos, por meio da qual são descritas as fases do processo.

Para a elaboração da metodologia, foi criada uma disciplina na área de processo do nível G do modelo MPS.BR: Gerência de Projetos, atuante como uma categoria padrão oferecida pela EPF e pode ser utilizada para categorizar as atividades e tarefas do processo.

De acordo com tais atividades, foi definido um ciclo de vida do processo contendo quatro fases, as quais têm objetivos bem definidos e fornecem uma base para o modo como o trabalho do projeto será estruturado.

Para criar a estrutura do processo e padronizar as atividades de desenvolvimento, optou-se por utilizar as denominações existentes no PMBOK: Iniciação, Planejamento, Acompanhamento/Controle e Encerramento, conforme apresentado na Figura 1:

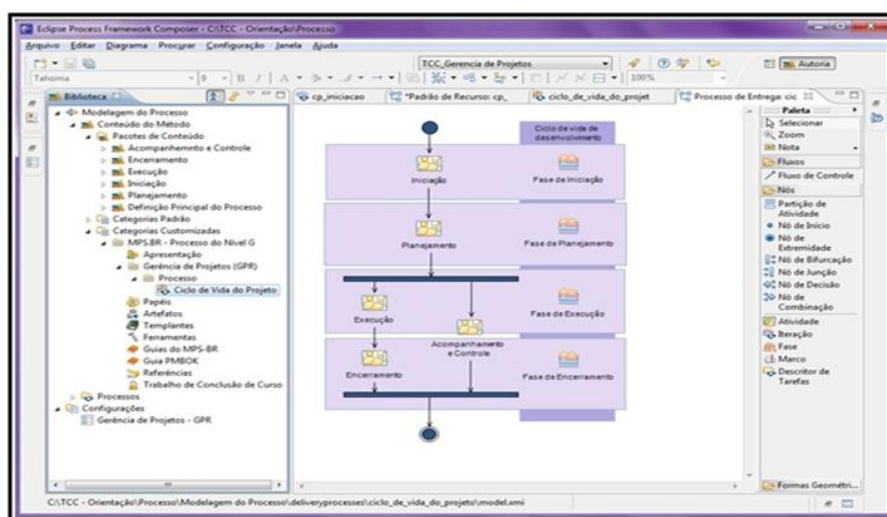


Figura 1. Ciclo de vida da modelagem do Processo

Podem ser visualizados na Figura 132 os principais elementos de métodos.



Figura 2. Elementos básicos do Conteúdo de Método

Para a definição do processo, optou-se por utilizar todos os conceitos sugeridos pelo EPF, pois o objetivo é proporcionar uma documentação organizada e clara. Dessa forma, ao usá-lo, pode-se criar o seu próprio processo de desenvolvimento de software e estruturá-lo em sentido específico. O processo define os métodos que podem ser reutilizados durante a definição de outros processos.

Na modelagem, optou-se em utilizar um *plugin* do Conteúdo de Método e foram criados os seguintes pacotes de métodos: Iniciação, Planejamento, Execução, Acompanhamento e Controle, e Encerramento, conforme apresentado na Figura 173.

Esses métodos são compostos por tarefas de acordo com o que está sendo ilustrado na Figura 183, e servem para descrever o que é necessário para a conclusão de cada tarefa.

Tais tarefas têm como finalidade a descrição dos objetivos desempenhados dentro do processo, podendo ser analisadas por todos os envolvidos no projeto. Vale destacar que as tarefas descritas definem o mínimo a ser realizado. Caso o responsável considere necessário realizar outras atividades, poderá agregá-las ao seu processo.

As tarefas são muito importantes e, com a utilização do EPF Composer, torna-se simples descrevê-las, pois são definidas por uma série de etapas, funções e produtos de trabalho. Dessa forma, o processo fornece uma compreensão intuitiva de como as tarefas podem ser planejadas, quem será o responsável pela criação ou atualização dos artefatos, oferecendo ainda um guia prático sobre o significado da tarefa e como ela deve ser executada.

Na Figura 193, são apresentados, como exemplo do processo definido, os artefatos de entrada e saída bem como o responsável pela tarefa “Alocar_Gerente_de_Projetos”, contendo também as descrições referentes à execução, finalidade e descrição principal.

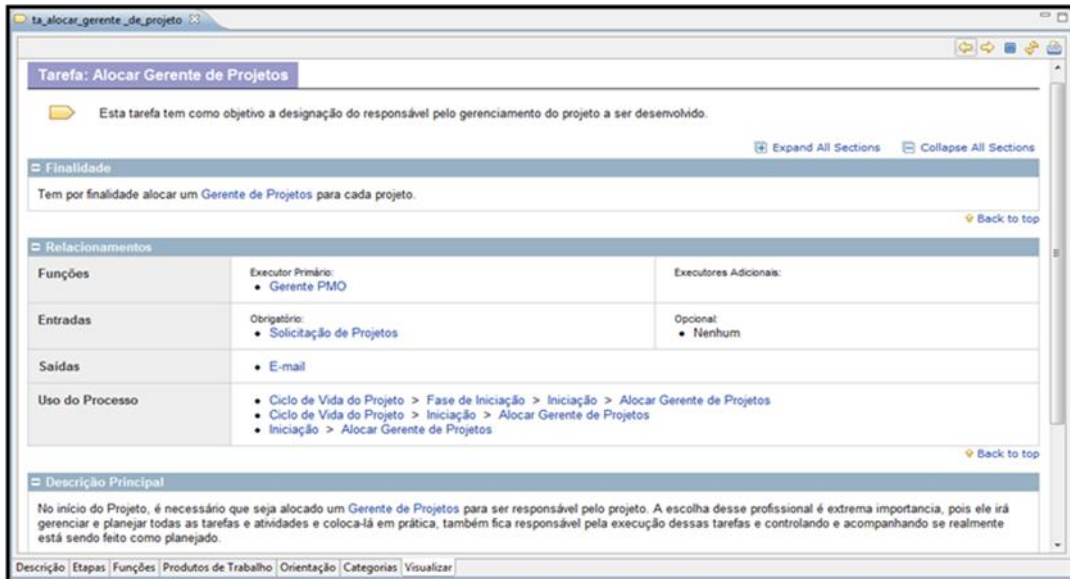


Figura 3. Visualização da Tarefa: Alocar Gerente de Projetos

Outro fator importante é a criação das Funções. Conforme ilustrado na Figura 204, foi criado um pacote de método chamado Descrição Principal do Processo, local em que foram designadas todas as funções envolvidas no projeto.

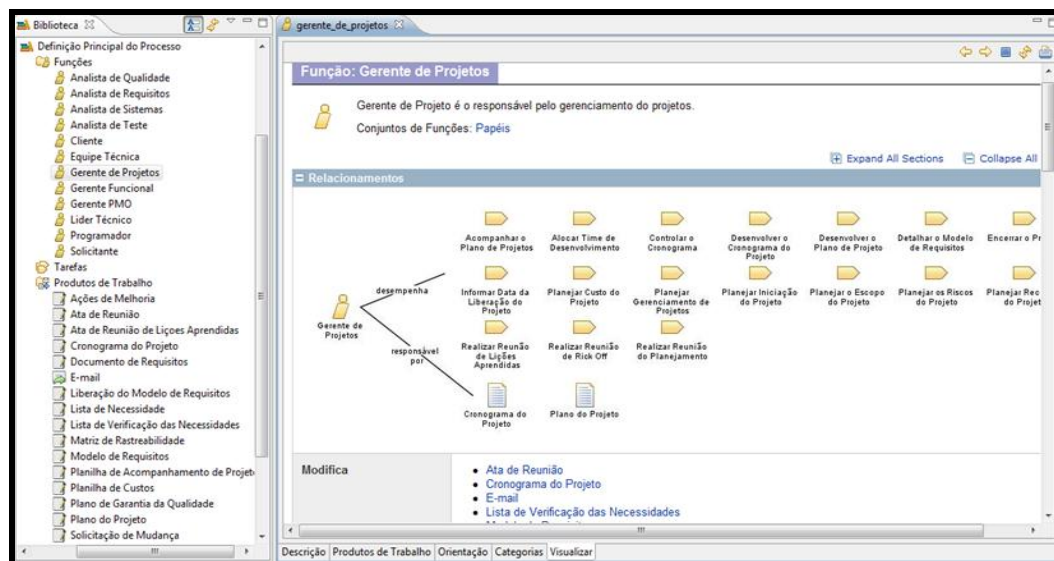


Figura 4. Funções disponíveis para definição do Processo

O modelo proposto ainda agrupa um conjunto de *templates* padrão que podem ser utilizados na implementação do processo.

Todas as funções foram descritas, destacando-se os papéis e as responsabilidades no processo. A descrição conta com informações claras e organizadas a fim de que todos possam entender o objetivo do projeto.

Dentro de uma organização, um profissional pode assumir várias funções, desde que possua as competências necessárias pra exercê-las, conforme estabelecido na definição do processo.

Para cada fase do processo de desenvolvimento de software, há uma série de atividades a serem executadas. As quais foram organizadas num conjunto lógico que, quando

aplicado em uma linha de desenvolvimento, apóia a criação de um produto de software com qualidade.

Na modelagem, o ciclo de vida do processo possui atividades que apontam e determinam as fases responsáveis por agrupar as atividades que, por sua vez, possuem tarefas cuja função é descrever um fluxo de trabalho para que os objetivos do projeto sejam alcançados.

3.3 Definição dos Fluxos de Atividades do Processo

O fluxo de atividades do processo de Gerência de Projetos foi definido com base nos resultados esperados pelo processo GPR do MPS.BR, além de ser influenciado pelo modelo de referência para o desenvolvimento da modelagem utilizada neste trabalho.

Por meio do fluxo de atividades, o processo Gerência de Projetos é sistematizado a fim de possibilitar que as atividades, os recursos e as responsabilidades do projeto sejam definidos. Faz isso também para fornecer informações acerca do andamento do projeto, capazes de permitir a realização de correções quando houver desvios significativos no seu desenvolvimento.

Ao passo que é feita a identificação dos assuntos pertinentes ao projeto, são analisadas e planejadas as atividades e tarefas a serem executadas para se alcançar os objetivos. São definidos também os recursos necessários para que o produto/serviço seja obtido. Ao se estabelecer as tarefas e as atividades, deve-se atribuir as responsabilidades pela execução delas bem como alocar os recursos necessários para que elas sejam feitas corretamente. A ação de coordenar atividades, tarefas e recursos de um projeto envolve designar prioridades e integrar atividades e tarefas e, desse modo, distribuir recursos corretamente.

Ao concluir os primeiros passos, é preciso haver monitoramento constante no que está sendo realizado e utilizado a fim de que as possíveis inconformidades sejam identificadas e tratadas antes de causarem danos ao plano estabelecido para a execução do projeto.

Os propósitos do processo de GPR constam nas atividades a serem realizadas dentro do processo e serão vistas nas subseções que detalham cada fase e suas respectivas atividades.

3.3.1. Iniciação

Na fase de iniciação, são concedidas várias atribuições ao projeto. Esta é a fase da integração de todos os envolvidos no processo, das definições e da autorização para dar-se início a ele. A fase tem por finalidade alocar o time de desenvolvimento bem como o gerente de projetos que assumirá a responsabilidade pelo gerenciamento e controle das atividades.

Após a definição das atividades, é importante reunir a equipe técnica para planejar o cronograma e o plano de projeto. Os participantes deverão assinar a ata da reunião e assumir o compromisso correspondente às responsabilidades que lhes cabem no desenvolvimento do projeto.

3.3.2. Planejamento

Na fase de planejamento, é estabelecida uma análise apurada das tarefas e atividades e também é planejada a ação necessária para atingir os objetivos do escopo do projeto.

Nesta fase, também são detalhadas as informações feitas na iniciação e é criada uma base para as decisões que serão tomadas ao longo do projeto. O Gerente de Projetos é responsável por desempenhar o gerenciamento das atividades descritas no plano de projetos e verificar se elas estão sendo atendidas, garantindo, assim, um gerenciamento controlado das atividades. É necessário definir-se o tempo de execução de cada atividade e, com base nisso, estimar-se o custo. Nesta fase, é desenvolvido o cronograma e o plano de projeto, o qual consiste em informações referentes ao projeto, sendo elas documentadas para serem controladas e monitoradas.

Em seguida, é realizada uma análise do planejamento criado, geralmente outro Gerente de Projetos irá analisar se as atividades planejadas estão de acordo com o escopo e, assim, efetuar a validação do projeto. Fazendo isso, é realizada uma reunião com todos os envolvidos no projeto, na qual o Gerente de Projetos responsável irá informar o resultado das análises feitas e entregar o planejamento do projeto, certificando-se ainda de que todos tenham entendido sobre suas funções e responsabilidades dentro do projeto. O intuito principal da reunião é esclarecer certos pontos e entrosar a equipe responsável por executá-lo, motivando-a a cumprir os objetivos do projeto ciente das funções estabelecidas

3.3.3. Execução

Esta fase é a etapa mais delicada do processo, pois as mudanças começam a surgir quando chega o momento de colocar em prática tudo que foi planejado. Como foi visto anteriormente, na fase de planejamento, são definidas quais atividades serão realizadas, definições estas que servirão de base para a condução da fase de execução.

3.3.4. Acompanhamento e Controle

Esta fase ocorre paralelamente à fase de execução do projeto. Ela tem como objetivo acompanhar e controlar o que está sendo realizado no projeto e propor ações corretivas e preventivas no menor tempo possível logo após a detecção dos problemas.

3.3.5. Encerramento

A fase de Encerramento é a finalização das atividades concluídas do processo. O encerramento do projeto é, então, formalmente informado para todos os envolvidos, especialmente o cliente, o qual é comunicado de que o projeto está concluído e de que as novas solicitações serão atendidas num novo projeto. Em casos de cancelamento, serão detalhadas as razões da suspensão.

4. Conclusão

Com base em tudo o que foi explanado no presente artigo, conclui-se que o MPS.BR é um modelo que apóia fortemente a prática de melhoria contínua em todas as etapas presentes no processo de desenvolvimento de software. Ele oferece recursos que identificam e minimizam os riscos, proporcionando maior flexibilidade no atendimento dos requisitos a serem implementados. De forma sistemática, organiza todas as etapas bem como suas respectivas atividades presentes ao longo do processo de desenvolvimento, facilitando, assim, o planejamento, a execução, o controle e a avaliação contínua.

A principal contribuição deste artigo foi a definição de uma metodologia de melhoria voltada para as organizações que desenvolvem software. O MPS.BR é visto como um modelo adequado às empresas brasileiras que buscam a evolução contínua da qualidade de seus produtos, projetos e da capacitação dos recursos empregados no desenvolvimento do software produzido.

A partir dessa metodologia, as organizações que desejarem padronizar seus processos podem utilizar, como base, a metodologia desenvolvida neste trabalho e adequá-la para atender à realidade da organização conforme a necessidade dela.

5. Referências

CAROSIA, Jaciara S. **Levantamento da qualidade do Processo de Software com foco em pequenas organizações**. 2004, 158f. . Dissertação (Mestrado em Computação Aplicada) - Curso da Pós-Graduação em Computação Aplicada – Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos. 2004.

ESPINHA, Rafael; SOUZA, João. Melhorando Processos Através da Análise de Riscos e Conformidades. **Engenharia de Software Magazine: Qualidade de software: Entenda os principais conceitos sobre Testes e Inspeção de Software**. Brasil, n. 1, p.10-21, 2007. Disponível em: <<http://www.devmedia.com.br>>. Acesso em: 28 out. 2009.

ECLIPSE, **Process Framework Project (EPF) Composer**. Version 1.5.0.4. 2009. Disponível em: <<http://www.eclipse.org/epf/>>. Acesso em: 05 ago. de 2010.

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

SOFTEX, MPS.BR - **Melhoria de Processo do Software Brasileiro**, Guia de Implementação, Parte 1 - Fundamentação para implementação do Nível G do MR – MPS. Maio de 2009. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_1_2009.pdf>. Acesso em: 03 nov. de 2009.
KOSCIANSKI, André; SOARES Santos, Michel dos. **Qualidade de Software**. Novatec, 2006.