

UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2005.1

ANÁLISE DAS CARACTERÍSTICAS ORGANIZACIONAIS PARA A DEFINIÇÃO DE PROCESSOS DE SOFTWARE

TRABALHO DE GRADUAÇÃO
EM
ENGENHARIA DE SOFTWARE

Aluno – Márcio Bernardino Fernandes Lima da Cunha, mbflc@cin.ufpe.br.

Orientador – Alexandre Marcos Lins de Vasconcelos, amlv@cin.ufpe.br.

Co-Orientador – Sandro Ronaldo Bezerra Oliveira, srbo@cin.ufpe.br
(Aluno de Doutorado).

19 de Agosto de 2005.

UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2005.1

MÁRCIO BERNARDINO FERNANDES LIMA DA CUNHA

ANÁLISE DAS CARACTERÍSTICAS ORGANIZACIONAIS PARA A
DEFINIÇÃO DE PROCESSOS DE SOFTWARE.

ESTE TRABALHO FOI APRESENTADO À 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: PROF. ALEXANDRE MARCOS LINS DE VASCONCELOS
CO-ORIENTADOR: SANDRO RONALDO BEZERRA OLIVEIRA

19 de Agosto de 2005.

Dedico,

Aos meus pais
Arnaldo e Fátima

A minha irmã
Geórgia

© Márcio Bernardino Fernandes Lima da Cunha, 2005.
Todos os direitos reservados.

AGRADECIMENTOS

Aos meus pais, Arnaldo e Fátima, minha irmã Geórgia assim como toda minha família pelo amor, carinho, apoio, compreensão e permanente incentivo.

Ao aluno de doutorado Sandro Oliveira pela sua amizade, co-orientação e constante disponibilidade no decorrer deste trabalho.

Ao professor Alexandre Vasconcelos pela orientação e validação de cada etapa deste trabalho.

E, finalmente, a todos aqueles que contribuíram direta ou indiretamente para a realização deste trabalho.

Márcio Bernardino Fernandes Lima da Cunha

RESUMO

A definição de um processo de software deve ocorrer em três níveis de caracterização: Padrão, Especializado e Instanciado. A definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software. Esse processo deverá ser especializado considerando-se as características relacionadas ao tipo de software e ao paradigma de desenvolvimento utilizado. Por fim, a instanciação na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades. O objetivo deste trabalho de graduação é fazer o levantamento e a análise das características organizacionais que são consideradas na definição de um processo de software.

SUMÁRIO

1. INTRODUÇÃO	10
1.1. Contexto do Trabalho	10
1.2. Motivação	11
1.3. Objetivos	12
1.4. Metodologia do Trabalho	13
1.5. Estrutura do Trabalho	13
2. IMPPROS: UM AMBIENTE DE IMPLEMENTAÇÃO DE PROCESSO DE SOFTWARE.....	15
2.1. Processo de Software	15
2.2. Arquitetura do Ambiente	21
2.3. Considerações Finais.....	40
3. EMPRESAS DE SOFTWARE: UMA VISÃO GERAL	42
3.1. Introdução.....	42
3.2. Fábricas de Software.....	42
3.4. Classificações de Organizações Desenvolvedoras de Software com Outros Critérios	46
3.5. Adequação das Classificações de Organizações Desenvolvedoras de Software.....	50
4. ANÁLISE DE CARACTERÍSTICAS ORGANIZACIONAIS NA DEFINIÇÃO DE PROCESSO DE SOFTWARE AO AMBIENTE IMPPROS	53
4.1. Descrição da Proposta	53
4.2. Atores Envolvidos	54
4.3. Visões da Proposta.....	55
4.4. Fluxo de Atividades	56
4.4.1. Fluxo Principal.....	56
4.4.2. Fluxo Auxiliar Avaliar Regras.....	58
4.5. Características Organizacionais.....	60
4.5.1. Características do Processo de Software	61
4.5.2. Características do Ambiente de Desenvolvimento.....	62
4.5.3. Características da Equipe.....	63
5. VALIDAÇÃO DA ADEQUAÇÃO.....	64
5.1. Arquitetura de Implementação	64
5.1.1. Camadada de Apresentação.....	64

5.1.2. Camadada de Negócios	65
5.1.3. Camadada de Acesso a Dados	66
5.2. Protótipo.....	67
6. CONCLUSÕES	71
6.1. Sumário do Trabalho	71
6.2. Trabalhos Futuros	71
APÊNDICE A – CARACTERÍSTICAS ORGANIZACIONAIS, VALORES E DEFINIÇÕES.....	76
APÊNDICE B – MODELO DE DADOS RELACIONAL.....	86

LISTA DE FIGURAS

Figura 1 - Arquitetura do Ambiente para Implantação de Processo de Software.....	22
Figura 2 - Modelo de Adaptação do Meta-modelo de Processo.....	32
Figura 3 -Fluxo Principal de Atividades.....	58
Figura 4 - Fluxo das Sub-Atividades que Representam o Fluxo Interno da Macro- Atividade Avaliar Regras.....	60
Figura 5 - Arquitetura Proposta para a Implementação do Protótipo de Validação da Proposta.....	65
Figura 6 - Tela Manter Avaliação Consulta.....	68
Figura 8 - Tela Manter Avaliação Individual Avaliar	69
Figura 9 - Tela Manter Avaliação Avaliar	70

LISTA DE QUADROS

Quadro 1 – Relação entre os Tipos de Organizações Contemplados pela Pesquisa.....	52
Quadro 2 – Características Organizacionais.....	76

1. INTRODUÇÃO

1.1. Contexto do Trabalho

O desenvolvimento de um software de qualidade, com elevada produtividade, dentro do prazo estabelecido e sem necessitar de mais recursos que aqueles alocados tornou-se um desafio para as organizações [MACHADO_2000]. O crescimento do tamanho e da complexidade dos produtos de software faz com que estes assumam papéis críticos nos negócios das organizações. A experiência da indústria de software mostra que a principal razão para o insucesso dos projetos está na falta de um processo de software disciplinado, ou seja, na falta de um mecanismo que habilite o gerenciamento e controle da qualidade dos produtos. Seguindo a mesma tendência, já é amplamente aceito que a qualidade de um produto de software seja fortemente determinada pela qualidade do processo utilizado durante o seu desenvolvimento e manutenção.

Conhecer os processos significa conhecer como os produtos são planejados e produzidos. Cabe ressaltar que, a partir da definição do processo, é possível definir-se medições e coletar dados de execução.

Isto dá visibilidade aos gerentes e técnicos sobre o andamento dos projetos, possibilitando ações para controlar as variações do projeto e dos processos por ele utilizados. Neste sentido, a definição de um processo de software é um requisito básico para a obtenção de produtos de software de qualidade.

Quando são definidos, processos devem levar em consideração especificidades do projeto em questão. Embora diferentes projetos requeiram processos com características específicas para atender às suas particularidades, é possível estabelecer um conjunto de ativos de processo de software (software process assets) comum a todos os processos de software de uma organização, constituindo um Processo Padrão da organização, a partir do qual podem ser definidos processos específicos de projetos, levando em conta suas características particulares. Contudo, não basta definir o processo padrão, é importante que se defina, também, os mecanismos de adaptação desse processo para cada projeto. Além disso, depois de definidos o processo padrão e os procedimentos de adaptação, é necessário que sejam definidos mecanismos de realimentação do processo padrão para garantir sua atualização contínua, seja por falhas encontradas nos ativos do processo ou por necessidade de incorporação de uma nova metodologia ou tecnologia.

1.2. Motivação

A definição de um processo de software deve ocorrer em níveis de caracterização (Processo Padrão, Processo Especializado e Processo Instanciado) para melhor permitir uma análise das diferentes características que levam a sua composição para um projeto ou organização específica. 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. 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). Por fim, a instanciação para projetos específicos consiste na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades.

Sendo assim, a identificação das Características Organizacionais é essencial para a definição do processo de software padrão de uma organização.

1.3. Objetivos

Este trabalho tem como principal meta fazer o levantamento e analisar as características organizacionais que devem ser levadas em consideração para a definição de um processo de software. Para tanto se faz necessário o uso de características como: práticas de engenharia de

software, cultura organizacional, características de desenvolvimento de software na organização (modelo de maturidade, nível de maturidade, tipo de ambiente de desenvolvimento de software). O resultado é um mapeamento das características organizacionais que influenciam em um processo de software.

1.4. Metodologia do Trabalho

As informações necessárias para a realização deste trabalho foram obtidas através de pesquisa que foram fundamentadas em artigos publicados recentemente, livros de autores consagrados na área e informações obtidas de instituições com competência na área.

1.5. Estrutura do Trabalho

Além deste capítulo introdutório, o trabalho é composto por mais cinco capítulos. O capítulo 2 descreve o contexto no qual está inserido este trabalho vislumbrando definições relacionadas a processos de software e detalhando o ImPProS, Ambiente para Implantação Progressiva de Processo Software. No capítulo 3 encontramos classificações para as organizações que desenvolvem software. No quarto capítulo apresentam-se as características organizacionais e todo o fluxo de atividades envolvido no módulo de definição de processos relativos a esse tipo de

característica. No capítulo 5 encontra-se a validação da proposta apresentada com um protótipo. No último capítulo a conclusão do trabalho será realizada mostrando um sumário do trabalho e citando possíveis trabalhos futuros.

2. IMPPROS: UM AMBIENTE DE IMPLEMENTAÇÃO DE PROCESSO DE SOFTWARE

2.1. Processo de Software

O primeiro passo em busca da produção de software em larga escala ocorreu na década de 70, quando o aumento da demanda e da complexidade dos produtos de software e conseqüente aumento das equipes de desenvolvimento resultaram no desenvolvimento e utilização de métodos estruturados e de ferramentas automatizadas. Com o passar do tempo a indústria de software percebeu que a simples utilização de ferramentas não significava, necessariamente, uma garantia de sucesso para um projeto de software. Os insucessos nos projetos, normalmente, não eram decorrentes da implantação de novas tecnologias e do uso de ferramentas, mas sim, da ausência de um processo de software disciplinado [MACHADO_00]. Experiências demonstram que o uso de novas tecnologias e ferramentas em um ambiente de desenvolvimento imaturo ou não disciplinado tem levado ao aumento dos problemas já existentes [ZAHRAN_98].

A produção de software é um processo que envolve atividades complexas as quais são realizadas por pessoas com capacidades distintas. Uma forma de analisar e amadurecer tal processo é através da sua descrição, a qual consiste de um modelo de processo de software [OLIVEIRA_00]. A descrição formal de um processo de software é uma atividade que permite que o mesmo seja analisado, compreendido e automatizado (executado). Neste sentido, a modelagem e a execução de processos de software são de fundamental importância para o aumento da qualidade do produto de software.

De acordo com [HUMPHREY_89], podemos definir processo de software como sendo o conjunto de tarefas de engenharia de software necessárias para transformar os requisitos dos usuários em software. Um processo de software é formado por um conjunto de passos de processo parcialmente ordenados, relacionados com conjuntos de artefatos, pessoas, recursos, estruturas organizacionais e restrições e tem como objetivo produzir e manter os produtos de software finais requeridos [REIS_98].

Os passos de processos são divididos em atividades ou tarefas (tarefas são passos de processo gerenciados e atividades são passos elementares, que conduzem à realização de uma tarefa, e não são gerenciadas). Uma atividade é um passo de processo que produz mudanças de estado visíveis externamente no produto de software. Atividades incorporam e implementam procedimentos, regras e políticas, e têm como objetivo gerar ou modificar um dado conjunto de artefatos.

Uma atividade aloca recursos (por exemplo, máquinas e orçamento), é escalonada, monitorada e atribuída a desenvolvedores (agentes), que podem utilizar ferramentas para executá-la. Uma atividade também pode ser executada somente por ferramentas automatizadas, sem intervenção humana. Neste caso, a atividade é automática. Toda atividade possui uma descrição, a qual pode especificar os artefatos necessários, as relações de dependência com outras atividades, a data de início e fim planejadas, os recursos a serem alocados e os agentes responsáveis pela mesma.

Um agente está relacionado com as atividades de um processo e pode ser uma pessoa ou uma ferramenta automatizada (quando a atividade é automática). Os agentes podem estar organizados em cargos. Diferentes agentes terão percepções acerca do que acontece durante o processo de software. Um agente, por exemplo, perceberá os aspectos de controle e alocação de recursos e cronogramas para atividades, enquanto um desenvolvedor perceberá as suas atividades como atribuições que devem ser feitas para produzir um resultado (possivelmente através de agendas, onde as atividades em que está envolvido são relacionadas).

Um artefato é um produto criado ou modificado durante um processo. Tal produto é resultado de uma atividade e pode ser utilizado posteriormente como matéria-prima para a mesma ou para outra atividade a fim de gerar novos produtos. Desta forma, uma atividade pode consumir produtos (de entrada) e gerar novos produtos (de saída).

Os produtos são freqüentemente persistentes e possuem versões [REIS_98].

A realização do processo é afetada pelas restrições, que podem atingir atividades, agentes, recursos, artefatos, papéis e seus relacionamentos. Uma restrição é uma condição definida que um passo de processo deve satisfazer antes ou depois de ser executado.

A tecnologia de processo de desenvolvimento de software (ou, simplesmente, processo de software) é um dos mais recentes tópicos de estudo, pesquisa e prática na Engenharia de Software. Esta tecnologia baseia-se no conceito de que o processo de software seja descrito formalmente, a fim de propiciar a sua automação, verificação, aperfeiçoamento e reutilização. Suas origens estão relacionadas com as primeiras propostas de um ciclo de vida para o software, isto é, a definição de um conjunto bem definido de estágios que forneçam uma referência para o acompanhamento do desenvolvimento.

A tecnologia CASE, que culminou no surgimento de ambientes integrados de desenvolvimento de software (ambientes de desenvolvimento de software - ADS), estabeleceu a cultura da utilização de ferramentas para auxiliar a automação do desenvolvimento. Assim, os ADSs fornecem uma base de integração para a utilização de ferramentas CASE durante todo o processo de software.

Mais recentemente, surgiu a necessidade de incorporar mecanismos para o controle do processo de desenvolvimento através dos ADSs: os PSEEs (Process-Centered Software Engineering Environments ou

ADSs Orientados ao Processo) surgiram neste contexto, permitindo a descrição do processo e o acompanhamento das atividades pelo ambiente.

Um modelo de processo de software é uma descrição abstrata do processo de software. Vários tipos de informações devem ser integradas em um modelo de processo de software para indicar quem, quando, onde, como e por que os passos são realizados. Para representar um modelo de processo de software é utilizada uma linguagem de modelagem do processo de software, a qual deve oferecer recursos para descrever e manipular os passos do processo.

Um modelo do processo instanciado ou processo executável é um modelo de processo pronto para execução. Este modelo pode ser interpretado por uma máquina de processo, a qual é um componente provido por um ambiente de desenvolvimento de software (ADS) com capacidade de suportar a modelagem e a execução de processos de software (ADS Orientado a Processo).

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 a 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_95]. Assim, na definição de um processo deve-se 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.

Em uma organização ou grupo de desenvolvimento multi-organizacional, diversos projetos podem coexistir possuindo características específicas. Porém, existe um conjunto de elementos fundamentais que se deseja sejam incorporados em qualquer processo definido. Estes elementos fundamentais são definidos como o processo padrão, ou seja, 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, 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.

2.2. Arquitetura do Ambiente

Conforme descrito em [Oliveira_05], o ImPProS, Ambiente para Implantação Progressiva de Processo Software, está sendo desenvolvido tendo como principal objetivo dar apoio à implantação de um processo de software nas organizações de forma progressiva. São ainda conferidos a ele alguns objetivos específicos, como:

- 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 tomando como base uma máquina de inferência;
- Possibilitar a avaliação dos critérios do processo de software;
- Apoiar a melhoria contínua do processo de software e o reuso através da realimentação e coleta de experiências;
- 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 existente, para uso do Ambiente em seus serviços providos.

Para alcançar estes objetivos o ambiente foi projetado com a arquitetura apresentada na Figura 1.

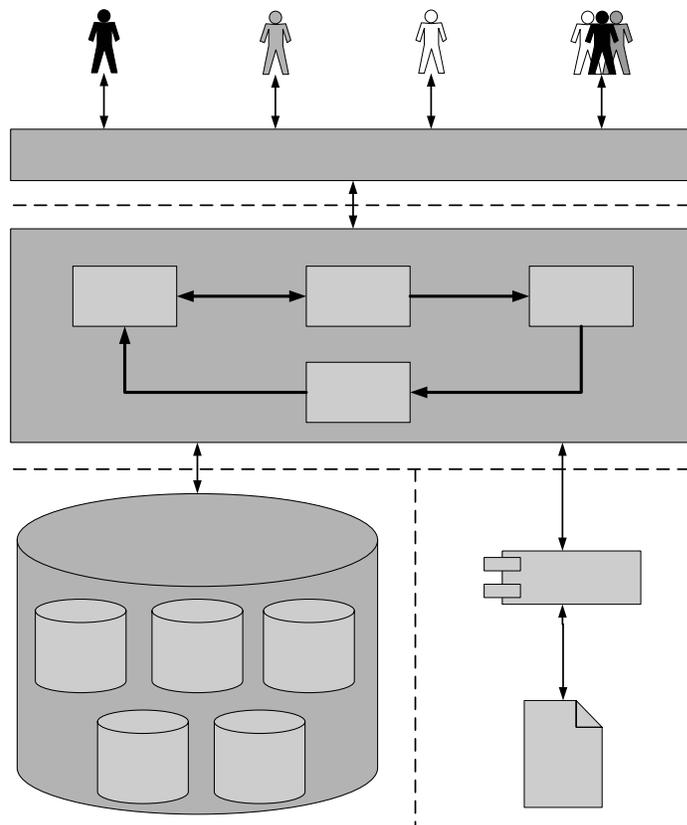


Figura 1 - Arquitetura do Ambiente para Implantação de Processo de Software

Conforme visto na Figura 1, o ambiente é composto por quatro mecanismos integrados. O Mecanismo de Interação com o Usuário é responsável pela interação do ambiente com o usuário, apresentando um único componente: o Módulo de Interação e Visualização do Ambiente que tem como responsabilidade provê mecanismos de interação para diferentes usuários do ambiente (por exemplo, agendas para equipe de desenvolvimento e facilidade de visualização de processo para projetistas/gerentes);

A persistência de todos os dados manipulados pelo ambiente é garantida pelo Mecanismo de Repositório do Ambiente, o qual possui os seguintes componentes:

- Meta-Modelo de Processo de Software: componente que contém a estrutura que indica as características de um processo de software (modelos de ciclo de vida, procedimentos, atividades, ferramentas, modelos de documentos e políticas) e seus relacionamentos baseada em definições de modelos e normas de qualidade para processo de software (CMMI, SPICE – ISO 15504, ISO 9000-3). Serve na verdade como um grande repositório indicando uma terminologia comum para a definição de processos de software de acordo com características dos projetos de software e da organização;

- Características de Definição do Processo: componente composto por regras que caracterizam projetos de software e aspectos organizacionais, possibilitando ajuda na definição de um processo de software de acordo com os aspectos especificados na Figura 2.
- Modelo de Processo da Organização: funciona como um repositório que armazena todos componentes de um processo de software para uma organização. Aqui o foco está em armazenar o resultado dos diferentes níveis de definição de um processo de software (Processo Padrão, Processos Especializados e Processos Instanciados), como visto na Figura 2, para prover reuso destes em diferentes projetos de software;
- Repositório de Experiências: repositório contendo os conhecimentos formal e informal obtidos nos projetos, organizados de forma a poderem ser reutilizados na definição de outros processos ou para melhoria contínua dos processos existentes. É importante realçar que a construção do repositório de experiências será apoiada em uma ontologia de processo de software. Essa ontologia foi utilizada para tornar homogêneo o vocabulário, facilitar o compartilhamento do conhecimento sobre processos, e simplificar a busca de experiências. Assim, a descrição dos objetos da base de experiência é feita usando os termos da ontologia e buscas feitas com sinônimos são transformadas para o vocabulário da ontologia;

- Conhecimento do Processo: componente que contém os termos da ontologia (mesma do Repositório de Experiências), usados para definir o processo de software especificado como guia para a execução do projeto de software específico da organização, pronto para ser gerenciado (executado) e / ou simulado. Este componente permite o controle das versões do processo de software a fim de possibilitar modificações nos seus componentes;

A necessidade de integração do ambiente ImPProS com o ambiente de desenvolvimento já existente nas organizações fez surgir o Mecanismo para a Integração de Ferramentas ao Ambiente, o qual dispõe dos seguintes componentes:

- Kernel do Ambiente: componente que possibilita a integração de ferramentas de apoio à execução do serviço no ambiente. Funciona como um tradutor das mensagens (funcionalidades executadas) a serem passadas entre o Ambiente e as Ferramentas de Apoio a fim de solicitar a execução de uma dada funcionalidade destas ferramentas integradas, bem como coletar as informações geradas por esta execução para garantir os serviços propostos pelo ambiente;
- Ferramentas de Apoio: representa as ferramentas que darão suporte automatizado à execução de alguns serviços providos pelo Ambiente. Assim, tem-se o entendimento de que os serviços

providos por alguns módulos definidos acima são de responsabilidade das ferramentas aqui especificadas, fazendo com que seja necessária à integração destas ao Ambiente. Como serviços do Ambiente desempenhados por estas ferramentas pode-se listar: Ferramenta de Avaliação da Organização para definir a maturidade desta no uso de processo de software; Ferramenta de Ontologia a fim de possibilitar a geração das ontologias que servirão como regras para a definição, execução, reuso e melhoria contínua do processo de software; Ferramenta de Modelagem de Processo a fim de possibilitar, através de uma notação de processos, a representação diagramática (visualização) dos componentes que representam o processo de software; Ferramenta de Simulação de Processo, capaz de antever resultados de execução dos processos definidos; Máquina de Inferência, capaz de interpretar os termos das ontologias usadas para definir o processo de software, as características dos projetos de software e da organização, e as experiências obtidas nos projetos. É importante ressaltar que todas as ferramentas usadas como apoio deverão possuir a característica de ser free e open source a fim de possibilitar uma difusão do ambiente independente de restrições orçamentárias, bem como para possibilitar uma integração adequada entre o ambiente e as ferramentas. As ferramentas definidas acima servem de suporte ao gerenciamento do Processo de Software no Ambiente, no entanto é válido especificar que além

dessas, serão de extrema relevância o apoio de ferramentas cujo foco seja o Desenvolvimento do Projeto de Software (Ferramentas de Edição, Documentação, Métricas, Gerenciamento de Projetos, etc.), as quais darão suporte total ao Módulo de Execução do Processo.

Além dos mecanismos acima citados, compõem a arquitetura desta ferramenta os Mecanismos para o Gerenciamento do Processo no Ambiente, subdivididos nos seguintes módulos:

- Módulo de Definição do Processo: possibilita a definição do processo de software nos seus níveis de caracterização (Processo Padrão, Processo Especializado e Processo Instanciado), conforme visto na Figura 2; a sua modelagem a partir de uma linguagem de representação diagramática no ambiente permitindo a visualização do processo definido e o relacionamento dos seus componentes; a sua instanciação através de regras de ontologias de processo de software que especificam o domínio de atuação; e possibilita funções para melhoria contínua do processo de software através da coleta e filtragem de experiências obtidas;
- Módulo de Simulação do Processo: permite o refinamento dos modelos de processos, permitindo antever resultados da realização do projeto a partir da condução de experimentos baseados em

computador a fim de descrever, explicar e prever o comportamento de um sistema real;

- Módulo de Execução do Processo: coordena as atividades do processo em execução, podendo executar processos incompletos e permitindo alteração do processo durante a execução;
- Módulo de Avaliação do Processo: provê uma avaliação em paralelo à execução do processo de software pela equipe de desenvolvimento, descrevendo o desempenho dos componentes definidos para a estrutura do processo.

A relação deste trabalho com o ambiente está inserida no Módulo de Definição do Processo, sendo assim esse módulo merece aqui um detalhamento especial. Abaixo se tem a finalidade de cada um dos quatro sub-componentes desse módulo:

- Definidor de Processos: componente responsável por gerenciar os requisitos dos usuários quanto à definição de um processo de software quanto ao domínio de um projeto específico ou às características de uma organização. Neste componente o Projetista do Processo possui duas visões de execução:
 - Manutenção do Processo e das Características dos Projetos e da Organização: uma vez que o meta-modelo é a base para a instanciação de processos de projeto de software, é

necessário manter seus ativos (ou componentes). Tomando por base alguns modelos e normas de qualidade para processos de software (CMMI, SPICE – ISO 15504, ISO 9000–3) o meta-modelo deve ser composto dos seguintes elementos:

- Modelos de ciclo de vida – estruturas que definem as macro-atividades de um processo, sua precedência e dependência. Para cada projeto, deve-se selecionar um modelo dentre o conjunto de modelos de ciclo de vida definidos como válidos no Ambiente;
- Atividades – ações que transformam artefatos de entrada (insumos) em artefatos de saída (produtos);
- Modelo de Documento – documentos padrão que contêm a formatação e as orientações para a elaboração de um artefato do projeto;
- Procedimentos – condutas bem estabelecidas e ordenadas para a realização de atividades;
- Ferramentas – recursos de software utilizados para apoiar a realização de atividades, ou seja, dizem respeito às ferramentas de apoio ao Desenvolvimento do Projeto;
- Políticas – direções que regem a organização.

Além dos ativos do processo, o Projetista do Processo tem a possibilidade de manter novas regras que possibilitarão a definição do processo. Estas regras são baseadas nas características dos projetos (Tipo de Software, Paradigma, etc.) e da organização (Ambiente de Desenvolvimento, Equipe, etc.), mantidas no repositório “Características de Definição do Processo”, visto na Figura 1.

- Adaptação do Processo: esta visão apóia o Projetista do Processo na adaptação do meta-modelo de processo de software para cada projeto, apresentando sugestões, tanto de modelo de ciclo de vida, quanto de atividades, procedimentos e ferramentas, entre outros. No entanto, o projetista é livre para acatar, ou não, as sugestões fornecidas pelo ambiente. Porém, para garantir a qualidade da adaptação, o Projetista de Processo deve realimentar o sistema com seus comentários, que podem ser críticas ou justificativas, sobre a orientação que recebeu. Sabe-se que, à medida que o produto de software vai sendo desenvolvido, uma visão mais precisa das necessidades reais do projeto, sejam ferramentas, documentos ou mesmo treinamentos, fica mais precisa. Portanto, oferece-se ao projetista, também, um meio para alterar o processo do projeto ao longo do desenvolvimento do produto. A Figura 2 mostra um

esquema do modelo utilizado pelo Ambiente na adaptação do meta-modelo de processo, composto por três atividades principais: Definição do Processo Padrão, Especialização do Processo e Instanciação do Processo.

Inicialmente enxerga-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 15504, ISO 9000-3). O objetivo deste Meta-modelo é determinar uma terminologia única para a definição de processos de software no Ambiente. Embora este meta-modelo surja da união dos modelos / normas de qualidade de processo de software especificados na Figura 3, o Ambiente deve ser capaz de permitir a manutenção deste meta-modelo a fim de atender as necessidades de futuras modificações (novas versões) nestes modelos / normas ou a inclusão de características de outros modelos / normas. Este meta-modelo é mantido no repositório “Meta-Modelo de Processo de Software”, visto na Figura 1.

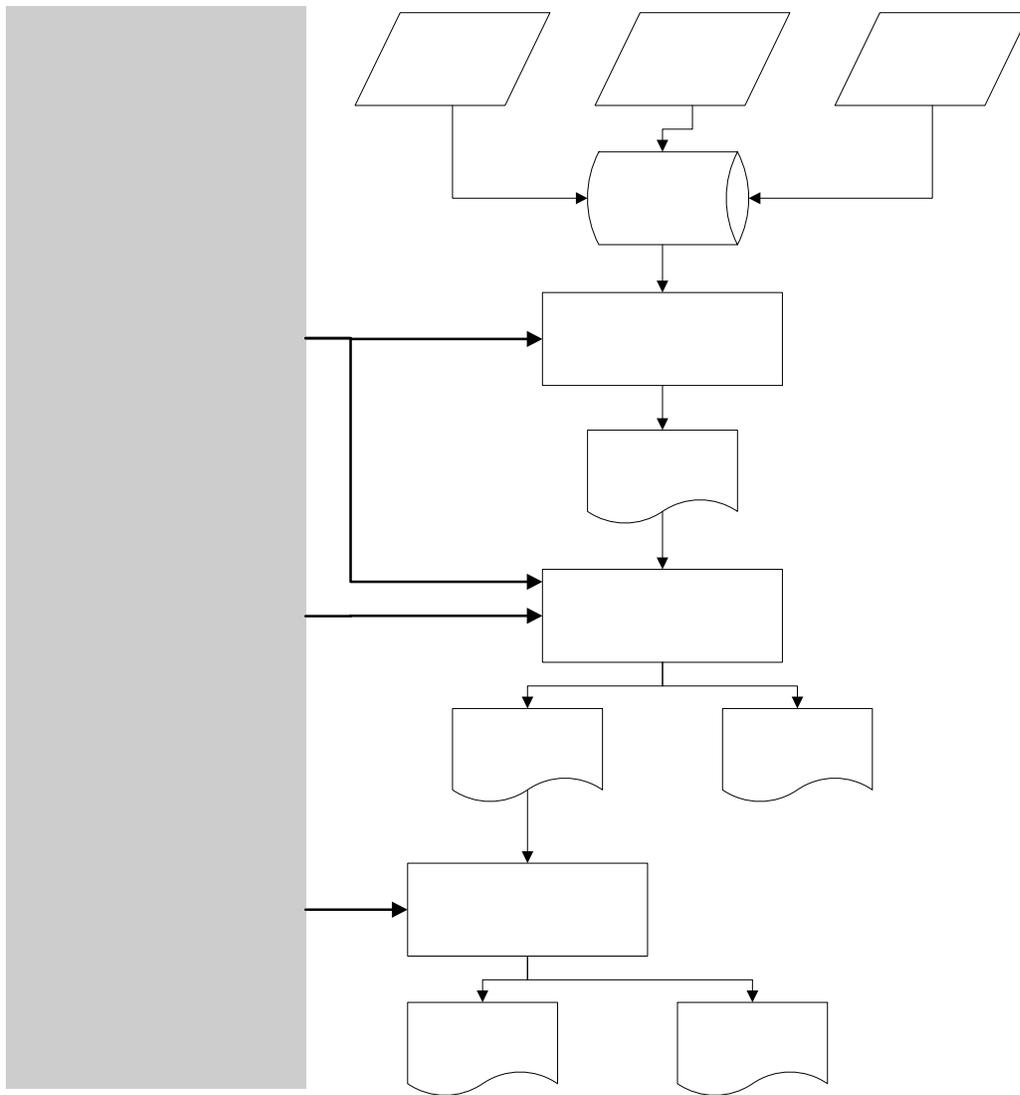


Figura 2 - Modelo de Adaptação do Meta-modelo de Processo

Prática de Engenharia de Software
Cultura Organizacional
Características de Desenvolvimento
de Software na Organização
(Modelo de Maturidade, Nível de
Maturidade, Tipo de Ambiente de
Desenvolvimento de Software)

Por sua vez, a definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software. Essa base para a definição de todos os processos. 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.

Nesse modelo, a definição do processo padrão pode ser realizada tendo como base o meta-modelo de processo de software e as características do desenvolvimento de software na organização. A definição deverá considerar uma Ferramenta de Avaliação da Organização a fim de inferir o status da maturidade e capacitação das atividades da organização, especificando as necessidades do processo a ser definido. O nível de maturidade deve ser selecionado para se saber que atividades do meta-modelo de processo considerar. É possível selecionar um nível fixo de maturidade escolhendo quais áreas-chave devem ser atingidas naquele projeto.

O tipo de software a ser desenvolvido tem impacto no processo de software a ser definido. Sistemas para a Web, por exemplo, requerem que o processo de desenvolvimento inclua atividades que tratem adequadamente a interação homem-máquina. No Ambiente, os produtos de software podem ser classificados nos seguintes tipos, segundo a literatura especializada [BORGES_02]: (i) Sistemas de tempo-real; (ii) Sistemas de informação; (iii) Software básico e (iv) Sistema para Web. Esta classificação foi baseada no histórico

de produtos desenvolvidos pela organização, assim como prospecções de seus rumos.

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). Assim, durante a etapa de especialização do processo padrão, atividades poderão ser adicionadas ou modificadas, de acordo com o contexto para qual se está realizando a especialização.

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 é utilizada 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. Assim, como resultado desta fase é gerada uma instância do processo contendo os componentes necessários para o processo de software a fim de representar o projeto específico a ser desenvolvido, atendendo desta forma todas as características deste desenvolvimento.

Na etapa de seleção de atividades, as atividades do processo de desenvolvimento devem ser selecionadas e mapeadas no modelo de ciclo de vida escolhido. A partir das atividades de construção do modelo de ciclo de vida, são definidas atividades de gerência e de controle da qualidade. Em todos os casos, para cada atividade, deve-se definir sub-atividades, procedimentos, recursos e ferramentas.

As atividades de gerência são aquelas relacionadas ao planejamento e acompanhamento gerencial do projeto e são selecionadas de acordo com o nível de qualidade e/ou áreas-chave escolhidas para o projeto. As atividades de construção estão diretamente relacionadas ao processo de construção do software. Sua seleção é, portanto, muito relacionada às características do projeto, assim como ao modelo de ciclo de vida selecionado.

As atividades de avaliação da qualidade dizem respeito à garantia da qualidade do produto em desenvolvimento ou ao processo utilizado nesse desenvolvimento e são selecionadas de acordo com o nível de qualidade e/ou áreas-chave escolhidas para o projeto.

Todas as características especificadas anteriormente que servem para a Definição do Processo Padrão, Especialização e Instanciação do Processo são mantidas como regras no repositório “Características de Definição do Processo”, conforme visto na Figura 1.

- Analisador dos Requisitos do Processo: Este componente possui duas finalidades-chave:
 - Prover melhoria contínua no processo de software;
 - Possibilitar o reuso de processos definidos para a organização.

Para o atendimento dessas finalidades, a coleta de experiências é realizada ao longo do desenvolvimento do software e é de responsabilidade do Projetista do Processo. Cabe a ele alimentar a base de experiências com o conhecimento formal, ou seja, aquele que está contido nos documentos do projeto, assim como o conhecimento informal, e as lições aprendidas.

O Ambiente fornece meios de armazenar as lições aprendidas, através do registro de idéias, fatos, questões, pontos de vista, conversas, discussões, decisões etc, que aconteceram no decorrer de um projeto. Uma vez que nem todos os relatos são úteis à organização, cabe ao Gerente do Processo fazer uma triagem dos relatos a serem disseminados e adaptar esse conhecimento, categorizando o problema e a solução adotada.

Os artefatos produzidos ao longo do desenvolvimento do projeto são o meio formal, normalmente utilizado para documentar o conhecimento adquirido ao longo do projeto. É importante que esse conhecimento formal seja coletado e compartilhado para que a organização aprenda coletivamente.

O conhecimento é armazenado em um componente “Repositório de Experiências”, visto na Figura 1, no seguinte formato:

- a) Projeto – projeto a qual pertence;
- b) Objeto – item do processo ao qual a lição está associada, tais como atividade, artefato, procedimento, ferramenta, etc.
- c) Tipo de lição aprendida – identifica se essa lição foi um ponto positivo (boa prática) ou um ponto negativo (oportunidade de melhoria);
- d) Categoria de problema – categorias fixas pré-definidas de problemas, tais como gerência de projeto, planejamento de projeto, gerência de requisitos, etc.

- e) Problema – descrição do problema;
- f) Solução – descrição da solução adotada para o problema;
- g) Contexto – descrição da situação onde o problema/solução é relevante.

Assim, são analisados os requisitos (componentes do processo) identificados para atividades de projeto de um novo processo, ou novos requisitos para um processo existente. Os requisitos resultantes especificam os recursos e propriedades que o processo deve oferecer. A partir da análise concluída e validada pelo usuário, o componente permite a incorporação de novos requisitos (experiências obtidas com a definição, especialização e instanciação do processo) a fim de possibilitar melhorias para processos existentes e novos processos.

Além da finalidade relacionada à melhoria contínua do processo de software a partir da coleta de experiências, este componente permite interagir juntamente com o componente “Definidor de Processo” provendo reutilização de processos de software já existentes no repositório “Modelo de Processo da Organização”, visto na Figura 1. Este procedimento dá-se a partir das características (de projetos e organizacionais) especificadas para a instanciação do processo, identificadas como comuns para novos e processos existentes.

- Modelador do Processo: este componente provê a arquitetura geral e detalhada do processo, usando a representação diagramática. Nesta etapa uma linguagem de modelagem de processo é utilizada, a qual deve ser capaz de representar através de suas notações os componentes definidos ao processo. Aqui, tem-se um serviço automatizado que permitirá ao final de sua execução uma validação do usuário acerca do objeto (processo) projetado. Embora seja automatizado, este serviço pode ter a intervenção do usuário (Projetista do Processo), caso sinta necessidade de modificação do processo projetado. Qualquer modificação no modelo gráfico do processo deve acionar o Ambiente a fim de prover alteração nas regras de composição dos componentes e dos relacionamentos do processo de software. Aqui, uma Ferramenta de Modelagem de Processo que permita o uso dessa linguagem deve ser integrada e utilizada ao Ambiente a fim de automatizar a execução deste componente.
- Instanciador do Processo: neste componente é gerado um modelo de processo instanciado, contendo informações detalhadas sobre os prazos, agentes e recursos utilizados por cada atividade definida no processo, ou seja, aqui um dos focos é o desenvolvimento de um plano de processo para um projeto específico de acordo com os componentes especificados para a sua composição. Todas estas informações, juntamente com o processo, são armazenadas no repositório “Conhecimento do Processo”, visto

na Figura 1, a fim de serem simulados e posteriormente executados. A instanciação se dá a partir do uso de ontologias que servirão como regras para execução e / ou simulação do processo de software. Desta forma, faz-se necessário à integração à estrutura do Ambiente de uma Ferramenta de Ontologias permitindo a definição das mesmas, para a automação deste procedimento.

Cada ontologia origina um módulo de conhecimento. Além disso, cada uma das instanciações de uma ontologia específica também dá origem a um componente. Uma instanciação de uma ontologia é uma descrição de fatos do domínio usando os conceitos e as relações descritos na ontologia. Estes módulos de conhecimento devem ser descritos em uma linguagem que possa ser interpretada por uma Máquina de Inferência que será integrada ao Ambiente para a execução do processo de software. A base de conhecimento de uma aplicação que se comprometa com uma ou várias ontologias deverá ser a conjunção dos módulos de conhecimento correspondentes, mais o conhecimento específico para esta aplicação.

2.3. Considerações Finais

O objetivo principal deste capítulo foi introduzir a temática e o contexto em que se encontra este trabalho. Na primeira seção foi

realizada uma descrição sobre processos de software, destacando a importância de um processo bem definido e de ferramentas que auxiliem as organizações que desenvolvem software nesse sentido. Em seguida, na segunda seção, foi visto um detalhamento da arquitetura do ambiente no qual este trabalho está envolvido, enfatizando claramente o Módulo de Definição do Processo.

3. EMPRESAS DE SOFTWARE: UMA VISÃO GERAL

3.1. Introdução

Neste capítulo serão encontradas classificações para organizações que desenvolvem software. Inicialmente, será visto um importante grupo dentro dessas organizações, as Fábricas de Software. Em seguida, classificações de acordo com: Atividades no Tratamento de Software, Principais Tipos de Software Desenvolvidos e Atividades Características das Empresas em Tecnologia da Informação serão vislumbradas em conjunto com uma relação entre esses diversos tipos de classificações.

3.2. Fábricas de Software

Segundo [CUSUMANO_89], o termo Fábrica de Software vem sendo discutido desde o final dos anos 60, e evoluindo e se refinando até os dias atuais.

Um processo fabril constitui-se na produção de produtos em massa, incluindo operações centralizadas de larga escala, tarefas simples e padronizadas, controles padronizados, trabalhadores especializados,

mas com poucas habilidades, divisão de trabalho, mecanização e automação do processo. Desta forma, a associação do termo fábrica ao desenvolvimento de software sugere que se apliquem técnicas para produção em larga escala, de forma coordenada e com qualidade.

Desde 1968 alguns estudos publicados associam ainda características como reusabilidade, utilização de ferramentas para suportar o desenvolvimento, sistemas de controle e gerenciamento, modularização e produção de famílias de produtos como básicas para uma organização que se intitula uma Fábrica de Software.

Mais recentemente, [GREENFIELD_03] nos apresenta uma visão semelhante, onde o conceito de Fábrica de Software está fundamentado no desenvolvimento baseado em componentes, direcionado a modelos e a linhas de produto de software que caracterizariam uma iniciativa de fábrica, visando tornar a montagem de aplicações mais barata através de reuso sistemático, possibilitando a formação de cadeias de produção.

Já [FERNANDES_04] apresenta fábricas de software como “Um processo estruturado, controlado e melhorado de forma contínua, considerando abordagens de engenharia industrial, orientado para o atendimento a múltiplas demandas de natureza e escopo distintas, visando à geração de produtos de software, conforme os requerimentos documentados dos usuários e/ou clientes, da forma mais produtiva e econômica possível”.

Este conceito baseia-se em alguns atributos básicos que o autor advoga como imprescindíveis em qualquer Fábrica de Software, seja qual

for a sua categorização. Alguns destes atributos são: processo definido e padrão (desenvolvimento, controle e planejamento); interação controlada com o cliente (entradas e saídas da fábrica); solicitações de serviço à fábrica devem ser padronizadas; estimativas de custos e prazos baseadas no conhecimento real da capacidade produtiva com métodos de obtenção baseados em dados históricos; controle rigoroso dos recursos envolvidos em cada demanda da fábrica; controle e armazenamento em bibliotecas de itens de software (documentos, código, métodos, etc); controle dos status e execução de todas as demandas; produtos gerados de acordo com os padrões estabelecidos pela organização; equipe treinada e capacitada nos processos organizacionais e produtivos; controle da qualidade do produto; processos de atendimento ao cliente; métricas definidas e controle dos acordos de nível de serviço definidos com o cliente.

3.3. Classificação das Fábricas de Software

Uma vez definido o conceito de Fábricas de Software e citadas algumas de suas características, as mesmas serão classificadas a fim de que possa ser caracterizado o mapeamento dos processos.

A classificação adotada será a de [FERNANDES_04], onde são apresentados quatro tipos de fábricas classificadas de acordo com o seu escopo de atuação ao longo das fases de desenvolvimento de um projeto de Software. Estes quatro tipos são: Fábrica de Projetos Ampliada,

Fábrica de Projetos de Software, Fábrica de Projetos Físicos e Fábrica de Programas.

O autor também cogita a existência de uma fábrica de testes, que englobaria apenas a fase de teste integrado. Porém, para o objetivo deste trabalho, a fábrica de testes se enquadraria na mesma classificação da fábrica de programas que será definida a seguir.

A Fábrica de Projetos Ampliada abrange além das atividades encontradas em uma Fábrica de Projetos de Software, soluções mais abrangentes de Tecnologia da Informação como arquitetura da solução, modelagem do negócio, auditoria de qualidade e outras atividades de suporte ao desenvolvimento de projetos.

A Fábrica de Projetos de Software abrange além das atividades inerentes à fábrica de programas, fases como projeto conceitual, especificação lógica, projeto detalhado da solução, realização de testes de integração e de aceitação. É ainda necessário o conhecimento do negócio do cliente.

A Fábrica de Projetos Físicos abrange as mesmas atividades encontradas na Fábrica de Projetos de Software exceto as atividades de projeto conceitual e especificação lógica.

A Fábrica de Programas consiste na menor unidade de fábrica, conseqüentemente a menos complexa. Tem por objetivo principal codificar e testar programas de computador. No seu processo produtivo engloba praticamente as fases de construção e testes unitários

O modelo de outsourcing de sistemas é citado como uma especialização da fábrica de projetos, onde a mesma é dedicada exclusivamente a um determinado cliente, diferenciando apenas na interface da fábrica com o cliente, que deve ser adaptada aos critérios e regras estabelecidas previamente entre ambos, normalmente através de um SLA - Service Level Agreement - que descreve estes critérios, restrições e procedimentos de mudança no escopo e na avaliação do serviço [ASSANO_02]. O modelo de outsourcing apresentado pelo autor é bem mais complexo e envolve outros processos auxiliares, que se tornam essenciais para esta modalidade.

Merece destaque ainda a fábrica de componentes, porém, segundo [BASILI_94] os processos de desenvolvimento atuais não suportam diretamente a utilização de componentes, uma vez que definem apenas as fases de desenvolvimento do projeto, que utilizam os componentes já catalogados. Desta forma, o mapeamento não se estenderá a esta classe de Fábricas de Software. Entretanto, os processos devem ser escolhidos de forma a utilizarem e tirem as vantagens inerentes da reutilização de artefatos durante todo o ciclo de vida.

3.4. Classificações de Organizações Desenvolvedoras de Software com Outros Critérios

Além da classificação mostrada na seção anterior, baseada no escopo de fornecimento de Fábricas de Software, essa pesquisa identificou outras classificações para as organizações que desenvolvem software. Baseado em [MCT_00] foram encontradas três formas de se caracterizar as organizações que desenvolvem software. Cada um dos itens dessas classificações foi definido e limitado de acordo com as necessidades deste trabalho e do Ambiente para Implantação Progressiva de Processo Software. Seguem abaixo essas classificações:

- Classificação das empresas, segundo Atividades no Tratamento de Software:
 - Pacote (packaged software): software destinado a atender a um grupo de clientes com o mesmo fim de gerenciamento das informações;
 - Sob encomenda (custom software): software destinado a atender um cliente específico;
 - Embarcado: sistema onde o software encontra-se embutido no próprio hardware a fim de satisfazer requisitos não-funcionais do mesmo;
 - Para Internet: software destinado a trabalhar servindo soluções web que necessitam de um considerável processamento de informações, ou seja, sistemas que não abrangem apenas manutenção do conteúdo;

- Para uso próprio: software desenvolvido para atender as necessidades internas de uma empresa com foco de gerenciamento das informações reduzido.
- Distribuição das empresas, segundo os principais tipos de software desenvolvidos:
 - Financeiro: software destinado ao controle de finanças de uma empresa ou organização;
 - Administração geral: software destinado ao monitoramento das áreas funcionais de uma empresa;
 - Automação comercial: software que gerencia o uso de informações, sob o foco de objetos físicos, das atividades de uma empresa que atua no setor de comércio;
 - Contabilidade: software destinado ao controle contábil de uma empresa ou organização;
 - Administração de recursos humanos: software destinado ao controle dos recursos humanos de uma empresa ou organização;
 - Página Web: Página ou software com baixo processamento de dados, abrangendo apenas manutenção de conteúdo;
 - Gestão integrada - ERP: software responsável pela gestão integrada de todas as informações inerentes à administração de uma empresa ou organização (financeiro, contábil, RH, etc.);

- Administração pública: software destinado às áreas relacionadas à administração pública em geral;
 - Administração de serviços: software destinado às áreas relacionadas à administração de serviços em geral;
 - Automação de escritórios: software que gerencia o uso de informações, sob o foco de objetos físicos, das atividades de um escritório;
 - Automação industrial: software que gerencia o uso de informações, sob o foco de objetos físicos, das atividades de uma indústria.
- Atividades características das empresas em Tecnologia da Informação:
 - Desenvolvimento de software: desenvolver as atividades inerentes à produção do software propriamente dito;
 - Consultoria e projetos: desenvolver atividades como: arquitetura da solução, modelagem do negócio, auditoria de qualidade e outras atividades de suporte ao desenvolvimento de projetos;
 - Treinamento: oferecer treinamento sobre o produto desenvolvido e todos os procedimentos necessários para este fim;
 - Distribuição ou Editoração de Software de terceiros: distribuir (implantar) ou editar software de terceiros;

- Manutenção e assistência técnica: executar manutenção e assistência técnica para os produtos desenvolvidos pela empresa;
- Serviços de automação comercial: desenvolver a atividade de prestação de serviços para automação de estabelecimentos comerciais;
- Distribuição ou revenda de produtos de hardware: distribuir (implantar) e revender produtos de hardware;
- Indústria de informática, telecomunicações ou automação: atuar na produção em larga escala (informática, telecomunicações ou automação);
- Serviços de processamento de dados: oferecer serviços de processamento, gerenciamento e manipulação de dados para uso em sistemas;
- Serviços de automação industrial: oferecer serviços na área de automação industrial que gerencie o uso de informações, sob o foco de objetos físicos.

3.5. Adequação das Classificações de Organizações Desenvolvedoras de Software

De acordo com as definições e limitações de cada um dos itens das classificações mostradas nas seções anteriores, foi realizado um

relacionamento entre todas, conforme mostrado no Quadro 1, indicando as possibilidades de relacionamento entre essas classificações (os possíveis relacionamentos estão indicados por um “X” no Quadro 1).

Nesse relacionamento percebemos que não é possível que uma Fábrica de Projetos de Software, Fábrica de Projetos Físicos ou Fábrica de Programas possa atuar em consultoria e projetos nem tampouco em distribuição ou revenda de produtos de hardware e indústria de informática, telecomunicações ou automação já que essas são atividades exclusivas de uma Fábrica de Projetos Ampliada. Não existe também a possibilidade de uma Fábrica de Projetos Físicos fazer treinamento ou distribuição e editoração de software de terceiros já que são atividades que fogem do escopo desse tipo de fábrica.

Quadro 1 – Relação entre os Tipos de Organizações Contemplados pela Pesquisa

	Fábrica de Projetos (Ampliada)	Fábrica de Projetos de Software	Fábrica de Projetos Físicos	Fábrica de Programas
Atividades no tratamento de software				
Pacote	X	X	X	X
Sob encomenda	X	X	X	X
Embarcado	X	X	X	X
Para Internet	X	X	X	X
Para uso próprio	X	X	X	X
Principais tipos de software desenvolvidos				
Financeiro	X	X	X	X
Administração geral	X	X	X	X
Automação comercial	X	X	X	X
Contabilidade	X	X	X	X
Administração de recursos humanos	X	X	X	X
Página Web	X	X	X	X
Gestão integrada – ERP	X	X	X	X
Administração pública	X	X	X	X
Administração de serviços	X	X	X	X
Automação de escritórios	X	X	X	X
Automação industrial	X	X	X	X
Atividades características das empresas em Tecnologia da Informação				
Desenvolvimento de software	X	X	X	X
Consultoria e projetos	X			
Treinamento	X	X		X
Distribuição / Editoração de SW de terceiros	X	X		X
Manutenção e assist. técnica	X	X	X	X
Serviços de automação comercial	X	X	X	X
Distribuição / revenda de produtos de hardware	X			
Indústria de informática, telecomunicações ou automação	X			
Serviços de processamento de dados	X	X	X	X
Serviços de automação industrial	X	X	X	X

4. ANÁLISE DE CARACTERÍSTICAS ORGANIZACIONAIS NA DEFINIÇÃO DE PROCESSO DE SOFTWARE AO AMBIENTE IMPPROS

4.1. Descrição da Proposta

A análise das características organizacionais que influenciam direta ou indiretamente na definição do processo de software padrão de uma organização é objetivo do estudo apresentado nesse capítulo. Aqui não apenas serão tratadas as características das organizações, mas também todo o fluxo de atividades envolvido nesse módulo de definição de processos do qual este trabalho faz parte, bem como os atores e as diferentes visões nas quais se encontram essas atividades.

O fluxo de atividades desenvolvido leva em consideração todo o procedimento relacionado à definição de regras organizacionais no ambiente, considerando sua manutenção, avaliação e até mesmo a sugestão de componentes do processo pelo ambiente.

Esse trabalho servirá como ponto de partida para o módulo de definição de processos no ImPPros. As características aqui identificadas

serão empregadas na concepção do processo padrão de uma organização.

4.2. Atores Envolvidos

No desenvolvimento do fluxo de atividades que contemple essa proposta foram identificados três tipos de atores: Projetista de Processo, Gerente de Processo e Ambiente.

O Projetista de Processo é responsável pela definição do processo, coleta e triagem 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, é este o ator que avalia individualmente uma nova regra inserida no ambiente.

O Gerente de Processo é o tipo de usuário acompanha a simulação e a avaliação do processo a fim de prover conhecimentos formal e informal (lições aprendidas) para possibilitar o reuso e a melhoria contínua dos processos de software, coordena as avaliações individuais de cada um dos Projetistas de Processo envolvidos na avaliação de uma nova regra inserida no ambiente.

O Ambiente é o ator que representa o próprio sistema nas atividades onde este recebe entradas dos outros atores envolvidos (Gerente de Processo, Projetista de Processo) e emite saídas ou eventos para os mesmos.

4.3. Visões da Proposta

No desenvolvimento dessa proposta foram identificadas estruturas que organizam funcionalmente as atividades que compõem o fluxo, essas estruturas são também conhecidas como visões. Seguem abaixo as visões encontradas:

- Gerenciamento: esta visão possui a finalidade de prover a entrada e manipulação de informações das regras que possibilitarão a sugestão de componentes do processo pelo ambiente. Dentro deste perfil, as atividades contempladas nesta visão são: Manter Regras, Montar Regras e Montar Formulários.
- Avaliação: esta visão objetiva possibilitar a verificação de consistência e conformidade das informações mantidas pelo ambiente. Dentro deste perfil, as atividades contempladas nesta visão são: Avaliar Regras que por sua vez possui Escolher Avaliadores, Receber Formulário Avaliação, Checar Pendência, Parecer das Avaliações, Inserir Conhecimento Adicional e Disseminar Regra como sub-atividades.
- Uso: esta visão tem como foco o processamento das informações das regras e a inferência de resultados. Dentro deste perfil, as atividades contempladas nesta visão são: Inferir Informações

(Organizacionais), Processar Informações Inferidas e Sugerir Componentes.

4.4. Fluxo de Atividades

Esta seção tem como objetivo descrever o fluxo de atividades que foi identificado neste trabalho. Esses fluxos estão apresentados da seguinte forma: um fluxo principal conforme apresentado na Figura 3 e um auxiliar, apresentado na Figura 4, onde encontramos as sub-atividades responsáveis pelo fluxo interno da atividade Avaliar Regras.

4.4.1. Fluxo Principal

Inicialmente visualizamos a atividade Manter Regra que será responsável pela manutenção de um cadastro de características organizacionais, onde o projetista de processos poderá inserir novas regras deste porte bem como modificar as existentes na base de regras.

De posse de novas regras introduzidas na base do ambiente, este necessitará que essas regras sejam avaliadas responsabilidade da atividade Avaliar Regras na qual o gerente de processo analisará as regras em conjunto com os projetistas de processo e com o suporte do ambiente, e esta, por se tratar de uma macro-atividade, possui um fluxo interno que será tratado na próxima subseção.

Após a avaliação o ambiente relaciona as regras, que foram anteriormente avaliadas, e os componentes do processo a fim de prover uma posterior sugestão do seu uso na atividade Montar Regras.

Em seguida, já na atividade Montar Formulários de Características Organizacionais, o ambiente coleta os tipos de organizações e as características organizacionais e monta os formulários para posterior inferência do projetista de processo.

Na atividade Inferir Informações Organizacionais: o projetista de processo realiza uma análise da organização, ou seja, avalia como a mesma encontra-se definida com base nos padrões de empresas de desenvolvimento de software. Posteriormente, ele define as características organizacionais e os tipos de organizações as quais a mesma se enquadra de acordo com os perfis descritos pelo ambiente.

E por fim, na atividade Processar Informações Inferidas as informações definidas no formulário são coletadas pelo ambiente e é realizado um processamento destas informações inferidas junto às regras organizacionais, para posterior sugestão dos componentes do processo.

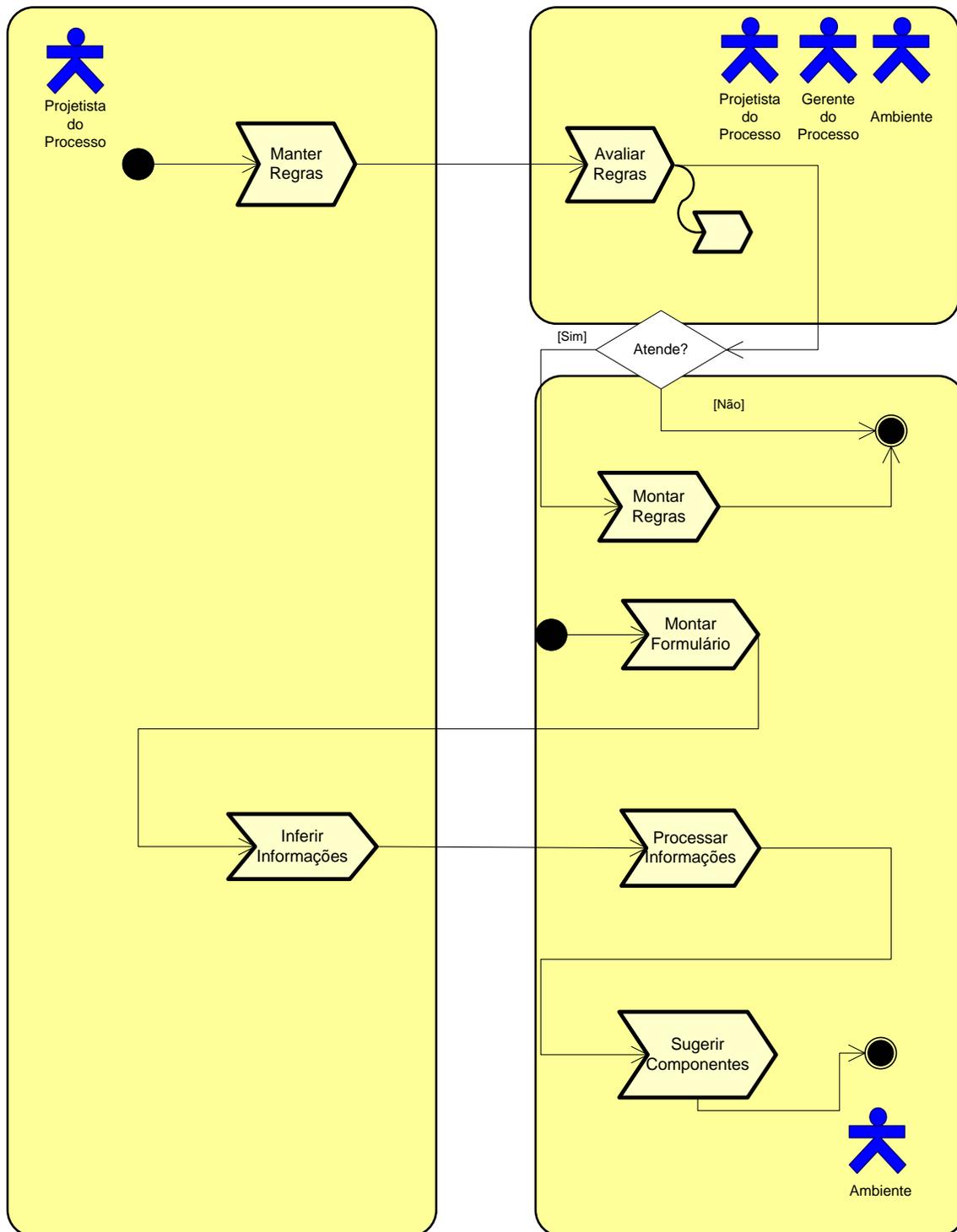


Figura 3 -Fluxo Principal de Atividades

4.4.2. Fluxo Auxiliar Avaliar Regras

Inicialmente, na atividade Escolher Avaliadores, o gerente de processos escolhe os projetistas de processos que irão participar da

avaliação individual, ou seja, darão um parecer sobre a regra analisada segundo critérios próprios.

Em seguida, os projetistas de processos recebem o formulário de avaliação da regra (atividade Receber Formulário Avaliação), a partir de um e-mail de comunicação. Posteriormente, cada projetista de processos faz uma avaliação individual da regra em questão (atividade Avaliar Individualmente).

Já na atividade Checar Pendência o ambiente verifica se a avaliação individual foi realizada, caso contrário ele notifica o Projetista de Processo da avaliação a ser executada.

Na atividade Parecer das Avaliações o gerente de processos faz uma avaliação geral a partir das avaliações individuais de cada projetista de processos e emite um parecer destas avaliações.

Caso a regra passe pela avaliação ela será adicionada à base de regras (atividade Inserir Conhecimento Adicional), podendo receber comentários adicionais aos de sua manutenção. O ambiente também disseminará via e-mail a existência de uma nova regra na sua base a todos os membros na atividade Disseminar Regra.

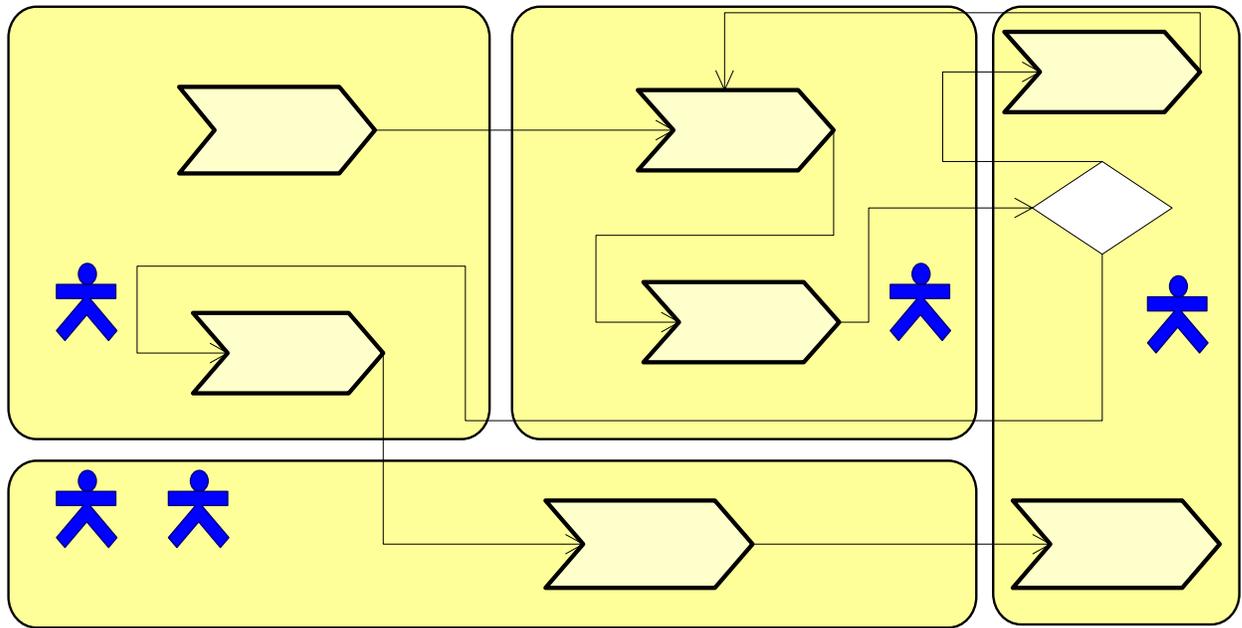


Figura 4 - Fluxo das Sub-Atividades que Representam o Fluxo Interno da Macro-Atividade Avaliar Regras

Escolhe
Avaliador

4.5. Características Organizacionais

No decorrer deste trabalho foram identificados três grupos de Características Organizacionais que influenciam na definição do processo padrão. Esta pesquisa foi realizada mediante análise da literatura especializada que relaciona estas possíveis características [MACHADO_00], assim como a sua influência na definição de um processo de software [REIS_98]. Estes grupos estão relacionados nas próximas subseções. No Apêndice A pode ser encontrado um quadro onde são listadas todas as características mapeadas, bem como os possíveis valores que essas características podem adquirir no momento em que o projetista de processo irá fazer a inferência da regra, conforme mostrado na seção anterior. Também serão mostradas definições pra

Gerente
do
Processo

Parece
das
Avaliaçõ

cada um dos valores pra que a escolha de um valor esteja adequada com as exigências do ambiente.

4.5.1. Características do Processo de Software

Nesta subseção foram relacionadas as Características do Processo de Software nas organizações. Procura-se que, de posse destas características se possa avaliar o processo de software institucionalizado na organização. Para este grupo as seguintes características foram identificadas:

- **Nível de Maturidade:** essa característica pode assumir os valores: Inicial, Repetível, Definido, Gerenciado ou Em Otimização (Dependendo do Modelo ou Norma de Qualidade a ser inferido no Ambiente e seu mapeamento para os valores definidos como padrão).
- **Modelo de Maturidade:** os possíveis valores para essa característica são Exato e Real, Autorizado, Crível ou Rápido de Usar (Dependendo do Modelo/Norma de Qualidade a ser inferido no Ambiente).
- **Institucionalização do Processo:** essa característica assumirá os valores Efetivo, Utilizável ou Consistentemente Aplicado de acordo com o que será inferido na organização.

4.5.2. Características do Ambiente de Desenvolvimento

Nesse grupo foram relacionadas inúmeras características relacionadas ao ambiente de desenvolvimento de uma organização conforme listado abaixo, seus possíveis valores como também definições podem ser obtidos no Apêndice A. As características contempladas foram:

- Nível de contribuição na facilidade do gerenciamento do projeto;
- Nível de contribuição na facilidade do controle de versões;
- Nível de facilidade no uso;
- Condição de suporte de uso a múltiplos usuários;
- Provimento de comunicação entre os membros da equipe de desenvolvimento;
- Suporte à cooperação da equipe de desenvolvimento;
- Suporta a múltiplas visões para diferentes usuários;
- Nível de automação da gerência do processo;
- Possibilidade de extensão;
- Possibilidade de integração entre as tecnologias utilizadas pela organização;
- Possibilidade de integração entre os paradigmas de desenvolvimento utilizados pela organização.

4.5.3. Características da Equipe

A exemplo do grupo de características mostrado na seção anterior, para esse grupo demais informações estão listadas no Apêndice A. Como características da equipe de uma organização são listadas:

- Nível de conhecimento de Engenharia de Software;
- Nível de aplicação dos conhecimentos de Engenharia de Software;
- Adequação dos perfis dos membros às suas habilidades;
- Quantidade de perfis agregados pelos membros no escopo do processo de software;
- Nível de treinamento nas tecnologias utilizadas;
- Nível de treinamento no ambiente de desenvolvimento utilizado;
- Nível de treinamento no processo de software institucionalizado.

5. VALIDAÇÃO DA ADEQUAÇÃO

5.1. Arquitetura de Implementação

Objetivando a validação da proposta apresentada no capítulo anterior, foi implementado um protótipo que contemplou todo o fluxo de atividades mostrado anteriormente. Conforme visto na Figura 3, para implementação do protótipo foi escolhida uma arquitetura em 3 camadas, atingindo assim um baixo acoplamento e melhorando a coesão do sistema.

Para a implementação foi utilizada a linguagem de programação Java [JAVA_05], assim como os ambientes de desenvolvimento Eclipse [ECLIPSE_05] e NetBeans [NETBEANS_05]. Visando a persistência dos dados foi utilizado o banco de dados relacional MySQL [MYSQL_05].

Nas próximas subseções serão vislumbrados todos os detalhes de cada uma das camadas desta arquitetura, a qual dispõe de vários padrões de projeto utilizados em todas as suas camadas. No Apêndice B do presente trabalho poderá ser visualizado o modelo de dados relacional projetado para esta proposta.

5.1.1. Camadada de Apresentação

É a camada responsável pela interação do usuário com o software onde encontramos os elementos gráficos do protótipo que capturam as entradas do usuário e eventualmente fornecem saídas para o mesmo. Na sua implementação foi realizada utilizando a biblioteca Swing. Mesmo se tratando de um software com objetivo de ser usado apenas como protótipo foram utilizados *Guidelines*¹ para que o mesmo se tornasse usável.

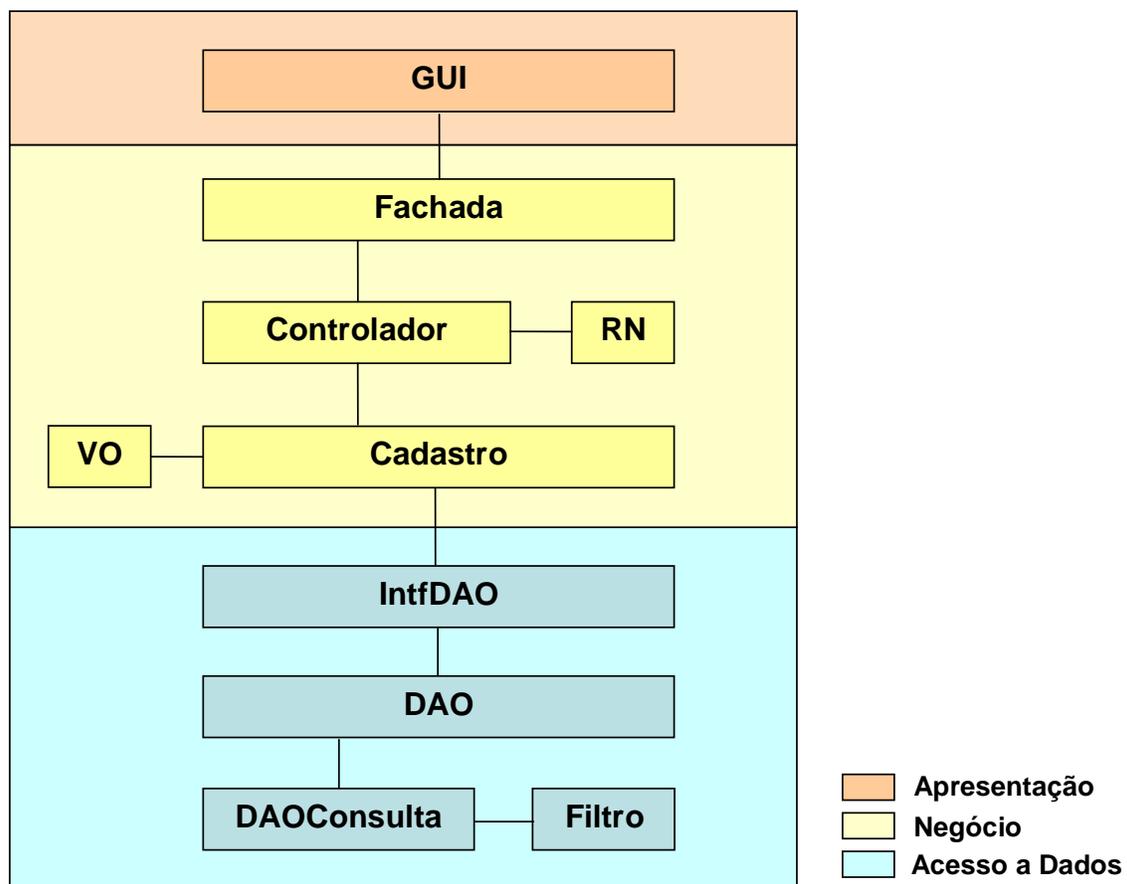


Figura 5 - Arquitetura Proposta para a Implementação do Protótipo de Validação da Proposta

5.1.2. Camadada de Negócios

¹ Padrões que devem ser seguidos na confecção de uma interface gráfica com o objetivo de torná-la mais atraentes e de fácil utilização.

Esta camada é responsável por retornar, incluir, modificar e excluir os dados da camada de dados além de repassar os dados da camada de dados para a camada de apresentação de forma transparente. É nesta camada que estão presentes todas as regras de negócios do sistema. O detalhamento de cada um dos elementos desta camada são vistos a seguir:

- Fachada: elemento responsável por tornar a camada de negócios acessível por apenas um ponto de entrada pela camada de aplicação.
- Controlador: responsável pela gerência os serviços do sistema, temos um controlador para cada serviço.
- RN (Regra de Negócios): classes utilizadas para retirar a complexidade dos controladores, todas as regras de negócios do sistema estarão nessas classes.
- Cadastro: classes que fazem a comunicação com a camada de dados. É independente da implementação da camada de dados.

5.1.3. Camadada de Acesso a Dados

A Camada de Acesso a Dados é responsável pela conexão com a fonte de dados do sistema, nela encontramos os seguintes elementos:

- InterfaceDAO: interface que determina quais serão os métodos existentes na classe que acessa diretamente os dados. São

determinantes para que a camada de negócios seja independente da implementação da camada de dados.

- DAO (Data Access Object): classe responsável pelo acesso aos dados. Implementa os métodos básicos de acesso, inclusão, alteração, exclusão e consulta por chave primária.
- DAOConsulta: classe responsável por fazer consultas à base de dados.
- Filtro: classes utilizadas para parametrizar as consultas. Nelas estarão concentrados os possíveis parâmetros da consulta bem como o nome de todas as colunas retornadas.
- VO (Value Object): representam as entidades na aplicação.

5.2. Protótipo

O objetivo desta seção é mostrar o funcionamento do protótipo após a implementação do mesmo. Para atingir essa meta será apresentada uma das atividades do fluxo apresentado no capítulo anterior, a atividade Avaliar Regra.

Com a finalidade de assegurar a consistência de informações na base de conhecimento, sempre que uma nova regra é adicionada ao ambiente é necessário que ocorra uma avaliação. Nesse caso, o ambiente assegura a existência de uma avaliação toda vez que ocorra a criação de uma nova regra. Esta avaliação pode ser vista na Figura X. (tela Manter Avaliação Consulta).

O próximo passo é a escolha de três avaliadores (Projetistas de Processo) por parte do Gerente de Processos para que estes participem do procedimento de avaliação, conforme visto na Figura 6.

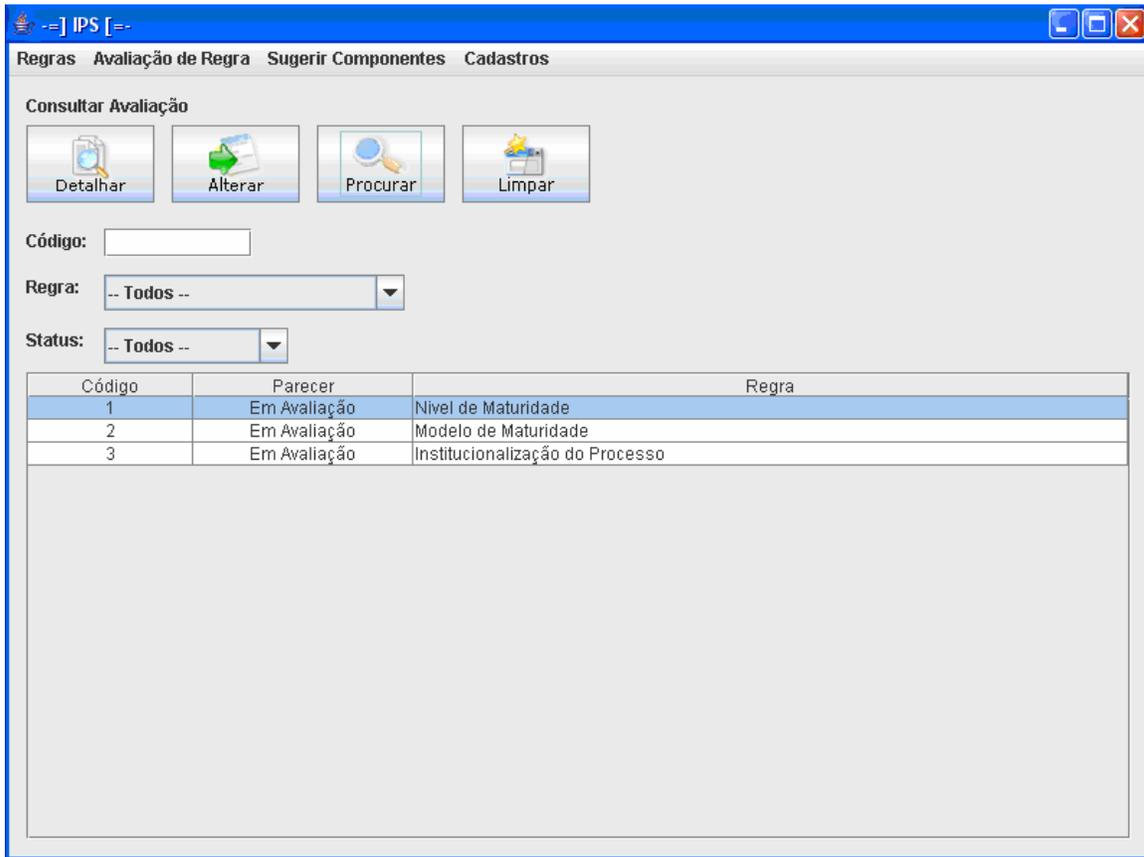


Figura 6 - Tela Manter Avaliação Consulta

Realizada a escolha dos avaliadores o ambiente se responsabilizará pela notificação através de um e-mail automático que será enviado para cada um dos avaliadores escolhidos.

Feito o aviso, cada um dos projetistas de processo acessa o ambiente e realiza sua avaliação individual, onde preenchem um formulário de avaliação e emitem os seus pareceres individuais em relação à regra analisada, veja isto na Figura 8.

Regras Avaliação de Regra Sugerir Componentes Cadastros

Processar Avaliação Individual

Ok Cancelar Limpar

Regra: Nivel de Maturidade

Membro: Márcio Bernardino Fernandes Lima da Cunha

Data: 18/08/2005

Parecer: Aceitação Forte

Comentário: De acordo.

Formulário de Avaliação

Grau de Qualificação

Valor: Muito Familiar

Comentário: De acordo.

Figura 7 - Tela Manter Avaliação Individual Avaliar

Uma vez finalizadas todas as avaliações individuais, o gerente de processos analisará os resultados emitidos pelos avaliadores e fornecerá um parecer final acerca da avaliação, veja a Figura 9.

[-] IPS [-]

Regras Avaliação de Regra Sugerir Componentes Cadastros

Alterar Avaliação

Ok Cancelar Limpar

Código: 1

Regra: Nivel de Maturidade

Status: Manter na Base

Conhecimento Adicional: Sem comentários adicionais.

Figura 8 - Tela Manter Avaliação Avaliar

6. CONCLUSÕES

6.1. Sumário do Trabalho

Como foi visto, a definição de Características Organizacionais que influenciem na definição de processos de software é de extrema importância para as organizações que desenvolvem software. Tanto a definição dessas características quanto o modo como o módulo de definição de processos do ImPProts deverá trabalhar de posse de regras que utilizem essas características foram contemplados por esse trabalho.

Espera-se então, que o resultado deste trabalho seja utilizado como base para outros que compartilhem de objetivos semelhantes a este, ou seja, atuem na análise de Características Organizacionais e na definição de processos de software.

6.2. Trabalhos Futuros

Mesmo não sendo vistos por esse trabalho surgem algumas sugestões ou necessidades que poderão ser aproveitadas posteriormente tomando o presente trabalho como ponto de partida, são elas:

- Embora na apresentação das características no capítulo 4 haja um indicativo do que cada característica influenciará no contexto da definição do processo de software, é necessário

haver um mapeamento direto dessas características com os componentes do processo de software (processos de software, atividades) a serem propostos;

- Integração da ferramenta proposta neste trabalho ao mecanismo de definição do processo de software e ferramentas de gestão do conhecimento (coleta de experiências, e análise e tomada de decisões);
- Incorporar ao contexto de proposição de componentes do processo de software o tipo de sugestão aprendida, ou seja, baseado em definições de processos de software, prover ao ImPProS a capacidade de coletar sugestões mediante ao aprendizado deste conhecimento.

• REFERÊNCIAS BIBLIOGRÁFICAS

[ASSANO_02] Assano, M. (2002) “Gerenciando Fábricas de Software Terceirizadas suportadas por Modelagem de Negócios”, Rational Software White Paper, 2002.

[BASILI_94] Basili, V.R., (1994) “Facts and Myths affecting Software Reuse”, IEEE, Maryland.

[BORGES_02] Borges, Ligia da Motta Silveira & Falbo, Ricardo de Almeida, Uma Ferramenta de Apoio à Instanciação de Processos de Software com Gerência de Conhecimento, Anais do I Simpósio Brasileiro de Qualidade de Software, Gramado – RS, 2002.

[CUSUMANO_89] Cusumano, M. A. (1989) “The software factory: a historical interpretation”. IEEE Software, 6(2), p.23–30, Cap. 14.

[ECLIPSE_05] Eclipse IDE. Disponível em: < <http://www.eclipse.org>>. Acesso em julho 2005.

[FERNANDES_04] Fernandes, A.A., Teixeira, D. S. (2004). “Fábrica de Software: Implantação e gestão de Operações”, Atlas, São Paulo.

[GREENFIELD_03] Greenfield J., Short K. (2003) "Software Factories – Assembling Applications with Patterns, Models, Frameworks and Tools", OOPSLA'03, ACM, California.

[HUMPHREY_89] Humphrey, W. S., 1989, Managing the Software Process, Addison Wesley Longman Inc.

[ISO_95] ISO/IEC 12207, 1995, Information Technology – Software Life-Cycle Processes.

[JAVA_05] Java Technology. Disponível em: <<http://java.sun.com>>. Acesso em julho 2005.

[MACHADO_00] Machado, L.F., 2000, "Modelo para Definição de Processos de Software na Estação Tabá", Tese de Mestrado, março.

[MCT_00] DSI/CGSA. Qualidade no Setor de Software Brasileiro: 1999. MCT/SEPIN, Brasília, 2000. [ISSN 1518-112X]. <http://www.mct.gov.br/sepim>.

[MYSQL_05] MySQL. Disponível em: <<http://www.mysql.org>>. Acesso em julho 2005

[NETBEANS_05] NetBeans IDE. Disponível em <<http://www.netbeans.org>>. Acesso em julho 2005.

[OLIVEIRA_02] Oliveira, S. R. B. (2000) "Gerência do Desenvolvimento de Software", Monografia apresentada a disciplina Engenharia de Software constante no programa de Mestrado em Ciência da Computação, orientador Prof. Alexandre Vasconcelos, Cin/UFPE, Recife-PE.

[OLIVEIRA_05] Oliveira, S., Vasconcelos, A., Rouiller, A. C. (2005) "Uma Proposta de um Ambiente de Implementação de Processo de Software", Artigo publicado na Revista InfoComp - Revista de Ciência da Computação da UFLA - vol. 4, n. 1, Lavras-MG.

[REIS_98] Reis, Carla Alessandra Lima: Um Gerenciador de Processos de Software para o Ambiente PROSOFT. Dissertação de Mestrado em Ciência da Computação: UFRGS, 1998.

[ZAHARAN_98] Zahran, S., 1998, Software Process Improvement, Addison Wesley Longman Inc.

APÊNDICE A – CARACTERÍSTICAS ORGANIZACIONAIS, VALORES E DEFINIÇÕES

Quadro 2 – Características Organizacionais

Características do Processo de Software	Valores	Definições
Nível de Maturidade.	Inicial, Repetível, Definido, Gerenciado ou Em Otimização (Dependendo do Modelo/Norma de Qualidade a ser inferido no Ambiente e seu mapeamento para os valores definidos como padrão)	Inicial: Os processos de desenvolvimento raramente são definidos e as práticas disponíveis são sacrificadas para atender a prazos incorretamente definidos.
		Repetível: As organizações de desenvolvimento de aplicações no nível Repetível devem possuir capacitação para liberar seus produtos dentro do cronograma e budget, evitando horas extras e custos além do budget.
		Definido: uma forte cultura é desenvolvida nesse nível com a utilização de processos comuns que cobrem os mais importantes elementos do desenvolvimento de aplicações.
		Gerenciado: as organizações estão capacitadas a gerar estatísticas que possam caracterizar o desempenho de seus processos.
		Em Otimização: Melhorias contínuas podem ser desenvolvidas através das lições de aprendizagem de

		<p>cada projeto podem ser desenvolvidos através de ações pró-ativas durante a avaliação de novos métodos de desenvolvimento, novos processos ou tecnologias.</p>
Modelo de Maturidade.	Exato e Real, Autorizado, Crível ou Rápido de Usar (Dependendo do Modelo/Norma de Qualidade a ser inferido no Ambiente)	Exato e Real: baseado nos testes executados em organizações voluntárias que validarão o modelo.
		Autorizado: quanto à concordância com a terminologia PMBOK, submissão de testes a ANSI e IEEE, etc.
		Crível: quanto ao uso de consistentes métodos para evidenciar e caracterizar habilidades organizacionais.
		Rápido de Usar: a avaliação deverá ser conduzida em no máximo cinco dias e dez dias para a montagem da elaboração de recomendações.
Institucionalização do Processo.	Efetivo, Utilizável ou Consistentemente Aplicado.	<p>Efetivo: coerente com o foco (missão, visão, valores) de atuação da organização e atende a todos os projetos desenvolvidos.</p>
		<p>Utilizável: especificado com base no foco da organização sendo que este sofre mudanças no seu contexto de acordo com o projeto o qual servirá como base para o desenvolvimento.</p>
		<p>Consistentemente Aplicado: possui as mesmas características do “Efetivo”, sendo que todos os projetos que fazem uso de serviços sub-contratados possuem um</p>

		processo consistente e aliado aos interesses da organização, ou seja, o processo organizacional faz-se comum a todas as sub-contratadas.
Características do Ambiente de Desenvolvimento	Valores	Definições
Facilita o gerenciamento do projeto?	Sim, Não e Parcial.	Sim: o ambiente de desenvolvimento possui ferramentas/técnicas que auxiliam todos os processos de gerenciamento de projetos da organização.
		Não: o ambiente de desenvolvimento não possui nenhuma ferramenta/técnica que auxiliam os processos de gerenciamento de projetos da organização.
		Parcial: o ambiente de desenvolvimento possui ferramentas/técnicas que auxiliam alguns dos processos de gerenciamento de projetos da organização.
Facilita o controle de versões?	Sim, Não e Parcial.	Sim: o ambiente de desenvolvimento agrega ferramentas/técnicas que automatizam todo o processo de controle de versões do software desenvolvido pela organização.
		Não: o ambiente de desenvolvimento não agrega nenhuma ferramenta/técnica que auxilie no processo de controle de versões do software desenvolvido pela organização.
		Parcial: o ambiente de desenvolvimento agrega

		ferramentas/técnicas que automatizam o processo de controle de versões do software desenvolvido pela organização de forma parcial
Nível de facilidade no uso.	Alto, Médio ou Baixo.	Alto: o uso do ambiente é intuitivo, agregando funções relacionadas ao uso de ajuda e de fácil compreensão ao usuário final.
		Médio: o uso do ambiente é intuitivo, no entanto o mesmo agrega funções de ajuda de uma forma não totalmente completa para tarefas sistematizadas ao usuário final.
		Baixo: o uso do ambiente é intuitivo, no entanto o mesmo não possui funções de ajuda ao usuário final.
Suporta o seu uso a múltiplos usuários?	Sim, Não e Parcial.	Sim: possui a habilidade de prover seu uso diferenciado para todos os perfis de usuário definidos como base da organização.
		Não: não possui a habilidade de prover seu uso diferenciado para todos os perfis de usuário definidos como base da organização.
		Parcial: possui a habilidade de prover seu uso diferenciado para alguns dos perfis de usuário definidos como base da organização, por exemplo, nos casos onde é muito comum a área gerencial fazer uso de um sistema de acompanhamento do processo em execução.
Provê a comunicação entre os membros da	Sim, Não e Parcial.	Sim: o ambiente disponibiliza o uso de

<p>equipe de desenvolvimento?</p>		<p>ferramentas/técnicas de comunicação (chat, fórum, e-mail, reuniões, etc.) entre todos os membros da equipe de desenvolvimento.</p>
		<p>Não: o ambiente não disponibiliza o uso de ferramentas/técnicas de comunicação (chat, fórum, e-mail, reuniões, etc.) entre todos os membros da equipe de desenvolvimento.</p>
		<p>Parcial: o ambiente disponibiliza o uso de ferramentas/técnicas de comunicação (chat, fórum, e-mail, reuniões, etc.) entre membros da equipe de desenvolvimento do mesmo nível, ou seja, entre desenvolvedores, entre analistas, entre designers e não permitindo a inter-relação entre estes.</p>
<p>Facilita a cooperação da equipe de desenvolvimento?</p>	<p>Sim, Não e Parcial.</p>	<p>Sim: o ambiente possibilita que os membros da equipe de desenvolvimento compartilhem a execução de todos os componentes do processo.</p>
		<p>Não: o ambiente não possibilita que os membros da equipe de desenvolvimento compartilhem a execução dos componentes do processo.</p>
		<p>Parcial: o ambiente possibilita que os membros da equipe de desenvolvimento compartilhem a execução de alguns dos componentes do processo, por exemplo: atividades e não artefatos.</p>

Suporta múltiplas visões para diferentes usuários?	Sim, Não e Parcial.	Sim: o ambiente provê a função de adaptar seus serviços e os objetos automatizados de acordo com o perfil dos membros da equipe de desenvolvimento.
		Não: o ambiente não provê a função de adaptar seus serviços e os objetos automatizados de acordo com o perfil dos membros da equipe de desenvolvimento.
Nível de automação da gerência do processo?	Alto, Médio ou Baixo.	Alto: a execução do processo é automatizada e o status (estado) dos componentes do processo é controlado por uma máquina de controle.
		Médio: a execução do processo é automatizada, porém o status (estado) dos componentes do processo não é controlado por uma máquina de controle.
		Baixo: o ambiente não é orientado a processos.
Possibilita a sua extensão?	Sim, Não e Parcial.	Sim: o ambiente permite a inserção ou criação de novas ferramentas ou já existentes.
		Não: o ambiente não permite a inserção ou criação de novas ferramentas ou já existentes.
		Parcial:
Possibilita a integração entre as tecnologias utilizadas pela organização?	Sim, Não e Parcial.	Sim: o ambiente provê com que as tecnologias usadas ao longo do desenvolvimento sejam integradas em todos os seus diferentes níveis de controle (interface, dados, serviços, etc.)
		Não: o ambiente não provê com que as tecnologias

		<p>usadas ao longo do desenvolvimento sejam integradas nos seus diferentes níveis de controle (interface, dados, serviços, etc.)</p> <p>Parcial: o ambiente provê com que as tecnologias usadas ao longo do desenvolvimento sejam integradas em alguns dos seus diferentes níveis de controle, por exemplo: uma ferramenta executa serviços da outra mas não compartilham modelos de dados.</p>
Possibilita a integração entre os paradigmas de desenvolvimento utilizados pela organização?	Sim ou Não	<p>Sim: o ambiente permite com que o desenvolvimento faça uso de diferentes paradigmas e proveja um controle na sua integração.</p> <p>Não: o ambiente não permite com que o desenvolvimento faça uso de diferentes paradigmas e não provê um controle na sua integração.</p>
Características da Equipe	Valores	Definições
Nível de conhecimento de engenharia de software.	Alto, Médio ou Baixo.	<p>Alto: a equipe de desenvolvimento possui conhecimento teórico e prático nas técnicas, modelos, normas, padrões, metodologias, etc. relacionadas ao desenvolvimento de software nas organizações.</p> <p>Médio: a equipe de desenvolvimento possui conhecimento apenas teórico nas técnicas, modelos, normas, padrões, metodologias, etc. relacionadas ao</p>

		desenvolvimento de software nas organizações.
		Baixo: a equipe de desenvolvimento não possui conhecimento nas técnicas, modelos, normas, padrões, metodologias, etc. relacionadas ao desenvolvimento de software nas organizações.
Os perfis dos membros se adequam às suas habilidades?	Sim, Não e Parcial.	Sim: todos os membros têm perfis que se adequam às suas habilidades, ou seja, suas atividades estão de acordo com as funções a serem realizadas.
		Não: nenhum dos membros da equipe tem perfil adequado às suas habilidades.
		Parcial: pelo menos um dos membros da equipe não tem perfil adequado às suas habilidades.
Os membros agregam mais de um perfil no escopo do processo de software?	Sim, Não e Parcial.	Sim: todos os membros da equipe agregam mais de um perfil no escopo do processo de software da organização.
		Não: nenhum dos membros da equipe agrega mais de um perfil no escopo do processo de software da organização.
		Parcial: alguns membros da equipe agregam mais de um perfil no escopo do processo de software da organização.
Nível de aplicação destes conhecimentos.	Alto, Médio ou Baixo.	Alto: os membros da equipe aplicam de forma diferenciada os modelos, normas, etc. relacionadas ao desenvolvimento de software, em projetos com características específicas.

		<p>Médio: os membros da equipe aplicam de forma igualitária os modelos, normas, etc. relacionadas ao desenvolvimento de software, em projetos com características específicas.</p> <p>Baixo: os membros da equipe não aplicam os modelos, normas, etc. relacionadas ao desenvolvimento de software, em projetos com características específicas.</p>
Possui treinamento nas tecnologias utilizadas?	Sim, Não e Parcial.	<p>Sim: a equipe recebe um acultramento e treinamento das tecnologias usadas no desenvolvimento de softwares, e acompanha esta aplicação em projetos reais.</p> <p>Não: a equipe não recebe um acultramento e treinamento das tecnologias usadas no desenvolvimento de softwares.</p> <p>Parcial: a equipe recebe um acultramento e treinamento das tecnologias usadas no desenvolvimento de softwares, mas não possui um acompanhamento desta aplicação em projetos reais.</p>
Possui treinamento no ambiente de desenvolvimento utilizado?	Sim, Não e Parcial.	<p>Sim: todos os membros da equipe possuem treinamento no ambiente de desenvolvimento utilizado na organização.</p> <p>Não: nenhum dos membros da equipe possui treinamento no ambiente de desenvolvimento utilizado na organização.</p> <p>Parcial: alguns dos membros da equipe possuem</p>

		treinamento no ambiente de desenvolvimento utilizado na organização ou até mesmo todos os membros possuem treinamento, só que incompleto.
Possui treinamento no processo de software institucionalizado?	Sim, Não e Parcial.	Sim: todos os membros da equipe possuem treinamento no processo de software institucionalizado na organização.
		Não: nenhum dos membros da equipe possui treinamento no processo de software institucionalizado na organização.
		Parcial: alguns dos membros da equipe possuem treinamento no processo de software institucionalizado na organização ou até mesmo todos os membros possuem treinamento só que incompleto, ou seja, não abrange todo o processo.

APÊNDICE B – MODELO DE DADOS RELACIONAL

