Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

UMA PERSPECTIVA DE MAPEAMENTO DO CMMI AO MPS.Br

Fábio Ferraz Pimentel

TRABALHO DE GRADUAÇÃO

Recife
05 de Outubro de 2006

Universidade Federal de Pernambuco
Centro de Informática

Fábio Ferraz Pimentel

UMA PERSPECTIVA DE MAPEAMENTO DO CMMI AO MPS.Br

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação

Orientador: Ph.D. Alexandre Marcos Lins de Vasconcelos
Co-orientador: M.Sc. Sandro Ronaldo Bezerra Oliveira

Recife
05 de Outubro de 2006

"A visão sem ação, não passa de um sonho.
A ação sem visão é só um passatempo.
A visão com ação pode mudar o mundo."
(Joel Baker)

AGRADECIMENTOS

Agradeço especialmente a meus pais e irmãos pelo amor e suporte dados durante toda minha trajetória. Agradeço ao prof. Alexandre Vasconcelos pelo apoio e orientação neste trabalho. Agradeço a Sandro Oliveira pela paciência, dedicação e co-orientação neste trabalho. Agradeço também a Tiago e demais colegas que compartilharam comigo as atividades do curso. Aos amigos da banda Lendários, do movimento de empresas juniores, do colégio, a Lipe e tantos outros que me ajudaram nos momentos de descontração e de stress. Deixo minha gratidão a Suzana pelo amor, carinho e compreensão. Por fim, a todos que não foram citados e que contribuíram direta ou indiretamente ao longo da minha graduação.

RESUMO

Este trabalho tem por objetivo definir um mapeamento entre os modelos CMMI (Capability Maturity Model Integration) e MPS.Br (Melhoria do Processo de Software Brasileiro) na sua versão 1.1. A perspectiva adotada visa atender ao ImPProS - Ambiente de Implementação Progressiva de Processo de Software - no sentido de relacionar um modelo de maturidade a um modelo de referência para permitir a rastreabilidade entre eles. Com o mapeamento definido, o ambiente é capaz de avaliar de forma automatizada a aderência de um processo definido em relação a um modelo especificado. O mapeamento realizado apresenta as atividades (sub-práticas) sugeridas pelo CMMI, na representação estagiada, para atingir os resultados esperados dos dois primeiros níveis de maturidade do MPS.Br.

Palavras-chaves: qualidade de software, processo de software, aderência entre modelos de qualidade, CMMI, MPS.Br.

ABSTRACT

This work defines a mapping between the CMMI (Capability Maturity Model Integration) and MPS.Br (Brazilian Software Process Improvement, version 1.1) models. The adopted perspective intend to work with ImPProS - Gradual Software Process Implementation Environment - to relate a maturity model with a reference model to provide rastreability between those. This perspective assures the environment can evaluate automatically adherence process to a specified model. The provided mapping presents activities (subpractices) suggested by CMMI, in the staged representation, to achieve wanted results of first and second MPS.Br maturity levels.

Keywords: software quality, software process, quality process adherence, CMMI, MPS.Br

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo inicial, é feita uma abordagem geral sobre o tema tratado ao longo deste trabalho. Será conhecido o contexto no qual o mapeamento proposto entre modelos de qualidade de software está inserido, a motivação para a proposição do tema, os objetivos a serem alcançados, a metodologia de trabalho utilizada ao longo do trabalho e a estrutura deste documento.

1.1. CONTEXTO

As organizações cada vez mais buscam a definição e melhoria contínua de seus processos, preocupadas com a qualidade dos softwares produzidos. Para colaborar com esta definição, foram desenvolvidos modelos de qualidade de software, permitindo que tais organizações sejam guiadas pelas melhores práticas no desenvolvimento e manutenção de softwares. Portanto, as organizações podem definir um processo padrão, apoiadas em um ou mais modelos consolidados, e moldá-los à sua realidade.

Comumente, a definição de um processo ocorre paralelamente à busca de alguma certificação em um modelo de qualidade. O processo padrão da organização, além de se adequar à cultura organizacional, deve ser aderente ao modelo para que haja uma avaliação positiva. No entanto, está definição costuma ser demorada o que acaba desmotivando funcionários que não visualizam rapidamente os ganhos de um processo estabelecido. Um grande desafio enfrentado, portanto, é conseguir dinamizar a definição e melhoria do processo que atenda a um modelo.

Além das normas, os modelos a serem seguidos para a definição de um processo padrão podem ser de dois tipos: de maturidade e de referência. Nos

modelos de maturidade, temos objetivos que devem ser cumpridos e, em um nível mais concreto, atividades a serem realizadas. Quando o modelo de referência é utilizado, não há qualquer especificação das atividades a serem executadas, mas apenas os resultados que devem ser obtidos. Se por um lado há um ganho de flexibilidade na definição do processo, por outro, a organização pode perder por não ter auxílio na estruturação das atividades de seu processo.

1.2. MOTIVAÇÃO

Uma das idéias utilizadas para a construção do mapeamento é sugerir atividades para as organizações que estão definindo um processo aderente a um modelo de referência. Como o modelo MPS.Br (Melhoria do Processo de Software Brasileiro) é o adotado neste trabalho, a organização que deseja implementá-lo pode ter o auxílio das atividades sugeridas pelo CMMI (Capability Maturity Model), através do mapeamento proposto, para definir um processo realmente aderente.

Além disso, o ImPProS, ambiente de implementação progressiva de processo de software, possui um componente em sua estrutura chamado meta-modelo (a ser descrito no capítulo 4) que permite relacionar modelos de maturidade, de referência e normas de qualidade, para auxiliar a implementação de um processo na organização. Portanto, um mapeamento entre modelos de qualidade focado neste ambiente, proporcionaria um apoio automatizado na definição de processos e avaliação da aderência dos mesmos.

1.3. OBJETIVOS

A perspectiva de mapeamento proposto, portanto, tem como objetivo o auxílio a organizações que pretendem implementar um processo seguindo o MPS.Br e que desejam se beneficiar das atividades sugeridas pelo CMMI. Além disso, o mapeamento também tem o foco no ImPProS, contribuindo para que o ambiente automatizado integre os modelos de qualidade, permitindo que a

organização aproveite os elementos destes para definir e avaliar a aderência de seus processos.

1.4. METODOLOGIA DE TRABALHO

Para obter a perspectiva de mapeamento apresentada neste trabalho, foi feito um estudo dos modelos, identificando características comuns que pudessem servir para associar os modelos escolhidos. Dessa forma, a estrutura de cada um dos modelos foi compreendida e a partir deste ponto, o mapeamento foi planejado. Além disso, foram consideradas as características de implementação, permitindo um entendimento mais detalhado das práticas e resultados dos modelos, e sendo útil para mapear estes modelos de forma mais ampla, apoiando-se em recomendações não documentadas, cobradas durante a avaliação do MPS.Br.

Após este edtudo, foi analisado o ambiente ImPProS, para compreender uma possível aplicação para o mapeamento pretendido. O ambiente mostrou-se ideal para utilizar o resultado deste trabalho e tornou-se o foco da perspectiva adotada.

Por fim, o mapeamento entre práticas específicas e resultados esperados foi efetivamente realizado para cada um dos processos dos dois primeiros níveis do MPS.Br. Com este mapeamento, foi possível encontrar as atividades sugeridas pelo CMMI para a implementação de um processo de nível F no modelo brasileiro.

1.5. ESTRUTRA DO TRABALHO

O presente trabalho foi estruturado da seguinte forma:

- O capítulo 1 introduz o documento, apresentando contexto, motivação e objetivos deste trabalho, metodologia utilizada e estrutura do documento;
- O capítulo 2 aborda o processo de software, sua definição e melhoria, baseado em um modelo que estrutura um processo padrão, processos especializados e processos instanciados em uma organização. Ainda

são apresentadas ferramentas automatizadas nesta área e as principais normas e modelos de qualidade de software;

- O capítulo 3 descreve brevemente a tecnologia de processo de software, os ambientes de desenvolvimento centrados no processo, e por fim, o ImPProS;
- O capítulo 4 apresenta a estrutura do ImPProS, com destaque para o meta-modelo, componente no qual o mapeamento será inserido;
- O capítulo 5 explica a estruturas dos 2 modelos adotados para o trabalho e apresenta o mapeamento entre eles;
- O capítulo 6 encerra o trabalho, apresentando as conclusões obtidas e os possíveis trabalhos futuros para o tema.

CAPÍTULO 2

DEFINIÇÃO E MELHORIA DO PROCESSO DE SOFTWARE

A definição, utilização e melhoria contínua de um processo de software correspondem aos principais objetivos de uma organização de software. Esse esforço geralmente considera apenas métodos e práticas da engenharia de software, sem contemplar suficientemente as restrições do ambiente de trabalho ou o conhecimento e a experiência das equipes de software. Ao definir um processo de software adequado a uma organização, é também importante ponderar as características peculiares da própria empresa e de seus grupos de trabalho. A alta rotatividade de pessoal, as equipes geograficamente distantes, a falta de experiência sobre o domínio do conhecimento da aplicação não correspondem a dificuldades técnicas, mas podem determinar o sucesso ou o fracasso de um projeto de software.

Este capítulo abordará a definição de um processo de software, experiências automatizadas nesta área, melhoria contínua de processo e por fim, normas e modelos para a definição e melhoria dos processos de software.

2.1. DEFINIÇÃO DE PROCESSO DE SOFTWARE

O processo de software é definido por [Humphrey.89] como o conjunto de tarefas de engenharia de software necessárias para transformar os requisitos dos usuários em software. Na definição de um processo de software devem ser consideradas as seguintes informações: atividades a serem realizadas, recursos utilizados, artefatos consumidos e gerados, procedimentos adotados, paradigma e tecnologia adotados, e o modelo de ciclo de vida utilizado ([Falbo.98]).

Os processos de software podem apresentar grande complexidade e possibilitar diversas alternativas de execução de suas atividades. Desta forma, um processo de software definido permite que profissionais de engenharia de software possam trabalhar de forma ordenada, possibilitando um melhor entendimento do seu trabalho, bem como de outras atividades executadas por outros membros da mesma equipe [Humphrey.89]. No entanto, não existe um processo de software que possa ser genericamente aplicado a diversos projetos, visto que nenhum projeto é idêntico ao outro. Variações nas políticas e procedimentos organizacionais, métodos e estratégias de aquisição, tamanho e complexidade do projeto, requisitos e métodos de desenvolvimento do sistema, entre outros fatores, influenciam na forma como um produto de software é adquirido, desenvolvido, operado e mantido [ISO.97].

2.1.1. Processo Padrão

Em uma organização ou grupo de desenvolvimento multi-organizacional, diversos projetos podem coexistir possuindo características específicas. Porém, existe um conjunto de elementos fundamentais os quais deseja-se que sejam incorporados em qualquer processo definido. A [ISO.98] define o conjunto destes elementos fundamentais como o processo padrão, ou seja, o processo básico que guia o estabelecimento de um processo comum na organização. Desta forma, um processo padrão define uma estrutura única a ser seguida por todas as equipes envolvidas em um projeto de software [Maidantchik.99], independente das características do software a ser desenvolvido. [Humphrey.89] define um conjunto de razões para a definição de um processo padrão:

- redução dos problemas relacionados a treinamento, revisões e suporte a ferramentas;
- as experiências adquiridas nos projetos são incorporadas ao processo padrão e contribuem para melhorias em todos os processos definidos;
- a utilização de padrões de processo, fornecendo as bases para medições de processos e qualidade;

- economia de tempo e esforço em definir novos processos adequados a projetos.

Na literatura atual, observa-se uma tendência à utilização de processos padrões na definição de processos. A norma ISO/IEC 12207 [ISO.97], o SPICE [ISO.98], o CMM [Paulk.93], o CMMI [Chrissis.03] e o MPS.Br [Softex.06], definem um processo padrão como um ponto base a partir do qual um processo especializado poderá ser obtido de acordo com as características de um projeto de software específico. A referida norma, em seus anexos A e B, descreve procedimentos a serem utilizados na especialização do seu modelo de processo. [Jacobson.98] define um template para processo que pode ser especializado em instâncias de processos para atender às necessidades das organizações e de projetos específicos. Desta forma, ser adaptável e configurável torna-se um importante requisito a ser atingido na definição de um processo padrão.

2.1.2. Modelo para Definição de Processo de Software

À medida que aumenta a preocupação com a qualidade de software, aumenta a procuração das organizações em seguir as orientações de modelos e os padrões de qualidade de processo, tais como normas ISO 9000-3 [ISO.91], CMMI [Chrissis.03], ISO/IEC 12207 [ISO.97] e SPICE (ISO/IEC TR 15504) [ISO.98]. Contudo, a definição de processos de software envolve várias e complexas facetas e são poucos os profissionais qualificados para realizar essa tarefa. De fato, para profissionais menos experientes, essa é uma atividade que requer alguma forma de ajuda, e o apoio automatizado é uma maneira de oferecer essa ajuda.

Variações nas políticas e procedimentos organizacionais, nos métodos e estratégias de aquisição, no tamanho e complexidade do projeto, nos requisitos e métodos de desenvolvimento do sistema, entre outros fatores, influenciam a forma como um produto de software é adquirido, desenvolvido, operado e mantido [ISO.97]. Assim, os processos devem ser definidos caso a caso, levando-se em consideração as particularidades de cada organização e/ou projeto. A definição de processo de software deve, portanto, considerar a sua

adequação às tecnologias envolvidas, ao tipo de software em questão, ao domínio de aplicação, ao grau de maturidade (ou capacitação) da equipe em engenharia de software, às características próprias da organização, às características do projeto e da equipe.

O modelo para definição de processos de software apresentado nesta seção, foi extraído de [Rocha.01]. Este modelo, baseado na experiência de definição de processo de software para diferentes domínios de aplicação e usando diferentes tecnologias de desenvolvimento de software, consiste de três etapas: definição do processo padrão, especialização do processo padrão e instanciação para projetos específicos. Resultam como produtos de tais etapas, processos de software em diferentes níveis de abstração.

A figura a seguir representa esquematicamente o modelo proposto, no qual são apresentadas as etapas envolvidas na definição de um processo de software para um projeto a partir de um processo padrão e os produtos intermediários obtidos, além de alocar os fatores que influenciam a definição do processo nos diferentes níveis de abstração.

Em uma organização, diferentes processos podem coexistir adequados a diferentes projetos. Para organizar e disciplinar o desenvolvimento de software é importante determinar as atividades fundamentais que deverão estar presentes em qualquer processo definido. A definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software e constitui a base para a definição de todos os processos [Rocha.01]. Dessa forma, estabelece-se um processo básico que servirá como ponto de partida para a posterior definição dos processos de software adequados às diferentes características de cada projeto, permitindo economia de tempo e esforço na definição de novos processos.

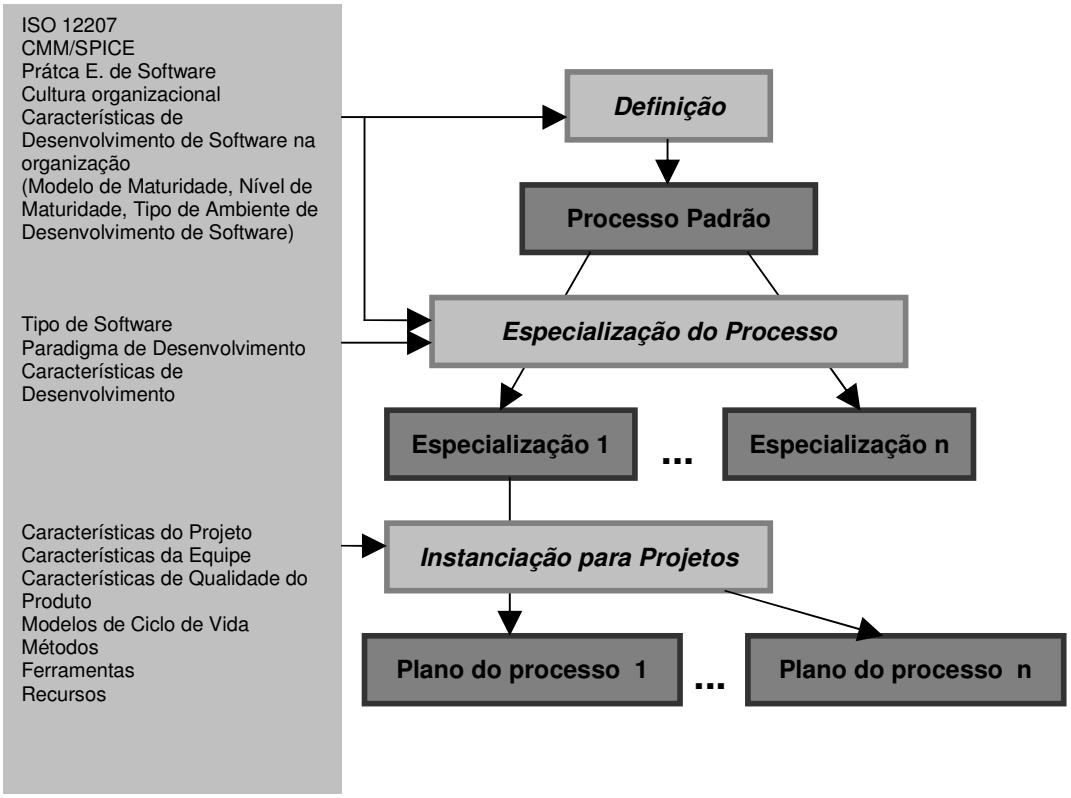

Figura 1 - Modelo para a definição de processos de software [Rocha.01]

Nesse modelo, a definição do processo padrão pode ser realizada tendo como base a norma ISO/IEC 12207 e as características do desenvolvimento de software na organização. A definição poderá também considerar um dos modelos de maturidade atualmente utilizados, tais como CMM e SPICE, associado à capacitação da organização.

Tendo em vista que tipos diferentes de software possuem características distintas e requerem diferentes abordagens de desenvolvimento, o processo de software padrão da organização deverá ser adaptado (especializado) considerando-se as características relacionadas ao tipo de software (por exemplo, sistemas de informação, sistema de missão crítica, etc.) e ao paradigma de desenvolvimento utilizado (por exemplo, orientação a objetos, lógico, etc.). Assim, durante a etapa de especialização do processo padrão, atividades poderão ser adicionadas ou modificadas, de acordo com o contexto para o qual se está realizando a especialização [Rocha.01].

A instanciação para projetos específicos consiste na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades.

Nesta etapa, são definidos o modelo de ciclo de vida, os métodos e as ferramentas que serão utilizadas no projeto, os recursos humanos e suas responsabilidades ao longo do processo e os artefatos (produtos) consumidos e gerados. A norma ISO/IEC 9126 [ISO.02] é utilizada para identificar os requisitos de qualidade do produto. Tais características irão influenciar o processo no que se referem às atividades, métodos e técnicas. As atividades do processo especializado deverão ser adaptadas ao modelo de ciclo de vida escolhido para o projeto e novas atividades poderão ser inseridas a um determinado ciclo de vida.

Embora esta seção tenha dado enfoque ao modelo definido por [Rocha.01], podemos encontrar na literatura especializada alguns outros modelos de definição do processo de software. O fato de que processos de software devem ser adaptados para os ambientes específicos nos quais serão aplicados, de forma a serem aceitos e terem seu uso maximizado, é bem conhecido. Basili et al [Basili.87], apresentam uma metodologia para melhoria do processo de software, através de sua adaptação para as metas de um ambiente e projeto específicos. A metodologia proposta é suportada pela ferramenta denominada TAME (Tailoring A Measurement Environment).

Posteriormente, trabalhos foram feitos no sentido de determinar o grau de adequação entre um processo de software e o ambiente no qual ele é usado. Pérez et al [Pérez.96] descrevem um sistema para avaliar tal adequação, baseado em um modelo de contingência, derivado empiricamente e consistindo dos atributos de um modelo de processo de software e de seu ambiente, e dos relacionamentos entre eles.

O desenvolvimento de uma instância de processo de software que é especificamente adaptada para o seu ambiente é uma tarefa difícil, em virtude dos diferentes tipos de informação que devem ser agregados e das alterações que devem ser realizadas no processo de software original. Conseqüentemente, este problema raramente tem sido verdadeiramente resolvido. Ao invés disso, grande parte das descrições de processos de software são representadas em um alto nível de abstração, de forma a poderem assumir diferentes variantes.

2.1.3. Experiências Automatizadas

Na literatura, é possível encontrar registros de experiências de automatização de processos. Tanto no meio acadêmico como no mercado, as ferramentas estão obtendo sucesso no auxílio à adaptação de processos definidos.

Configurando-se como uma notável exceção à regra geral, o processo padrão do governo alemão “Das Vorgehensmodell” (ou V-Model) oferece meios explícitos para ser adaptado para projetos específicos. Entretanto, a adaptação é limitada à remoção de elementos do processo padrão, o que restringe demais as possibilidades de customização. A ferramenta ProcePT (Process Programming and Testing), apresentada em [Welzel.95], automatiza a adaptação do V-Model para diferentes projetos, através das regras de deleção de atividades e produtos do processo. Assim como o V-model, a norma ISO/IEC 12207 e o framework RUP (Rational Unified Process) também são adaptáveis para diferentes organizações e projetos de software, sendo que a adaptação do RUP é suportada pela ferramenta RUP-Builder.

Nos ambientes de pesquisa, métodos mais sofisticados para a adaptação de processos de software têm sido desenvolvidos, tais como a ferramenta ProTail, que suporta adaptação semi-automática de modelos de processo usando regras de adaptação [Münch.97]. Todavia, as regras definidas por esta ferramenta aplicam-se apenas ao V-Model, o que lhe atribui uma aplicabilidade restrita e pouca flexibilidade.

Abordagens mais recentes têm dado bastante destaque ao papel do conhecimento no processo de adaptação, utilizando técnicas de Inteligência Artificial para representar, armazenar, recuperar e utilizar corretamente esse conhecimento. Berger, em [Berger.03], define um método de instanciação de processos de software apoiado pela ferramenta AdaptPro, através da qual é disponibilizado ao gerente de projetos (pessoa responsável por gerenciar o andamento de determinado projeto de software) o conhecimento sobre instanciação de processos de software acumulado pela organização de software em projetos anteriores. Esta abordagem fundamenta-se nos conceitos de gerência de conhecimento e de ambientes de desenvolvimento de software orientados à organização.

Rupprecht et al, em [Rupprecht.00], apresentam uma abordagem fundamentada em um kit de construção de processos de software, que consiste de um repositório para o gerenciamento de blocos básicos para construção de processos de software e um vetor de operadores para adaptar tais blocos. Este kit foca no projeto conceitual de processos de software em termos de ontologias, enquanto ao mesmo tempo oferece suporte automatizado para adaptação de processos de software.

Henninger, em [Henninger.98], apresenta uma abordagem que utiliza um sistema baseado em regras, organizadas sob a forma de árvores de decisão, para adaptar processos de software para as necessidades específicas de projetos individuais e usa técnicas de aprendizagem organizacional para modificar processos de software, de forma a atender às necessidades da organização de software. Esta abordagem é suportada pela ferramenta GUIDE, uma aplicação Web que utiliza o repositório baseado em casos BORE para capturar experiências em projetos anteriores.

Vale destacar ainda a abordagem bastante recente, apresentada em [Ahn.03], que defende a adaptação de processos de software baseada em características do projeto a ser conduzido, experiências anteriores de adaptação e iniciativas de melhoria. Esta abordagem utiliza a combinação de inferência baseada em conhecimento com raciocínio baseado em casos para adaptar processos de software, em virtude dos diferentes tipos de conhecimento envolvidos na adaptação.

Coelho, por sua vez, em [Coelho.03], apresenta o MAPS (Modelo de Adaptação de Processos de Software), o qual tem por objetivo adaptar o processo padrão de uma organização para os projetos conduzidos na mesma, baseado nas características desses projetos e em adaptações anteriores, utilizando uma abordagem orientada a artefatos. Soluções baseadas em políticas de instanciação, tal como descrito no modelo APSEE [Reis.03], adotam uma postura diferente por fornecer solução automatizada para a escolha de agentes e recursos a partir de critérios genéricos definidos a priori.

Segundo Reis, em [Reis.02], idealmente a adaptação de processos de software pode se valer de soluções baseadas em conhecimento, com o objetivo de fornecer sugestões de adaptações em função de soluções semelhantes

adotadas em situações similares, levando em consideração as características ambientais e organizacionais correntes.

2.2. MELHORIA DE PROCESSO DE SOFTWARE

Com o intuito de obter níveis de qualidade adequados no desenvolvimento de software, houve uma mudança no mercado de TI, principalmente na última década, onde o enfoque passou a ser no processo de software. Tem-se, então, uma nova abordagem na qual o foco principal das atenções está na garantia da qualidade do próprio processo produtivo, visto que este tem se mostrado um fator determinante para o alcance da qualidade do produto final.

A partir desta mudança de foco, intensificou-se a pesquisa sobre o processo de desenvolvimento e várias normas e padrões foram definidos a fim de auxiliar na definição e melhoria de processos de software. Com a intensificação dos estudos, constatou-se que, para alcançar níveis cada vez mais altos de qualidade, era necessário melhorar cada passo do ciclo de vida de desenvolvimento. Porém, para que isso se tornasse possível, dados quantitativos, que pudessem descrever a realidade do processo, precisavam ser obtidos e devidamente analisados.

A melhoria de processo convoca à análise dos resultados e a propor sempre uma maneira mais efetiva e eficiente de se trabalhar [Oliveira.05]. Os principais objetivos a serem alcançados são:

- Melhorar o entendimento sobre o processo, produto, recursos e ambiente de desenvolvimento e, assim, estabelecer bases para comparação entre medições;
- Avaliar o andamento do projeto comparando com dados planejados;
- Fazer previsões sobre o futuro andamento do projeto baseado em comportamentos passados;
- Promover melhorias identificando falhas, ineficiências e outras oportunidades para melhorar a qualidade do produto e o desempenho do processo.

Conradi e Fuggetta em [Conradi.02] apresentam 6 teses em como aumentar e melhor aplicar *frameworks* de melhoria contínua. A descrição de cada uma delas se encontra a seguir:

Tese 1: Técnicas de melhoria devem suportar estratégias que foquem na orientação a objetivos e inovação do produto

- Fazer com que a iniciativa de melhoria esteja consistente com os objetivos e estratégias de negócios para garantir que terão os efeitos desejados na performance da empresa;
- Começar com atividades de melhoria que lidem com as necessidades mais imediatas (estimativas, inspeções e desenvolvimento incremental);
- Definir objetivos realistas e visíveis de curto e longo prazo.

Tese 2: Desenvolvedores são motivados para mudanças; se possível, começar *bottom-up* com iniciativas concretas

- Usar um cartão de marcação (para estimativas, tipo *checklist*) simples e focado para começar, e não uma avaliação ampla;
- Contar com e aumentar a participação dos desenvolvedores no aprendizado;
- Estabelecer feedback das iniciativas de melhorias e fazer com que elas sejam visíveis para o alto escalão de gerenciamento.

Tese 3: Suporte automatizado a processos de software tem sido enfatizado de forma exagerada

- Apenas processos estáveis - para inspeção, testes ou gerência de configuração - são bem indicados para suporte de tecnologias de processo de software, especialmente com a ajuda do computador;
- Uma ferramenta processo de software deve se adaptar para as necessidades específicas da aplicação; construir uma ferramenta avançada para uma aplicação errada é desperdício técnico.

Tese 4: Análises de custo-benefício requer modelos de novos modelos de amortização

- Desenvolver modelos de custo-benefício modernos e incrementais (Cocomo II [BOEHM 2000]);
- Aspirar por um modelo de custo-benefício comum a toda companhia;
- Executar um esforço (experimento) de melhoria de cada vez, possivelmente baseados em projetos com *baselines*;
- Avisar à gerência que resultados reprodutíveis e convincentes virão com o tempo.

Tese 5: Melhorias assumem mudanças culturais, então é necessário algum conhecedor de ciências sociais

- Utilizar equipes multidisciplinares;
- Estabelecer engajamento participativo em todas as mudanças no processo e atividades associadas;
- Enfatizar o lado cultural, de aprendizado, e dimensões de longo-prazo do trabalho de melhoria contínua;
- Começar humildemente com passos curtos.

Tese 6: Melhoria contínua é aprendizado, não controle, como em QA

- Criar uma equipe de melhoria em separado do departamento de qualidade, ou melhor ainda, combinar SPI e QA num único time;
- Preparar estudos empíricos em como as pessoas trabalham na verdade;
- Fazer um sistema de recompensas para reportagem de problemas ou idéias de sugestões de melhoria.

2.3. MODELOS E NORMAS DE QUALIDADE PARA PROCESSO DE SOFTWARE

Alguns dos modelos e normas existentes ganharam destaque devido à sua qualidade e aplicação. Estes modelos foram amplamente difundidos no

mundo, sendo o último, exclusivamente brasileiro. A seguir, serão descritos de forma sucinta os modelos ISO/IEC 12207, ISO/IEC 15504 (SPICE), CMMI e MPS.Br.

2.3.1. ISO/IEC 12207

Em 1988, foi proposto o desenvolvimento da Norma ISO/IEC 12207 (NBR ISO/IEC 12207, 1998) dentro de um esforço conjunto da ISO – International Organization for Standardization e do IEC – International Electrotechnical Commission em estabelecer uma estrutura comum para os processos de ciclo de vida de software como forma de ajudar as organizações a compreenderem todos os componentes presentes na aquisição e fornecimento de software e, assim, conseguirem firmar contratos e executarem projetos de forma mais eficaz. A Norma foi publicada internacionalmente em 1995 e no Brasil em 1998.

Em 2002, foi feita uma atualização na norma ISO/IEC 12207 (ISO/IEC PDAM 12207, 2002) em forma de anexo que visava representar a evolução da engenharia de software, as necessidades vivenciadas pelos usuários da norma e a harmonização com a série de normas ISO/IEC 15504 – Avaliação de Processos de Software. Essa atualização inseriu processos e acrescentou na sua descrição propósito e resultados de implementação o que possibilita a avaliação da capacidade do processo. A norma, incluindo o seu anexo, é composta por 22 processos, 95 atividades, 325 tarefas e 254 resultados. Todos esses processos, executados durante o projeto de software, conduzem à qualidade tanto do produto quanto do processo. Entretanto, a norma deixa para a organização definir “como” os processos serão executados conservando dessa forma a flexibilidade necessária para que os países e as organizações a implementem de acordo com a cultura local e a tecnologia empregada.

A ISO/IEC 12207 tem sido amplamente utilizada em muitos países como referência para contratação de serviços de desenvolvimento e manutenção de software. No Brasil, muitas organizações têm tomado conhecimento da existência da norma e algumas já a utilizam para definição de processos de desenvolvimento de software. Muitos trabalhos de pesquisa têm utilizado a norma o que vislumbra uma ampla utilização da mesma no futuro.

2.3.2. ISO/IEC 15504 (SPICE)

A ISO/IEC 15504 nasceu em janeiro de 1993 quando a ISO iniciou o projeto SPICE (Software Process Improvement and Capability dEtermination) com o objetivo de produzir inicialmente um Relatório Técnico que fosse, ao mesmo tempo, mais geral e abrangente que os modelos existentes e mais específico que a norma ISO 9001 [Salviano.01]. Uma versão do SPICE foi aprovada em 1998 como Relatório Técnico e, em 2003, foi publicada a Norma ISO/IEC 15504 [ISO.03].

A ISO/IEC 15504 (SPICE) presta-se à realização de avaliações de processos de software com dois objetivos: a melhoria de processos e a determinação da capacidade de processos de uma organização. Se o objetivo for a melhoria de processos, a organização pode realizar a avaliação gerando um perfil dos processos que será usado para a elaboração de um plano de melhorias. A análise dos resultados identifica os pontos fortes, os pontos fracos e os riscos inerentes aos processos. No segundo caso, a empresa tem o objetivo de avaliar um fornecedor em potencial, obtendo o seu perfil de capacidade. O perfil de capacidade permite ao contratante estimar o risco associado à contratação daquele fornecedor em potencial para auxiliar na tomada de decisão de contratá-lo ou não [Salviano.01], [ISO.03]

2.3.3. CMMI

O CMMI é um framework que possui os elementos necessários para tornar um processo de desenvolvimento de software mais eficiente e controlado [Bartié.02] e foi construído para integrar os diferentes modelos criados a partir do sucesso alcançado pelo CMM para software.

Existem dois tipos de representação no CMMI: em estágios e contínua. Tem-se, assim, um único modelo que pode ser visto de duas perspectivas distintas. A representação em estágios é a representação usada no SW-CMM. Esta representação define um conjunto de áreas de processo para definir um caminho de melhoria para a organização, descrito em termos de níveis de maturidade. A representação contínua é o enfoque utilizado no SECM, no IPD-

CMM e também no SPICE. Este enfoque permite que uma organização selecione uma área de processo específica e melhore com relação a esta área.

O CMMI fornece um método de avaliação rigoroso para benchmarking, chamado SCAMPI, que implementa os sete princípios das avaliações CMMI [Chrissis.03]: patrocínio da gerência senior, foco nos objetivos de negócio da organização, confidencialidade das entrevistas, uso de um método de avaliação documentado, uso de um modelo de referência de processo como base (CMMI), enfoque de equipe colaborativa, e, foco em ações para melhoria do processo. O método de avaliação SCAMPI está em conformidade com o método de avaliação da ISO/IEC 15504 (SPICE).

2.3.4. MPS.Br

O modelo MPS.Br – Melhoria do Processo de Software Brasileiro [Softex.06] é um modelo brasileiro que vem sendo desenvolvido desde de Dezembro de 2003 pelo projeto MPS.Br, o qual é uma iniciativa envolvendo universidades, grupos de pesquisa e empresas, sob a coordenação da Sociedade SOFTEX (Sociedade para Promoção da Excelência do Software Brasileiro). O projeto MPS.Br visa promover a qualificação de um grupo amplo de empresas compatível com os padrões de qualidade aceitos internacionalmente pela comunidade de software, a custos acessíveis, sendo adequado ao perfil e cultura da grande maioria das empresas brasileiras.

O MPS.Br tem como objetivo definir um modelo de melhoria e avaliação de processo de software, preferencialmente para as micro, pequenas e médias empresas, de forma a atender as suas necessidades de negócio e a ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software. O MPS.Br também define regras para sua implementação e avaliação, dando sustentação e garantia que o MPS.Br está sendo empregado de forma coerente com suas definições. O MPS.Br baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de produtos de software e serviços correlatos.

O programa mobilizador MPS.BR está dividido em três (3) componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de

Negócio (MN-MPS). Cada componente é descrito por meio de Guias e/ou de Documentos do MPS.BR.

CAPÍTULO 3

IMPPROS – UM AMBIENTE DE IMPLEMENTAÇÃO PROGRESSIVA DE PROCESSO DE SOFTWARE

Para ajudar uma organização na implementação progressiva de um processo de software, é útil fornecer apoio automatizado por meio de um ambiente capaz de suportar as fases (Definição, Simulação, Execução e Avaliação) que a literatura especializada propõe como necessárias. O termo "progressiva" decorre do fato de que a implementação do processo é aperfeiçoada com as experiências aprendidas na sua definição, simulação, execução e avaliação.

Esse capítulo descreve brevemente a tecnologia de processo de software e os ambientes de desenvolvimento centrados no processo. Por fim, é apresentado o ImPProS, um ambiente de implementação progressiva de processos de software que serviu com base para este trabalho.

3.1. TECNOLOGIA DE PROCESSO DE SOFTWARE

Segundo Carla Reis [Reis.98], um processo de software é o conjunto de todas as atividades necessárias para criar um produto de software a partir dos requisitos de um usuário.

Um processo é organizado em etapas parcialmente ordenadas, que estão relacionadas a artefatos, pessoas, recursos, estruturas organizacionais e restrições.

As etapas de execução de um processo podem ser classificadas como atividades ou tarefas. As primeiras são etapas bem gerenciadas, já as ultimas são etapas elementares, que combinadas levam à execução de uma atividade.

As atividades possuem como objetivo criar ou alterar um conjunto de artefatos. Elas podem ainda possuir relacionamentos entre si, estando

associadas também a papéis e ferramentas. Atividades podem ser executas por agentes humanos, com o apoio de ferramentas, ou executadas sem intervenção humana, de maneira totalmente automatizada.

Artefatos são quaisquer produtos criados ou modificados durante a execução do processo. Artefatos podem servir como requisito para a execução de uma tarefa, ou como resultado da execução desta.

Um modelo de processo de software é uma descrição abstrata do processo de software. Vários tipos de informação devem ser integradas em um modelo de processo de software para indicar quem, quando, onde, como e por que os passos são realizados.

Um modelo do processo instanciado ou processo executável é um modelo de processo pronto para execução. Por sua vez, um projeto, segundo [18], é a instância de um processo, com objetivos e restrições específicos.

3.2. AMBIENTES DE DESENVOLVIMENTO DE SOFTWARE CENTRADOS NO PROCESSO

O surgimento da tecnologia CASE (Computer Aided Software Engineering) – Engenharia de Software Auxiliada por Computador, provocou um grande impacto na área de engenharia de software. Surgiram várias ferramentas capazes de auxiliar a execução de processo de produção de software. Como uma evolução dessa tecnologia, surgiu o conceito de ADS (Ambiente de Desenvolvimento de Software). Esse novo conceito baseia-se na integração das ferramentas de apoio, fornecendo um suporte contínuo em todas as etapas do ciclo de vida de um processo de desenvolvimento.

O principal objetivo de um ADS é prover um ambiente no qual produtos de software de grande porte possam ser desenvolvidos através da integração de um conjunto de ferramentas que suportam métodos de desenvolvimento, apoiados por uma estrutura que permite a comunicação e cooperação entre as ferramentas [Reis.00a]. O conceito de ADS é considerado bem mais amplo que o de ferramentas CASE por prover uma estrutura unificadora de serviços onde várias ferramentas podem ser integradas.

Uma evolução significativa nos ADS foi conseguida com a tecnologia de processos de software. A automação do processo de software foi incorporada aos ADSs mais recentes tornando-os ADS centrados em processo (ou orientados a processo), também conhecido na literatura como PSEE - Process-Centered Software Engineering Environment. Estes ambientes constituem uma nova geração de ADS que suportam além da função de desenvolvimento de software, também as funções associadas à gerência e garantia da qualidade durante o ciclo de vida do software. Um ADS centrado em processo baseia-se em uma definição explícita do processo de desenvolvimento de software. Por isso o processo de software utilizado na organização deve estar formalizado e ser obedecido.

Segundo [Reis.00b], uma das implicações da utilização de ambientes orientados a processos é a necessidade de definir a execução do processo de forma mais rigorosa. Isso permite uma melhor comunicação entre os membros da organização no que diz respeito ao processo, permitindo que pessoas que desconheçam o processo tenham uma maior facilidade em seu aprendizado. Mesmo os desenvolvedores mais experientes, podem ter seu trabalho facilitado uma vez que, durante a execução do processo, o ambiente provê informações que podem guiá-lo em suas atividades. Além disso, o ambiente pode realizar algumas atividades mais simples de forma automática, permitindo que os desenvolvedores ocupem seu tempo em outras atividades, aumentando assim a produtividade da organização. Pode-se ainda, configurar o ambiente para armazenar informações sobre o desenvolvimento, tais como métricas do processo, que podem ser consultadas, quando necessário.

Outra grande vantagem da utilização de um ADS centrado no processo é a possibilidade de reunir as definições do processo em uma biblioteca reusável. Assim, pode haver a padronização de um processo organizacional, que é adaptado para cada projeto específico.

Esta característica faz com que a organização não somente economize em recursos, mas também possa atingir o nível 3 de maturidade do modelo CMM, descrito em [Humphrey.89]. Outro recurso disponível é a coleta automática de métricas. Segundo [Reis.00a], os ambientes centrados no processo devem oferecer as seguintes funcionalidades:

- Suporte a múltiplos usuários concorrentemente;
- Prover um módulo responsável por controlar o acesso e a evolução de objetos compartilhados
- Apoio à edição cooperativa de documentos e itens de software de forma gerenciada;
- Suporte à modelagem e execução de processos de software;
- Permitir extensão do ambiente através da inclusão de novas ferramentas;
- Integração dos módulos em todos os níveis.

3.3. IMPLEMENTAÇÃO DE PROCESSO DE SOFTWARE

A mudança no processo de desenvolvimento de software em uma organização não é um processo trivial e pode levar algum tempo até que os resultados sejam percebidos. Esta mudança de um processo de software é um procedimento muito mais complexo do que a utilização de uma nova ferramenta de desenvolvimento [Balduino.02]. Para utilizar uma nova ferramenta, basta aprender a instalá-la e entender as instruções de como operá-la. Já a implantação de um novo processo de desenvolvimento afeta a maneira como os indivíduos trabalham, como eles vêem, e dão valor ao resultado de seu trabalho. Tal mudança afeta os indivíduos e a empresa muito mais profundamente do que a mudança de tecnologia ou ferramentas. Portanto, os resultados dessa mudança só são percebidos a longo prazo, o que reforça a necessidade de que essa mudança seja cuidadosamente planejada e gerenciada.

Uma abordagem de adoção gradual do processo de desenvolvimento e ferramentas de apoio, onde cada passo seja planejado, executado e avaliado com critério, permite que a implantação do processo seja validada em etapas, permitindo a correção de erros e minimizando os riscos associados.

Um projeto de implementação de um processo de software pode ser dividido em quatro etapas, conforme ilustrado na Figura 2:

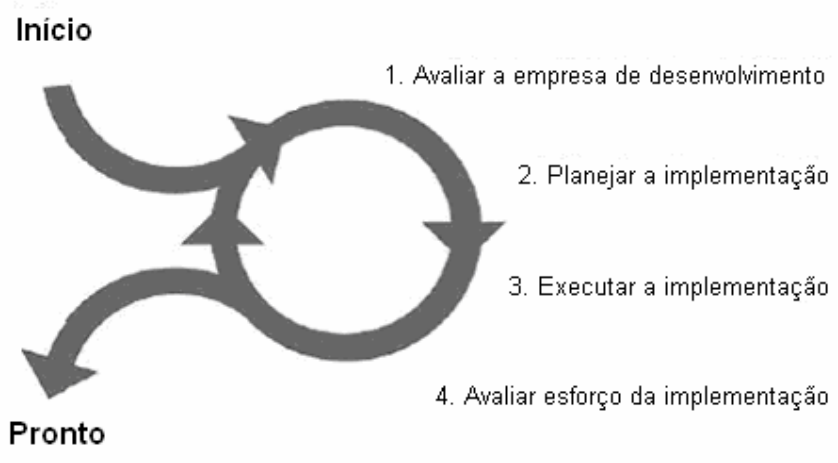

Figura 2 - Passos para implementação de um processo de software [Oliveira.05]

Na primeira etapa, o intuito é coletar informações de pessoas-chave internas ou externas à organização para obter uma lista dos problemas existentes e entender como essas pessoas vêem esses problemas e os priorizam. Este levantamento significa identificar as áreas com problemas na empresa que possuem maior prioridade para serem mudadas.

Afinal, pode não ser o foco da empresa mudar o processo todo e implantar todas as ferramentas de uma só vez, mas sim fazê-lo de forma interativa e incremental, começando com as áreas que têm a maior necessidade e maior potencial para melhorias, podendo, desta forma, provar aos patrocinadores a necessidade de mudança e benefícios esperados,e estabelecer o nível de capacitação técnica que as pessoas na empresa necessitam.

Em uma segunda etapa, a implantação do novo processo deve ser cautelosamente planejada. Um planejamento efetivo deve conter: uma definição clara dos objetivos a serem alcançados e escopo de atuação; um plano de gerência de riscos eficiente; a utilização de um projeto piloto para testar a implantação; estabelecer um plano de treinamentos, visando a capacitação dos desenvolvedores; definir pessoas para servir de disseminadores de conhecimento.

Na etapa seguinte, o processo é efetivamente implantado. Isso significa executar alguns projetos de software escolhidos para adotar o processo e as ferramentas. Do ponto de vista organizacional, isso significa: monitorar os

projetos de desenvolvimento de software; gerenciar a adoção de processo e ferramentas nos projetos; monitorar a criação e uso de um ambiente de desenvolvimento organizacional. A escolha de como deve-se implementar o processo e as ferramentas em uma empresa depende dos problemas que foram identificados e priorizados para o projeto e qual a capacidade de mudança que a equipe pode suportar.

Deve-se ter a capacidade de atacar os maiores riscos desde o início, e de observar se estes estão sendo eliminados de forma efetiva. Uma vez concluído o projeto piloto, o uso do processo e ferramentas deve então ser avaliado com o intuito de serem adotados pelos demais projetos da empresa.

A última etapa é feita a validação de todo o esforço de implantação, isto é, os resultado são validados em relação ao plano proposto no passo na etapa Planejar a implementação. O ciclo pode ser reiniciado, caso uma análise dos resultados proponha uma maneira mais efetiva e eficiente de se trabalhar.

3.4. O IMPPROS

Como descrito nas seções anteriores, para apoiar a modelagem e a execução de processo, têm sido propostos ambientes de desenvolvimento de software centrados no processo, os quais englobam, além das ferramentas de apoio ao desenvolvedor, ferramentas que permitem modelagem do processo de software e execução do mesmo. Dessa forma, o próprio ambiente armazena conhecimento sobre o processo e pode auxiliar desenvolvedores na execução de suas tarefas ou realizar algumas tarefas específicas de forma automática.

Apesar dos benefícios já citados, que essa tecnologia traz, percebe-se ao é comum que alguns processos, implementados não se adaptam às características da organização, ou não são adequados para o projeto para o qual estão sendo implementados.

Esse fato ocorre, pois muitas vezes os responsáveis pela definição do processo não conhecem profundamente as reais necessidades do ambiente onde o processo estará inserido, e acabam indicando, de forma arbitrária, as melhores práticas a serem instanciadas a partir de um processo padrão.

Assim, para ajudar uma organização na implementação progressiva de um processo de software, é útil fornecer apoio automatizado por meio de um ambiente capaz de suportar as fases que a literatura especializada propõe como necessárias. O ImPProS (Ambiente de Implementação Progressiva de Processos de Software) [Oliveira.05] é um ambiente proposto e aprovado em um plano do programa de doutorado do CIn/UFPE. O termo “progressiva” decorre do fato de que a implementação do processo é aperfeiçoado com as experiências aprendidas na sua definição, simulação, execução e avaliação.

O ImPProS está sendo concebido a fim de apoiar a implementação de um processo de software em uma organização. Seus principais objetivos são:

- Especificar um meta-modelo de processo de software a fim de definir uma terminologia única entre os vários modelos de qualidade de processo de software existentes, para uso do ambiente em seus serviços providos. Essa necessidade decorre da existência de vários padrões de modelos diferentes para especificar processos de software, que são encontrados na literatura;
- Apoiar a definição de um processo de software para organização;
- Permitir a modelagem e instanciação deste processo;
- Permitir a simulação do processo a partir das características instanciadas para um projeto específico;
- Dar apoio à execução do processo de software;
- Permitir avaliar se o processo atende aos critérios da organização;
- Apoiar a melhoria contínua do processo de software e o reuso através da realimentação e coleta das experiências aprendidas.

A próxima figura ilustra a arquitetura do ambiente proposto, apresentando quatro tipos de usuários distintos:

- Projetista do Processo: responsável pela definição do processo e coleta de experiência acerca da execução de projetos. Este tipo de usuário interage com o ambiente recebendo orientações e identificando melhorias para processos existentes ou em concepção;

- Gerente de Processo: acompanha a simulação e avaliação do processo a fim de prover conhecimento que possibilite o reúso e a melhoria contínua dos processos;
- Gerente de Projetos: atua nas fases de instanciação do processo para um projeto específico, acompanhando a execução do processo e a sua avaliação para posterior coleta de experiências;
- Equipe de Desenvolvimento: todos os perfis relacionados à execução de um projeto de software.

O mecanismo de interação com o usuário, composto pelo Módulo de Interação e Visualização do Ambiente, possui como objetivo prover, aos usuários envolvidos com os serviços do ambiente, diferentes visões da mesma informação sendo definida e especificada.

Cada perfil deve ter uma visão da informação que lhe seja mais útil.

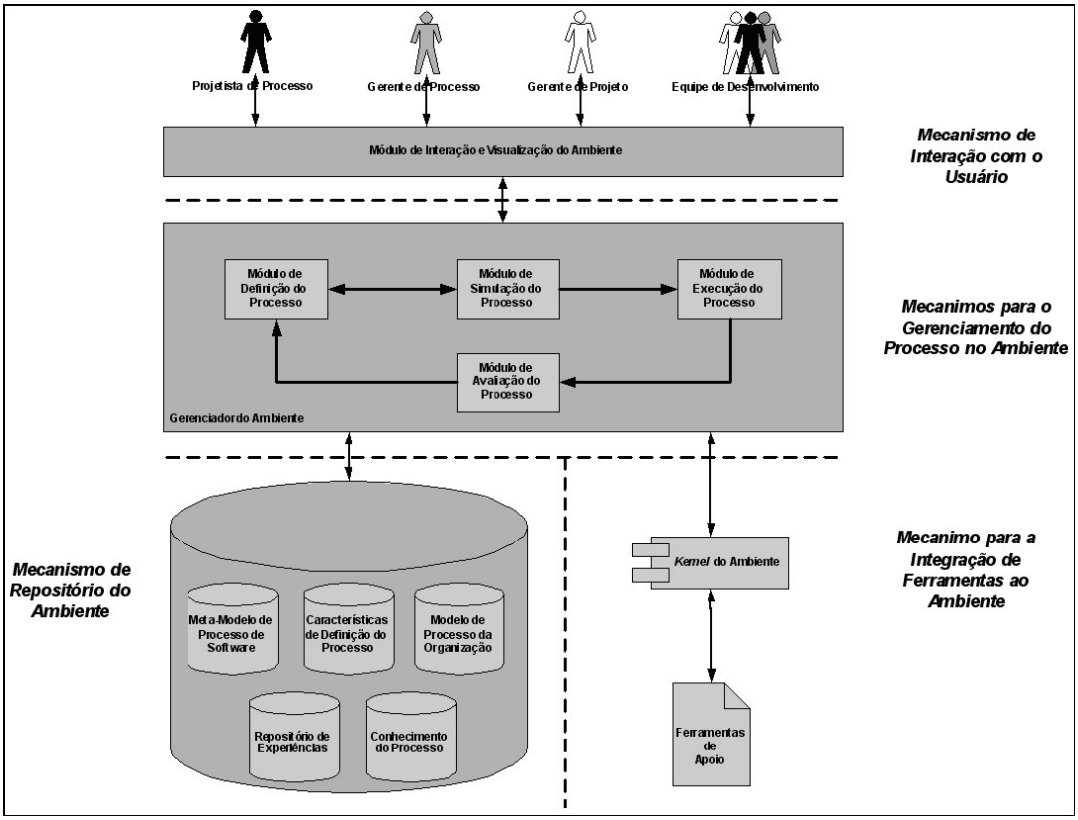

Figura 3 - Arquitetura do ImPProS

O mecanismo para o gerenciamento do processo no ambiente é composto pelo módulo de definição do processo, módulo de simulação do processo, módulo de execução do processo e pelo módulo de avaliação do processo. Este mecanismo possui a responsabilidade de prover serviços para ao ambiente de forma automatizada, ou seja, possibilitar que os usuários do ambiente executem suas funções tendo como referencial um guia que possibilite monitorar as suas definições.

O mecanismo de repositório provê ao ambiente um sistema de gerenciamento dos seus objetos a partir de bases de dados que permitam o controle de evolução e manutenção dos componentes do processo de software.

No mecanismo de integração de ferramentas do ambiente, o objetivo é prover a integração do ambiente com outras ferramentas de apoio tanto ao processo de software, quanto à execução do projeto de software, possibilitando desta forma a automação de atividades definidas no processo de software.

CAPÍTULO 4

ESTRUTURA DO META-MODELO

A estrutura do ImPProS, no qual o meta-modelo está inserido, tem como base o modelo de Falbo [Falbo.98]. A estrutura de definição, especialização e instanciação de processos é ampliada, recebendo o meta-modelo no topo do diagrama, e planos de execução como folhas do esquema. O meta-modelo foi definido contemplando componentes de um processo de software sem restringir a sua composição para algumas normas/modelos de processo de software.

4.1. ESTRUTURA PADRÃO DO PROCESSO DE SOFTWARE DO IMPPROS

O ImPProS por possuir como uma de suas características a definição do processo de software sob a forma de um modelo de processo de software e sua representação diagramática, possui como estrutura geral de composição dos processos de software o modelo baseado nas definições de ontologias de processo de software de Falbo [Falbo.98].

Processos são coleções de atividades relacionadas que têm lugar durante o desenvolvimento de um produto. Basicamente, um processo consiste de um conjunto estruturado de atividades e, por conseguinte, toda a infra-estrutura envolvida na realização destas (artefatos, procedimentos e recursos). Por sua vez, Atividades são as tarefas ou trabalhos a serem realizados. Uma atividade requer recursos e pode consumir ou produzir artefatos. Para sua realização, uma atividade pode adotar um procedimento. Uma atividade pode ser decomposta em outras atividades. Além disso, atividades, em qualquer nível, podem depender da finalização de outras atividades, denominadas pré-atividades. O conceito de atividade está presente em todos os modelos de processo de software e são consideradas primitivas que geram artefatos a partir

de artefatos de entrada auxiliados por recursos. Atividades podem corresponder a diferentes níveis, seja uma tarefa ou uma etapa do processo de desenvolvimento.

Para descrever as etapas do processo e as atividades a serem realizadas em cada etapa, surge o conceito de Modelo de Ciclo de Vida, que estrutura atividades e define abordagem de como organizar um projeto em fases. O ciclo de vida é iniciado quando um software é concebido até quando entra em desuso, ou seja, contém um conjunto de atividades de desenvolvimento, operação e manutenção. Aliado a este conceito temos a Combinação, a qual define a forma como um conjunto de fases de um modelo de ciclo de vida deve ser realizado e especifica o tipo de ordenação em que a estrutura pode ser: seqüencial ou iterativa.

Os Artefatos são produtos de software produzidos ou consumidos por atividades durante a sua realização. São exemplos de artefatos: manuais de qualidade, manuais de revisão, diagramas de fluxos de dados, diagramas de objetos, código fonte, etc. Um artefato pode ser decomposto em outros artefatos (composição de artefatos). É a entrada ou produto de uma atividade, podendo ser artefatos de código, documentos ou componentes de software.

Já os Procedimentos são condutas bem estabelecidas e ordenadas para a realização de atividades. Alguns procedimentos podem ser parcialmente automatizados por ferramentas de software. São utilizados para auxiliar a realização das atividades, podendo ser direcionados a um tipo específico de atividade, devendo ser adequados a uma tecnologia de desenvolvimento e a um paradigma. Aliado a este conceito, temos o Padrão de Atividades que um procedimento deve sugerir para a execução de uma atividade. Representa um comportamento em que decomposições de uma atividade têm em comum.

As pessoas, as ferramentas de software, os equipamentos, ou quaisquer outras infra-estruturas necessárias à execução de uma atividade, recebem o nome de Recurso. Um recurso humano, especificamente, desempenha um papel na execução das atividades do processo. São elementos necessários para a realização de uma atividade, tais como agentes humanos, equipamentos de hardware e ferramentas de software. Apóiam ou atuam na realização da atividade, mas não podem ser consideradas “matérias-primas” para a atividade,

ou seja, apenas auxiliam o processo, mas não são incorporados ao produto de software sendo considerados recursos para a atividade.

Podemos encontrar, ainda, os conceitos de Paradigma de Desenvolvimento, que são princípios e conceitos que orientam o desenvolvimento (por exemplo: o estruturado e o orientado a objetos), e a Tecnologia de Desenvolvimento que representa a tecnologia a ser empregada no desenvolvimento do software (é o caso das tecnologias convencional de processamento de dados e de sistemas baseados em conhecimento). Por fim, para limitar e/ou restringir a execução das atividades definidas no processo, tem-se as Restrições, que especificam as regras de definição do processo de software.

4.2. META-MODELO DE PROCESSO DE SOFTWARE DO IMPPROS

Na definição de processos de software do ImPProS adaptada do modelo definido por [Rocha.01], inicialmente encontra-se o Meta-modelo de processo de software, composto de componentes e dos relacionamentos entre esses que são oriundos do mapeamento de algumas normas e modelos de qualidade para processo de software (CMMI, SPICE – ISO/IEC 15504, ISO/IEC 12207, etc.).

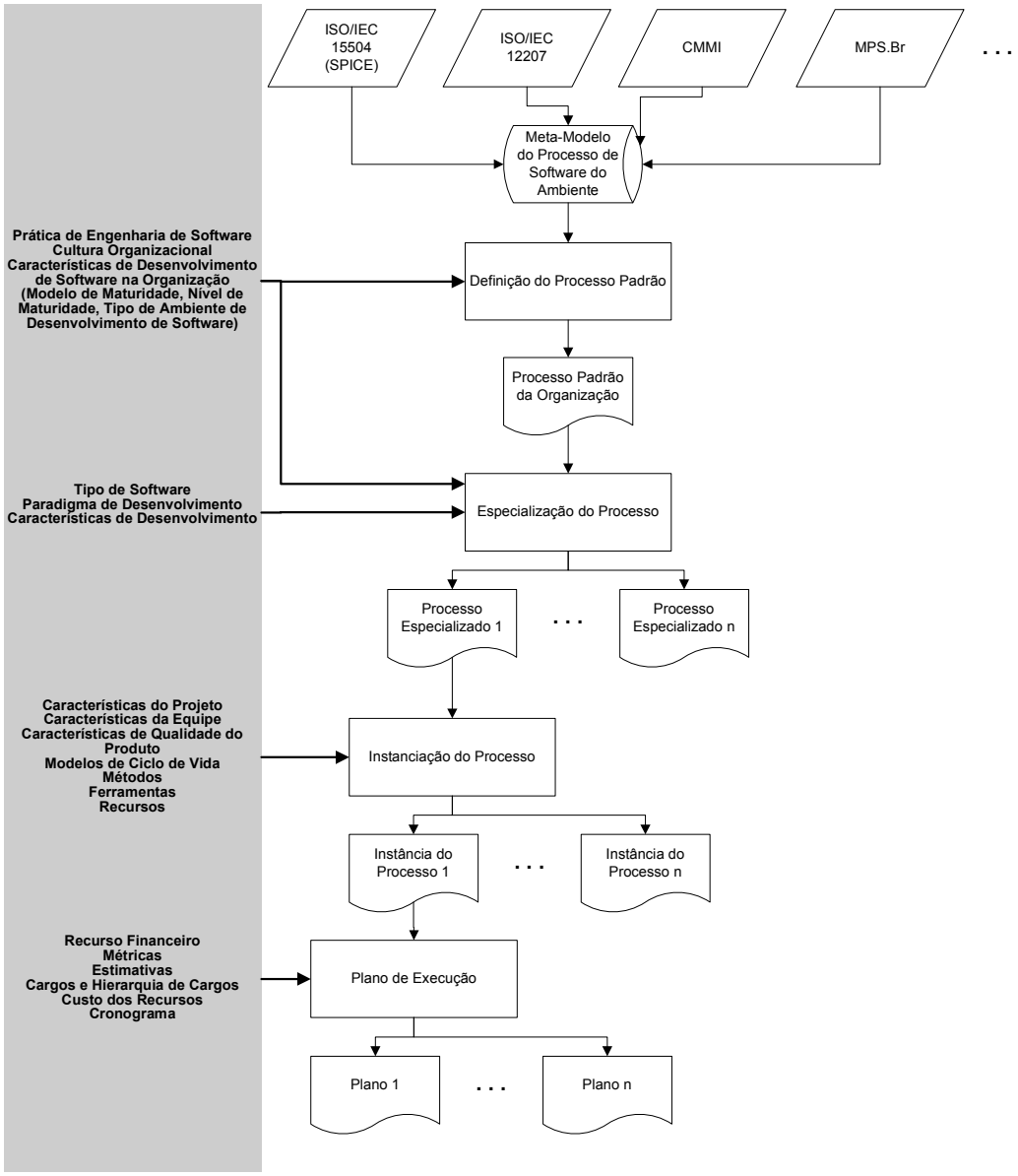

Figura 4 - Estrutura de Definição do Processo de Software no ImPProS [Oliveira.05]

O objetivo deste meta-modelo é determinar uma terminologia única para a definição de processos de software no ImPProS. Vale ressaltar que a estrutura do meta-modelo foi definida de forma a contemplar todos os componentes de um processo de software (processo, atividades, artefatos, etc.), mas não restringir a sua composição para algumas normas/modelos de processo de software (comum nos trabalhos relacionados na seção anterior), ou seja, dependendo da norma/modelo a ser usada, o usuário pode fazer o

mapeamento da mesma usando como base a terminologia da ISO/IEC 12207 e definir os seus processos a partir do uso deste novo meta-modelo.

Por sua vez, a definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software e constitui a base para a definição de todos os seus processos. Tendo em vista que tipos de software diferentes possuem características distintas e requerem diferentes abordagens de desenvolvimento, o processo de software padrão da organização deverá ser adaptado (especializado) considerando-se as características relacionadas ao tipo de software (por exemplo, sistemas de informação) e ao paradigma de desenvolvimento utilizado (por exemplo, orientação a objetos). A instanciação para projetos específicos consiste na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades. Nesta etapa, são definidos o modelo de ciclo de vida, os métodos e as ferramentas que serão utilizadas no projeto, os recursos humanos e suas responsabilidades ao longo do processo e os artefatos (produtos) consumidos e gerados.

O ImPProS propôs três frentes de contribuição que, a partir de análises feitas em definições de processo cotidianamente e o que a literatura especializada propõe como uso, aperfeiçoaram a especificação do processo de software nos três níveis definidos e possibilitaram com que os componentes definidos ao processo pudessem ser simulados a fim de antever problemas na execução do mesmo pela equipe do projeto, a saber:

- Inclusão de um conjunto de novas características organizacionais, de projetos de software e de produtos de software;
- Sugestão de componentes de processo de software a partir de definições de processos anteriormente feitas e conhecimentos aprendidos ao longo destas definições;
- Nível de Planejamento do Processo Instanciado para que este possa servir como base para a simulação deste processo a um projeto específico.

A definição do plano de execução de um processo instanciado consta de uma especificação de algumas características do planejamento do processo

que possibilitem a sua prévia execução. Desta forma, o usuário pode modelar um ou mais planos para verificar como o processo instanciado se comporta a partir das características de execução de um projeto específico: recurso financeiro; métricas; estimativas; cargos e sua hierarquia; custo; e cronograma. Estes atributos são parametrizados para a ferramenta de simulação analisar a execução do processo de software instanciado.

4.2.1. ESPECIFICAÇÃO DA ESTRUTURA DO META-MODELO DO ImPProS

A partir do entendimento prévio da real importância de um meta-modelo de processo de software em uma definição progressiva de processo de software de forma automatizada, com base na estrutura adaptada na figura, buscou-se projetar um repositório que pudesse agregar os conceitos definidos anteriormente com os propostos pelas normas/modelos de qualidade de processos de software. Como retratado por Falbo [Falbo.98] a definição de uma ontologia de um processo de software composta de todo o conteúdo padrão deste tipo de processo, não levando em consideração termos associados às normas/modelos de qualidade de processo de software.

Com o objetivo de atender às necessidades propostas pelo ImPProS quanto à sua definição de processo de software com base em normas/modelos de qualidade, analisou-se inicialmente dois tipos de padrões de qualidade para este fim:

- Normas/Modelos de Maturidade, as quais descrevem orientações para a definição e implantação de processos através de práticas especificadas (atividades, tarefas, produtos gerados, etc.) que são usadas diretamente no processo de software, por exemplo o CMMI, ISO/IEC 15504, etc.;
- Modelos de Referência, semelhante às normas/modelos de maturidade porém não indicam claramente práticas para a composição de processos e sim propósitos (objetivo a ser atingido), resultados esperados (artefato produzido, uma mudança significativa de estado e o atendimento das especificações) e informações adicionais (referências que podem ajudar na definição e implementação do processo) que são descritos através de

um relacionamento com as normas/modelo de maturidade, podemos citar o MPS.Br [Softex.05].

Além destes dois tipos de normas/modelos de qualidade de processo de software, ao meta-modelo do ImPProS foi usada a norma ISO/IEC 12207 por estabelecer uma arquitetura comum para o ciclo de vida de processos de software com uma terminologia bem definida, contendo processos, atividades e tarefas a serem aplicadas durante o fornecimento, desenvolvimento, operação e manutenção de produtos de software. Isso permite com que os processos sejam especificados com uma terminologia unificada e esta norma sirva de base para promover o relacionamento entre as normas/modelos de maturidade a partir do mapeamento dos processos e práticas especificados por estas normas/modelos aos processos, atividades e tarefas constantes na norma ISO/IEC 12207.

Além da norma ISO/IEC 12207, a norma ISO/IEC 9126 [ISSO.02] foi utilizada, também, para identificar os requisitos de qualidade do produto. Tais características irão influenciar o processo no que se refere a atividades, métodos e técnicas. As atividades do processo especializado deverão ser adaptadas ao modelo de ciclo de vida escolhido para o projeto e novas atividades poderão ser inseridas em um determinado ciclo de vida. A influência de atividades ao processo a partir das características desejadas ao produto é fruto do relacionamento entre atividades e tarefas da norma ISO/IEC 12207 e do grau de relevância das características de qualidade apresentadas na norma ISO/IEC 9126 proposta por ODDO et al. [Oddo 2003].

A figura a seguir apresenta, a partir do uso das notações do SPEM – Software *Process Engineering Metamodel* [OMG 2005], o relacionamento existente entre os tipos de normas/modelos de qualidade usados no ImPProS, a norma ISO/IEC 12207 e a norma ISO/IEC 9126. Vale perceber os tipos de relacionamentos existentes entre eles, os quais produzirão os resultados para a composição das terminologias do meta-modelo de processo de software do ImPProS.

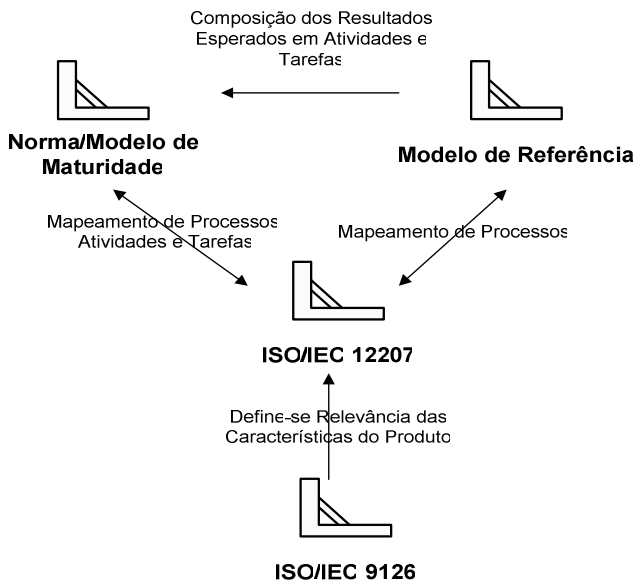

Figura 5 - Relacionamento entre as normas/modelos de qualidade adotados pelo meta-modelo de processo de software do ImPProS.

Pode-se notar na figura acima que há quatro tipos de relacionamento existentes entre as normas/modelos de qualidade, a saber: o mapeamento dos processos e atividades das normas/modelos de qualidade aos processos e atividades da norma ISO/IEC 12207, com o objetivo de padronizar/unificar os conceitos definidos pelos dois tipos de normas no que tange o ciclo de desenvolvimento de software, a fim de prevalecer os termos usados pela norma ISO/IEC 12207 por se tratar da arquitetura comum do ciclo de vida de processo de software; o mapeamento de processos do modelo de referência aos processos da norma ISO/IEC 12207, a fim proporcionar o mesmo objetivo do relacionamento anteriormente especificado, vale ressaltar que não se faz mapeamento de atividades neste caso uma vez que os modelos de referência apresentam resultados esperados e não tarefas a serem executadas em um processo de software; a composição dos resultados esperados dos modelos de referência às atividades das normas/modelos de maturidade, o qual orienta como uma referência do modelo pode ser implementada em um processo de software, ou seja, como o modelo de referência não define o “como” deve ser implementado um processo e sim o “que” este processo deve gerar como resultado, este tipo de relacionamento possibilita especificar o detalhamento destes resultados com base nas tarefas definidas pelas normas/modelos de

maturidade; e a definição da relevância das atividades da norma ISO/IEC 12207 em função das características de qualidade do produto da norma ISO/IEC 9126, o que permite enfatizar que determinadas tarefas da norma ISO/IEC 12207 devem estar obrigatoriamente contempladas no processo de software a partir da análise da relevância das características de qualidade do produto.

A partir dos relacionamentos propostos na última figura, a próxima apresenta, a partir das notações do SPEM, uma adaptação dos conceitos propostos por Falbo, e os padrões adotados pelas normas/modelos de qualidade usadas para a especificação de processos do ImPProS. Esta adaptação define todos os elementos que compõem o meta-modelo de processos de software do ImPProS.

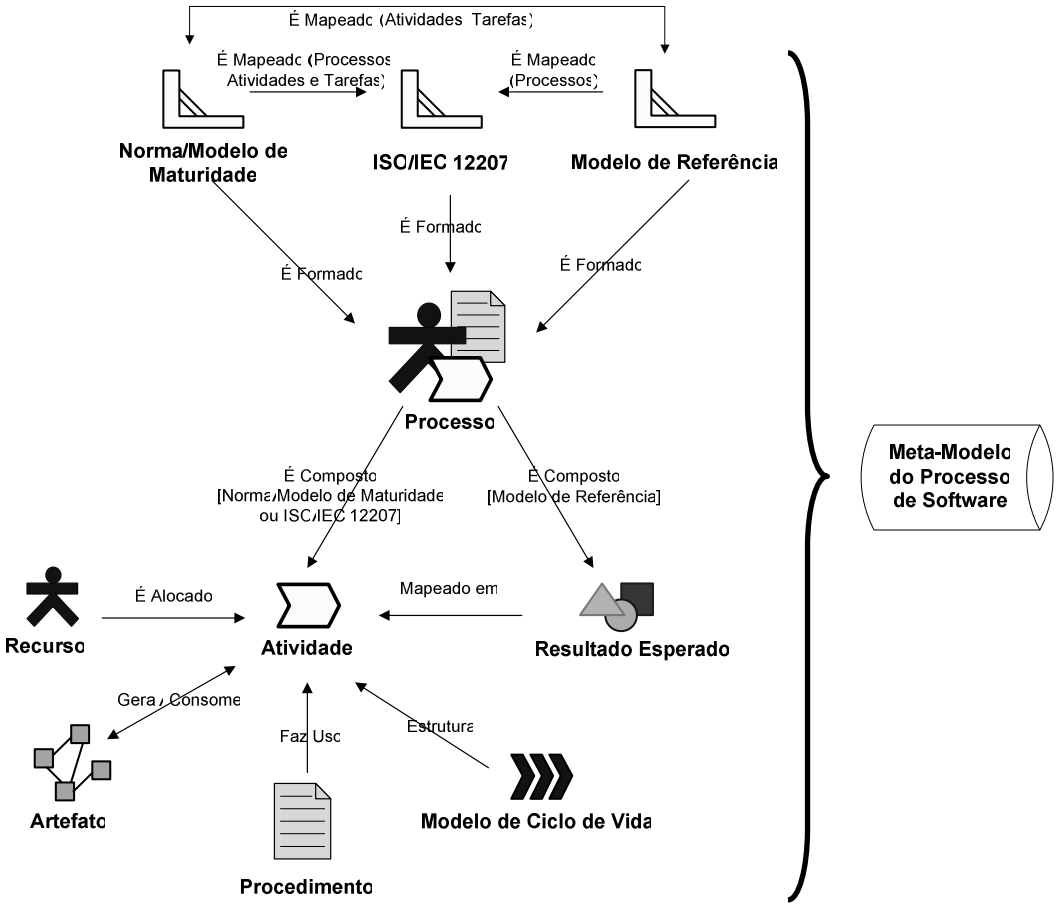

Figura 6 - Elementos de composição do Meta-Modelo de Processo de Software do ImPProS

Pode-se notar a representação dos tipos de mapeamentos entre as normas/modelos de qualidade e a norma ISO/IEC 12207, os quais produzirão uma grande base para a definição e implementação de processos de software no ImPProS. Observa-se que todos são formados por processos de software e estes possuem atividades, se suas origens forem normas/modelos de maturidade, e resultados esperados, se são resultantes de modelos de referência. Por sua vez, os resultados esperados são mapeados em atividades a fim de descrever o processo de software. Estas atividades possuem: uma origem, a qual não restringe que seja exclusivamente oriunda de uma norma/modelo de maturidade ou norma ISO/IEC 12207, podendo originar-se de um tipo de organização, um tipo de projeto de software ou simplesmente genérica; uma granularidade, que define a composição das atividades, ou seja, possibilita descrever se a atividade pode ser decomposta em outras ou se o seu valor é atômico (elementar); e um tipo, que a classifica de acordo com o seu objetivo de implementação (Construção, Gerência ou Garantia de Qualidade).

CAPÍTULO 5

ESTUDO DE CASO

O meta-modelo apresentado na seção 4.2.1 define relacionamentos entre normas/modelos de qualidade, como ilustrado na figura 5. Como é possível observar, o modelo de referência pode ser associado às atividades existentes em normas/modelos de maturidade. Nesta seção, será exibido uma perspectiva de mapeamento entre os modelos MPS.Br (modelo de referência) e CMMI (modelo de maturidade), com o qual será possível, ao final do trabalho, conhecer quais atividades sugeridas pelo modelo de maturidade poderão ser utilizadas para o alcance dos resultados esperados no modelo de referência. Para tanto, é necessário conhecer a estrutura dos dois modelos e compreender como ocorreu a associação entre eles (atividade descrita na subseção 5.1).

Para delimitar o escopo do mapeamento, foram escolhidos os níveis G e F. A decisão foi baseada na maior aplicação deste trabalho, necessidade atual da indústria de software nacional, dado que estes são os primeiros níveis do modelo brasileiro e equivalem ao nível 2 do CMMI (nível inicial de estruturação). O detalhamento dos níveis e o mapeamento entre eles serão feitos logo a seguir.

5.1. ESTRUTURA DO MPS.BR E SUA COMPOSIÇÃO EM FUNÇÃO DO CMMI

O MPS.Br [Softex.06] é o programa de Melhoria de Processo do Software Brasileiro. A coordenação deste programa conta com duas estruturas de apoio para o desenvolvimento de suas atividades: o Fórum de Credenciamento e Controle (FCC), e a Equipe Técnica do Modelo (ETM). Esta última é a responsável sobre os aspectos técnicos relacionados ao Modelo de Referência (MR-MPS) e Método de Avaliação (MA-MPS).

Quando se faz referência ao modelo MPS.Br, na realidade, refere-se ao Modelo de Referência (MR-MPS). Este modelo contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o MR-MPS. Ele contém as definições dos níveis de maturidade, processos e atributos do processo, e está descrito no Guia Geral.

É interessante notar que muitos dos conceitos utilizados no MR-MPS são identificados na subseção referente ao CMMI. Isto acontece pois a definição do MR-MPS foi baseada no modelo americano.

5.1.1. Estrutura MR-MPS

O Modelo de Referência possui níveis de maturidade que são combinações entre processos e suas capacidades. Um processo é definido declarando-se seu propósito e os resultados esperados para sua execução. Dessa forma é possível avaliar e atribuir graus de efetividade na execução dos processos. É de grande importância salientar que as atividades e tarefas necessárias para atender ao propósito e aos resultados esperados não são definidas.

A capacidade do processo é a caracterização da habilidade do processo para alcançar os objetivos de negócio, atuais e futuros; estando relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade.

5.1.1.1. Níveis de Maturidade

Os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos na organização. O MR-MPS define sete níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). O progresso e o alcance de um determinado nível de maturidade MPS se obtém quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e dos atributos de processo estabelecidos para aquele nível.

A divisão em estágios, embora baseada nos níveis de maturidade do CMMI-SE/SW tem uma graduação diferente, com o objetivo de possibilitar uma implementação e avaliação mais adequada às micros, pequenas e médias empresas.

5.1.1.2. Processo

Os processos no MR-MPS são descritos em termos de propósito, resultados e informações adicionais. O propósito descreve o objetivo geral a ser atingido durante a execução do processo. Os resultados esperados do processo estabelecem os resultados a serem obtidos com a efetiva implementação do processo. Estes resultados podem ser evidenciados por um artefato produzido ou uma mudança significativa de estado ao se executar o processo. As informações adicionais são referências que podem ajudar na definição do processo pela organização. Os processos que compõem o MR-MPS.Br serão ilustradas na figura 7.

5.1.1.3. Capacidade do Processo

A capacidade do processo é representada por um conjunto de atributos de processo descrito em termos de resultados esperados. A capacidade do processo expressa o grau de refinamento e institucionalização com que o processo é executado na organização.

O atendimento aos atributos do processo (AP), através do atendimento aos resultados esperados dos atributos do processo (RAP) é requerido para todos os processos no nível correspondente ao nível de maturidade, embora eles não sejam detalhados dentro de cada processo. Os níveis são acumulativos, ou seja, se a organização está no nível F, esta possui o nível de capacidade do nível F que inclui os atributos de processo dos níveis G e F para todos os processos relacionados no nível de maturidade F (que também inclui os processos de nível G). Isto significa que, ao passar do nível G para o nível F, os processos do nível de maturidade G passam a ser executados no nível de capacidade correspondente ao nível F.

A capacidade do processo no MPS possui cinco (5) atributos de processos (AP) que são: AP 1.1 (o processo é executado), AP 2.1 (o processo é gerenciado), AP 2.2 (os produtos de trabalho do processo são gerenciados), AP 3.1 (o processo é definido) e AP 3.2 (o processo está implementado).

Na figura 7, é possível identificar quais atributos de processo são requisitadas em cada um dos níveis de maturidade.

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

Figura 7 – Níveis de Maturidade do MR-MPS

5.1.2. Estrutura CMMI

Como visto na descrição desse modelo, nas seções iniciais do trabalho, o CMMI [Chrissis.03] possui 2 formas de representação: contínua e estagiada. Na representação estagiada, escolhida para ser analisada neste trabalho, com exceção do nível 1 (considerado imaturidade organizacional), todos os níveis restantes (gerenciado, definido, quantitativamente gerenciado e em otimização) são compostos de PAs (Process Areas) que tratam aspectos de cada componente do processo. A PA é um conjunto de práticas relacionadas em uma determinada área que, quando executadas coletivamente, satisfazem um conjunto de objetivos considerados importantes para executar significantes melhorias naquela área. O nome "estagiada" advém da idéia de ganhar maturidade atingindo níveis sucessivos. Cada nível atingido é utilizado como base para o próximo, e assim sucessivamente. A este nível, dá-se o nome de nível de maturidade.

O nível 2 do CMMI (objeto deste trabalho) é composto das seguintes PAs: Gerência de Requisitos, Planejamento de Projetos, Acompanhamento e Controle de Projetos, Gerenciamento de Acordos com Fornecedores, Medição e Análise, Garantia de Qualidade de Produto e Processo e Gerenciamento de Configuração.

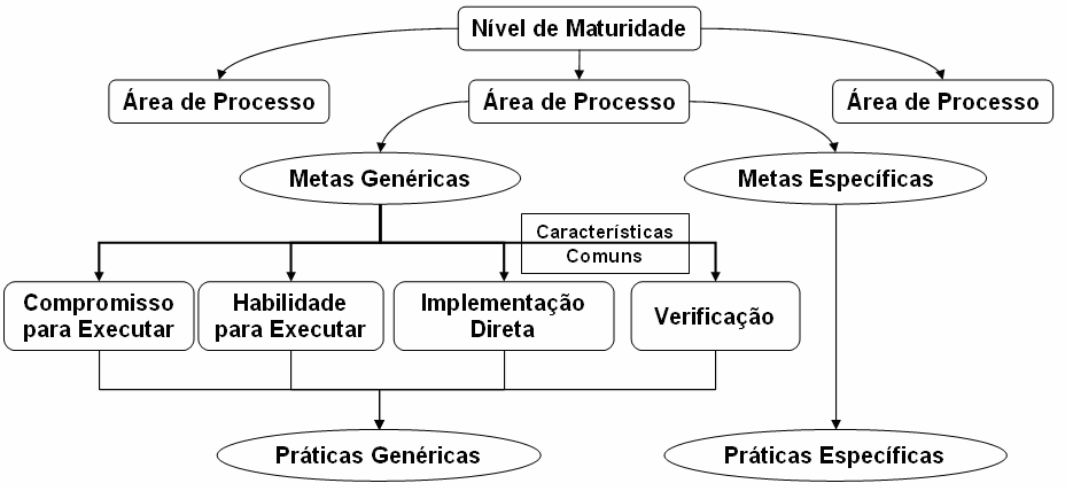

Figura 8 – Componentes do CMMI

Com base na figura acima, é possível identificar os componentes que formam cada uma das PAs: as metas genéricas e as metas específicas. As metas genéricas são objetivos de caráter mais abstrato, únicos para todas as PAs do mesmo nível de maturidade. Para alcançar cada uma das metas genéricas, é preciso executar as práticas genéricas, reunidas em grupos de características comuns. As práticas representam objetivos de caráter mais concreto, próximo do nível de atividade.

Diferentemente das genéricas, as metas específicas são aquelas associadas especificamente a uma única PA. Para atingi-las, de forma análoga, é preciso executar as práticas específicas.

Complementando a ilustração acima, as práticas específicas possuem sub-práticas. Estas são atividades/tarefas sugeridas pelo modelo para atingir as evidências definidas pelas práticas específicas. Observando com atenção, estas atividades definem como uma determinada prática específica pode ser obtida. Isto caracteriza a diferença entre um modelo de maturidade e um modelo de referência. Este último apenas especifica as evidências, sem nenhum aprofundamento (detalhamento).

É válido salientar o caráter sugestivo dessas sub-práticas. Isto implica que o não seguimento destas sub-práticas não representa o descumprimento ao modelo. Portanto, são opcionais para as organizações que desejam a avaliação.

5.1.3. Mapeamento entre modelos

Após um breve detalhamento sobre os modelos, objetos de estudo deste trabalho, é possível explicar como o mapeamento proposto acontece. Antes de adentrar a tal mapeamento, é fato que o MPS.Br originou-se tomando como referência o CMMI. Portanto, nada mais natural do que identificar estruturas semelhantes nos modelos.

Visto que o MR-MPS possui processos para alcançar um objetivo geral de uma área, e que este alcance depende da obtenção de resultados esperados associados ao processo, é possível identificar uma estrutura semelhante no CMMI. De forma análoga, uma PA busca alcançar um objetivo

por meio da execução de práticas específicas. A idéia, portanto, é associar os processos existentes no MR-MPS às PAs do CMMI. Consequentemente, os resultados esperados de cada processo do modelo brasileiro podem ser associados às práticas específicas (SPs) do modelo americano.

No mapeamento proposto, a associação existente entre resultados e práticas é mais importante que associação entre processos e PAs visto que a obtenção destes últimos são definidas em conseqüência dos primeiros. Desta forma, para conseguir a associação a algum resultado esperado do MR-MPS, é possível que a prática específica do CMMI possa estar em uma PA diferente da associada ao processo relativo ao resultado desejado. Assim, focando apenas na associação entre resultados e práticas, poderemos obter um resultado significativo com o mapeamento.

5.2. MAPEAMENTO DOS NÍVEIS 'G' E 'F' DO MPS.BR

Definido o procedimento de associação entre os modelos, podemos enfim apresentar o mapeamento obtido. Para efeito de organização, cada processo dos 2 níveis definidos serão tratados em sub-seções diferentes.

5.2.1. GRE – Gerenciamento de Requisitos (Nível G)

Para o processo de Gerenciamento de Requisitos do MPS.Br, foi realizada a associação com a PA REQM (Requirements Management). A seguir, os resultados esperados e as práticas específicas (SPs) dos respectivos processo e área de processo (PA).

- *GRE 1. Uma comunicação contínua com os fornecedores de requisitos é estabelecida*
- *GRE 2. O entendimento dos requisitos é obtido*
- *GRE 3. A aceitação dos requisitos é estabelecida por meio de critérios objetivos*
- *GRE 4. O comprometimento com os requisitos é estabelecido e mantido*
- *GRE 5. A rastreabilidade entre os requisitos, os planos do projeto e os produtos de trabalho é estabelecida e mantida*
- *GRE 6. Inconsistências entre os planos do projeto, os produtos de trabalho e os requisitos são identificadas e corrigidas*
- *GRE 7. Mudanças nos requisitos são gerenciadas ao longo do projeto*

Figura 9 – GRE (resultados esperados)

- ***SG 1 Manage Requirements***
- *SP 1.1 Obtain an Understanding of Requirements*
- *SP 1.2 Obtain Commitment to Requirements*
- *SP 1.3 Manage Requirements Changes*
- *SP 1.4 Maintain Bidirectional Traceability of Requirements*
- *SP 1.5 Identify Inconsistencies between Project Work and Requirements*

Figura 10 – REQM SG1 (práticas específicas)

Após analisadas as possíveis associações, a tabela seguinte apresenta o mapeamento obtido:

Tabela 1 – GRE x REQM (práticas específicas)

MPS.Br	CMMI
GRE 1	
GRE 2	SP 1.1 (REQM)
GRE 3	SP 1.1 (REQM)
GRE 4	SP 1.2 (REQM), SP 2.1 (REQD)
GRE 5	SP 1.4 (REQM)
GRE 6	SP 1.5 (REQM)
GRE 7	SP 1.3 (REQM)

Como se pode perceber, o primeiro resultado esperado (GRE1) não obteve nenhuma associação. Portanto, não há no CMMI qualquer referência que esteja relacionada com este resultado. Um outro ponto de destaque é a inclusão da SP 2.1, da PA de REQD (Requirements Development) do nível 3 do CMMI. Está prática específica está enunciada da seguinte forma: *Establish Product and Product-Component Requirements*.

De acordo com o mapeamento obtido, é possível também obter as atividades (sub-práticas) sugeridas pelo CMMI para o alcance das práticas específicas, e consequentemente, dos resultados esperados do MPS.Br associados. Abaixo, a tabela que representa o mapeamento das atividades sugeridas.

Tabela 2 – GRE x REQM (sub-práticas)

MPS.Br	Sub-práticas sugeridas pelo CMMI
GRE 1	Sem referências
GRE 2	▪ Establish criteria for distinguishing appropriate requirements providers (SP 1.1-1 – REQM) ▪ Establish objective criteria for the acceptance of requirements (SP 1.1-2 – REQM) ▪ Analyze requirements to ensure that the established criteria are met (SP 1.1-3 – REQM) ▪ Reach an understanding of the requirements with the requirements provider so the project participants can commit to them (SP 1.1-4 – REQM)
GRE 3	▪ Establish objective criteria for the acceptance of requirements (SP 1.1-2 – REQM) ▪ Analyze requirements to ensure that the established criteria are met (SP 1.1-3 – REQM)
GRE 4	▪ Assess the impact of requirements on existing commitments (SP 1.2-1 – REQM) ▪ Negotiate and record commitments (SP 1.2-2 – REQM) ▪ Establish and maintain relationships between requirements for consideration during change management and requirements allocation (SP 2.1-3 – REQD)
GRE 5	▪ Maintain requirements traceability to ensure that the source of lower level (derived) requirements is documented (SP 1.4-1 – REQM) ▪ Maintain requirements traceability from a requirement to its derived requirements as well as to its allocation of functions, objects, people, processes, and work products (SP 1.4-2 – REQM) ▪ Maintain horizontal traceability from function to function and across interfaces (SP 1.4-3 – REQM) ▪ Generate the requirements traceability matrix (SP 1.4-4 – REQM)

A tabela acima identifica quais sub-práticas estão associadas aos resultados do MPS.Br. Ao final de cada sub-prática, há uma pequena

identificação exibindo a prática específica, o ordinal daquela sub-prática, juntamente com a PA de origem da prática específica.

5.2.2. GPR – Gerenciamento de Projetos (Nível G)

Para o processo de Gerenciamento de Projetos do MPS.Br, foi realizada a associação com as PAs PP (Project Planning) e PMC (Project Monitoring and Control). A seguir, os resultados esperados e as práticas específicas (SPs) dos respectivos processo e área de processo (PA).

- *GPR 1. O escopo do trabalho para o projeto está definido*
- *GPR 2. O escopo, os produtos de trabalho e as tarefas do projeto são estimados, através de métodos apropriados*
- *GPR 3. As fases do ciclo de vida do projeto são definidas*
- *GPR 4. A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário ajustes são realizados*
- *GPR 5. As tarefas, os recursos e a infra-estrutura necessários para completar o trabalho são planejados*
- *GPR 6. O cronograma e o orçamento do projeto são estabelecidos e mantidos*
- *GPR 7. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridades de tratamento são determinados e documentados*
- *GPR 8. 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 9. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo*
- *GPR 10. O esforço e o custo para os produtos de trabalho e tarefas são estimados baseados em dados históricos ou referências técnicas*
- *GPR 11. O envolvimento dos interessados no projeto é planejado*
- *GPR 12. O planejamento do projeto é revisado com todos os interessados e o compromisso com o mesmo é obtido*
- *GPR 13. O planejamento do projeto é monitorado no que se refere a cronograma, custos, recursos, riscos, envolvimento dos interessados e dados*
- *GPR 14. Revisões são realizadas em marcos do projeto conforme estabelecido no planejamento*
- *GPR 15. Registros e análise dos problemas identificados nas monitorações são estabelecidos*
- *GPR 16. Ações corretivas são estabelecidas quando necessário e gerenciadas até a sua conclusão*

Figura 11 – GPR (resultados esperados)

- ***SG 1 Establish Estimates***
- *SP 1.1 Estimate the Scope of the Project*
- *SP 1.2 Establish Estimates of Work Product and Task Attributes*
- *SP 1.3 Define Project Life Cycle*
- *SP 1.4 Determine Estimates of Effort and Cost*

Figura 12 – PP SG1 (práticas específicas)

- ***SG 2 Develop a Project Plan***
- *SP 2.1 Establish the Budget and Schedule*
- *SP 2.2 Identify Project Risks*
- *SP 2.3 Plan for Data Management*
- *SP 2.4 Plan for Project Resources*
- *SP 2.5 Plan for Needed Knowledge and Skills*
- *SP 2.6 Plan Stakeholder Involvement*
- *SP 2.7 Establish the Project Plan*

Figura 13 – PP SG2 (práticas específicas)

- ***SG 3 Obtain Commitment to the Plan***
- *SP 3.1 Review Plans that Affect the Project*
- *SP 3.2 Reconcile Work and Resource Levels*
- *SP 3.3 Obtain Plan Commitment*

Figura 14 – PP SG3 (práticas específicas)

- ***SG 1 Monitor Project Against Plan***
- *SP 1.1 Monitor Project Planning Parameters*
- *SP 1.2 Monitor Commitments*
- *SP 1.3 Monitor Project Risks*
- *SP 1.4 Monitor Data Management*
- *SP 1.5 Monitor Stakeholder Involvement*
- *SP 1.6 Conduct Progress Reviews*
- *SP 1.7 Conduct Milestone Reviews*

Figura 15 – PMC SG1 (práticas específicas)

- ***SG 2 Manage Corrective Action to Closure***
- *SP 2.1 Analyze Issues*
- *SP 2.2 Take Corrective Action*
- *SP 2.3 Manage Corrective Action*

Figura 16 – PMC SG2 (práticas específicas)

O mapeamento obtido para este processo do MPS.Br foi o seguinte:

Tabela 3 – GPR x PP/PMC (práticas específicas)

MPS.Br	CMMI
GPR 1	SP 1.3 (PP)
GPR 2	SP 1.1, 1.2 (PP)
GPR 3	SP 1.3 (PP)
GPR 4	SP 3.2 (PP)
GPR 5	SP 2.1, 2.4 (PP)
GPR 6	SP 2.1 (PP), SP 1.1 (PMC)
GPR 7	SP 2.2 (PP)
GPR 8	SP 2.3 (PP)
GPR 9	SP 2.5 (PP)
GPR 10	SP 1.4 (PP)
GPR 11	SP 2.6 (PP)
GPR 12	SP 3.1, 3.3 (PP)
GPR 13	SP 1.1, 1.2, 1.3, 1.4, 1.5 (PMC)
GPR 14	SP 1.7 (PMC)
GPR 15	SP 2.1 (PMC)
GPR 16	SP 2.2, 2.3 (PMC)

O processo de GPR é o que possui o maior número de resultados esperados. Para cobri-lo completamente, foi necessária a associação às PAs de PP e PMC, que têm como foco o planejamento de projeto e o controle e monitoramento do mesmo, respectivamente. É interessante notar que alguns resultados esperados reúnem várias práticas específicas do CMMI. A exemplo do GPR 13, são necessárias 5 práticas específicas para tratar dos diversos aspectos a serem monitorados em um projeto (cronograma, custos, recursos, riscos, envolvimento dos interessados e dados).

Abaixo, a tabela que representa o mapeamento das atividades sugeridas.

Tabela 4 – GPR x PP/PMC (sub-práticas)

MPS.Br	Sub-práticas sugeridas pelo CMMI
GPR 1	-
GPR 2	▪ Develop a WBS based on the product architecture (SP 1.1-1 – PP) ▪ Identify the work packages in sufficient detail to specify estimates of project tasks, responsibilities, and schedule (SP 1.1-2 – PP) ▪ Identify work products (or components of work products) that will be externally acquired (SP 1.1-3 – PP) ▪ Identify work products that will be reused (SP 1.1-4 – PP) ▪ Determine the technical approach for the project (SP 1.2-1 – PP) ▪ Use appropriate methods to determine the attributes of the work products and tasks that will be used to estimate the resource requirements (SP 1.2-2 – PP) ▪ Estimate the attributes of the work products and tasks (SP 1.2-3 – PP) ▪ Estimate, as appropriate, the labor, machinery, materials, and methods that will be required by the project (SP 1.2-4 – PP)
GPR 3	-
GPR 4	▪ Revised methods and corresponding estimating parameters better tools, use of off-the-shelf components) (SP 2.3-1 – PP) ▪ Renegotiated budgets (SP 2.3-2 – PP) ▪ Revised schedules (SP 2.3-3 – PP) ▪ Revised requirements list (SP 2.3-4 – PP) ▪ Renegotiated stakeholder agreements (SP 2.3-5 – PP)
GPR 5	▪ Identify major milestones (SP 2.1-1 – PP) ▪ Identify schedule assumptions (SP 2.1-2 – PP) ▪ Identify constraints (SP 2.1-3 – PP) ▪ Identify task dependencies (SP 2.1-4 – PP) ▪ Define the budget and schedule (SP 2.1-5 – PP) ▪ Establish corrective action criteria (SP 2.1-6 – PP) ▪ Determine process requirements (SP 2.4-1 – PP) ▪ Determine staffing requirements (SP 2.4-2 – PP) ▪ Determine facilities, equipment, and component requirements (SP 2.4-3 – PP)
GPR 6	▪ Identify major milestones (SP 2.1-1 – PP) ▪ Identify schedule assumptions (SP 2.1-2 – PP) ▪ Identify constraints (SP 2.1-3 – PP) ▪ Identify task dependencies (SP 2.1-4 – PP) ▪ Define the budget and schedule (SP 2.1-5 – PP) ▪ Establish corrective action criteria (SP 2.1-6 – PP) ▪ Monitor progress against the schedule (SP 1.1-1 – PMC) ▪ Monitor the project's cost and expended effort (SP 1.1-2 – PMC) ▪ Document the significant deviations in the project planning parameters (SP 1.1-6 – PMC)
GPR 7	▪ Identify risks (SP 2.2-1 – PP) ▪ Document the risks (SP 2.2-1 – PP) ▪ Review and obtain agreement with relevant stakeholders on the completeness and correctness of the documented risks (SP 2.2-1 – PP) ▪ Revise the risks as appropriate (SP 2.2-1 – PP)
GPR 8	▪ Establish requirements and procedures to ensure privacy and security of the data (SP 2.3-1 – PP) ▪ Establish a mechanism to archive data and to access archived (SP 2.3-2 – PP) ▪ Determine the project data to be identified, collected, and distributed (SP 2.3-3 – PP)
GPR 9	▪ Identify the knowledge and skills needed to perform the project (SP 2.5-1 – PP) ▪ Assess the knowledge and skills available (SP 2.5-2 – PP) ▪ Select mechanisms for providing needed knowledge and skills (SP 2.5-3 – PP) ▪ Incorporate selected mechanisms in the project plan (SP 2.5-4 – PP)

GPR 10	▪ Collect the models or historical data that will be used to transform the attributes of the work products and tasks into estimates of the labor hours and cost (SP 1.4-1 – PP) ▪ Include supporting infrastructure needs when estimating effort and cost (SP 1.4-2 – PP) ▪ Estimate effort and cost using models and/or historical data (SP 1.4-3 – PP)
GPR 11	▪ -
GPR 12	▪ Identify needed support and negotiate commitments with relevant stakeholders (SP 3.3-1 – PP) ▪ Document all organizational commitments, both full and provisional, ensuring appropriate level of signatories (SP 3.1-2 – PP) ▪ Review internal commitments with senior management as appropriate (SP 3.1-3 – PP) ▪ Review external commitments with senior management as appropriate (SP 3.1-4 – PP)
GPR 13	▪ Monitor progress against the schedule (SP 1.1-1 – PMC) ▪ Monitor the project's cost and expended effort (SP 1.1-2 – PMC) ▪ Monitor the attributes of the work products and tasks (SP 1.1-3 – PMC) ▪ Monitor resources provided and used (SP 1.1-4 – PMC) ▪ Monitor the knowledge and skills of project personnel (SP 1.1-5 – PMC) ▪ Document the significant deviations in the project planning parameters (SP 1.1-6 – PMC) ▪ Regularly review commitments (both external and internal) (SP 1.2-1 – PMC) ▪ Identify commitments that have not been satisfied or which are at significant risk of not being satisfied (SP 1.2-2 – PMC) ▪ Document the results of the commitment reviews (SP 1.2-3 – PMC) ▪ Periodically review the documentation of the risks in the context of the project's current status and circumstances (SP 1.3-1 – PMC) ▪ Revise the documentation of the risks, as additional information becomes available, to incorporate changes (SP 1.3-2 – PMC) ▪ Communicate risk status to relevant stakeholders (SP 1.3-3 – PMC) ▪ Periodically review data management activities against their description in the project plan (SP 1.4-1 – PMC) ▪ Identify and document significant issues and their impacts (SP 1.4-2 – PMC) ▪ Document the results of data management activity reviews (SP 1.4-3 – PMC) ▪ Periodically review the status of stakeholder involvement (SP 1.5-1 – PMC) ▪ Identify and document significant issues and their impacts (SP 1.5-2 – PMC) ▪ Document the results of the stakeholder involvement status reviews (SP 1.5-3 – PMC)
GPR 14	▪ Conduct reviews at meaningful points in the project's schedule, such as the completion of selected stages, with relevant stakeholders (SP 1.7-1 – PMC) ▪ Review the commitments, plan, status, and risks of the project (SP 1.7-2 – PMC) ▪ Identify and document significant issues and their impacts (SP 1.7-3 – PMC) ▪ Document the results of the review, action items, and decisions (SP 1.7-4 – PMC) ▪ Track action items to closure (SP 1.7-5 – PMC)
GPR 15	▪ Gather issues for analysis (SP 2.1-1 – PMC) ▪ Analyze issues to determine need for corrective action (SP 2.1-2 – PMC)
GPR 16	▪ Determine and document the appropriate actions needed to address the identified issues (SP 2.2-1 – PMC) ▪ Review and get agreement with relevant stakeholders on the actions to be taken (SP 2.2-2 – PMC) ▪ Negotiate changes to internal and external commitments (SP 2.2-3 – PMC) ▪ Monitor corrective actions for completion (SP 2.3-1 – PMC) ▪ Analyze results of corrective actions to determine the effectiveness of the corrective actions (SP 2.3-2 – PMC) ▪ Determine and document appropriate actions to correct deviations from planned results for corrective actions (SP 2.3-3 – PMC)

5.2.3. MED – Medição (Nível F)

Para o processo de Medição do MPS.Br, foi realizada a associação com a PA MA (Measurement and Analysis). A seguir, os resultados esperados e as práticas específicas (SPs) dos respectivos processo e área de processo (PA).

- *MED 1. Objetivos e atividades de medição são estabelecidos a partir das necessidades de informação e objetivos da organização*
- *MED 2. Um conjunto adequado de medidas, orientado pelas necessidades de informação e objetivos de medição, é identificado e/ou desenvolvido, priorizado, documentado, revisado e atualizado*
- *MED 3. As atividades coleta e armazenamento são especificadas, incluindo-se métodos e ferramentas*
- *MED 4. As atividades de análise são especificadas, incluindo-se métodos e ferramentas*
- *MED 5. Os dados requeridos são coletados e analisados*
- *MED 6. Os dados e os resultados são armazenados*
- *MED 7. As informações produzidas são usadas para apoiar decisões e para fornecer uma base objetiva para comunicação aos interessados*

Figura 17 – MED (resultados esperados)

- ***SG 1 Align Measurement and Analysis Activities***
- *SP 1.1 Establish Measurement Objectives*
- *SP 1.2 Specify Measures*
- *SP 1.3 Specify Data Collection and Storage Procedures*
- *SP 1.4 Specify Analysis Procedures*

Figura 19 – MA SG1 (práticas específicas)

- ***SG 2 Provide Measurement Results***
- *SP 2.1 Collect Measurement Data*
- *SP 2.2 Analyze Measurement Data*
- *SP 2.3 Store Data and Results*
- *SP 2.4 Communicate Results*

Figura 20 – MA SG2 (práticas específicas)

Após analisadas as possíveis associações, a tabela seguinte apresenta o mapeamento obtido:

Tabela 5 – MED x MA (práticas específicas)

MPS.Br	CMMI
MED 1	SP 1.1 (MA)
MED 2	SP 1.2 (MA)
MED 3	SP 1.3 (MA)
MED 4	SP 1.4 (MA)
MED 5	SP 2.1, 2.2 (MA)
MED 6	SP 2.3 (MA)
MED 7	SP 2.4 (MA)

O mapeamento entre o processo de Medição e a PA de MA é bastante direto. Cada um dos resultados esperados possui uma prática específica associada, exceto o MED 5, que reúne 2 práticas específicas para ser coberto.

Abaixo, a tabela que representa o mapeamento das atividades sugeridas.

Tabela 6 – MED x MA (sub-práticas)

MPS.Br	Sub-práticas sugeridas pelo CMMI
MED 1	▪ Document information needs and objectives (SP 1.1-1 – MA) ▪ Prioritize information needs and objectives (SP 1.1-2 – MA) ▪ Document, review, and update measurement objectives (SP 1.1-3 – MA) ▪ Provide feedback for refining and clarifying information needs and objectives as necessary (SP 1.1-4 – MA) ▪ Maintain traceability of the measurement objectives to the identified information needs and objectives (SP 1.1-5 – MA)
MED 2	▪ Identify candidate measures based on documented measurement objectives (SP 1.2-1 – MA) ▪ Identify existing measures that already address the measurement objectives (SP 1.2-2 – MA) ▪ Specify operational definitions for the measures (SP 1.2-3 – MA) ▪ Prioritize, review, and update measures (SP 1.2-4 – MA)
MED 3	▪ Identify existing sources of data that are generated from curren work products, proccess, or transaction (SP 1.3-1 – MA) ▪ Identify measures for which data are needed, but are not currently available (SP 1.3-2 – MA)

	▪ Specify how to collect and store the data for each required measure (SP 1.3-3 – MA) ▪ Create data collection mechanisms and process guidance (SP 1.3-4 – MA) ▪ Support automatic collection of the data where appropriate and feasible (SP 1.3-5 – MA) ▪ Prioritize, review, and update data collection and storage procedures (SP 1.3-6 – MA) ▪ Update measures and measurement objectives as necessary (SP 1.3-7 – MA)
MED 4	▪ Specify and prioritize the analyses that will be conducted and the reports that will be prepared (SP 1.4-1 – MA) ▪ Select appropriate data analysis methods and tools (SP 1.4-2 – MA) ▪ Specify administrative procedures for analyzing the data and communicating the results (SP 1.4-3 – MA) ▪ Review and update the proposed content and format of the specified analyses and reports (SP 1.4-4 – MA) ▪ Update measures and measurement objectives as necessary (SP 1.4-5 – MA) ▪ Specify criteria for evaluating the utility of the analysis results, and of the conduct of the measurement and analysis activities (SP 1.4-6 – MA)
MED 5	▪ Obtain the data for base measures (SP 2.1-1 – MA) ▪ Generate the data for derived measures (SP 2.1-2 – MA) ▪ Perform data integrity checks as close to the source of the data as possible (SP 2.1-3 – MA) ▪ Conduct initial analyses, interpret the results, and draw preliminary conclusions (SP 2.2-1 – MA) ▪ Conduct additional measurement and analysis as necessary, and prepare results for presentation (SP 2.2-2 – MA) ▪ Review the initial results with relevant stakeholders (SP 2.2-3 – MA) ▪ Refine criteria for future analyses (SP 2.2-4– MA)
MED 6	▪ Review the data to ensure their completeness, integrity, accuracy, and currency (SP 2.3-1 – MA) ▪ Make the stored contents available for use only by appropriate groups and personnel (SP 2.3-2 – MA) ▪ Prevent the stored information from being used inappropriately (SP 2.3-3 – MA)
MED 7	▪ Keep relevant stakeholders apprised of measurement results on a timely basis (SP 2.4-1 – MA) ▪ Assist relevant stakeholders in understanding the results (SP 2.4-2 – MA)

5.2.4. GCO – Gerência de Configuração (Nível F)

Para o processo de Gerência de Configuração do MPS.Br, foi realizada a associação com a PA CM (Configuration Management). A seguir, os resultados esperados e as práticas específicas (SPs) dos respectivos processo e área de processo (PA).

- *GCO 1. Os itens de configuração são identificados*
- *GCO 2. Os itens de configuração gerados pelo projeto são definidos e colocados sob uma linha base*
- *GCO 3. É estabelecido e mantido um Sistema de Gerência de Configuração*
- *GCO 4. As modificações e liberações dos itens de configuração são controladas*
- *GCO 5. As modificações e liberações são disponibilizadas para todos os envolvidos*
- *GCO 6. A situação dos itens de configuração e as solicitações de mudanças são registradas, relatadas e o seu impacto é analisado*
- *GCO 7. A completeza e a consistência dos itens de configuração são asseguradas*
- *GCO 8. O armazenamento, o manuseio e a entrega dos produtos de trabalho são controlados*
- *GCO 9. A integridade das linhas bases (baselines) é estabelecida e mantida, através de auditoria da configuração e de registros da Gerência de Configuração*

Figura 21 – GCO (resultados esperados)

- ***SG 1 Establish Baselines***
- *SP 1.1 Identify Configuration Items*
- *SP 1.2 Establish a Configuration Management System*
- *SP 1.3 Create or Release Baselines*

Figura 22 – CM SG1 (práticas específicas)

- ***SG 2 Track and Control Changes***
- *SP 2.1 Track Change Requests*
- *SP 2.2 Control Configuration Items*

Figura 23 – CM SG2 (práticas específicas)

- ***SG 3 Establish Integrity***
- *SP 3.1 Establish Configuration Management Records*
- *SP 3.2 Perform Configuration Audits*

Figura 24 – CM SG3 (práticas específicas)

Após analisadas as possíveis associações, a tabela seguinte apresenta o mapeamento obtido:

Tabela 7 – GCO x CM (práticas específicas)

MPS.Br	CMMI
GCO 1	SP 1.1 (CM)
GCO 2	SP 1.3 (CM)
GCO 3	SP 1.2 (CM)
GCO 4	SP 1.3, 2.2 (CM)
GCO 5	SP 1.3, 2.1 (CM)
GCO 6	SP 2.1 (CM)
GCO 7	SP 3.1 (CM)
GCO 8	SP 2.2 (CM)
GCO 9	SP 3.2 (CM)

No mapeamento acima, notamos que a prática específica 1.3 (Create or Release Baseline) acaba satisfazendo vários resultados esperados. Quando não é observado o mapeamento direto no propósito da prática, investigando-se a fundo é possível identificar relacionamentos em suas subpráticas e produtos esperados.

Abaixo, a tabela que representa o mapeamento das atividades sugeridas.

Tabela 8 – GCO x CM (sub-práticas)

MPS.Br	Sub-práticas sugeridas pelo CMMI
GCO 1	▪ Select the configuration items and the work products that compose them based on documented criteria (SP 1.1-1 – CM) ▪ Assign unique identifiers to configuration items (SP 1.1-2 – CM) ▪ Specify the important characteristics of each configuration item (SP 1.1-3 – CM) ▪ Specify when each configuration item is placed under configuration management (SP 1.1-4 – CM)

GCO 2	▪ Obtain authorization from the configuration control board (CCB) before creating or releasing baselines of configuration items (SP 1.3-1 – CM) ▪ Create or release baselines only from configuration items in the configuration management system (SP 1.3-2 – CM) ▪ Document the set of configuration items that are contained in a baseline (SP 1.3-3 – CM) ▪ Make the current set of baselines readily available (SP 1.3-4 – CM)
GCO 3	▪ Establish a mechanism to manage multiple control levels of configuration management (SP 1.2-1 – CM) ▪ Store and retrieve configuration items in the configuration management system (SP 1.2-2 – CM) ▪ Share and transfer configuration items between control levels within the configuration management system (SP 1.2-3 – CM) ▪ Store and recover archived versions of configuration items (SP 1.2-4 – CM) ▪ Store, update, and retrieve configuration management records (SP 1.2-5 – CM) ▪ Create configuration management reports from the configuration management system (SP 1.2-6 – CM) ▪ Preserve the contents of the configuration management system (SP 1.2-7 – CM) ▪ Revise the configuration management structure as necessary (SP 1.2-8 – CM)
GCO 4	▪ Obtain authorization from the configuration control board (CCB) before creating or releasing baselines of configuration items (SP 1.3-1 – CM) ▪ Create or release baselines only from configuration items in the configuration management system (SP 1.3-2 – CM) ▪ Document the set of configuration items that are contained in a baseline (SP 1.3-3 – CM) ▪ Make the current set of baselines readily available (SP 1.3-4 – CM) ▪ Control changes to configuration items throughout the life of the product (SP 2.2-1 – CM) ▪ Obtain appropriate authorization before changed configuration items are entered into the configuration management system (SP 2.2-2 – CM) ▪ Check in and check out configuration items from the configuration management system for incorporation of changes in a manner that maintains the correctness and integrity of the configuration items (SP 2.2-3 – CM) ▪ Perform reviews to ensure that changes have not caused unintended effects on the baselines (e.g., ensure that the changes have not compromised the safety and/or security of the system) (SP 2.2-4 – CM) ▪ Record changes to configuration items and the reasons for the changes as appropriate (SP 2.2-5 – CM)
GCO 5	▪ Obtain authorization from the configuration control board (CCB) before creating or releasing baselines of configuration items (SP 1.3-1 – CM) ▪ Create or release baselines only from configuration items in the configuration management system (SP 1.3-2 – CM) ▪ Document the set of configuration items that are contained in a baseline (SP 1.3-3 – CM) ▪ Make the current set of baselines readily available (SP 1.3-4 – CM) ▪ Initiate and record change requests in the change request database (SP 2.1-1 – CM) ▪ Analyze the impact of changes and fixes proposed in the change requests (SP 2.1-2 – CM) ▪ Review change requests that will be addressed in the next baseline with those who will be affected by the changes and get their agreement (SP 2.1-3 – CM) ▪ Track the status of change requests to closure (SP 2.1-4 – CM)
GCO 6	▪ Initiate and record change requests in the change request database (SP 2.1-1 – CM) ▪ Analyze the impact of changes and fixes proposed in the change requests (SP 2.1-2 – CM) ▪ Review change requests that will be addressed in the next baseline with those who will be affected by the changes and get their agreement (SP 2.1-3 – CM) ▪ Track the status of change requests to closure (SP 2.1-4 – CM)

GCO 7	▪ Record configuration management actions in sufficient detail so the content and status of each configuration item is known and previous versions can be recovered (SP 3.1-1 – CM) ▪ Ensure that relevant stakeholders have access to and knowledge of the configuration status of the configuration items (SP 3.1-2 – CM) ▪ Specify the latest version of the baselines (SP 3.1-3 – CM) ▪ Identify the version of configuration items that constitute a particular baseline (SP 3.1-4 – CM) ▪ Describe the differences between successive baselines (SP 3.1-5 – CM) ▪ Revise the status and history (i.e., changes and other actions) of each configuration item as necessary (SP 3.1-6 – CM)
GCO 8	▪ Control changes to configuration items throughout the life of the product (SP 2.2-1 – CM) ▪ Obtain appropriate authorization before changed configuration items are entered into the configuration management system (SP 2.2-2 – CM) ▪ Check in and check out configuration items from the configuration management system for incorporation of changes in a manner that maintains the correctness and integrity of the configuration items (SP 2.2-3 – CM) ▪ Perform reviews to ensure that changes have not caused unintended effects on the baselines (e.g., ensure that the changes have not compromised the safety and/or security of the system) (SP 2.2-4 – CM) ▪ Record changes to configuration items and the reasons for the changes as appropriate (SP 2.2-5 – CM)
GCO 9	▪ Assess the integrity of the baselines (SP 3.2-1 – CM) ▪ Confirm that the configuration records correctly identify the configuration of the configuration items (SP 3.2-2 – CM) ▪ Review the structure and integrity of the items in the configuration management system (SP 3.2-3– CM) ▪ Confirm the completeness and correctness of the items in the configuration management system (SP 3.2-4– CM) ▪ Confirm compliance with applicable configuration management standards and procedure (SP 3.2-5– CM) ▪ Track action items from the audit to closure (SP 3.2-6 – CM)

5.2.5. GQA – Gerência de Qualidade (Nível F)

Para o processo de Medição do MPS.Br, foi realizada a associação com a PA PPQA (Process and Product Quality Assurance). A seguir, os resultados esperados e as práticas específicas (SPs) dos respectivos processo e área de processo (PA).

- *GQA 1. A aderência dos produtos aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente*
- *GQA 2. A aderência dos processos executados aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente*
- *GQA 3. Os produtos de trabalho são avaliados antes de serem entregues ao cliente e em marcos predefinidos ao longo do ciclo de vida do projeto*
- *GQA 4. Os problemas e as não-conformidades são identificados, registrados e comunicados*
- *GQA 5. Ações corretivas para não-conformidades são estabelecidas e acompanhadas até as suas efetivas conclusões*
- *GQA 6. O escalonamento das ações corretivas para níveis superiores é realizado, quando necessário, de forma a garantir a solução das mesmas*
- *GQA 7. A aderência ao processo de Garantia da Qualidade e de seus produtos de trabalho aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente*

Figura 25 – GQA (resultados esperados)

- ***SG 1 Objectively Evaluate Processes and Work Products***
- *SP 1.1 Objectively Evaluate Processes*
- *SP 1.2 Objectively Evaluate Work Products and Services*

Figura 26 – CM SG1 (práticas específicas)

- ***SG 2 Provide Objective Insight***
- *SP 2.1 Communicate and Ensure Resolution of Noncompliance Issues*
- *SP 2.2 Establish Records*

Figura 27 – CM SG2 (práticas específicas)

Após analisadas as possíveis associações, a tabela seguinte apresenta o mapeamento obtido:

Tabela 9 – GQA x PPQA (práticas específicas)

MPS.Br	CMMI
GQA 1	SP 1.2 (PPQA)

GQA 2	SP 1.1 (PPQA)
GQA 3	SP 1.2 (PPQA)
GQA 4	SP 1.2, 2.1 (PPQA)
GQA 5	SP 2.1 (PPQA)
GQA 6	SP 2.1 (PPQA)
GQA 7	SP 1.1 (PPQA)

A área de processo PPQA possui apenas 4 práticas específicas, no entanto todos os resultados de GQA são completamente mapeados pelas práticas apresentadas.

Abaixo, a tabela que representa o mapeamento das atividades sugeridas.

Tabela 10 – GQA x PPQA (sub-práticas)

MPS.Br	Sub-práticas sugeridas pelo CMMI
GQA 1	▪ Select work products to be evaluated, based on documented sampling criteria if sampling is used (SP 1.2-1 – PPQA) ▪ Establish and maintain clearly stated criteria for the evaluation of work products (SP 1.2-2 – PPQA) ▪ Use the stated criteria during the evaluations of work products (SP 1.2-3 – PPQA) ▪ Evaluate work products before they are delivered to the customer (SP 1.2-4 – PPQA) ▪ Evaluate work products at selected milestones in their development (SP 1.2-5 – PPQA) ▪ Perform in-progress or incremental evaluations of work products and services against process descriptions, standards, and procedures (SP 1.2-6 – PPQA) ▪ Identify each case of noncompliance found during the evaluations (SP 1.2-7 – PPQA) ▪ Identify lessons learned that could improve processes for future products and services (SP 1.2-8 – PPQA)
GQA 2	▪ Promote an environment (created as part of project management) that encourages employee participation in identifying and reporting quality issues (SP 1.1-1 – PPQA) ▪ Establish and maintain clearly stated criteria for the evaluations (SP 1.1-2 – PPQA) ▪ Use the stated criteria to evaluate performed processes for adherence to process descriptions, standards, and procedures (SP 1.1-3 – PPQA) ▪ Identify each noncompliance found during the evaluation (SP 1.1-4 – PPQA) ▪ Identify lessons learned that could improve processes for future products and services (SP 1.1-5 – PPQA)
GQA 3	▪ Select work products to be evaluated, based on documented sampling criteria if sampling is used (SP 1.2-1 – PPQA) ▪ Establish and maintain clearly stated criteria for the evaluation of work products (SP 1.2-2 – PPQA) ▪ Use the stated criteria during the evaluations of work products (SP 1.2-3 – PPQA) ▪ Evaluate work products before they are delivered to the customer (SP 1.2-4 –

	PPQA) ▪ Evaluate work products at selected milestones in their development (SP 1.2-5 – PPQA) ▪ Perform in-progress or incremental evaluations of work products and services against process descriptions, standards, and procedures (SP 1.2-6 – PPQA) ▪ Identify each case of noncompliance found during the evaluations (SP 1.2-7 – PPQA) ▪ Identify lessons learned that could improve processes for future products and services (SP 1.2-8 – PPQA)
GQA 4	▪ Identify each case of noncompliance found during the evaluations (SP 1.2-7 – PPQA) ▪ Document noncompliance issues when they cannot be resolved within the project (SP 2.1-2 – PPQA) ▪ Analyze the noncompliance issues to see if there are any quality trends that can be identified and addressed (SP 2.1-4 – PPQA) ▪ Ensure that relevant stakeholders are aware of the results of evaluations and the quality trends in a timely manner (SP 2.1-5 – PPQA)
GQA 5	▪ Resolve each noncompliance with the appropriate members of the staff where possible (SP 2.1-1 – PPQA) ▪ Document noncompliance issues when they cannot be resolved within the project (SP 2.1-2 – PPQA) ▪ Analyze the noncompliance issues to see if there are any quality trends that can be identified and addressed (SP 2.1-4 – PPQA) ▪ Ensure that relevant stakeholders are aware of the results of evaluations and the quality trends in a timely manner (SP 2.1-5 – PPQA) ▪ Track noncompliance issues to resolution (SP 2.1-7 – PPQA)
GQA 6	▪ Escalate noncompliance issues that cannot be resolved within the project to the appropriate level of management designated to receive and act on noncompliance issues (SP 2.1-3 – PPQA)
GQA 7	▪ Promote an environment (created as part of project management) that encourages employee participation in identifying and reporting quality issues (SP 1.1-1 – PPQA) ▪ Establish and maintain clearly stated criteria for the evaluations (SP 1.1-2 – PPQA) ▪ Use the stated criteria to evaluate performed processes for adherence to process descriptions, standards, and procedures (SP 1.1-3 – PPQA) ▪ Identify each noncompliance found during the evaluation (SP 1.1-4 – PPQA) ▪ Identify lessons learned that could improve processes for future products and services (SP 1.1-5 – PPQA)

5.2.6. AQU – Aquisição (Nível F)

Para o processo de Aquisição do MPS.Br, foi realizada a associação com a PA SAM (Supplier Agreement Management). A seguir, os resultados esperados e as práticas específicas (SPs) dos respectivos processo e área de processo (PA).

- *AQU 1. As necessidades de aquisição, as metas, os critérios de aceitação do produto e/ou serviço, os tipos e estratégia de aquisição são definidos*
- *AQU 2. Os critérios de seleção do fornecedor são estabelecidos e usados para avaliar os potenciais fornecedores;*
- *AQU 3. O fornecedor é selecionado com base na avaliação das propostas e dos critérios estabelecidos*
- *AQU 4. Um acordo que expresse claramente a expectativa, as responsabilidades e as obrigações de ambos (cliente e fornecedor) é estabelecido e negociado entre o cliente e o fornecedor*
- *AQU 5. Um produto e/ou serviço que satisfaz a necessidade expressa pelo cliente é adquirido baseado na análise dos potenciais candidatos*
- *AQU 6. A aquisição é monitorada de forma que as condições especificadas são atendidas, tais como: custo, cronograma e qualidade e, se necessário, ações corretivas são conduzidas*
- *AQU 7. O produto e/ou serviço de software entregue é avaliado em relação ao acordado e os resultados da aceitação são documentados*
- *AQU 8. O produto adquirido (caso pertinente) é incorporado ao projeto*

Figura 28 – AQU (resultados esperados)

- ***SG 1 Establish Supplier Agreements***
- *SP 1.1 Determine Acquisition Type*
- *SP 1.2 Select Suppliers*
- *SP 1.3 Establish Supplier Agreements*

Figura 29 – SAM SG1 (práticas específicas)

- ***SG 2 Satisfy Supplier Agreements***
- *SP 2.1 Review COTS Products*
- *SP 2.2 Execute the Supplier Agreement*
- *SP 2.3 Accept the Acquired Product*
- *SP 2.4 Transition Products*

Figura 30 – SAM SG2 (práticas específicas)

Após analisadas as possíveis associações, a tabela seguinte apresenta o mapeamento obtido:

Tabela 11 – AQU x SAM (práticas específicas)

MPS.Br	CMMI
AQU 1	SP 1.1 (SAM)
AQU 2	SP 1.2 (SAM)
AQU 3	SP 1.2 (SAM)
AQU 4	SP 1.3 (SAM)
AQU 5	SP 2.2 (SAM)
AQU 6	SP 2.2 (SAM)
AQU 7	SP 2.3 (SAM)
AQU 8	SP 2.4 (SAM)

As práticas específicas da PA de SAM conseguem atender a cada um dos resultados esperados. Como podemos observar, existem práticas específicas, como a SP 2.1 que trata da revisão dos chamados produtos de prateleira, que não são mapeadas a nenhum resultado esperado.

Abaixo, a tabela que representa o mapeamento das atividades sugeridas.

Tabela 12 – AQU x SAM (sub-práticas)

MPS.Br	Sub-práticas sugeridas pelo CMMI
AQU 1	▪ -
AQU 2	▪ Establish and document criteria for evaluating potential suppliers (SP 1.2-1 – SAM) ▪ Identify potential suppliers and distribute solicitation material and requirements to them (SP 1.2-2 – SAM) ▪ Evaluate proposals according to evaluation criteria (SP 1.2-3 – SAM) ▪ Evaluate risks associated with each proposed supplier (SP 1.2-4 – SAM) ▪ Evaluate proposed suppliers' ability to perform the work (SP 1.2-5 – SAM)
AQU 3	▪ Identify potential suppliers and distribute solicitation material and requirements to them (SP 1.2-2 – SAM) ▪ Evaluate proposals according to evaluation criteria (SP 1.2-3 – SAM) ▪ Evaluate risks associated with each proposed supplier (SP 1.2-4 – SAM) ▪ Evaluate proposed suppliers' ability to perform the work (SP 1.2-5 – SAM) ▪ Select the supplier (SP 1.2-6 – SAM)
AQU 4	▪ Revise the requirements to be fulfilled by the supplier to reflect negotiations with the supplier when necessary (SP 1.3-1 – SAM) ▪ Document what the project will provide to the supplier (SP 1.3-2 – SAM) ▪ Document the supplier agreement (SP 1.3-3 – SAM) ▪ Ensure all parties to the agreement understand and agree to all requirements before implementing the agreement (SP 1.3-4 – SAM) ▪ Revise the supplier agreement as necessary (SP 1.3-5 – SAM) ▪ Revise the project's plans and commitments as necessary to reflect the supplier

	agreement (SP 1.3-6 – SAM)
AQU 5	▪ Monitor supplier progress and performance (schedule, effort, cost, and technical performance) as defined in the supplier agreement (SP 2.2-1 – PPQA) ▪ Monitor selected supplier processes and take corrective action when necessary (SP 2.2-2 – PPQA) ▪ Conduct reviews with the supplier as specified in the supplier agreement (SP 2.2-3 – PPQA) ▪ Conduct technical reviews with the supplier as defined in the supplier agreement (SP 2.2-4 – PPQA) ▪ Conduct management reviews with the supplier as defined in the supplier agreement (SP 2.2-5 – PPQA) ▪ Use the results of reviews to improve the supplier's performance and to establish and nurture long-term relationships with preferred suppliers (SP 2.2-6 – PPQA) ▪ Monitor risks involving the supplier and take corrective action as necessary (SP 2.2-7 – PPQA) ▪ Revise the supplier agreement and project plans and schedules as necessary (SP 2.2-8 – PPQA)
AQU 6	▪ Monitor supplier progress and performance (schedule, effort, cost, and technical performance) as defined in the supplier agreement (SP 2.2-1 – PPQA) ▪ Monitor selected supplier processes and take corrective action when necessary (SP 2.2-2 – PPQA) ▪ Conduct reviews with the supplier as specified in the supplier agreement (SP 2.2-3 – PPQA) ▪ Conduct technical reviews with the supplier as defined in the supplier agreement (SP 2.2-4 – PPQA) ▪ Conduct management reviews with the supplier as defined in the supplier agreement (SP 2.2-5 – PPQA) ▪ Use the results of reviews to improve the supplier's performance and to establish and nurture long-term relationships with preferred suppliers (SP 2.2-6 – PPQA) ▪ Monitor risks involving the supplier and take corrective action as necessary (SP 2.2-7 – PPQA) ▪ Revise the supplier agreement and project plans and schedules as necessary (SP 2.2-8 – PPQA)
AQU 7	▪ Define the acceptance procedures (SP 2.3-1 – SAM) ▪ Review and obtain agreement with relevant stakeholders on the acceptance procedures before the acceptance review or test (SP 2.3-2 – SAM) ▪ Verify that the acquired products satisfy their requirements (SP 2.3-3 – SAM) ▪ Confirm that the nontechnical commitments associated with the acquired work product are satisfied (SP 2.3-4 – SAM) ▪ Document the results of the acceptance review or test (SP 2.3-5 – SAM) ▪ Establish and obtain supplier agreement on an action plan for any acquired work products that do not pass their acceptance review or test (SP 2.3-6 – SAM) ▪ Identify, document, and track action items to closure (SP 2.3-7 – SAM)
AQU 8	▪ Ensure that there are appropriate facilities to receive, store, use, and maintain the acquired products (SP 2.4-1 – SAM) ▪ Ensure that appropriate training is provided for those involved in receiving, storing, using, and maintaining the acquired products (SP 2.4-2 – SAM) ▪ Ensure that storing, distributing, and using the acquired products are performed according to the terms and conditions specified in the supplier agreement or license (SP 2.4-3 – SAM)

CAPÍTULO 6

CONCLUSÕES

Neste capítulo, são feitas as considerações finais sobre o trabalho realizado. É apresentado um sumário contendo os pontos abordados no documento, o relato dos conhecimentos obtidos, bem como as lições aprendidas. Além disso, algumas propostas de trabalho futuro são lançadas, esperando que a discussão possa ser ampliada em um novo trabalho.

6.1. SUMÁRIO DO TRABALHO

Este trabalhou apresentou em seu capítulo inicial uma introdução sobre tema, fazendo uma abordagem geral sobre o trabalho, destacando os objetivos a serem alcançados. Em seguida, o processo de software foi abordado com breve explicação sobre sua definição e melhoria, e foi apresentado um modelo para definição de software. Também foi apresentado o ImPProS, sua estrutura, e em destaque, o meta-modelo. A perspectiva de mapeamento proposta foi demonstrada no penúltimo capítulo e por fim, as considerações finais.

A perspectiva de mapeamento atingiu seu objetivo no sentido de fornecer apoio para a implementação do MPS.Br através de atividades sugeridas. Com o resultado obtido, o ambiente ImPProS poderá mapear os resultados esperados do modelo de referência nas atividades do modelo de maturidade, auxiliando a definição automatizada de um processo de software pelos seus usuários.

A atividade de mapeamento entre modelos mostrou-se complexa em alguns momentos devido às interpretações geradas na leitura dos modelos. Apenas com alguma vivência prática de implementação e avaliação de processos, é possível compreender quais são os objetivos pretendidos pelos modelos em suas práticas e resultados esperados.

A realização deste trabalho permitiu também um conhecimento abrangente sobre processos de software, ao apresentar seus conceitos e experiências automatizadas. O conhecimento sobre ambientes de desenvolvimento de software centrado no processo permitiu conhecer a base do ImPProS, e por conseqüência, entender na prática como o ambiente influi no desenvolvimento de projetos de software.

6.2. TRABALHOS FUTUROS

Alguns trabalhos podem surgir a partir do tema abordado. Uma primeira e evidente proposta é o mapeamento do MPS.Br por completo, já que aqui, devido à grande demanda de trabalho, foram abordados apenas os primeiros dois níveis do modelo. Esta possibilidade representaria um guia completo de apoio às organizações que tivessem como objetivo a implementação de um processo aderente ao MPS.Br.

Para comprovar na prática os benefícios gerados por este trabalho, a aplicação desta perspectiva de mapeamento a partir da implementação de melhoria de processo de software na indústria nacional seria de grande valia. Com está aplicação, seria possível verificar concretamente os resultados obtidos pelas organizações participantes, podendo inclusive otimizar este mapeamento para torná-lo mais eficiente no apoio à definição e melhoria de processos.

Devido a perspectiva adotada, focada no ambiente ImPProS, é clara a necessidade da aplicação do mapeamento obtido no próprio ambiente. Com esta proposta, seria possível verificar efetivamente os ganhos pretendidos com o mapeamento, ao torná-lo parte integrante do ambiente automatizado. Sendo o mapeamento utilizado no ImPProS, seria possível de forma automatizada avaliar a aderência de um processo aos modelos MPS.Br e CMMI, bem como obter a rastreabilidade entre eles.

REFERÊNCIAS BIBLIOGRÁFICAS

[Ahn.03] AHN, Y. W., AHN, H. J. & PARK, S. J., Knowledge and Case-Based Reasoning for Customization of Software Processes - A Hybrid Approach, International Journal of Software Engineering and Knowledge Engineering, vol. 13, n. 3, 2003.

[Bartié.02] BARTIÉ, Alexandre. Garantia da qualidade de software: adquirindo maturidade organizacional. Rio de Janeiro, Elsevier, 2002.

[Basili.87] BASILI, V. R. & ROMBACH, H. D., Tailoring the Software Process to Project Goals and Environments, In Proc. International Conference on Software Engineering, 1987.

[Balduino.02] BALDUINO, Ricardo. Implementação de um processo de desenvolvimento de software: uma abordagem passo-a-passo. Rational Software White Paper, 2002.

[Berger.03] BERGER, P. M., Instanciação de Processos de Software em Ambientes Configurados na Estação TABA, Dissertação de Mestrado, COPPE, Universidade Federal do Rio de Janeiro, 2003.

[Chrissis.03] CHRISSIS, M. B., KONRAD, M. and SHRUM, S., CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003.

[Coelho.03] COELHO, C. C., MAPS: Um Modelo de Adaptação de Processos de Software, Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco, 2003.

[Conradi.02] CONRADI, R e FUGGETTA, A - Improving Software Process Improvement, (2002).

[Falbo.98] FALBO, Ricardo de Almeida, Integração de Conhecimento em um Ambiente de Desenvolvimento de Software, Orientadora: Ana Regina Cavalcanti da Rocha. Tese de Doutorado, COPPE/UFRJ, 1998.

[Falbo.99] FALBO, Ricardo de Almeida & MENEZES, Crediné Silva de &

ROCHA, Ana Regina Cavalcanti da, Assist-Pro: um assistente inteligente para apoiar a definição de processos de software, Anais do XIII Simpósio Brasileiro de Engenharia de Software. Florianópolis, SC, 1999.

[Henninger.98] HENNINGER, S., An Environment for Reusing Software Processes, In Proc. International Conference on Software Reuse, 1998.

[Humphrey.89]] HUMPHREY, Watts S., Managing the Software Process, The SEI Series in Software Engineering. Addison-Wesley, 1989.

[ISO.91] ISO 9000-3, Quality management and quality assurance standards – parte 3: guidelines for the application of ISO 9001:1994 to the development, supply and maintenance of computer software. International Organization for Standardization, 1991.

[ISO.97] NBR ISO/IEC 12207:1995, Tecnologia de informação – processos de ciclo de vida de software. Associação Brasileira de Normas Técnicas, 1997.

[ISO.98] ISO/IEC TR 15504, Information technology – software process assessment. International Organization for Standardization, 1998.

[ISO.02] ISO/IEC TR 9126, Software engineering – product quality. International Organization for Standardization, 2002.

[ISO.03] ISO/IEC 15504 –1 Information Technology – Process Assessment, - Part 1: Concepts and Vocabulary, 2003.

[Jacobson.98] JACOBSON, I., BOOCH, G., RUMBAUGH, J., The Unified Software Development Process, Addison Wesley Longman, Inc, 1998.

[Maidantchik.99] MAIDANTCHIK, C. L. L. M., Gerência de Processos de Software para Equipes Geograficamente Dispersas, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, 1999.

[Münch.97] MÜNCH, J., SCHMITZ, M. & VERLAGE, M., Tailoring groβer Prozeβmodelle auf der Basis von MVP-L, In Proc. Workshop der Fachgruppe: Vorgehensmodelle – Einführung, betrieblicher

Einsatz, Werkzeug- Unterstützung und Migration, 1997.

[Olveira.05] OLIVEIRA, Sandro e VASCONCELOS, Alexandre. Proposta de um ambiente de implementação de processo de software, 2005. Proposta de doutorado submetida ao Centro de Informática da Universidade Federal de Pernambuco.

[Paulk.93] PAULK, Mark C. & CURTIS, Bill & CHRISSIS, Mary Beth & WEBER, Charles V., Capability Maturity Model for Software, Version 1.1. Technical Report CMU/SEI-93-TR-024. Software Engineering Institute - Carnegie Mellon University, 1993.

[Pérez.96] PÉREZ, G., EL EMAN, K. & MADHAVJI, N. H., Evaluating the Congruence of a Software Process Model in a Given Environment, In Proc. International Conference on the Software Process, 1996.

[Reis.98] REIS, Carla Alessandra Lima. Um gerenciador de processos de software para o ambiente PROSOFT. Tese de Mestrado, PPGC-UFRGS, 1998.

[Reis.00a] REIS, Carla Alessandra Lima. Reutilização de processos de software. PPGC-UFRGS. Exame de Qualificação do Doutorado, 2000.

[Reis.00b] REIS, Carla Alessandra Lima. Ambientes de desenvolvimento de software e seus mecanismos de execução de processos de software. PPGC-UFRGS. Exame de Qualificação do Doutorado, 2000.

[Reis.02] REIS, R. Q., APSEE-Reuse: Um Meta-Modelo para Apoiar a Reutilização de Processos de Software, Tese de Doutorado, Instituto de Informática, Universidade Federal do Rio Grande do Sul, 2002.

[Rocha.01] ROCHA, Ana Regina Cavalcanti da & MALDONADO, José Carlos & WEBER, Kival Chaves, Qualidade de Software: Teoria e Prática, São Paulo: Prentice Hall, 2001.

[Rupprecht.00] RUPPRECHT, C., FÜNFFINGER, M., KNUBLAUCH, H. & ROSE, T., Capture and Dissemination of Experience About the Construction of Engineering Processes, In Proc. Conference on

Advanced Information Systems Engineering, 2000.

[Salviano.01] SALVIANO, C., CUNHA, M.A.V.C., CÔRTES, M.L, OLIVEIRA, W.L. SPICE in ROCHA, A.R.C., MALDONADO, J.C, WEBER, K.C. (eds) Qualidade de Software: Teoria e Prática. São Paulo, Prentice Hall, 2001

[Softex.06] Softex - Sociedade para Promoção da Excelência do Software Brasileiro, MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral, versão 1.1, 2006.